AIML

QUESTION BANK

MODULE-1

Short Questions:

1. What is a AI?
2. What are goals of AI?
3. What are types of AI
4. What is Weak and strong AI
5. What are agents
6. What is rationality
7. What is an ideal rational agent?
8. What is an environment
9. What are properties of Search Algorithm.
10. What are the types of uninformed search?
11. What is difference between uninformed and informed search.
12. What is uniform cost search.
13. What is Heuristics function.
14. What are features of hill climbing algorithm?
15. What is local maximum and global maximum.

Descriptive Questions:

1. Write the advantages and disadvantages of AI
2. Describe any three applications of AI
3. Describe simple rational agent and Goal based agents.
4. Describe properties of Environment.
5. Describe types of search algorithm.
6. Describe DFS and BFS algorithm with their advantages and disadvantages.
7. Describe uniformed cost search with advantages and disadvantages.
8. Explain Greedy search algorithm with examples
9. Explain A* search algorithm with examples
10. Describe simple hill climb algorithm with example.
11. Describe mean-end analysis algorithm with examples
12. Describe Constraint satisfaction problem with examples.

Compiler Design
Question Bank

UNIT 1

1. What is Compiler? Design the Analysis and Synthesis Model of Compiler.
2. Write down the five properties of compiler.
3. What is translator? Write down the steps to execute a program.
4. Discuss all the phases of compiler with a with a diagram.
5. Write a short note on:
    a. YACC
    b. Pass
    c. Bootstrapping
    d. LEX Compiler
    e. Tokens, Patterns and Lexemes
6. Write the steps to convert Non-Deterministic Finite Automata (NDFA) into Deterministic Finite Automata (DFA).
7. Let M=({q0,q1}, {0,1}, δ, q0, {q1}).
   Be NFA where δ(q0,0)={q0,q1}, δ(q1,1) = {q1}
   δ(q1, 0)=φ, δ(q1, 1)={q0, q1}
   Construct its equivalent DFA.

8. Convert the given NFA to DFA:

Input/State	0	1
➔ q0	{q0, q1}	q0
q1	q2	q1
q2	q3	q3
q3 (final state)	φ (null character)	q2

9. What is Regular Expression? Write the regular expression for:
    a. R=R1+R2 (Union operation)
    b. R=R1.R2 (concatenation Operation)
    c. R=R1* (Kleen Clouser)
    d. R=R+ (Positive Clouser)
    e. Write a regular expression for a language containing strings which end with “abb” over Σ = {a,b}.
    f. Construct a regular expression for the language containing all strings having any number of a’s and b’s except the null string.
10. Construct Deterministic Finite Automata to accept the regular expression :
    (0+1)* (00+11) (0+1)*

11. Derivation and Parse Tree:
    a. Let G be a Context Free Grammar for which the production Rules are given below:

       S -> aB|bA

       A -> a|aS|bAA

       B -> b|bS|aBB

Drive the string *aaabbabbba* using the above grammar (using Left Most Derivation and Right most Derivation).

UNIT 2

1. Explain the parsing techniques with a hierarchical diagram.
2. What are the problems associated with Top Down Parsing?
3. Write the production rules to eliminate the left recursion and left factoring problems.
4. Consider the following Grammar:

   A-> ABd|Aa|a

   B-> Be|b

   Remove left recursion.

5. Do left factoring in the following grammar:

   A-> aAB|aA|a

   B-> bB|b

6. Write a short note on:
   a. Ambiguity (with example)
   b. Recursive Descent Parser
   c. Predictive LL(1) parser (working)
   d. Handle pruning
   e. Operator Precedence Parser
7. Write Rules to construct FIRST Function and FOLLOW Function.
8. Consider Grammar:

   E-> E+T|T

   T-> T*F|F

   F-> (E)|id

9. Write the algorithm to create Predictive parsing table with the scanning of input string.
10. Show the following Grammar:

    S-> AaAb|BbBa

    A-> €

    B-> €

    Is LL(1) and parse the input string “ba”.

11. Consider the grammar:

    E-> E+E

    E-> E*E

    E->id

    Perform shift reduce parsing of the input string “id1+id2+id3”.

12. Write the properties of LR parser with its structure. Also explain the techniques of LR parser.
13. Write a short note on:
    a. Augmented grammar
    b. Kernel items
    c. Rules of closure operation and goto operation
    d. Rules to construct the LR(0) items
14. Consider the following grammar:

    S-> Aa|bAc|Bc|bBa

    A-> d

    B-> d

    Compute closure and goto.

15. Write the rules to construct the SLR parsing table.
16. Consider the following grammar:

    E-> E+T|T

    T-> TF|F

    F-> F*|a|b

    Construct the SLR parsing table and also parse the input “a*b+a”

17. Write the rules to construct the LR(1) items.

18. What is LALR parser? Construct the set of LR(1) items for this grammar:

    S-> CC

    C-> aC

    C->d

19. Show the following grammar

    S->Aa|bAc|Bc|bBa

    A->d

    B->d

    Is LR(1) but not LALR(1).

20. Write the comparison among SLR Parser, LALR parser and Canonical LR Parser.

UNIT 3

1. What is syntax directed translation (SDD)?
2. Write short note on:
    a. Synthesized attributes
    b. Inherited attributes
    c. Dependency graph
    d. Evaluation order
    e. Directed Acyclic Graph (DAG)
3. Draw the syntax tree and DAG for the following expression:

    (a*b)+(c-d)*(a*b)+b

4. Differentiate between synthesized translation and inherited translation.
5. What is intermediate code and write the two benefits of intermediate code generation.
6. Write the short note on:
    a. Abstract syntax tree
    b. Polish notation
    c. Three address code
    d. Backpatching
7. Construct syntax tree and postfix notation for the following expression:

(a+(b*c)^d-e/(f+g)

8. Write quadruples, triples and indirect triples for the expression:

   -(a*b)+(c+d)-(a+b+c+d)

9. Write the three address statement with example for:
   a. Assignment
   b. Unconditional jump (goto)
   c. Array statement (2D and 3D)
   d. Boolean expression
   e. If-then-else statement
   f. While, do-while statement
   g. Switch case statement

UNIT 4

1. Write the definition of symbol table and procedure to store the names in symbol table.
2. What are the data structures used in symbol table?
3. What are the limitations of stack allocation?
4. Write two important points about heap management.
5. Write the comparison among Static allocation, Stack allocation and Heap Allocation with their merits and limitations.
6. What is activation record? Write the various fields of Activation Record.
7. What are the functions of error handler?
8. Write a short note on Error Detection and Recovery.
9. Classify the errors and discuss the errors in each phase of Compiler.

UNIT 5

1. What are the properties of code generation phase? Also explain the Design Issues of this phase.
2. What are basic blocks? Write the algorithm for partitioning into Blocks.
3. Write a short note on:
   a. Flow graph (with example)
   b. Dominators
   c. Natural loops
   d. Inner loops
   e. Reducible flow graphs

4. Consider the following program code:

```
Prod=0;

I=1;

Do{

Prod=prod+a[i]*b[i];

I=i+1;

}while (i<=10);
```

    a. Partition in into blocks
    b. Construct the flow graph
5. What is code optimization? Explain machine dependent and independent code optimization.
6. What is common sub-expression and how to eliminate it? Explain with example.
7. Write a short note with example to optimize the code:
    a. Dead code elimination
    b. Variable elimination
    c. Code motion
    d. Reduction in strength
8. What is control and data flow analysis? Explain with example.

TYPICAL QUESTIONS & ANSWERS

PART - I

OBJECTIVE TYPE QUESTIONS

Each Question carries 2 marks.

Choose correct or the best alternative in the following:

Q.1 In a virtual memory system, the addresses used by the programmer belongs to
(A) memory space. (B) physical addresses.
(C) address space. (D) main memory address.

Ans: C
An address used by programmers in a system supporting virtual memory concept is called virtual address and the set of such addresses are called address space.

Q.2 The method for updating the main memory as soon as a word is removed from the Cache is called
(A) Write-through (B) write-back
(C) protected write (D) cache-write

Ans: B
In this method only cache location is updated during write operation.

Q.3 A control character is sent at the beginning as well as at the end of each block in the synchronous-transmission in order to
(A) Synchronize the clock of transmitter and receiver.
(B) Supply information needed to separate the incoming bits into individual character.
(C) Detect the error in transmission and received system.
(D) Both (A) and (C).

Ans B
As the data are sent continuously as a block of data at the rate dictated by the clock frequency, so the receiver should be supplied with the same function about the same bit length in order to interrupt the information.

Q.4 In a non-vectored interrupt, the address of interrupt service routine is
(A) Obtained from interrupt address table.
(B) Supplied by the interrupting I/O device.
(C) Obtained through Vector address generator device.
(D) Assigned to a fixed memory location.

Ans: D
The source device that interrupted the processor supply the vector address which helps processor to find out the actual memory location where ISR is stored for the device.

Q.5 Divide overflow is generated when
(A) Sign of the dividend is different from that of divisor.
(B) Sign of the dividend is same as that of divisor.
(C) The first part of the dividend is smaller than the divisor.
(D) The first part of the dividend is greater than the divisor.

Ans: B
If the first part of the dividend is greater than the deviser, then the result should be of greater length, then that can be hold in a register of the system. The registers are of fixed length in
any processor.

Q.6 Which method is used for resolving data dependency conflict by the compiler itself?
(A) Delayed load. (B) operand forwarding.
(C) Pre fetch target instruction. (D) loop buffer.

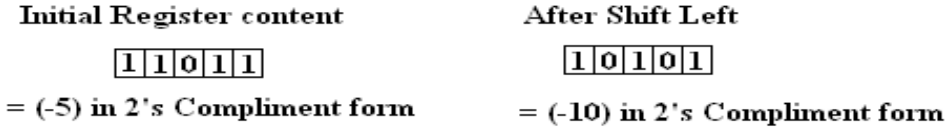
Ans: A
In case of delayed load technique the complier detects the data conflict and reorder the instruction as necessary to delay the loading of the conflicting data by inserting no operation instructions.

Q.7 Stack overflow causes
(A) Hardware interrupt.
(B) External interrupt.
(C) Internal interrupt.
(D) Software interrupt.

Ans: C
Stack overflow occurs while execution of a program due to logical faults. So it is a program dependent, hence interrupt activated.

Q.8 Arithmetic shift left operation
(A) Produces the same result as obtained with logical shift left operation.
(B) Causes the sign bit to remain always unchanged.
(C) Needs additional hardware to preserve the sign bit.
(D) Is not applicable for signed 2's complement representation.

Ans: A
If the register hold minus five in two's compliment form than in arithmetic shift left the contents of the register shall be



It is found that the register contents multiplied by two after logical shift left operation. Hence arithmetic shift left operation is same as logical shift operation.

Q.9 Zero address instruction format is used for
(A) RISC architecture.
(B) CISC architecture.
(C) Von-Neuman architecture.
(D) Stack-organized architecture.

Ans: D
In stack organized architecture push and pop instruction is needs a address field to specify the location of data for pushing into the stack and destination location during pop operation but for logic and arithmetic operation the instruction does not need any address field as it operates on the top two data available in the stack.

Q.10 Address symbol table is generated by the
(A) memory management software.
(B) assembler.
(C) match logic of associative memory.
(D) generated by operating system

Ans: B
During the first pass of assembler address symbol table is generated which contains the label used by the programmer and its actual address with reference to the stored program.

Q.11 The ASCII code for letter A is
(A) 1100011 (B) 1000001
(C) 1111111 (D) 0010011

Ans. (B)

Q.12 The simplified expression of $\overline{(A+B)} + \overline{C}$ is
(A) (A + B)C (B) A(B + C)
(C) (C+A + B) (D) None of these

Ans. (A)

Q.13 The negative numbers in the binary system can be represented by
(A) Sign magnitude (B) I's complement
(C) 2's complement (D) All of the above

Ans. (C)

Q.14 ABCD - seven segment decoder / driver in connected to an LED display. Which segments are illuminated for the input code DCBA = 0001.
(A)b, c (B) c, b
(C)a, b, c (D) a, b, c, d

Ans. (A)

Q.15 How many flip-flops are required to produce a divide-by-32 device?
(A)4 (B) 6

(C)5 (D) 7

Ans. (C)

Q.16 The content of a 4-bit register is initially 1101. The register is shifted 2 times to the right with the serial input being 1011101.
What is the content of the register after each shift?
(A)1110, 0111 (B) 0001, 1000
(C)1101, 1011 (D) 1001, 1001

Ans. (A)

Q.17 How many different addresses are required by the memory that contain 16K words?
(A)16,380 (B) 16,382
(C)16,384 (D) 16,386

Ans. (C)

Q.18 What is the bit storage capacity of a ROM with a 512' 4-organization?
(A) 2049 (B) 2048
(C) 2047 (D) 2046

Ans. (B)

Q.19 DMA interface unit eliminates the need to use CPU registers to transfer data from
(A) MAR to MBR (B) MBR to MAR
(C) I/O units to memory (D) Memory to I/O units

Ans. (D)

Q.20 How many 128 x 8 RAM chips are needed to provide a memory capacity of 2048 bytes?
(A) 8 (B) 16
(C) 24 (D) 32

Ans. (B)

Q.21 Which of the following is a self complementing code?
(A) 8421 code (B) 5211
(C) Gray code (D) Binary code

Ans. (A)

Q.22 Which gate can be used as anti-coincidence detector?
(A) X-NOR (B) NAND
(C) X-OR (D) NOR
Ans. (C)

Q.23 Which of the following technology can give high speed RAM?
(A) TTL (B) CMOS
(C) ECL (D) NMOS

Ans. (C)

Q.24 In 8085 microprocessor how many I/O devices can be interfaced in I/O mapped I/O technique?
(A) Either 256 input devices or 256 output devices.
(B) 256 I/O devices.
(C) 256 input devices & 256 output devices.
(D) 512 input-output devices.

Ans. (C)

Q.25 After reset, CPU begins execution of instruction from memory address
(A) 0101_H (B) 8000_H
(C) 0000_H (D) $FFFF_H$

Ans. (C)

Q.26 Which is true for a typical RISC architecture?
(A) Micro programmed control unit.
(B) Instruction takes multiple clock cycles.
(C) Have few registers in CPU.
(D) Emphasis on optimizing instruction pipelines.

Ans. (A)

Q.27 When an instruction is read from the memory, it is called
(A) Memory Read cycle (B) Fetch cycle
(C) Instruction cycle (D) Memory write cycle

Ans. (B)

Q.28 Which activity does not take place during execution cycle?
(A) ALU performs the arithmetic & logical operation.
(B) Effective address is calculated.
(C) Next instruction is fetched.
(D) Branch address is calculated & Branching conditions are checked.
Ans. (D)

Q.29 A circuit in which connections to both AND and OR arrays can be programmed is called
(A) RAM (B) ROM
(C) PAL (D) PLA
Ans. (A)

Q.30 If a register containing data $(11001100)_2$ is subjected to arithmetic shift left operation, then the content of the register after 'ashl' shall be

(A) $(11001100)_2$ (B) $(1101100)_2$

(C) $(10011001)_2$ (D) $(10011000)_2$

Ans. (D)

Q.31 Which logic is known as universal logic?

(A) PAL logic. (B) NAND logic.

(C) MUX logic. (D) Decoder logic.

Ans. (B)

Q.32 The time for which the D-input of a D-FF must not change after the clock is applied is known as

(A) Hold time. (B) Set-up time.

(C) Transition time. (D) Delay-time.

Ans. (A)

Q.33 How many memory chips of (128 x 8) are needed to provide a memory capacity of 4096 x 16?

(A)64 (B) A B

(C)32 (D) None

Ans. (A)

Q.34 In addition of two signed numbers, represented in 2' s complement form generates an overflow if

(A) A. B = 0 (B) A = 0

(C) $A \oplus B = 1$ (D) A + B = 1

Ans. (C)

Where A is the carry in to the sign bit position and B is the carry out of the Sign bit position.

Q.35 Addition of $(1111)_2$ to a 4 bit binary number 'a' results:-

(A) Incrementing A (B) Addition of $(F)_H$

(C) No change (D) Decrementing A

Ans. (C)

Q.36 In a microprocessor system, suppose. TRAP, HOLD, RESET Pin got activated at the same time, while the processor was executing some instructions, then it will first respond to

(A) TRAP (B) HOLD

(C) RESET (D) None

Ans. (D)

Q.37 Pseudo instructions are
(A) Machine instructions (B) Logical instructions
(C) Micro instructions (D) instructions to assembler.

Ans. (A)

Q.38 An attempt to access a location not owned by a Program is called
(A) Bus conflict (B) Address fault
(C) Page fault (D) Operating system fault

Ans. (B)

Q. 39 Dynamic RAM consumes _____ Power and _____ then the Static RAM.
(A) more, faster **(B)** more, slower
(C) less, slower **(D)** less, faster

Ans. (C)

Q.40 The flag register content after execution of following program by 8085 microprocessor shall be

Program
SUB A
MVI B, $(01)_H$
DCR B
HLT

(A) $(54)_H$ **(B)** $(44)_H$
(C) $(45)_H$ **(D)** $(55)_H$

Ans. (A)

Q.41 Which flag of the 8085's flag register is not accessible to programmer directly?
(A) Zero flag
(B) Carry flag
(C) Auxiliary carry flag
(D) Parity flag

Ans. (C)

Q.42 Cache memory works on the principle of
(A) Locality of data.
(B) Locality of reference
(C) Locality of memory
(D) Locality of reference & memory

Ans. (B)

Q.43 Which of the following is a Pseudo instruction?
(A) SPHL (B) LXI

(C) NOP (D) END

Ans. (D)

Q.44 A demultiplexer can be used as
(A)Encoder (B)Decoder
(C)Multiplexer (D)None of the above

Ans. (B)

Q.45 Excess-3 equivalent representation of $(1234)_H$ is
(A) $(1237)_{Ex-3}$ (B) $(4567)_{Ex-3}$
(C) $(7993)_{Ex-3}$ (D) $(4663)_{Ex-3}$

Ans. (B)

Q.46 Which of the memory holds the information when the Power Supply is switched off?
(A) Static RAM (B) Dynamic RAM
(C) EEROM (D) None of the above

Ans. (C)

Q.47 Minimum no. of NAND gate required to implement a Ex-OR function is
(A)2 (B)3
(C)4 (D)5

Ans. (C)

Q.48 Which of the following interrupt is maskable?
(A)INTR (B)RST 7.5
(C)TRAP (D)Both (A) and (B)

Ans. (B)

Q.49 Which of the following expression is not equivalent to x?
(A) x NAND x (B) x NOR x
(C) x NAND 1 (D) x NOR 1

Ans. (D)

Q.50 Word 20 contains 40
Word 30 contains 50
Word 40 contains 60
Word 50 contains 70
Which of the following instructions does not, load 60 into the Accumulator
(A) Load immediate 60
(B) Load direct 30
(C) Load indirect 20

(D) both (A) & (C)

Ans. (B)

Q.51 An interrupt for which hardware automatically transfers the program to a specific memory location is known as
(A) Software interrupt
(B) Hardware interrupt
(C) Maskable interrupt
(D) Vector interrupt

Ans. (B)

Q.52 Synchronous means _____
(A) At irregular intervals
(B) At same time
(C) At variable time
(D) None of these

Ans. (B)

Q.53 'n' Flip flops will divide the clock frequency by a factor of

(A) n^2 (B) n

(C) 2^n (D) log (n)

Ans. (B)

Q.54 In DMA the data transfer is controlled by

(A) Microprocessor (B) RAM

(C) Memory (D) I/O devices

Ans. (D)

Q.55 The number of instructions needed to add a numbers an store the result in memory using only one address instruction is

(A) n (B) n - 1

(C) n +1 (D) Independent of n

Ans. (D)

Q.56 Negative numbers cannot be represented in
(A) Signed magnitude form
(B) I's complement form
(C) 2's complement form
(D) 8-4-2-1 code

Ans. (C)

Q.57 Which of the following architecture is/are not suitable for realizing SIMD

(A) Vector Processor (B) Array Processor

(C) Von Neumann (D) All of the above

Ans. (C)

Q.58 In Boolean expression A+BC equals
(A) (A+B)(A+C) (B) (A'+B)(A'+C)
(C) (A+B)(A'+C) (D) (A+B)C

Ans. (A)

Q.59 A JK flip-flop can be implemented using D flip-flop connected such that
(A) $D=J\overline{Q}+\overline{K}Q$ (B) $D=\overline{J}Q+K\overline{Q}$
(C) $D=\overline{J}\overline{Q}+KQ$ (D) $D=J\overline{Q}+K\overline{Q}$

Ans. (A)

Q.60 An effective solution to the power consumption problem lies in using _____ transistors to implement ICs.
(A) NMOS (B) TTL shottky
(C) PMOS (D) both NMOS & PMOS

Ans. (D)

Q.61 Memory interleaving technique is used to address the memory modules in order to have
(A) higher average utilization
(B) faster access to a block of data
(C) reduced complexity in mapping hardware
(D) both (A) & (B)

Ans. (C)

Q.62 In a multiprogramming system, which of the following is used
(A) Data parallelism (B) Paging concept
(C) L1 cache (D) None of the above

Ans. (B)

Q.63 Cycle stealing technique is used in
(A) Interrupt based data transfer
(B) Polled mode data transfer
(C) DMA based data transfer
(D) None of these

Ans. (C)

Q.64 Manipulation of individual bits of a word is often referred to as
(A) Bit twidding (B) Bit swapping
(C) Micro-operation (D) None of these

Ans. (A)

Q.65 Which of the following is not a characteristic of a RISC architecture.

(A) Large instruction set (B) One instruction per cycle
(C) Simple addressing modes (D) Register-to-register operation

Ans. (A)

Q.66 When CPU is not fully loaded, which of the following method of data transfer is preferred

(A) DMA (B) Interrupt
(C) Polling (D) None of these

Ans. (D)

Q.67 Associative memory is some times called as

(A) Virtual memory (B) Cache memory
(C) Main memory (D) Content addressable memory

Ans. (D)

Q.68 BCD equivalent of Two's complement is

(A) nine's complement (B) ten's complement
(C) one's complement+1 (D) none of these
Ans. (C)

Q.69 PAL circuit consists of

(A) Fixed OR & programmable AND logic
(B) Programmable OR & Fixed AND Logic
(C) Fixed OR & fixed AND logic
(D) Programmable OR & programmable AND logic
Ans. (A)

Q.70 8085 microprocessor carryout the subtraction by

(A) BCD subtraction method
(B) Hexadecimal subtraction method
(C) 2's complement method
(D) Floating Point subtraction method
Ans. (C)

Q.71 CPU checks for an interrupt signal during

(A) Starting of last Machine cycle
(B) Last T-State of instruction cycle
(C) First T-State of interrupt cycle
(D) Fetch cycle

Ans. (B)

Q.72 During DMA acknowledgement cycle, CPU relinquishes

(A) Address bus only **(B)** Address bus & control bus
(C) Control bus & data bus **(D)** Data bus & address bus

Ans. (D)

Q.73 If the clock input applied to a cascaded Mod-6 & Mod-4 counter is 48KHz. Than the output of the cascaded arrangement shall be of
(A) 4.8 KHz **(B)** 12 KHz
(C) 2 KHz **(D)** 8 KHz

Ans.(C)

Q.74 If the stack pointer is initialised with $(4FEB)_H$, then after execution of Push operation in 8085 microprocessor, the Stack Pointer shall be
(A) 4FEA **(B)** 4FEC
(C) 4FE9 **(D)** 4FED

Ans. (D)

Q.75 A more efficient way to organise a Page Table is by means of an associative memory having
(A) Number of words equal to number of pages
(B) Number of words more than the number of pages
(C) Number of words less than the number of pages
(D) Any of the above

Ans. (A)

Q.76 If there are four ROM ICs of 8K and two RAM ICs of 4K words, than the address range of Ist RAM is (Assume initial addresses correspond to ROMs)
(A) $(8000)_H$ to $(9FFF)_H$ **(B)** $(6000)_H$ to $(7FFF)_H$
(C) $(8000)_H$ to $(8FFF)_H$ **(D)** $(9000)_H$ to $(9FFF)_H$

Ans. (C)

Q.77 $A \oplus B \oplus C$ is equal to $A \odot B \odot C$ for
(A) A=0, B=1, C=0 **(B)** A=1, B=0, C=1
(C) A=1, B=1, C=1 **(D)** All of the above

Ans. (D)

Q.78 Gray code equivalent of $(1000)_2$ is
(A) $(1111)_G$ **(B)** $(1100)_G$
(C) $(1000)_G$ **(D)** None of these

Ans. (A)

PART- II

DESCRIPTIVES

Q.1 Use K-maps to find the simplest Sum of Products (SOP) form of the function
F = f . g, where

$$f = wx\bar{y} + \bar{y}z + \bar{w}y\bar{z} + \bar{x}y\bar{z}$$

$$g = (w + x + y' + z')(x' + y' + z) \quad (7)$$

Ans:

$$f = wx\bar{y} + \bar{y}z + \bar{w}y\bar{z} + \bar{x}y\bar{z}$$

$$= wxy'(z + z') + (x + x')y'z(w + w') + w'(x + x')\ yz' + (w + w')x'yz'$$

$$= wxy'z + wxy'z' + wxy'z + w'xy'z + wx'y'z + w'x'y'z + w'xyz' + wx'y'z$$

$$wx'yz' + w'x'yz'$$

$$= m13 + m12 + m13 + m5 + m9 + m1 + m6 + m12 + m10 + m2$$

$$= m1 + m2 + m5 + m6 + m9 + m10 + m12 + m13$$

$$now, g = (w + x + y' + z')(x' + y' + z)\ (w' + y + z')$$

$$= (w + x + y' + z')(w' + x' + y' + z)\ (w + x' + y' + z)(w' + x' + y' + z)\ (w' + x + y + z')$$

$$= M3M14M6M13M9$$

The K map for f and g are given below

f

wx \ yz	y'z'	y'z	yz	yz'
w'x'	0	1	0	1
w'x	0	1	0	1
wx	1	1	0	0
wx'	0	1	0	1

g

wx \ yz	y'z'	y'z	yz	yz'
w'x'	1	1	0	1
w'x	1	1	1	0
wx	1	0	1	0
wx'	1	0	1	1

Therefore the K-map for F=fg is given below

wx \ yz	y'z'	y'z	yz	yz'
w'x'	0	1	0	1
w'x	0	1	0	0
wx	1	0	0	0
wx'	0	0	0	1

F= m1+m2+m5+m10+m12

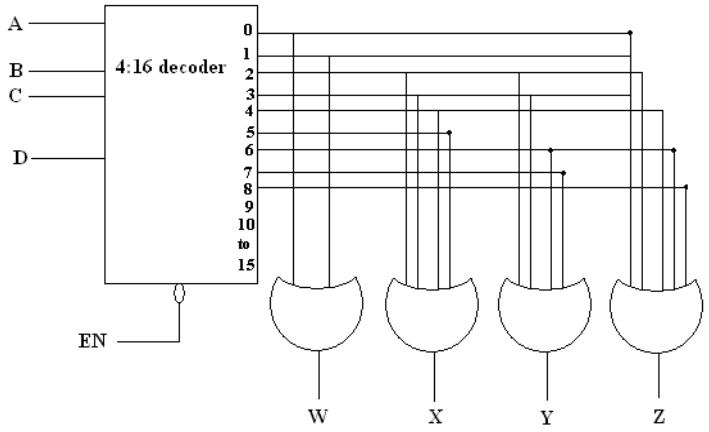
Simplification of above K-map gives-
F = w'y'z + x'yz' + wxy'z'

Q.2 Design a combinational circuit that generates 9's complement of a BCD digit. (6)

Ans:

Decimal No.	BCD input				output			
	A	B	C	D	W	X	Y	Z
0	0	0	0	0	1	0	0	1
1	0	0	0	1	1	0	0	0
2	0	0	1	0	0	1	1	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	0	1
5	0	1	0	1	0	1	0	0
6	0	1	1	0	0	0	1	1
7	0	1	1	1	0	0	1	0
8	1	0	0	0	0	0	0	1
9	1	0	0	1	0	0	0	0

w=m0+m1
x=m2+m3+m4+m5
y=m2+m3+m6+m7
z=m0+m2+m4+m6+m8



Q.3 Show that the dual of EX-OR is also its complement. (3)

Ans:
Logic equation for EX-OR operation is: $A \oplus B = A'B + AB'$
dual of (A ⊕ B)=dual of [A'B+AB']
=(A+B')(A'+B)
=A.A'+A.B+A'.B'+B.B'
=AB+A'B'=(A ⊕ B)
Hence dual of EX-OR is also its complement.

Q.4 Explain with the help of an example, the use of hamming code as error detection and correction code. (7)

Ans:

Hamming code is generated by adding k- parity bits to n - bit data word, forming the new word of (n+k) bits. The bit positions are numbered from l to (n + k) from left to right. Those positions numbered as a power of 2 are reserved for the parity bit. The remaining bits are data bits.

The relation between k & n are as:-

$2^k - 1 - k \geq n$,

If n = 4 then k = 3 and for n = 8 then k =4

Let's consider 8 bit data word 11000100, for which parity generation, error detection and correction capability of Hamming code shall be discussed. For n = 8, k = 4, therefore n+k =12

Bit position:

1	2	3	4	5	6	7	8	9	10	11	12
p1	p2	1	p3	1	0	0	p4	0	1	0	0

The 4 parity bits, p1, p2 p3 and p4 are in position 1, 2, 4 and 8 respectively. The 8-bit data word is in remaining positions. Each parity bit is calculated as follows:-

P1 = XOR of bits (3. 5, 7, 9, 11) = 1+1+0+0+0=0

P2 = XOR of bits (3, 6, 7, 10, 11) =1 + 0 + 0 + 1 + 0 = 0

P4 = XOR of bits (5, 6, 7, 12) = 1 + 0 + 0 + 0 = 1

PS = XOR of bits (9, 10, 11, 12) = 0 +1 + 0 + 0 = 1

Therefore the code generated is : - 0 0 1 1 1 0 0 1 0 1 0 0

Bit position 1 2 3 4 5 6 7 8 9 10 11 12

This new code is transmitted and at receiver side parity is checked over the same combination including parity bit. the four check bits are so generated as follows -

Cl = XOR of bits {l, 3, 5, 7, 9, 11}

C2 = XOR of bits {2, 3, 6, 7, 10, 11}

C4 = XOR of bits {4, 5, 6, 7, 12)

C8 = XOR of bits (8, 9, 10, 11, 12)

If the result C = C8 C4 C2 C 1 = 0000, it indicates there is no error in received data. However, if C is not equal to zero, then the binary number formed by the check bits C8 C4 C2 C1 gives the position of the erroneous bit. Consider the following three cases for error detection:-

Received data

Bit Position	1 2 3 4 5 6 7 8 9 10 11 12	C8	C4	C2	C1	Remarks
A	0 0 1 1 1 0 0 1 0 1 0 0	0	0	0	0	No error
B	1 0 1 1 1 0 0 1 0 1 0 1	0	0	0	1	Error in bit position 1
C	0 0 1 1 0 0 0 1 0 1 0 0	0	1	0	1	Error in bit position 5

The error can be corrected by complementing the bit in the position as dictated by C8 C4 C2 C1.

Hence, in this way hamming code can detect and correct one bit error only.

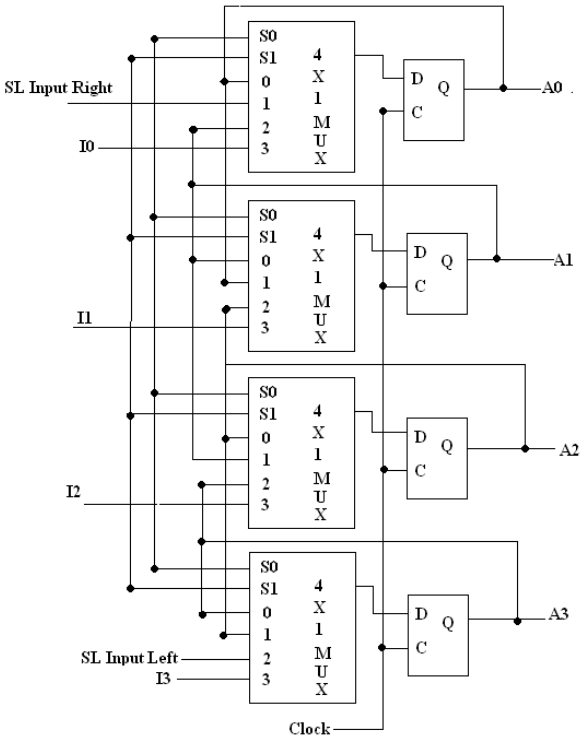
Q.5 With the help of a neat sketch, explain the working of a 4-bit universal shift register. (6)

Ans:

A register that can shift the data in both directions and has the capacity of parallel load is called Universal shift register. All the possible operation lo the register can be made with it. It can work as serial in parallel out, serial in serial out, parallel in parallel out and parallel in serial out fashion.

A 4- bit universal shift register is shown below:-

Each stage consists of a D- FF and a 4x1 Multiplexer. The two select in puts SO, Sl select one of the multiplexer data input for the DFF's. The selection lines control the mode of operation of the register according to the function table given below



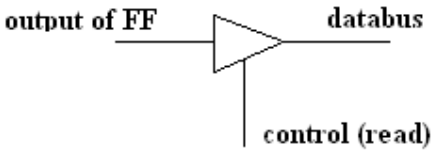
Function Table

Mode control		Register Operation
S1	SO	
0	0	No Change '-
0	1	Shift Right
1	0	Shift Left
1	1	Parallel Load

For Sl SO = 00, data input 0 of each multiplexer is selected. This condition forms a path from the output of each FF into the input of the same FF. The clock pulse transfers the binary value it held previously and no change of state occurs.

When S 1 SO = 01, this cases a shift right operation. The data available at serial input (Shift Right) enters into the First FF and the output of one FF is transferred to the other next to it causing the data stored, in the register to shift right by one bit with

each clock pulse. If we want that the data stored in register should rotate right, then the output A3 can be connected to the serial input (Shift Right).
When Sl SO = 10, shift left operation takes place. Here also in order to rotate the data in left direction by one bit with each clock pulse, the output AO can be connected to serial input (Shift Left) terminal.
When Sl SO = 11, the data available on the line 10, 11, 12, 13 get loaded in to the register with one clock pulse.
Further, the output of the register can also be available at AO, Al, A2 and A3 lines. These lines can be connected to with an tri- state buffer so as to read the data of the register only when the control input is 1.



Q.6 State the condition in which overflow occurs in case of addition & subtraction of two signed 2's complement number. How is it detected? (3)

Ans:
Overflow may occur when two n-bits number of same sign are added or when two n-bit I the time of the numbers of different sign are subtracted. If the result of addition / Subtraction is of (n + l) bit or out of permissible range than it is said to be overflow. For Example:-
+70-----01000110
+80----- 01010000
+150----- 10010110
Here the result of addition is in 8-bit only, but as +150 is out of the range that a number can be represented by 8-bit signed 2's Compliment form. So, overflow has occurred giving the wrong result. Now

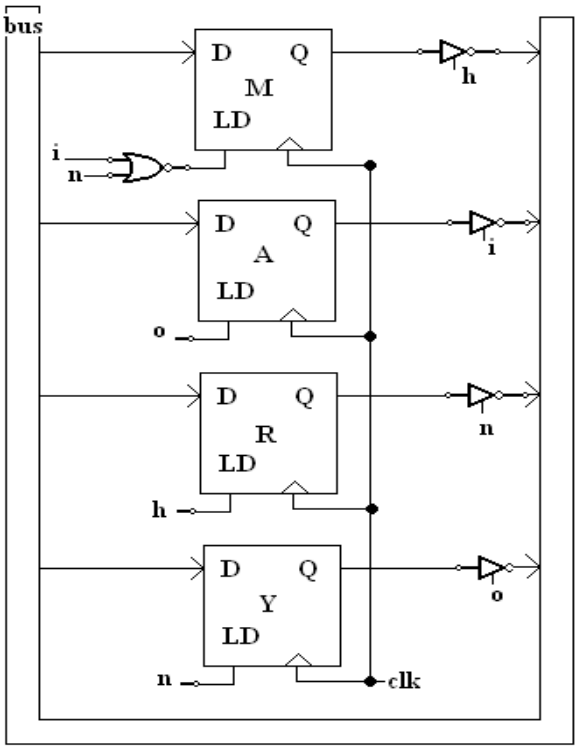
-70 2's 10111010
-————-»
-80 2's . 10110000
———>>
-150 101101010

Here the result of nine bits, clearly shows the over flow, as the largest negative number that can be represented by 8 signed 2'compliments number is -128 only.
An overflow condition can be detected by observing the carry in to the sign bit position and the carry out of sign bit position. If these two carries are not equal, then over flow is produced. If these two carries applied to an XOR gate, an overflow will be detected when the output of the gate is equal to 1.

Q.7 Implement the following RTL code, using common bus and tri-state buffers.
J : M←A
o: A←Y
b: R← M

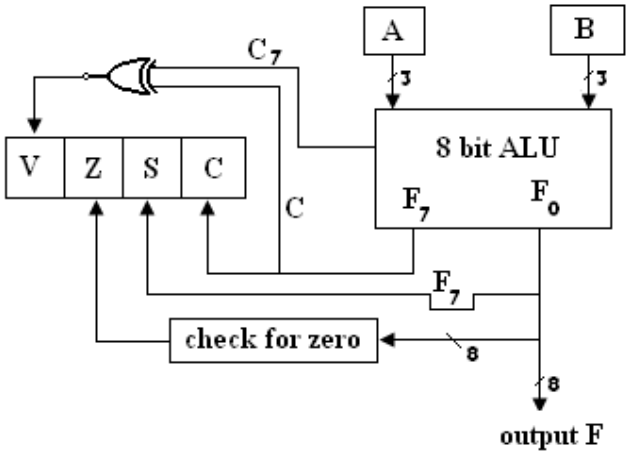
n: Y ←R, M ←R
Assume M, A, R and Y are to be one bit D – flip- flop. (6)

Ans:
The hardware realisation of R.TL behavior is shown below by using common bus and tri state buffer. All the FFs get the same clock pulse. Depending on the control signal of tri state buffer, the source FF is selected, and depending on control signals connected to 'LD' of FF's the destination FF is selected.



Q.8 What do you mean by program control instructions? With a neat diagram, explain how the status register containing overflow, zero, sign and carry flags works with the status of the accumulator content obtained from ALU. (3+4)

Ans:
Program control type instructions, when executed by the processor, may change the address value of the Program Counter and cause type flow of control to be altered. Program control instruction specifies conditions for altering the content of Program Counter. This causes break in the sequence of instruction execution. This Instruction also gives the capability for branching to different Program segments.
Examples - Branch., Jump, Skip, Call, Return etc.
Status bits are set or reset depending on the result of a logical or arithmetic manipulation of accumulator data. So status bits are called condition - code bits or flag bits. These status bits constitute status register.
The hardware realization of status register containing overflow, Zero, Sign, Carry flag is shown below –



(i) Bit 'c' (carry) is *set* to 1. if the end carry c8 is 1. It is cleared to zero if the carry is 0
(ii) Bit's' (sign) is set to one if' the highest order bit F7 is 1. It is set zero, if thebitF7=0
(iii) Bit 'z'(zero) is set to 1. if the output of 'ALU' contains all zeros. Otherwise it is set to zero.
(iv) Bit 'v' (overflow) is set to 1, if the Ex – OR of the last two carries i.e. c7 & c8 is equal to 1, and cleared to 0 otherwise. This is the condition for an overflow when negative numbers are in 2's complement form.

Q.9 What are interrupts? Explain different types of interrupts. (6)

Ans:
In a microprocessor system, there are three major types of interrupt that cause a break in the normal executing of a program. These are -
(i) External Interrupt: interrupt signal came from input-output devices connected external to processor. These interrupts depend on external conditions that are independent of the program being executed at the time. The examples that causes external Interrupt are - I/0 device requesting transfer of data, elapsed time of an event, power failure. time out mechanism for a program etc.
(ii) Internal Interrupt: Cause due to illegal or erroneous use of an instruction or data. Internal interrupts are also called traps. Internal interrupts are initiated due to some exceptional condition caused by the program itself rather than by an external event. If the program is rerun, the internal interrupts will occur in the same place each time. Example of cause of internal interrupts are - attempt to divide by zero, stack overflow, Invalid opcode, protection violation etc.
(iii) Software interrupts: It is initiated by executing an instruction. These are special call instructions that behaves like an interrupt rather than subroutine call. These can be used by the programmer to initiate an interrupt procedure at any designed point of the program. These interrupts arc usually used for switching to supervisor made from user mode.

Q.10 How stack is implemented in a general microprocessor system. (3)

Ans:

In a general microprocessor system, there is a special register known as stack pointer, which holds the address of the top of the stack. In some microprocessor, register stack is provided. In order to indicate the stack full condition and stack empty condition, two flags are used. These two flags are known as EMPTY flag & FULL flag. The empty flag is set when the stack is completely empty. Full flag is set only when all the stack locations are filled with data. Stack is essential for implementing subroutine call and interrupts.

Stacks operate in two principles

(1) LIFO i.e. Last in First Out

(2) FIFO i.e. first in first out.

These principles of operation depends on stack architecture. Most of general purpose processor use LIFO principle for their stack.

If the stack is organized in R/W memory, than the stack pointer is loaded with same address to initialize. The memory stack grow down word i.e. with each Push operations, stack pointer is decremented. The situation is just reverse on register stack.

Q.11 What are the advantages of assembly language? How is it different from high-level language? (6)

Ans:

Writing program for a computer consists of specifying, directly or indirectly, a sequence of machine instructions- The machine instruction stored in RAM of the computer is in binary format. This binary format is very difficult to use and to – troubleshoot. So programs are written by user by using English like symbols of the alpha-numeric character set, which is known as assemble language. The assembler converts these assembly language programs to binary form.

Advantages of assemble language program is it is easy to use. It is easy to troubleshoot, it is fast to execute than high level language program.

A programming language is defined by a set of rules. Users must conform to all format rules of the assembly language if it is to be translated correctly. Each microprocessor has its own assembly language format. The assembly language use predefined rules that specify the symbols that can be used & how they may be combined to form a line of code.

Some of the common rules are

(i) The label field may be empty or it may specify a symbolic address.

(ii) The instruction field specify a machine instruction or a Pseudo instructions.

(iii) The comment field may be empty or it may include a comment.

(iv) The symbolic address consists of up to four alphanumeric characters.

(v) Symbolic address in the label field is terminated by a comma so that it will be recognized as a label by the assembler.

(vi) The comment field is preceded by a slash foe assembler to recognize the beginning of a comment field.

Q.12 What is vertical micro code? State the design strategy of a vertical micro coded control unit. (6)

Ans:

In vertical microcode, the micro-operators are grouped in to fields. Each micro-operation is assigned a unique encoded value in this field. For example, 16 micro operations could be encoded using four bits, with each microoperation assigned with a unique binary field value from 0000 to 1111, These 16 micro-operation also include the 'NOP" i.e. No operation. Vertical micro instructions require fewer bits than their equivalent horizontal micro instructions, however the micro sequencer incorporate a decoder for each microoperation field to generate the actual micro-operation signals.

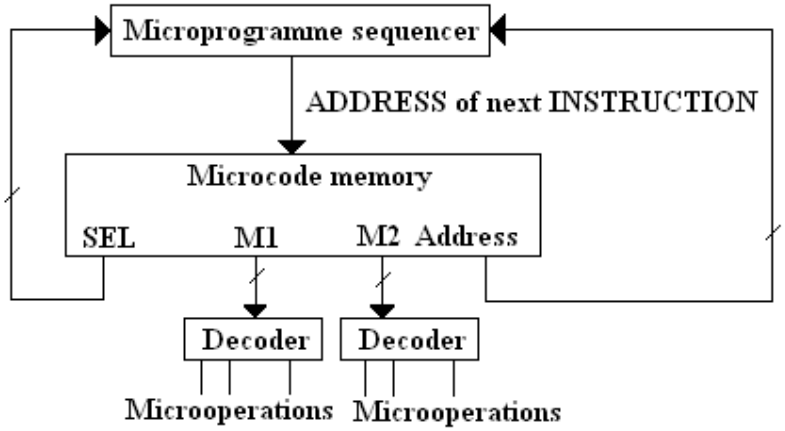
The design strategy used for vertical microcode is as follows:

Whenever two microoperations occur during the same state, assign them to different fields.

(i) Include NOP in each field if necessary.

(ii) Distribute the remaining micro-operations to make the best use of the micro-operation field bits.

(iii) Group together micro-operation that modify the same registers in the same field.



Q.13 What is a microprogram sequencer? With block diagram, explain the working of microprogram sequencer. (2+8)

Ans:

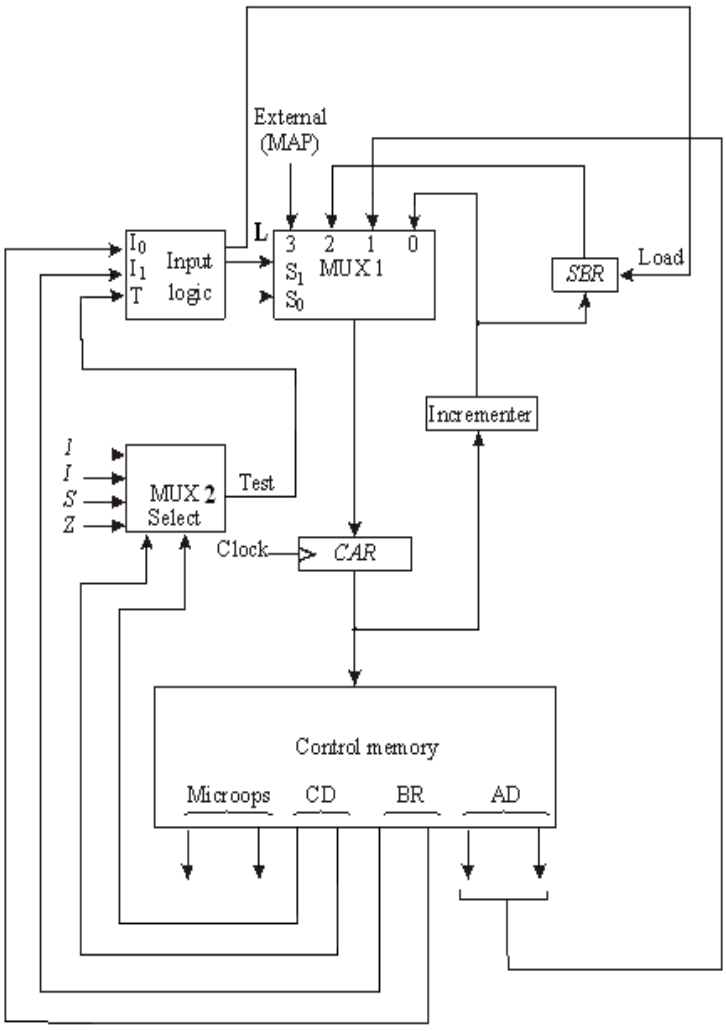
The function of control unit in a digital computer is to initiate sequences of micro-operations. When the control signals are generated by hardware using conventional logic design techniques, the control unit is said to be hard wired. Microprogramming is the second alterative. The control function that specifies microoperation is a binary variable. These binary control variables are stored in memory is called a microprogrammed control unit. A sequence of microinstructions constitutes a microprogram. Each machine instruction initiates a search of micro instruction in control memory. These microinstructions generates the microoperations to fetch the instruction from main memory, to evaluate the effective address, to execute the operation specified by the instruction and to return control to the fetch phase, in order to repeat the cycle for new instruction. Control memory address register specifies the address of the microinstruction to be read from memory. The micro instruction contains a control word that specifics one or more micro operations for the data processes. Once these operations are executed, the

control must determine the new address. The location of the next microinstruction may be the one next in sequence or it may be located somewhere else in the control memory. For this reason it is necessary to use some bits of the present micro instruction to control the generation of the address of the next micro instruction. The next address may also be a function of external input condition. The next address is computed by the circuit is called microprogram sequencer. The typical functions of a micro program sequencer are

(i) Incrementing the control address registers by one.
(ii) Loading into the control address register an address from control memory
(iii) Transferring an external address loading an initial address to start control operations.

The sequencer should also have a facility for subroutine call and return. This is shown in the following diagram:-

Fig. Microprogram sequencer for a control memory

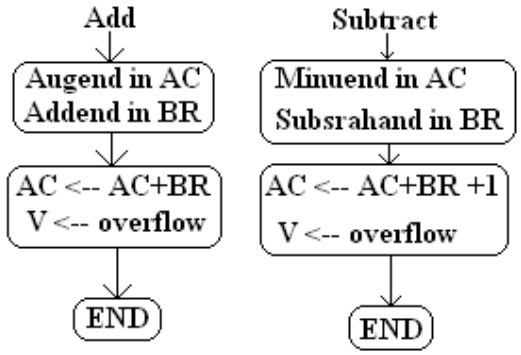


Q.14 Give the flow chart for add and subtract operation of two signed 2's complement data. Explain the logic of each operation. (4+6)

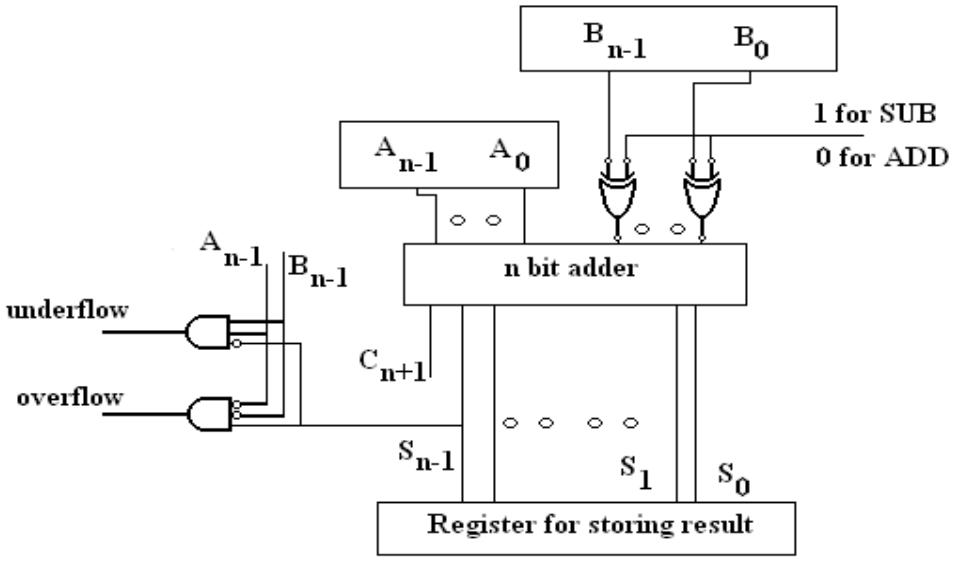
Ans:

In signed 2's complement representation, the left most bit of a binary number represents the sign bit. '0' for + ve & 1 for – ve. If the sign bit is 1, the entire number is represented in 2's complement form.

Flow chart:



Addition: - both the operands are added up along with the sign bit. A carry out of the sign bit position is discarded.
Subtraction:-Take 2's complement form of the subtracted including the sign bit and add it to the minuend including the sign bit. A carry out of the sign bit position is discarded.



Example: If we want to carry out -35 - (+40) in signed 2's complement representation then ,the binary representation of + 35 = 010001 and + 40 = 0101000 In signed 2's complement representation; both the operands are represented as

$-35 \rightarrow 0010011 \xrightarrow{2'\text{scompl}} 11011101$
$+40 \rightarrow 00101000 \xrightarrow{2'\text{scompl}} 11011000$
As we have to subtract (+40) from-35

So the result of substraction = Minuend in signed 2's complement from plus 2's | complement of subtrahend.

-35-(+40) = 11011101
11011000
10110101

The overflow carry is neglected. So the answer is (10110101) which is in signed 2's complement from, which is equal to (-75).

Q.15 Explain different methods used for establishing the priority of simultaneous interrupts. (6)

Ans:

To establish the priority of simultaneous interrupts can be done by software or hardware. Polling procedure is a software method. It is used for identifying the highest-priority source by executing a program. In this method there is one common branch address for all interrupts. The programme polls the interrupt sources in sequence. The order in which the sources are polled determines the priority of each interrupt. Thus the initial service routine for all interrupts consist of a program that tests the interrupts sources in sequence and to branch to one of many possible service routines.

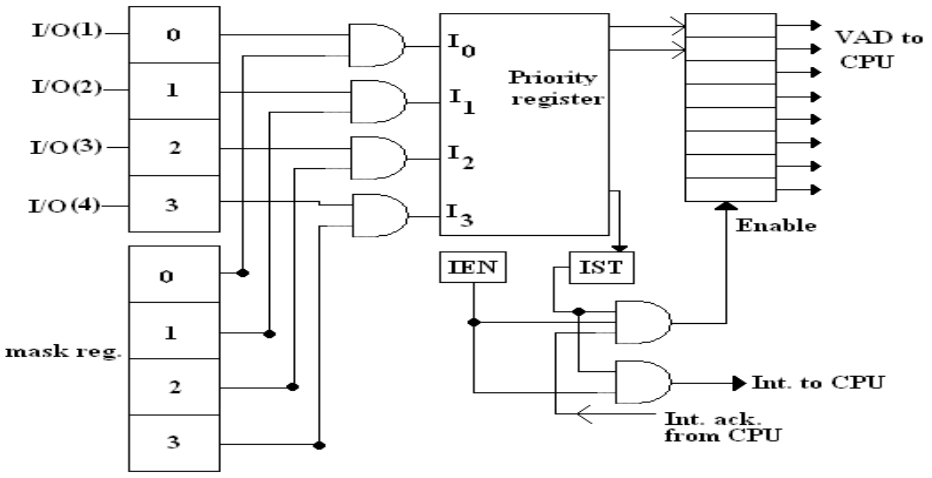
There are two hardware methods for establishing priority. These are

(i) Daisy- chaining priority and (ii) Parallel priority interrupts.

Daisy chaining is a hardware implementation of polling procedure, whereas parallel priority method uses a priority encoder and is the fastest method for establishing the priority of interrupt sources.

The parallel priority interrupt method uses a register whose bits are set separately by the interrupt signal from each device. Priority is established according to the position of the bus in the register. In addition to the interrupt register, the circuit may include a mask register whose purpose is to control the status of each interrupt request. The mask register can be programmed to disable lower priority interrupts while a higher priority device is being serviced. It can also provide a facility that allows a high priority device to interrupt the CPU, while a lower priority device is being serviced.

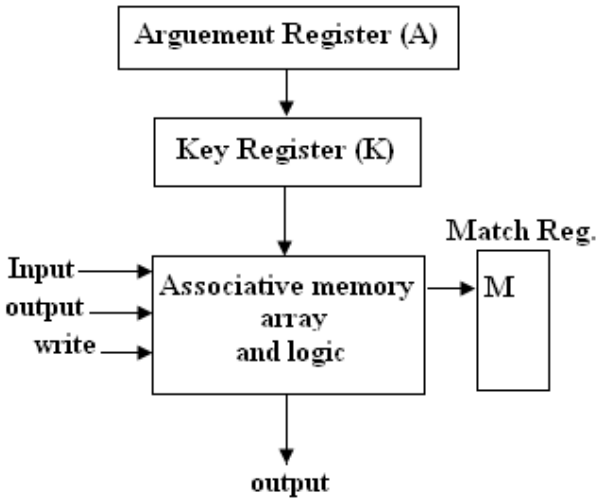
The priority logic for a system of four interrupt sources is shown below.



in the interrupt register, individual bits are set by internal I/O device requesting the service of CPU and is cleared by program instructions. The I/O devices are given with some priority value depending on their nature of devices and the services rendered by them. For example magnetic disk may get higher priority than a printer.
The mask register has same no. of bits as that of interrupt register. By means of program, it is possible to set or reset any bit of the mask register. If it is 1, then the associated interrupt is recognized, otherwise it is treated to be masked. Each interrupt bit along with its mask bit are applied to a AND gate lo produce four inputs to a priority encoder.
In this way the interrupts are recognized by CPU. The priority encoder output decides the vector address of the interrupt service subroutine. (ISR), which is to be loaded in to PC for execution of ISR during interrupt cycle. Another output of priority encoder sets an interrupt status flip-flop (IST FF). When an interrupt is recognized, the interrupt enable FF(IEN) can be set or cleared by the program to provide an overall control over the interrupt system. If $I_{EN} = 1$, then interrupt is recognised by CPU otherwise not. If IST= 1 & IEN = 1, then the interrupt signal goes to CPU, in return CPU sends interrupt acknowledgement signal, which enables the vector address register to place the vector address of ISR into program computer.

Q.16 Give the hardware organization of associative memory. Why associative memory is faster than other memories. Deduce the logic equation used to find the match in the associative memory. Explain how four-bit argument register is realized. (3+2+5)

Ans:
The hardware organization of the cell of one word in associative memory including the read and write logic is shown below:-



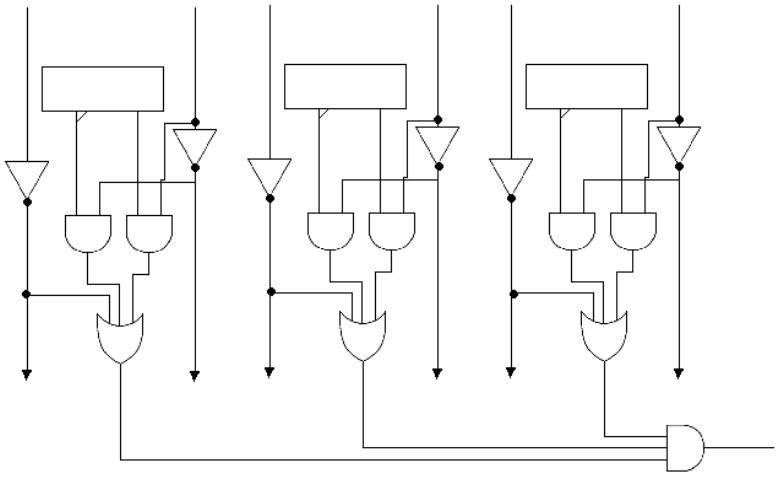
This consists of a memory array of logic for 'M' words with n-bits per word. The argument register A. The key register K, each has n-bits, one for each bit of a word. The match register 'M' has m bits, one for each memory word. Each word in the memory is compared in parallel with the content of the argument register. Words that match the bits of the argument register set a corresponding bit in the match register. After the matching process, those bits in the match register that have been set indicate

the fact that their corresponding words have been matched. As the identification and search of the data is done parallel by the hardware circuit so it is faster than other mapping logic.
Let A1, A2 .. ... An are n-bits of arguments register and K1, K2 „ . .Kn are n- bits of key register.
Let there be m-words in the memory, each of n- bits arrange in the matrix term having row 1 from 1 to m and column from 1 to n.
The output of comparison of each bit in a particular row i is given by xj.
Than Xj +kj' = if kj =1= 1if kj =0
The match logic register bits be M1, M2, Mn. As there is one bit in match register for each word .
Mi = (x1+k1') (x2 + k2')...(xn + kn')

$$Mi = \prod_{j=1}^{n}(Aj + Fij + Aj'Fij' + Kj')$$

As xj = AjFij+Aj'Fij'

The circuit for match for one word of associative memory is given below-



Q.17 Why page-table is required in a virtual memory system. Explain different ways of organizing a page table. (4+2)

Ans:
In any computer the address space is larger than memory space i.e. secondary memory is larger than the main memory, physically available to processor for execution of program. So programs and data are transferred to and from auxiliary memory and main memory based on demand imposed by the CPU. As the address of virtual memory is of larger bit then that of main memory, so mapping technique is required. To obtain the actual main memory address of the data from its virtual memory address. For this purpose a page table is required which holds the page number of virtual memory and the block number of the main memory. Further, each word of page table also has 'presence bit' to donate whether this page is presently available in main memory or not.
The different ways of organizing a page table are:
(i) In the R/W memory: it is called memory page table. But it is inefficient w.r.t. storage utilization and it required two main memory references to read a data,

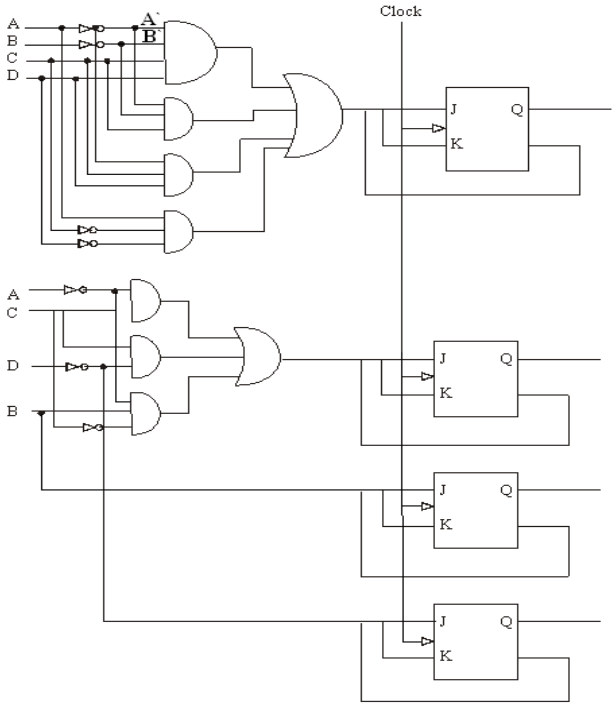
thus reducing the speed of execution of program.

(ii) By using associative logic: It is more efficient way to organize the page table, as it can be constructed with no. of words equal to no. of blocks in main memory.

Q. 18 Design a sequential circuit with JK flip-flop to satisfy the following state equations.

A(t+1) = A'B'CD + A'B'C + ACD +AC'D'
B(t+1) = A'C +CD' + + A'BC'
C(t+1) = B
D (t+1) = D'

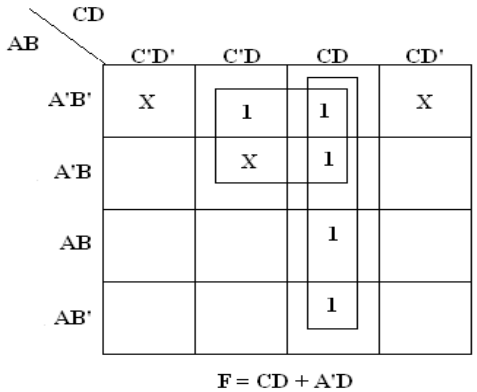
Ans.



Q.19 Simplify the Boolean function F together with don't care condition.

F (A,B, C, D) = m (1, 3, 7,11,15) + d(0,2,5)

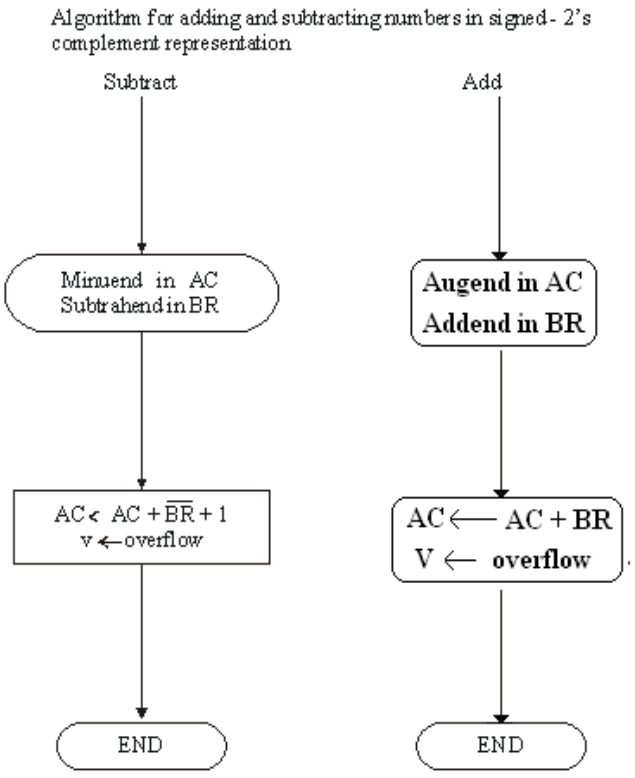
Ans.



F = CD + A'D

Q. 20 Explain the Adder-Subtractor with the help of 2's complement.

Ans.
The addition of two numbers in signed 2's complement form consists of adding the numbers with the sign bits treated the same as the other bits of the number. A carry-out of the sign-bits position is discarded. The subtraction consists of first taking the 2's complement of the subtrahend and then adding it to the minuend. When two numbers of a



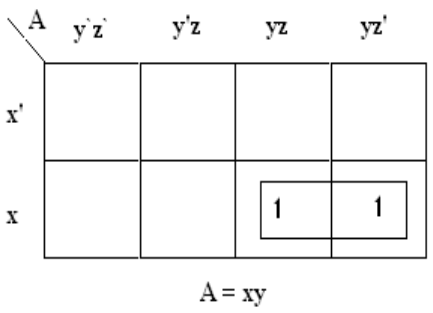
n digits each are added and the sum occupies n. + 1 digits then an overflow occurred. An overflow can be detected by inspecting the last two carries out of the addition. When the two carries are applied to an exclusive OR gate, the overflow is detected when, the output of the gate is equal to 1.
The sum is obtained by adding the contents of AC and BR (including their sign bits). The overflow bit V is set to 1 if the exclusive-OR of the last two carries is 1, and it is cleared to 0 otherwise. The subtraction operation is accomplished by adding the content of AC to the 2's complement of BR. Taking the 2's complement of BR has the effect of changing a positive number to negative, and vice versa. An overflow must be checked during this operation because the two numbers added could have the same sign. The programmer must realized that if an overflow occurs, there with be an erroneous result in the AC register.

Q.21 Design a combinational circuit using a ROM. The circuit accepts a 3-bit number and generates an output in binary number equal to the square of the input number.

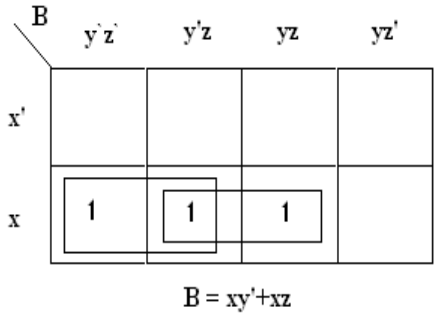
Ans.

The three bit number can represent 8 number of variables combination from 0-7. For this type of circuit, we need six bits of output. The truth table for the circuit is shown below:

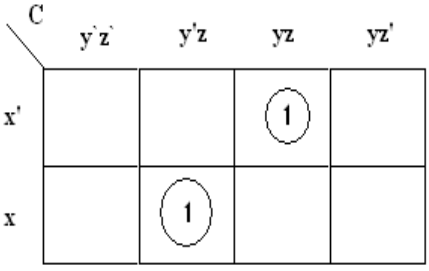
Input			Output					
X	Y	Z	A	B	C	D	E	F
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1
0	1	0	0	0	0	1	0	0
0	1	1	0	0	1	0	0	1
1	0	0	0	1	0	0	0	0
1	0	1	0	1	1	0	0	1
1	1	0	1	0	0	1	0	0
1	1	1	1	1	0	0	0	1



A = xy



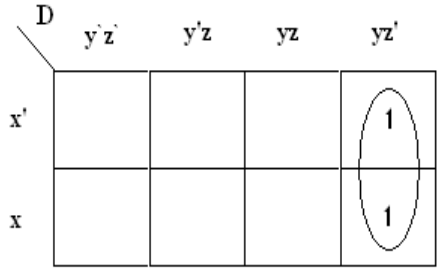
B = xy'+xz



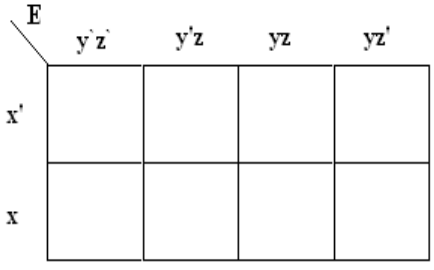
C = x'yz + xy'z

= z (x'y + xy')

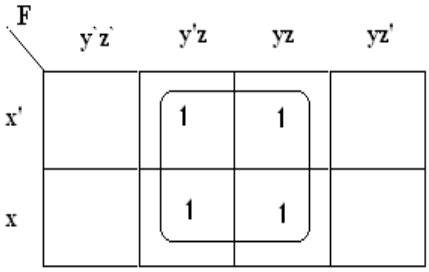
= z (x⊕y)



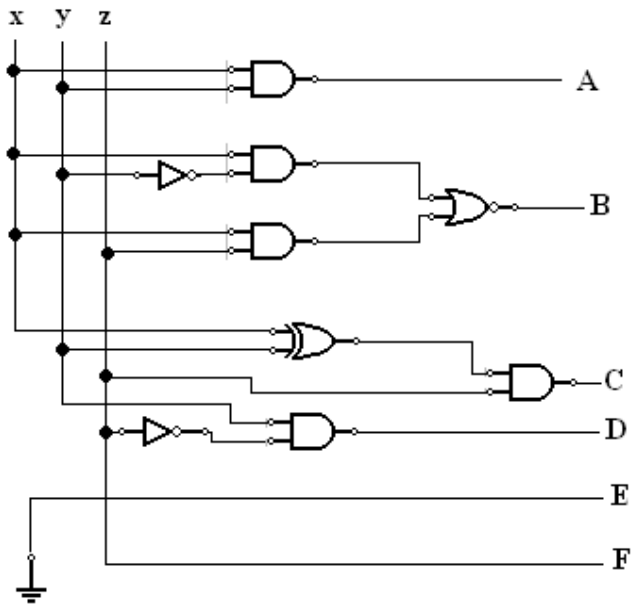
D = yz'



E = 0



F = Z



Q.22 Represent microinstructions for a microprogram of LD r_1, (r_2) instruction at control memory addresses a_j to a_{j+5}. How will be program counter increment f 9 for the next, instruction?

Ans.

(Z_{DR} = 1 if DR = 0 ; Z_{AC} = 1 if AC = 0)

INR (PC) = R' T1 + RT_2 + D_6 T_6 Z_{DR} + PB_9 (FGI) + PB_8 (FGO)
+ rB_4 (AC_{15}) ' rB_3 (AC_{15}) + rB_2 Z_{AC} +RB_1 E'

LD (PC) = D_4 T_4 + D_5 T_5

CLR (PC) = RT_1.

Q.23 Evaluate the arithmetic statement X = (A+B)*(C+D) using a general register computer with three address, two address and one address instruction format

Ans.

Three-Address Instructions

Computers with three-address instruction formats can use each address field to specify either a processor register or a memory operand. The program in assembly language that evaluates *X* = evaluates *X* = (A+B) * (C+D) is shown below, together with comments that explain the register transfer operation of each instruction.

ADD	R1, A, B	R1 M[A] + M[B]
ADD	R2, C, D	R2 M[C] + M[D]
MUL	X, R1, R2	M[X] R1 * R2.

It is assumed that the computer has two processor registers, *R*1and *R*2. The symbol *M*[A] denotes the operand at memory address symbolized by *A*.

Two-Address Instructions

Two-address instructions are the most common in commercial computers. Here again each address field can specify either a processor register or a memory word. The program to evaluate $X = (A + B) * (C + D)$ is as follows:

MOV	R1, A	R1 <-M[A]
ADD	R1, B	R1 <-R1 + M[B]
MOV	R2, C	R2 <-M[C]
ADD	R2, D	R2<- R2 + M[D]
MUL	R1, R2	R1 <-R1 * R2
MOV	X, R1	M[X] <- R1

The MOV instruction moves or transfers the operands to and form memory and processor registers.

One-Address Instructions

One-address instructions use an implied accumulator (*AC*) register for all data manipulation. For multiplication and division there is a need for a second register. However, here we will neglect the second register and assume that the *AC* contains the result of all operations. The program to evaluate

$X = (A + B) * (C + D)$ is

LOAD	A	AC <-M[A}
ADD	B	AC<-A[C] + M[B]
STORE	T	M[T] <-AC
LOAD	C	AC<-M[C]
ADD	D	AC<-AC+M[D]
MUL	T	AC<-AC * M[T]
STORE	X	M[X] <-AC

Q. 24 Write an assembly language program to convert a digital string into respective exactly opposite value.

Ans.

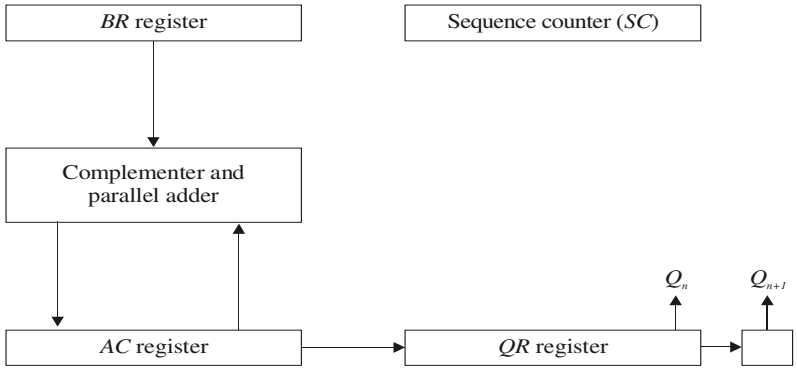
Address	Machine Code	Labels	Memories	Operands	Comments
2000	11,01,26		LXI	$D_1$2500H	Memory location for storing result.
2003	21,0025		LXI	$H_1$2500H	Address for count in H-L pair.
2006	46		MOV	B_1M	Ist number is accumulator
2007	21,0025	LOOP	INX	H	Decrement Court.
2008	DA, 14,20		JC	AHEAD	Yes, next digit in accumulator go to AHEAD
2009	00	AHEAD	DCR	C	Decrement Court.
2010	12		STAX	D	Store the result.

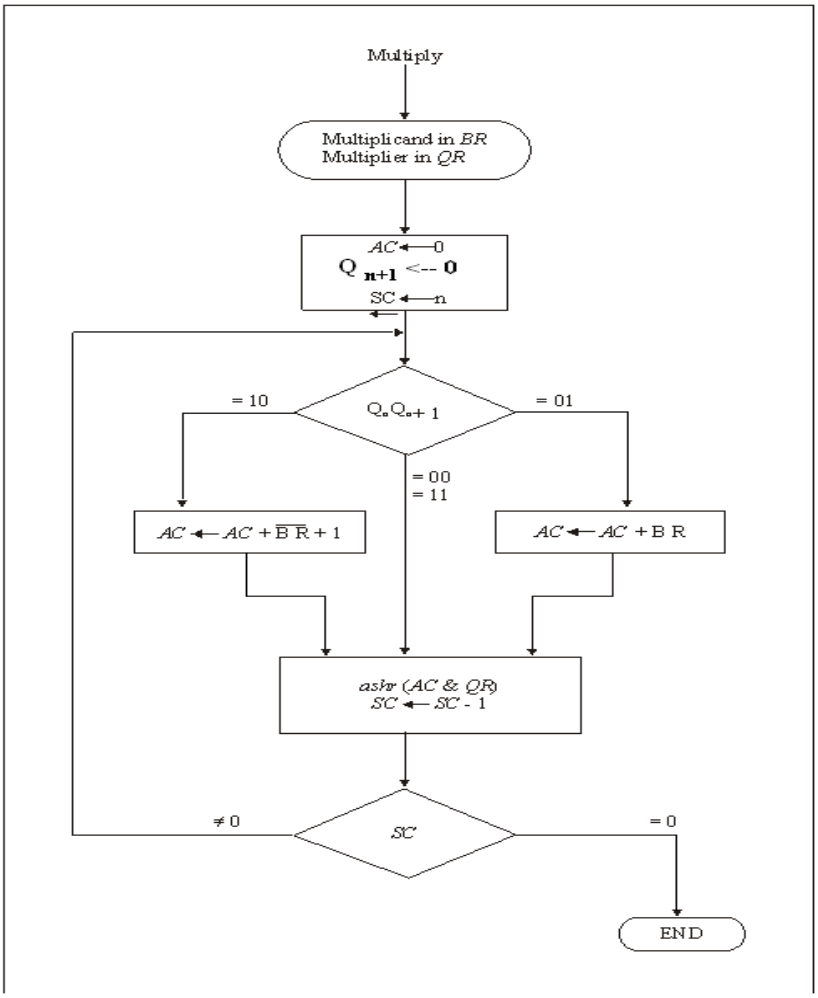
Q.25 Explain Booth's multiplications algorithm through an example. Give an example of multiplicant and multiplier for which this algorithm takes the maximum time.

Ans.

The hardware implementation of Booth algorithm requires the register configuration shown. The sign bits are not separated from the rest of the registers. Registers *A, B,* and *Q,* as *AC*, *BR,* and *QR,* respectively. Q_n designates the least significant bit of the multiplier in register QR. An extra flip-flop Q_{n+1} is appended to *QR* to facilitate a double bit inspection of the multiplier. The flowchart for Booth algorithm is shown.

Hardware for Booth algorithm





Example: Refer table 10-3 from page 348, Morris mano (3rd Edition)

Q. 26 Using 8-bit 2's complement representation of negative numbers, perform the following computations: .

(i) - 35 + (-11) (ii) 19 - (- 4)

Ans.

35 = 00100011
-35 = 11011100
11011101
11 = 00001011
-11 = 11110100
= 11110101
-35+ (-11) = 11011101
+ 11110101

111010010

(ii) 19 = 0001 0011
4 = 00000100
-4 = 11111100
19 - (- 4) = 19 + 4
00010011
00000100

00010111

Q. 27 Consider a cache (M_1) and memory (M_2) hierarchy with the following characteristics:

M_1 : 16 K words, 50 ns access time

M_2 : 1 M words, 400 ns access time

Assume 8 words cache blocks and a set size of 256 words with set associative mapping.

(i)Show the mapping between M_2 and M_1.

(ii)Calculate the Effective Memory Access time with a cache hit ratio of h = .95.

Ans.

(i) Main Memory = 1M words.
$= 2^{20}$ words.
Block size = 8 words.

main memory = $\frac{2^{20}}{8} = 2^{17}$ blocks

Cache memory = 16 k words.

Therefore Cache memory = $\frac{16K}{8} = \frac{2^{14}}{2^3} = 2^{11}$ blocks.

Set size = 265 words.

$= \frac{256}{8} = \frac{2^8}{2^3} = 2^5$ blocks

Tag = 17-11+5 = 11 bits

Set = 11-5 = 6 bits

word = 3 bits.

(ii) Given, t_c = 50ns tm = 400 ns

h= 0.95

Memory access time = $ht_c + (1- h)(t_c + t_m)$

$= 0.95 \times 50 + (1-0.95)(50+400)$

$= 47.5 + 22.5$

$= 70$ns

Q. 28 Write short notes on the following:

(i) Flip-Flops.

(ii) Multiplexer.

Ans.

(i) **Flip-Flops**:-

Flip-Flops is another name for a bistable multivibrator. A flip-flop is capable of storing 1 bit of binary data. It has two stable states- 'one' and 'zero'. The output stays low or high, to change it, the circuit must be drived by an input called trigger. Until the trigger arrives, the output voltage remains low or high indefinitely.

Edge Triggered Flip-flops:- An edge triggered flip-flop responds only during the brief instant the clock switches from one voltage level to another. When the triggering occurs on the positive going edge of the clock, it is called positive-edge triggering. Sometimes, triggering on the negative edge is better suited to the application. This means the trailing edge of the clock activates the gates, allowing data to be recognized. This is called negative-edge triggering.

Preset and Clear :- When power is first applied, flip-flops come up in random states. To get some computers started, an operator has to push a reset button. This sends a reset or CLEAR signal to all flip-flops. Also, it is necessary in some digital system to PRESET (synonymous with set) certain flip-flops.

In a clocked flip-flop PRESET and CLEAR inputs are called asynchronous, because they activate the flip-flops independently of the clock.

Types of Flip-flops

1)R-S. Flip-Flop

2)Clocked R-S Flip-FLop

3)D Flip-FLop.

4)J-K Flip-Flop

5)Master-Slave Flip-Flop

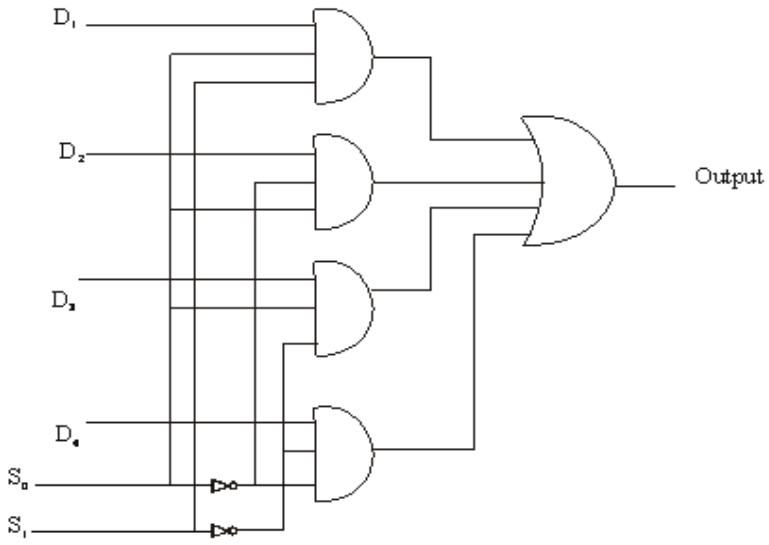
6)T-Flip-Flop

(ii) **Multiplexer**

The multiplexer (MUX) is a combinational logic circuit that selects binary information from one of the multiple input lines (D_{n-1}.....D_1, D_0) and directs it to an output line according to a received select code ($S = S_{n-1}$..... S_1 S_0) and directs it. A block diagram of a four input multiplexer is shown in Fig. it's truth table given below.

Truth Table		
S_0	S_1	O/P
0	0	D_4
0	1	D_a
1	0	D_2
1	1	D_1

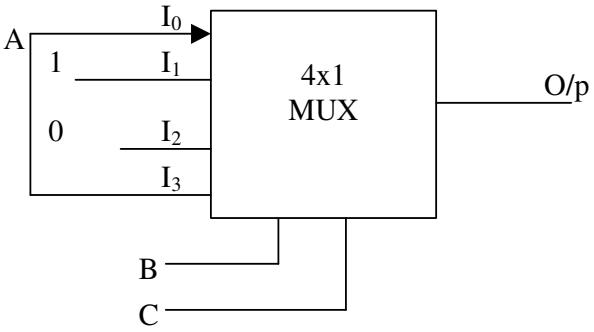
Output = $D_4 S_0 S_1 + D_3 S_0 S_1 + D_2 S_0 S_1 + D_1 S_0 S_1$



Q.29 Implement the following by using 4:1 multiplexer
$P = \Pi(M_0, M_1, M_5, M_7)$.

Ans.

	I_0	I_1	I_2	I_3
A´	0	1	2	3
A	4	5	6	7
	A´	1	0	A



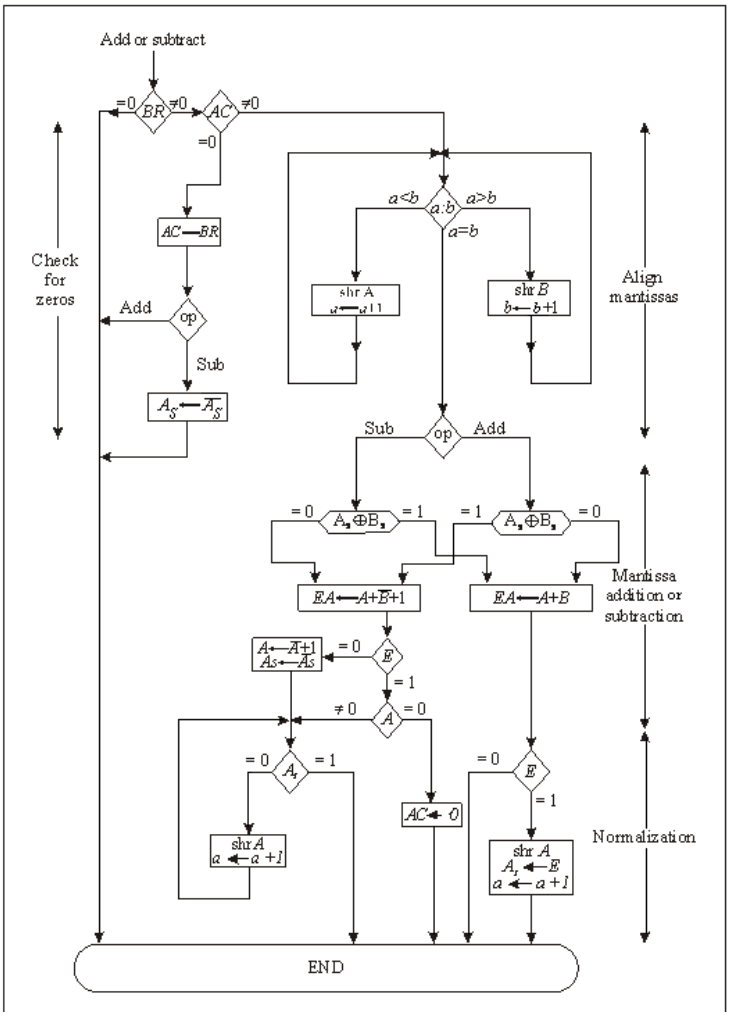
B and C are the selection lines.

Q. 30 Explain with neat flow chart the addition and subtraction of floating point numbers.

Ans:

In addition and subtraction the two floating point operands are in AC and BR. The sum or difference is formed in AC. The algorithm can be divided into four parts.

Flow Chart



(1) Check for zeros.
(2) Align the mantissas
(3) Add or subtract the mantissas
(4) Normalize the result.

The flowchart for adding or subtracting two floating point binary numbers is shown in fig. If BR is equal to zero, the operation is terminated, with the value in the AC being the result. If AC is equal to zero, we transfer the content of BR into AC and also complement its sign if the numbers are to be subtracted. If neither number is equal to zero, we proceed to align the mantissas.

The magnitude comparator attached to exponents a and b provides three outputs that indicate their relative magnitude. If the two exponents are equal, then perform the

arithmetic operation. If the exponents are not equal, the mantissa having the smaller exponent is shifted to the right and its exponent incremented. This process repeated until the two exponents are equal.

Q.31 Multiply $(-7)_{10}$ with $(3)_{10}$ by using Booth's multiplication. Give the flow table of the multiplication.

Ans.
Binary equivalent of 7 = 0111, -7 = 1001
Binary equivalent of 3 = 0011.

Q_n	Q_{n+1}	BR=1001 $\overline{BR}$+1=0111	AC	QR	Q_{n+1}	SE
		Initial	0000	0011	0	100
1	0	Subtract BR	0111 0111			
		ashr	0011	1001	1	011
1	1	ashr	0001	1100	1	010
0	1	Add BR	1001 1010			
		ashr	1101	0110	0	001
0	0	ashr	1110	1011	0	000
Final product = 11101011						

Q.32 Design a hardware circuit by using common bus architecture to implement the following Register Transfer Languages.

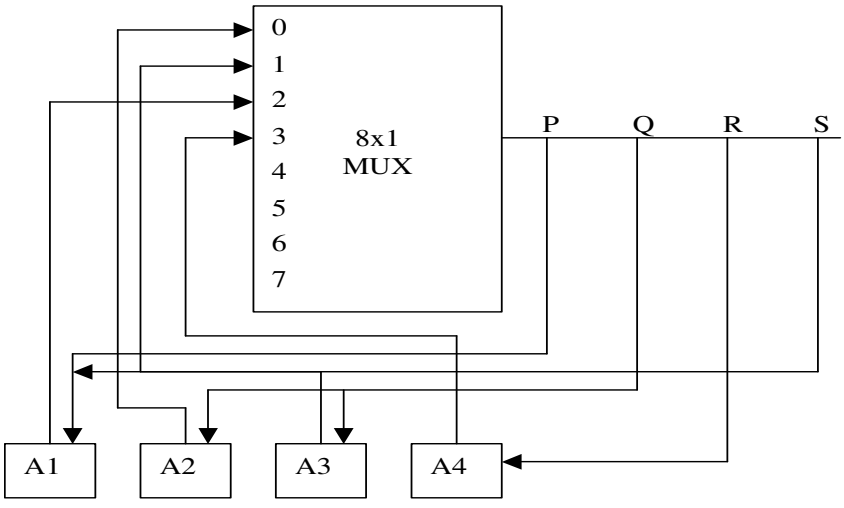
P: $A_1 \leftarrow A_2$

Q: $A_2 \leftarrow A_3$

R: $A_4 \leftarrow A_1$

S: $A_3 \leftarrow A_4$, $A_1 \leftarrow A_4$

Where A_1, A_2, A_3, A_4 are one bit register



Input			Output
S_2	S_1	S_1	
0	0	0	P
0	0	1	Q
0	1	0	R
0 1	1 0	1 0	S_1 S_2]s
X	X	X	X
X	X	X	X
X	X	X	X
X	X	X	X

Q.33 Explain hardware polling method for data transfer.

Ans.

In a bus system that uses polling, the bus grant signal is replaced by a set of lines called poll lines which are connected to all units. These lines are used by the bus controller to define an address for each device connected to the bus. The bus controller sequences through the addresses in a prescribed manner. When a processor that requires access recognizes its address, it activates the bus busy line and then accesses the bus. After number of bus cycles, the polling process continues by choosing a different processor. The polling sequence is normally programmable, and as a result, the selection priority can be altered under program control.

There are so many ways to boot alternative OSs to common PCs these days. There are many ways to run multiple operating systems on a single piece of hardware. Below we count the pros & cons of the three most popular methods: re-partitioning, emulation and virtualization.

1. Partitioning

With the re-partitioning method the user must manually re-partition his hard drive and then install the OSs one after the other on the right partition and then use a boot manager to boot between OSs.

Pros:

- OSs run full speed
- Full access to the hardware
- Partition resizing is possible (depending on the file system used)

Cons:

- Manually partitioning can be tricky for newbies
- Frustrating if you 'lose' your boot manager after an OS update
- Requires a full reboot to run another OS

2. Virtualization

Virtualization is the new kid on the block and it's gaining ground very fast. To the eyes of a simple user it looks a lot like straight emulation, but in reality it's not. The virtualizer "shares" more hardware resources with the host OS than an emulator does.

Pros:

- Slower than the re-partitioning method but much faster than emulation
- Support for all host hardware, including 3D support
- Virtual clustering made-easy

Cons:
- Requires enough RAM
- Only runs on the same architecture as the host OS

3. Emulation

Emulators will completely emulate the target CPU and hardware (e.g. sound cards, graphics cards, etc). Emulators are the "old way" of running multiple OSs on a single computer. Emulation on PCs these days is only good for non-OS usages (e.g. game consoles, embedded systems) or specific OS/CPU development purposes.

Pros:
- Best solution for embedded/OS development
- Doesn't interfere with the underlying host OS
- Can be ported to any architecture

Cons:
- Can be very slow
- No 3D or other exotic PC hardware support
- Requires enough RAM

Being a traditional geek chick myself, I still prefer the manual re - partioning method for my PCs (I like the clean nature of it), but for a MacTel I would much prefer Boot Camp's special portioning scheme (if Vista, Linux are supported properly — otherwise, Virtualization is my next best option on MacTels). Tell us what's your preferred method is below

Q. 34 Explain with an example, how effective address is calculated in different types of addressing modes.

Ans.

To explain the difference between the various modes, the two word instruction at address 200 and 201 is a "load to AC" instruction with an address field equal to 500. The first word of the instruction specifies the operation code and mode, and the second word specifies the address part. PC has the value 200 for fetching this instruction. The content of processor register R1 is 400, and the content of an index register XR is 100. AC receives the operand after the instruction is executed. The figure lists a few pertinent addresses and shows the memory content at each of these addresses for each possible mode. We calculate the effective address and the operand that must be loaded into AC. In the direct address mode the effective address is the address part of the instruction 500 and the operand to be loaded into AC is 800. In the immediate mode the second word of the instruction is taken as the operand rather than an address, so 500 is loaded into AC. In the indirect mode the effective address is stored in memory at address 500. Therefore the effective address is 800 and the operand is 300. In the Index mode the effective address is XR + 500 = 100 + 500 = 600 and the operand is 900. In the register mode the operand is in R1 and 400 is loaded into AC.

The Autoincrement mode is the same as the register indirect mode except that R1 is incremented to 401 after the execution of the instruction. The Autodecrement mode decrements R1 to 399 prior to the execution of the instruction. In the relative mode the effective address is 500+202=702 and the operand is 325.

In the register indirect made the effective address is 400, equal to the content of R1 and the operand loaded into AC is 700.

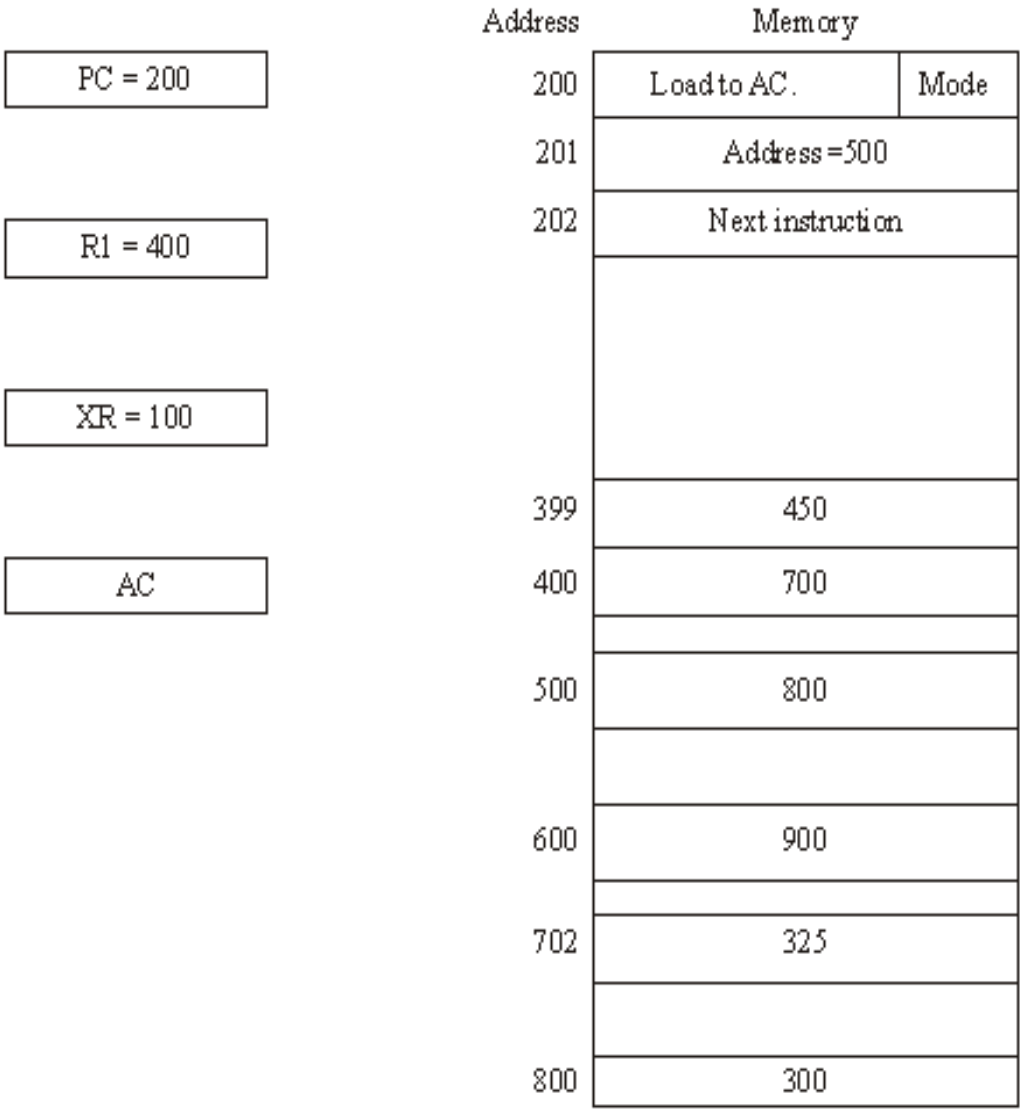


Fig. Numerical example for addressing modes

Q.35 How an interrupt is recognized? Explain the interrupt cycle.

Ans.

In the parallel priority interrupt method uses a register whose bits are get separately by the interrupt signal from each device. Priority is established according to the position of the bits in the register. In the interrupt register the circuit may include a mask register whose purpose is to control the status of each interrupt request. The mask register can be programmed to disable low priority interrupts while a higher priority device is being carried.

The mask register has the same number of bits as the interrupt register. Each interrupt bit and its corresponding mask bit are applied to an AND get to produce the four inputs to a priority encoder. In this way an interrupt is recognized only if its corresponding mask bit is get to 1 by the program.

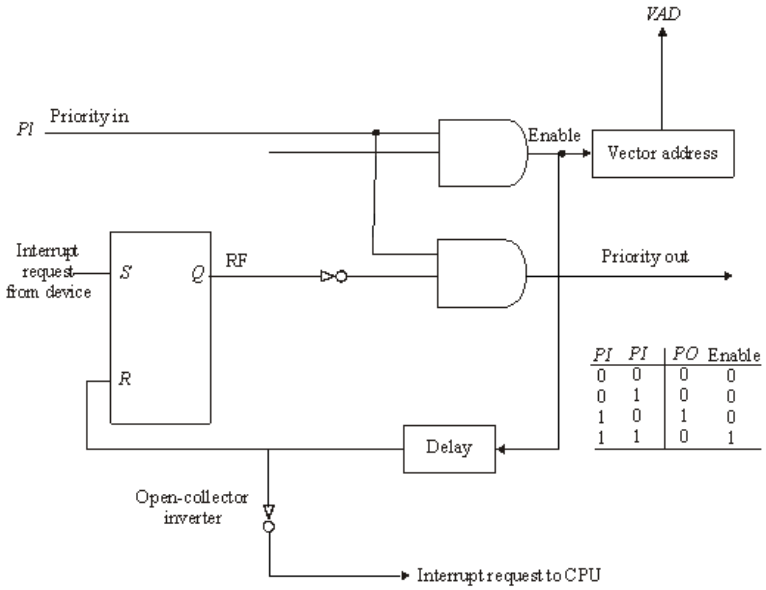


Fig. One stage of the daisy-chain priority arrangement

Interrupt Cycle:-

The interrupt enable flip-flop *IEN* shown it can be set or cleared by program instructions. When *IEN* is cleared, the interrupt request coming from *IST* is neglected by the CPU. The program-controlled *IEN* bit allows the programmer to choose whether to use the interrupt facility. If an instruction to clear *IEN* has been inserted in the program, it means that the user does not want his program to be interrupted. An instruction to set *IEN* indicates that the interrupt facility will be used while the current program is running. Most computers include internal hardware that clears *IEN* to 0 every time an interrupt is acknowledged by the processor. At the end of each instruction cycle the CPU checks *IEN* and the interrupt signal from *IST*. If either is equal to 0, control continues with the next instruction. If both *IEN* and *IST* are equal to 1, the CPU goes to an interrupt cycle. During the interrupt cycle the CPU performs the following sequence of micro-operations:

SP	*SP* –1	Decrement stack pointer
M[*SP*]	*PC*	Push *PC* into stack
INTACK	1	Enable interrupt acknowledge
PC	*VAD*	Transfer vector address to *PC*
IEN	0	Disable further interrupts

Go to fetch next instruction.

The CPU pushes the return address from *PC* into the stack. It then acknowledges the interrupt by enabling the *INTACK* line. The priority interrupt unit responds by placing a unique interrupt vector into the CPU data bus. The CPU transfers the vector address into *PC* and clears *IEN* prior to going to the next fetch phase. The instruction read from memory during the next fetch phase will be the one located at the vector address.

Q. 36 Compare assembly language with high level language. Write a program using assembly language of 8085 microprocessor to check whether a given number is

odd or even. If the given number is even then display '1' on its SOD line. Give the flow chart also.

Ans.

High Level Language	Assembly Language
(1) Programs developed in High level language are most understandable	(1) Program are less under-stand able than high level language but more than machine language
(2) Program are portable	(2) Not portable, portable to the processor of same architecture only
(3) Debugging is easier	(3) Debugging is more complex
(4) Most suited for software development	(4) Not good for large programs
(5) Program are not machine dependent	(5) Program are machine dependent
(6) Provides flexible construct for program development	(6) Does not provide flexible construct for development
(7) Programs are translated using compiler and/or interpreter to generate object code	(7)Uses assembler to generate object code

```
DATA SEGMENT
NUMBER DB 11
EVE DB      ' ENTERED NUMBER IS EVEN'
ODD DB      'ENTERED NUMBER IS ODD'

 DATA ENDS
 CODE SEGMENT
 ASSUME CS: CODE, DS: DATA
                START : MOV    DX, DATA
                        MOV    DS, DATA
                        MOV    AL, NUMBER
                        SHR    AL, 1
                        JL     LABEL 1
                 MOV    DX, OFFSET EVE
                      INC      LABEL 2

   LABEL1 :   MOV DX, OFFSET ODD
   LABEL 2 :  MOV AH, 09H
              INT   21H
              MOV  AH, 4CH
              INT    21H
   CODE       ENDS
   END        START
```

Q.37 Compare horizontal microcode with vertical microcode. State the advantage of micro programmed control unit. **(6)**

Ans.

Horizontal Microcode	**Vertical Micro-code**
(1) Control signal directly in micro-code (2) All control signals always there. (3) Lots of signals many bits in micro-instruction	(1) Each action encoded density. (2) Actions need to be decoded to signal at execution time. (3) Takes less space but may be slower.

Advantage of micro programmed control unit is that once the hardware configuration is established. There should be no need for further hardware or wiring changes. If establish a different control sequence for the system, is specify a different set of micro instructions for control memory.

Q. 38 Explain in detail the different mappings used for cache memory. Compare them.

Ans.
Three types of mapping procedures used for cache memory:
(i) Associative mapping
(ii) Direct mapping
(iii) Set-associative mapping

(i) **Associative mapping:-**

The fastest and most flexible cache organization uses an associative memory. The organization is illustrated. The associative memory stores both the address and content (data) of the memory word. This permits any location in cache to store any word from main memory. The diagram shows three words presently stored in the cache. The address value of 15 bits is shown as a five-digit octal number and its corresponding 12-bit word is shown as a four-digit octal number and its corresponding 12-bit word is shown as a four-digit octal number. A CPU address of 15 bits is placed in the argument register and the associative memory is searched for a matching address. If the address is found, the corresponding 12-bit data is read and sent to the CPU. If no match occurs, the main memory is accessed for the word. The address-data pair is then transferred to the associative cache memory. If the cache is full, an address-data pair must be displaced to make room for a pair that is needed and not presently in the cache. The decision as to what pair is replaced is determined from the replacement algorithm that the designer chooses for the cache. A simple procedure is to replace cells of the cache is round-robin order whenever a

new word is requested from main memory. This constitutes a first-in first-out (FIFO) replacement policy.

Fig. Associative mapping cache (all numbers in octal)

CPU address (15 bits)

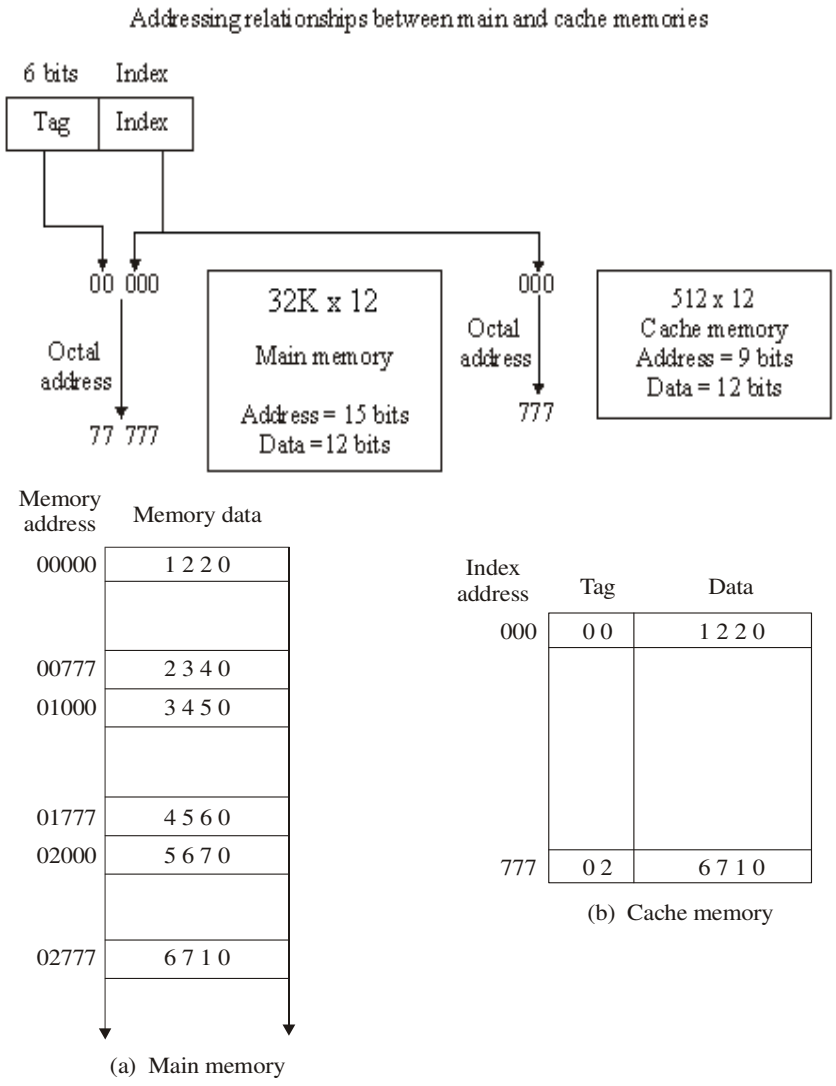
↓

Argument register

← Address →	← Data →
01000	3450
02777	6710
22345	1234

(ii) **Direct Mapping** :-

Associative memories are expensive compared to random-access memories because of the added logic associated with each cell. The possibility of using a random-access memory for the cache is investigated. The CPU address of 15 bits is divided into two fields. The nine least significant bits constitute the *index* field and the remaining six bits form the *tag* field. The figure shows that main memory needs an address that includes both the tag and the index bits. The number of bits in the index field is equal to the number of address bits required to access the cache memory. In the general case, there are 2^k words in cache memory and 2^n words in main memory. The n bit memory address is divided into two fields: k bits for the index field and the *n*-k bits for the tag field. The direct mapping cache organization uses the *n*-k bits for the tag field. The direct mapping cache organization uses the *n*-bit address to access the main memory and the k-bit index to access the cache. The internal organization of the words in the cache memory is as shown. Each word in cache consists of the data word and its associated tag. When a new word is first brought into the cache, the tag bits are stored alongside the data bits. When the CPU generates a memory request, the index field is used for the address to access that cache. The tag field of the CPU address is compared with the tag in the word read from the cache. If the two tags match, there is a hit and the desired data word is in cache. If there is no match, there is a miss and the required word is read from main memory. It is then stored in the cache together with the new tag, replacing the previous value. The disadvantage of direct mapping is that the hit ratio can drop considerably if two or more words whose addresses have the same index but different tags are accessed repeatedly. However, this possibility is minimized by the fact that such words are relatively far apart in the address range.



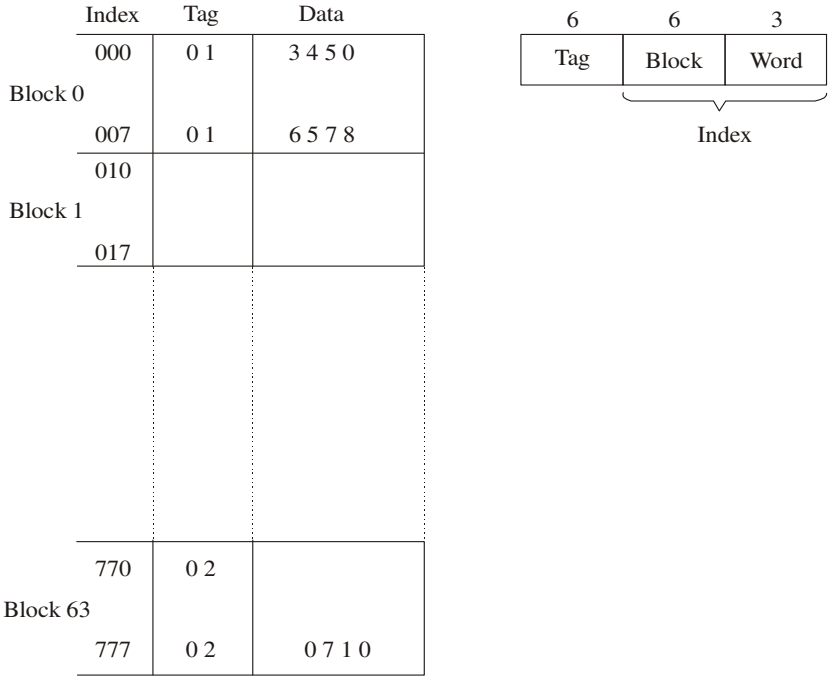
Direct mapping cache organization

To see how the direct-mapping organization operates, consider the numerical example shown. The word at address zero is presently stored in the cache (index = 000, tag = 00, data = 1220). Suppose that the CPU now wants to access the word at address 02000. The index address is 000, so it is used to access the cache. The two tags are then compared. The cache tag is 00 but the address tag is 02, which does not produce a match. Therefore, the main memory is accessed and the data word 5670 is transferred to the CPU. The cache word at index address 000 is then replaced with a tag of 02 and data of 5670.

The direct-mapping example just described uses a block size of one word. The same organization but using a block size of 8 words is shown.

The index field is now divided into two parts: the block field and the word field. In a 512-word cache there are 64 blocks of 8 words cache, since 64 x 8 = 512. The block number is specified with a 6-bit field and the word within the block is specified with a 3-bit field. The tag field stored within the cache is common to all eight words of the same block. Every time a miss occurs, an entire block of eight words must be transferred from main memory to cache memory. Although this takes extra

time, the hit ratio will most likely improve with a larger block size because of the sequential nature of computer programs.



Direct mapping cache with block size of 8 words

(ii) **Set-Associative Mapping** :-

It was mentioned previously that the disadvantage of direct mapping is that two words with the same index in their address but with different tag values cannot reside in cache memory at the same time. A third type of cache organization, called set-associative mapping, is an improvement over the direct-mapping organization in that each word of cache can stored two or more words of memory under the same index address. Each data word is stored together with its tag and the number of tag-data items in one word of cache is said to form a set. An example of a set-associative cache organization for a set size of two is shown. Each index address refers to two data words and their associated tags. Each tag requires six bits and each data words has 12 bits, so the word length is 2(6+12) = 36 bits. An index address of nine bits can accommodate 512 words. Thus the size of cache memory is 512 x 36. It can accommodate 1024 words of main memory since each word of cache contains two data words. In general, a set-associative cache of set size k will accommodate *k* words of main memory in each word of cache.

The octal numbers listed are with reference to the main memory contents illustrated in the fig. The words stored at addresses 01000 and 02000 of main memory are stored in cache memory at index address 000. Similarly, the words at addresses *02777* and *00777* are stored in cache at index address *777*. When the CPU generates a memory request, the index value of the address is used to access the cache. The tag field of the CPU address is then compared with both tags in the cache to determine if a match occurs. The comparison logic is done by an associative search of the tags in the set similar to an associative memory search: thus the name "set-associative." The

Index	Tag	Data	Tag	Data
000	0 1	3 4 5 0	0 2	5 6 7 0
777	0 2	6 7 1 0	0 0	2 3 4 0

hit ratio will improve as the set size increases because more words with the same index but different tags can reside in cache. However, an increase in the set size increases the number of bits in words of cache and requires more complex comparison logic. When a miss occurs in a set-associative cache and the set if full, it is necessary to replace one of the tag-data items with a new value. The most common replacement algorithms used are: random replacement, first-in, first-out (FIFO), and least recently used (LRU). With the random replacement policy the control chooses one tag-data item for replacement at random. The FIFO procedure selects for replacement the item that has been in the set the longest. The LRU algorithm selects for replacement the item that has been least recently used by the CPU. Both FIFO and LRU can be implemented by adding a few extra bits in each word of cache.

Q.39 State how different policies of writing into cache are implemented.

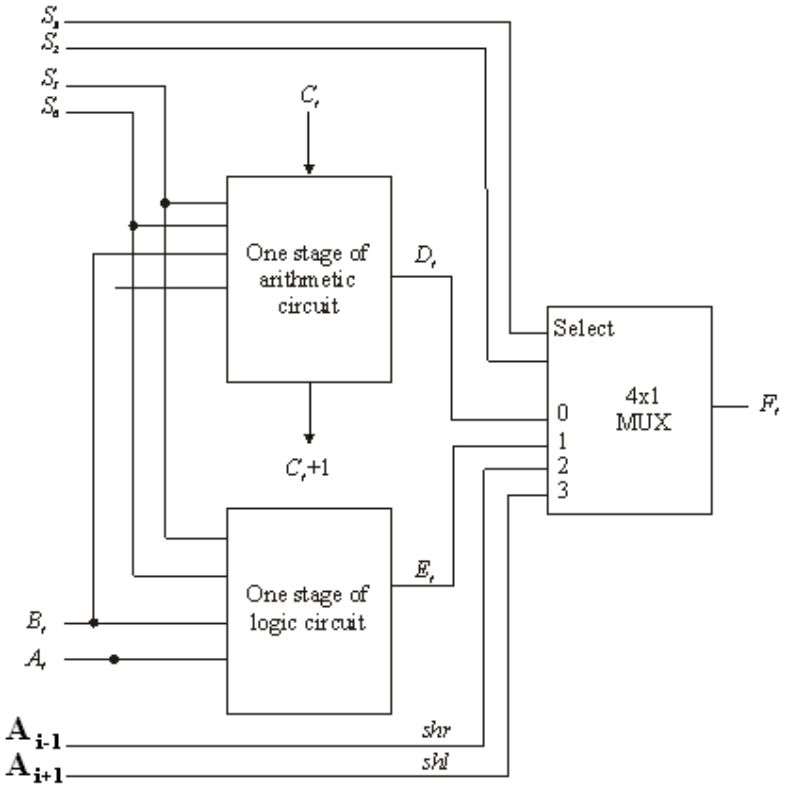
Ans.

The simplest and most commonly used procedure is to update main memory with every memory write operation, with cache memory being updated in parallel if it contains the word at the specified address. This is called the write-through method. This method has the advantage that main memory always contains the same data as the cache. This characteristic is important in systems with direct memory access transfers. It ensures that the data residing in main memory are valid at all times so that an I/O device communicating through DMA would receive the most recent updated data. The second procedure is called the write-back method. In this method only the cache location is updated during a write operation. The location is then marked by a flag so that later when the word is removed from the cache it is copied into main memory. The reason for the write-back method is that during the time a word resides to the cache, it may be updated several times; however, as long as the word remains in the cache, it does not matter whether the copy in main memory is out of date, since requests from the word are filled from the cache. It is only when the word is displaced from the cache that an accurate copy need be rewritten into main memory.

Q. 40 Design a hardware circuit to implement logical shift, arithmetic shift and circular shift operations. State your design specifications.

Ans.

In computer systems a number of storage registers connected to a common operational unit called an arithmetic logic unit (ALU). To perform a micro-operation, the contents of registers are placed in the inputs of a common ALU. The ALU performs an operation and the result of the operation is then transferred to a destination register. The ALU is a combination circuit so that the entire register transfer operation from the source register through the ALU and into the destination register can be performed during one clock pulse period. The shift micro-operation are often performed in a separate unit, but sometimes the shift unit is made part of the overall ALU. The arithmetic, logic and shift circuit can be combined into one ALU with common selection variables. One stage of an arithmetic logic and shift unit is shown in fig.



Function Table for Arithmetic Logic Shift Unit

Operation select						
S_3	S_2	S_1	S_0	c_{in}	Operation	Function
0	0	0	0	0	$F = A$	Transfer A
0	0	0	0	1	F = A+1	Increment A
0	0	0	1	0	F = A + B	Addition
0	0	0	1	1	F = A + B + 1	Add with carry
0	0	1	0	0	F = A + $\overline{B}$	Subtract with borrow
0	0	1	0	1	F = A + $\overline{B}$ +1	Subtraction
0	0	1	1	0	F = A - 1	Decrement A
0	0	1	1	1	F = A	Transfer A
0	1	0	0	x	F = A^ B	AND
0	1	0	1	x	F = A v B	*OR*
0	1	1	0	x	F = A $\oplus$ B	*XOR*
0	1	1	1	x	F = $\overline{A}$	Complement A
1	0	x	x	x	F = shr A	Shift right A into F
1	1	x	x	x	F = shl A	Shift left A into F

(i) Input A_i and Bi are applied to both the arithmetic and logic units. A particular micro-operation is selected with inputs S_1 and S_0.

(ii) A 4x1 MUX at the output chooses between an arithmetic output in D_i and a logic output in E_i.

(iii) The data inputs to the multiplexer are selected with inputs S_3 and S_2.

(iv) The other two data inputs to the MUX receive inputs A_{i-1} for the shift right operation and A_{i+1} for the shift left operation.

(v) C_{in} is the selection variable for the arithmetic operation.

(vi) The circuit provides eight arithmetic operation, four logic operations and two shift operations. Each operation is selected with the five variables S_3, S_2, S_1, S_0 and C_{in}.

(vii) The table lists the 14 operations of the ALU. The first eight are arithmetic operation and are selected with S_3 S_2 = 00. The next four are logic operation and are selected with S_3 S_2 = 01 and last two operation are shift operation and are selected with S_3 S_2 = 10 and 11.

Q.41 Discuss different techniques used for interfacing I/O units with the processor.

Ans.

Isolated I/O:- In the isolated I/O configuration the CPU has distinct input and output instructions, and each of these instructions is associated with the address of an interface register. When the CPU fetches and decodes the operation code of an input or output instruction, it places the address associated with the instruction into the common address lines. At the same time, it enables the I/O read or I/O write control line. When the CPU is fetching an instruction or an operand from memory, it places the memory address on the address lines and enables the memory read or memory write control line. The isolated I/O method isolates memory and I/O

addresses so that memory address values are not affected by interface address assignment since each has its own address space.

Memory Mapped I/O :- In computers that employ only one set of read and write signals and do not distinguish between memory and I/O addresses. The configuration is referred to as memory-mapped I/O. In a memory-mapped I/O organization there are no specific input or output instructions. The CPU can manipulate I/O data residing in interface registers with the same instructions that are used to manipulate memory words. The advantage is that the load and store instructions used for reading and writing from memory can be used to input and output data from I/O registers.

Q. 42 Write short notes on:-

(i) Sequential circuit.
(ii) Priority encoder.
(iii) Virtual memory.
(iv) Program control instructions.

Ans.

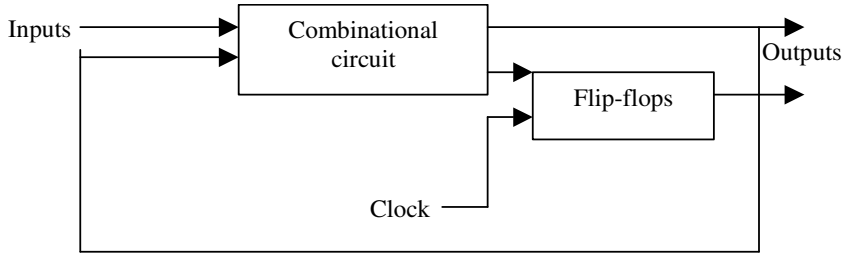
(i) **Sequential circuit**:-

A sequential circuit is an interconnection of flip-flops and gates. The gates by themselves constitute a combinational circuit, but when included with the flip-flops, the overall circuit is classified as a sequential circuit. The block diagram of a clocked sequential circuit is shown in Figure below. It consists of a combinational circuit and a number of clocked flip-flops. In general, any number or type of flip-flops may be included. In the diagram, the combinational circuit block receives binary signals from external inputs and from the outputs of flip-flops. The gates in the combinational circuit determine the binary value to be stored in the flip-flops after each clock transition. The outputs of flip-flops, in turn, are applied to the combinational circuit inputs and determine the circuit's behavior. The next state of flip-flops is also a function of their present state and external inputs. Thus a sequential circuit is specified by a time sequence of external inputs, external outputs, and internal flip-flop binary states.

Block diagram of a clocked synchronous sequential circuit

(ii) **Priority encoder**:-

The priority encoder is a circuit that implements the priority



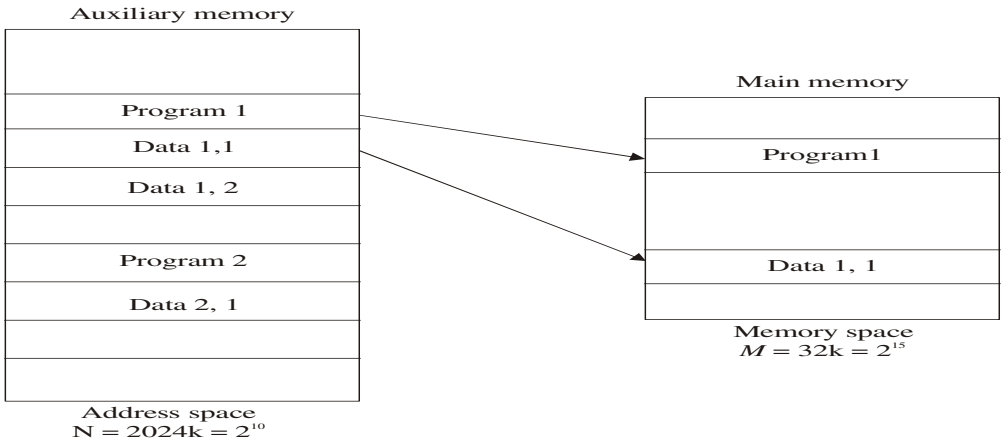
function. The logic of the priority encoder is such that if two or more inputs arrive at the same time, the input having the highest priority will take precedence. The truth table of a four-input priority encoder is given in Table below. The X's in the table designate don't care conditions. Input *Io* has the highest priority; so regardless of the values of other inputs, when this input is 1, the output generates an output

xy = 00. I_1 has the next priority level. The output is 01 if $I_{1=}$ 1 provided that $I_o = 0$ regardless of the values of the other two lower - priority inputs. The output for I_2 is generated only if higher-priority inputs are 0, and so on down the priority level. The interrupt status IST is set only when one or more inputs are equal to 1.The interrupt status IST is cleared to 0 and the other outputs of the encoder are not used, so they are marked with don't-care conditions. This is because the vector address is not transferred to the CPU when IST = 0. The output of the priority encoder is used to form part of the vector address for each interrupt source. The other bits of the vector address can be assigned any value.

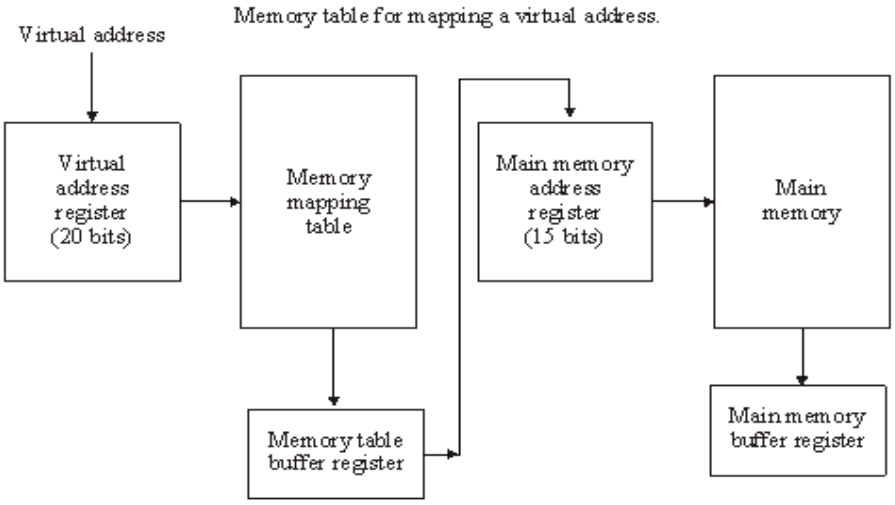
Inputs				Output			Boolean function
I_0	I_1	I_2	I_3	X	Y	IST	
1	X	X	X	0	0	1	
0	1	X	X	0	1	1	$x=I_0'I_1'$
0	0	1	X	1	0	1	$y=I_0'I_1+I_0'I_2'$
0	0	0	1	1	1	1	$(IST)=I_0+I_1+I_2+I_3$
0	0	0	0	X	X	0	

(iii) **Virtual memory**:- Virtual memory is a concept used in some large computer that permit the user to construct programs as through a large memory space were available, equal to the totality of auxiliary memory. Virtual memory is used to give programmes the illusion that they have a very large memory at their disposal, even through the computer actually has a relatively small main memory. A virtual memory system provides a mechanism for translating program generated addresses into correct main memory locations.

In a virtual memory system, programmes are told that they have the total address space at their disposal. The address field of the instruction code has a sufficient number of bits to specify all virtual address. In our example, the address field of an instruction code will consist of 20 bits but physical memory addresses must be specified with only 15 bits. Thus CPU will reference instructions and data with a 20-bit address, but the information at this address must be taken from physical memory because access to auxiliary storage for individual words will be prohibitively long.



Relation between address and memory space in a virtual memory system.



(iv) **Program Control Instruction**: – A program control type of instruction, when executed, may change the address value in the program counter and cause the flow of control to be altered. In other words, program control instructions specify conditions for altering the content of the program counter, while data transfer and manipulation instructions specify conditions of data-processing operations. Some typical program control instructions are list in table below. The branch and jump instructions are used interchangeably to mean the same thing. It is written in assembly language as BR ADR, where ADR is a symbolic name for an address.

Name	Mnemonic
Branch	BR
Jump	JMP
Skip	SKP
Call	CALL
Return	RET
Compare (by subtraction)	CMP
Test (by ANDing)	TST

Branch and jump instructions may be conditional or unconditional. An unconditional branch instruction causes a branch to the specified address without any condition. The conditional branch instruction specifies a condition such as branch, if positive or branch if zero. If the condition is meet, the program counter is loaded with the branch address and the next instruction is taken from this address.
The skip instruction does not need an address field and is therefore a zero-address instruction. A conditional skip instruction will skip the next instruction if the condition is met. If the condition is not met, control proceeds with the next instruction in sequence where the programmer inserts an unconditional branch instruction. The call and return instruction are used in conjunction with subroutines. The compare and text instructions do not change the program sequence directly. The compare instruction performs a subtraction between two operands, the result of the operation is not retained. Similarly, the test instructions performs the

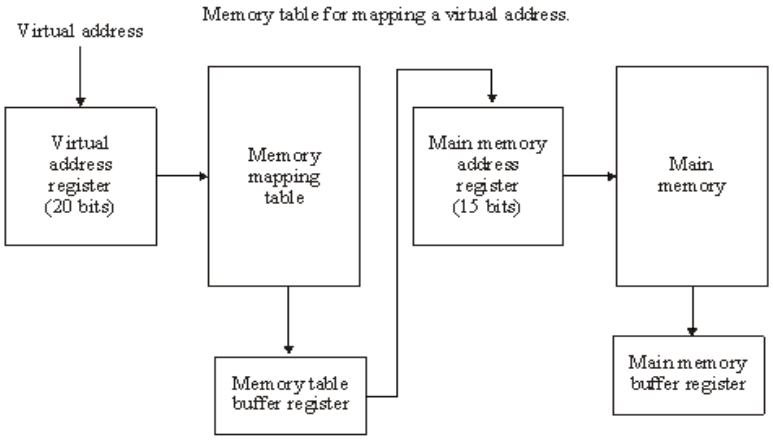
logical AND of two operands and updates certain status bits without retaining the result or changing the operands.

Q.43 Discuss the main features of associative memory Page Table. How does it work in mapping the virtual address into Physical memory address? **(7)**

Ans.

An address generated by user program is called virtual address and the set of virtual addresses make the virtual address space. A main memory address is called a location or physical address and set of such locations are called memory space or physical address space. However, in a system that uses a virtual memory, the size of virtual address space is usually longer than the available physical address space. Consider a computer of main – memory capacity of 32 K words. Since 32K = 2^{15} , 15- bits will be needed to specify a physical address. Suppose the computer has available auxiliary memory for strong 2^{20} words. Let N the address space and M be the memory space. Thus, N = 1024K and M=32K.

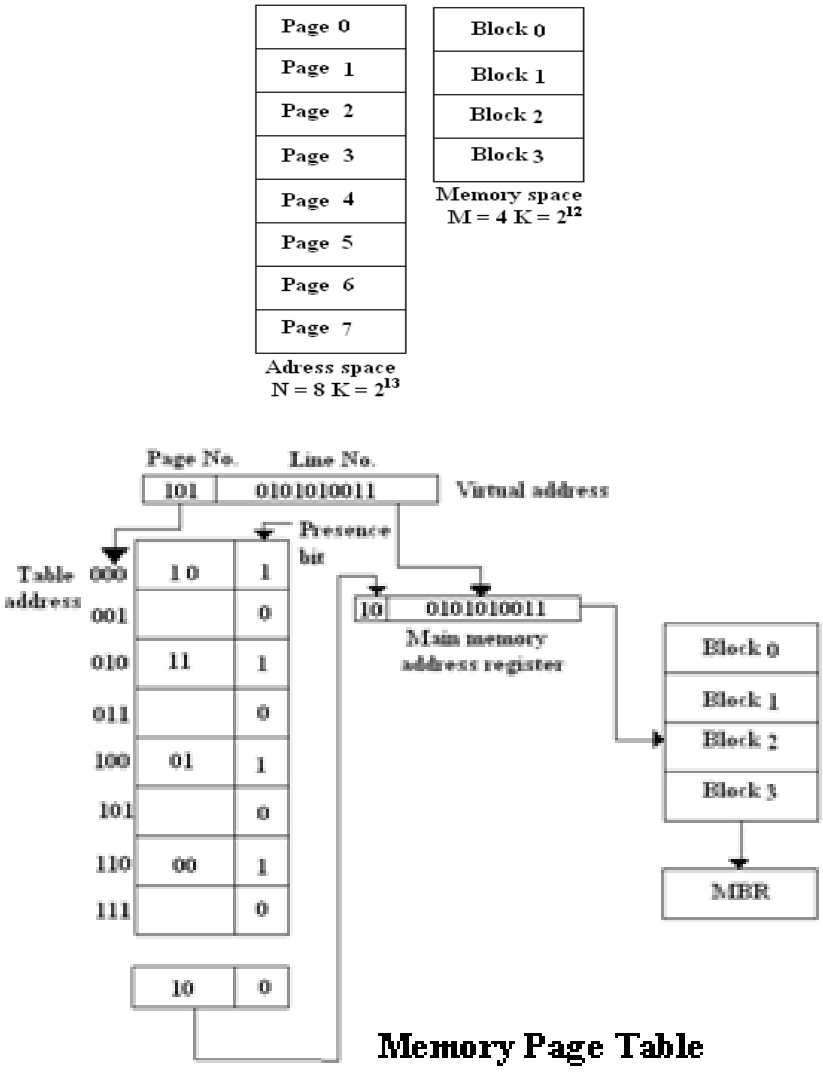
The address bit of the instruction code will consist of 20 bits put physical memory addresses must be specified using only 15 bits. Thus, the CPU will reference instructions and data with 20- bit addresses, but the information at this address must be taken from physical memory rather than auxiliary memory. Thus, it is required to map a virtual address of 20 bit to a physical address of 15 bit. For this a memory mapping table is needed which is shown in the fig. below. This mapping is a dynamic operation and every address is translated immediately as a word is referenced by CPU. The mapping table can be stored in main memory.



Address mapping can be further simplified if the information in address space and memory space can be divided into groups of equal size. The address space is broken into groups of equal size known as page and the memory space is broken into groups of same size known as blocks. These blocks can range from 64 to 4096 words. If a page or block consists of 1K words than the address space of 1024K consists of 1024 pages and the memory space of 32K consists of 32 blocks. Consider a computer with an address space of 8K and memory space of 4K. Thus, if the group of 1K words, we

have 8 pages and 4 blocks. This division of address space and memory space is shown in figure.

Every address generated by the CPU is divided into two parts: a page number address and a line within the page. In a computer with 2^p words per page, p bits are used for line-address and remaining high- order bits of the virtual address specify page number. Virtual address is 13 bits as in fig. Each page consists of 1K words i.e. 2^{10} words. Thus, 10 bits will be used to specify line number address and three high – order bits of the virtual address will specify one of the eight pages of the address space. The line address in address space and memory space is same and hence only mapping required is from a page number to a block number.



The mapping table in a paged system is shown in Figure. The memory page table consists of eight words. The table address denotes the page number and content of words give the block number where that page is stored in main memory. Thus this shows that pages 0, 2, 4 and 6 are stored in main memory in blocks 2, 3, 1 and 0, respectively. The present bit signifies that whether the page is in main memory or not. If presence bit is 1 the page is available in main memory and if it is 0 the page is not available in main

memory. When a CPU references a word in memory with virtual address of 13 bits, the lower – order 10 bits specifies the line number and three high – order specifies a page number which is also used as an address of the memory-page table. Then the content of the word at this address is read-out into the memory table buffer register along with the presence bit. If presence bit is 1, the content thus read (the block number) is transferred into main memory address register and the line number is also transferred into the main memory address register as 10 lower order bits. A read signal thus transfers the content to the main memory buffer register, to be used by the CPU. If the presence bit is 0, the content of the word referenced by the CPU does not reside in main memory. Then a call to operating system is generated to fetch the required page from auxiliary memory to main memory before resuming computation.

Q.44 A Virtual memory has a Page Size of 1K words. There are eight Pages and four blocks. The associative memory page table contains the following entries.

Page	Block
6	0
1	1
4	2
0	3

Give the list of virtual addresses in decimal that will cause a Page fault if used by CPU. **(6)**

Ans.
The pages which are not in main memory are:

Page	Address	Address that will cause fault
2	2K	2048-3071
3	3K	3072-4095
5	5K	5120-6143
7	7K	7168-8191

Q.45 How LRU technique is implemented ? **(3)**

Ans.
The LRU policy is more difficult to implement but has been more attractive on the assumption that the least recently used page is a better candidate for removal than the least recently used page is a better candidate for removal than the least recently loaded page as in FIFO. The LRU algorithm can be implemented by associating a counter with every page that is in the main memory. When a page is referenced, its associated counter is set to zero. At fixed interval of time, the counters associated with all pages presently in memory are incremented by 1. The least recently used page in the page with the highest count. The counters are often called aging registers, as their count indicates their age, that is how long ago their associated pages have been referenced.

Q.46 What is cycle stealing DMA operation? **(3)**

Ans.

Cycle Stealing: In this method, the DMA controller transfers one data word at a time, after which it must return control of the buses to the CPU. The CPU merely delays its operation for one memory cycle to allow the direct memory input/output transfer to 'Steal' one memory cycle.

Q.47 What do you understand by the term micro-operation. Explain Register and Arithmetic types of micro-operation. Show the hardware realization of decrement micro-operation.

i.e. $T1: X \leftarrow X - 1$ **(6)**

Ans.

A miocrooperation is an elementary operation performed with the data stored in registers. The operations executed on data stored in registers are called microoperations. A microoperation is an elementary operation performed on the information stored in one or more registers. The result of the operation may replace the previous binary information of a register or may be transferred to some another registers. Examples are clear, shift, count, load etc. A bidirectional shift register is capable of performing the shift right and shift left microoperations.

The microoperations must often encounter in digital computers are classified into following categories:

(1) Register transfer microoperations transfer binary information from one register to another.

(2) Arithmetic microoperations perform arithmetic operations on numeric data stored in registers.

(3) Logic microoperations perform bit manipulation operations on non numeric data stored in registers.

(4) Shift microoperations perform shift operations on data stored in registers.

The register transfer microoperation does not change the information content when the binary information moves from source register to destination register.

Arithmetic microoperations are those microoperations that are used to perform arithmetic operations. The basic arithmetic microoperations are addition, subtraction, increment, decrement and shift.

Let the arithmetic microoperations defined by the statement

$$R3 \leftarrow R1+R2$$

specifies an add microoperation. It states that add the content of register R1 and that of register R2 and stored the result in register R3. Thus to implement this statement with hardware we need three registers and the digital component that performs the addition operation.

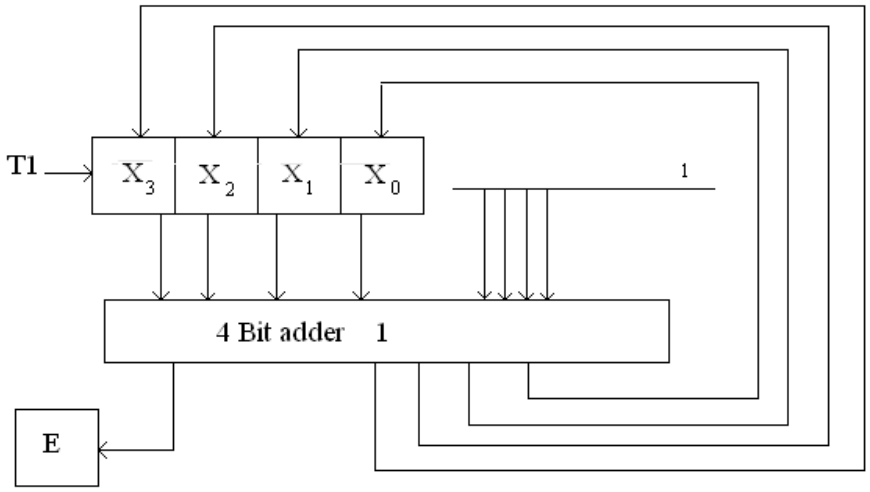
Subtraction is basically implemented through complementation and addition and can be specified by the statement

$$R \leftarrow A+B'+1$$

where B' is the 1's complement of B. When we add 1 to the 1's complement it will give 2's complement of B and adding A to the 2's complement of B gives A minus B(A-B). The increment and decrement microoperations are implemented with the help of combinational circuit or with a binary up-down counter. They are symbolized by plus-one or minus-one operation executed on the contents of a register.

The multiplication operation and division are valid arithmetic operations. Multiplication operation is basically implemented with a sequence of subtract and shift microoperations.
Hardware realization of decrement microoperation

T1: X ← X -1



Q.48 A Register 'A' holds on 8-bit binary number 11011001. Determine the operand 'B' and the logic micro-operation to be performed in order to change the value of 'A' to
(i) 01101101
(ii) 11111101 **(4)**

Ans.

A	=	11011001
B	=	10110100
A ⊕ B		01101101
A	=	11011001
B	=	11111101
A ∨ B		11111101

Q.49 Give the flow chart of division of two signed magnitude data. Discuss the logic of the flow chart. **(10)**

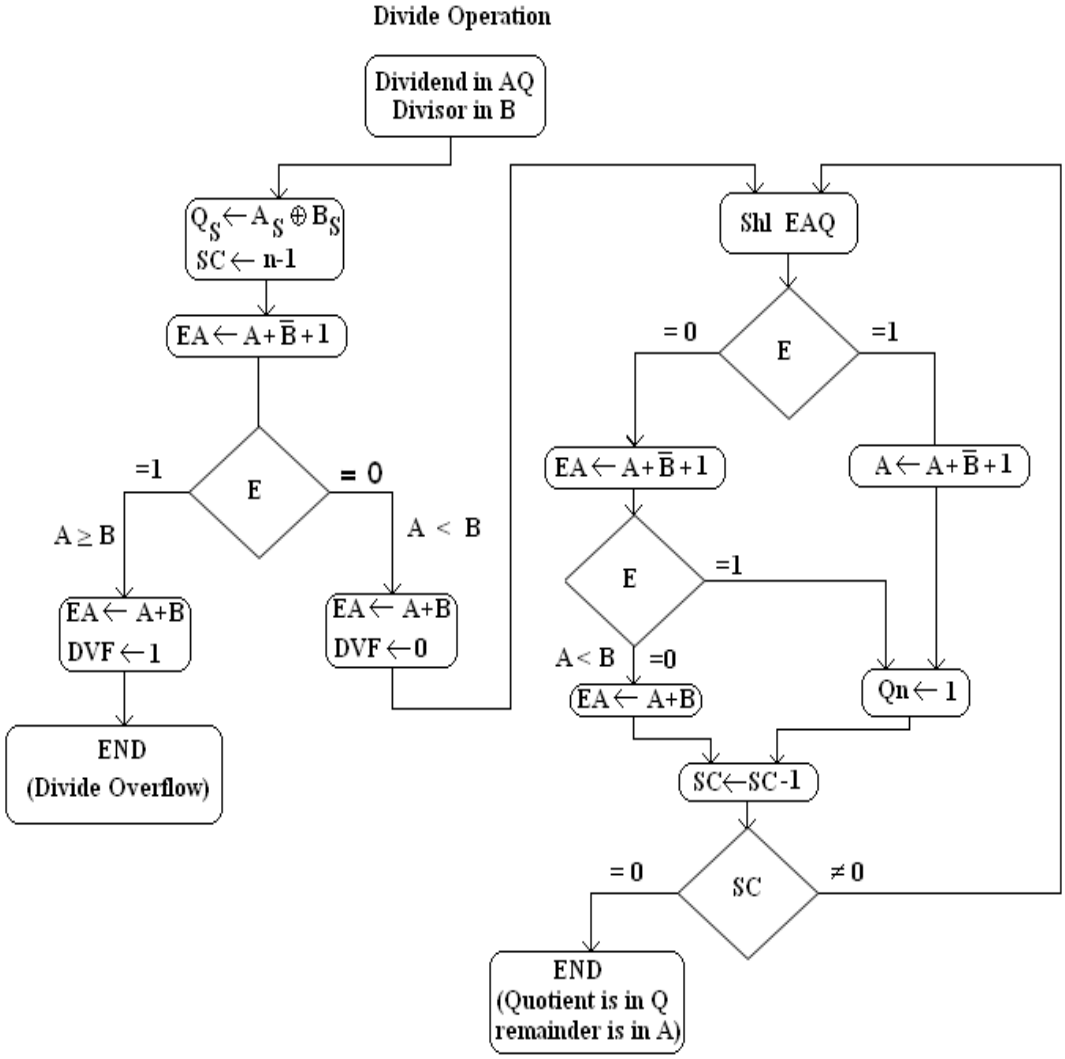
Ans.

The dividend is in A and Q and the divisor in B. The sign of the result is transferred into Q_s to be part of the quotient. The operands are transferred to registers from a memory unit that has words of n bits. Since an operand must be stored with its sign, one bit of the word will be occupied by the sign and the magnitude will consist of n-1 bits. A divide overflow condition is tested by subtracting the divisor in B from half of the bits of the dividend stored in A. If A ≥ B, the divide overflow flip flop DVF is set and the operation is terminated prematurely. If A<B, no divide overflow occurs so the value of the dividend is restored by adding B to A.

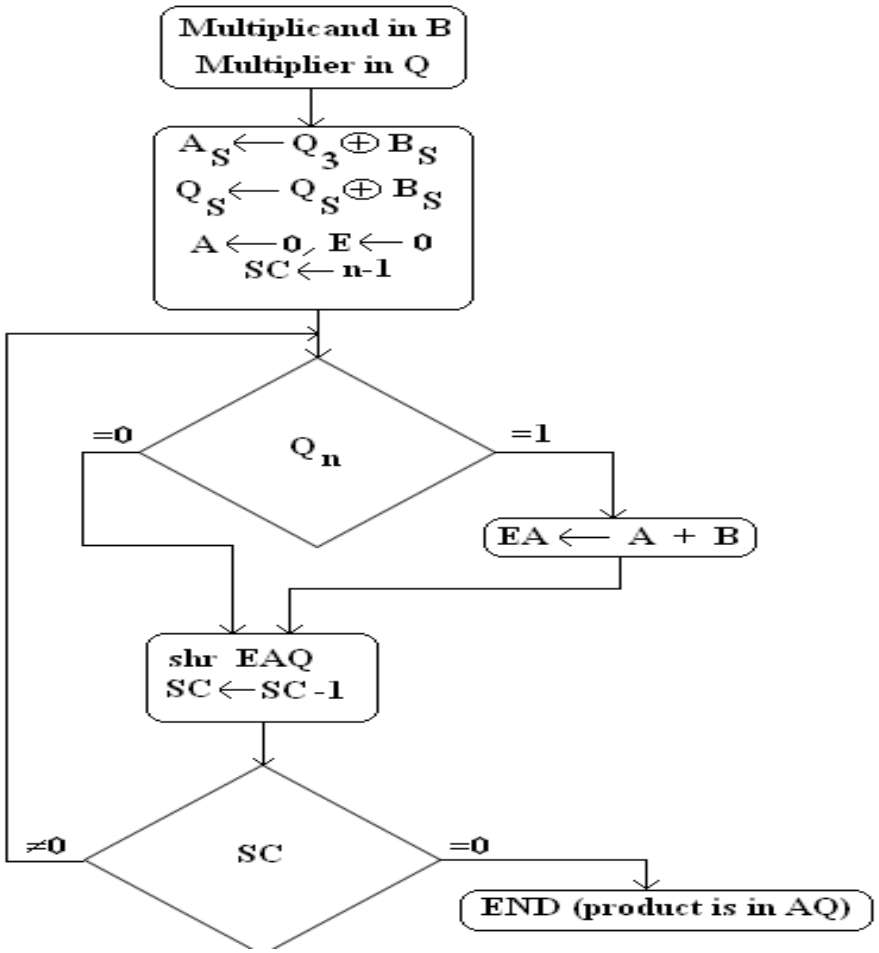
The division of the magnitudes starts by shifting the dividend in AQ to the left with the high order bit shifted into E. If the bit shifted into E is 1, then EA>B because EA consists of a 1 followed by n-1 bits while B consists of only n-1 bits. In this case B must be subtracted from EA and 1 inserted into Q_n for the quotient bit. Since register A is missing the high order bit of the dividend (which is in E), its value EA -2^{n-1}. Adding to this value the 2's complement of B results in

$$(EA-2^{n-1})+(2^{n-1}-B)=EA-B$$

The carry from this addition is not transferred to E if we want E to remain a 1.
If the shift left operation inserts a 0 into E, the divisor is subtracted by adding its 2's complement value and the carry is transferred into E. If E = 1, it signifies that $A \geq B$; therefore Q_n is set to 1. If E=0, it signifies that A<B and the original number is restored by adding B to A. This process is repeated again and again with register A holding the partial remainder. After n-1 times, the quotient magnitude is formed in register Q and the remainder is found in register A. The quotient sign is in Q_S and the sign of the remainder in A_S is the same as the original sign of the dividend.



Flow chart for divide operation.



Multiplication of (-3) with (+4)

Multiplicand (-3) B=101	E	A	Q	SC
Multiplier in Q	0	000	100	11
Q_n=0; shift right EAQ	0	000	010	10
Q_n=0; shift right EAQ	0	000	001	01
Q_n=1; Add B		101		
First partial product		101		
Shift right EAQ	0	010	100	00

Final product in AQ = 10100

Q.50 Convert the following arithmetic expression from reverse polish notation to infix notation:
ABXYZ+*–/
Write a program using three address instruction to evaluate the same. **(4)**

Ans.
The given expression is ABXYZ + * - /
ABXYZ + * - / = A / (BXYZ+*-)
= A / [B-(XYZ+*)]

= A / [B-{X*(YZ+)}]
= A / [B – {X* (Y+Z)}]
The arithmetic expression is
$\delta = A / \{X * (Y + Z)\}]$
PROGRAM USING THREE ADDRESS INSTRUCTIONS
ADD R1, Y, Z R1← M[Y]+M[Z]
MUL R1, X, R1 R1← M[X] * R1
SUB R1, B, R1 R1 ← M[B]-R1
DIV δ, A, R1 M[δ] ← M[A] / R1

Q.51 What are the interrupts? Explain different types of interrupts. **(8)**

Ans.
An interrupt signal is a signal sent by an input output interface to CPU when it is ready to send information to the memory or receive information from the memory.
The word interrupt is used from any exceptional event that causes the CPU to temporarily transfer the control from its current program to another program, an interrupt handler which will service the interrupt. This program is known as Interrupt Service Routine (ISR). Interrupts are the primary means by which input/ output devices obtain the services of the CPU. Various sources, internal and external to the CPU can generate interrupts. Input output interrupts are external requests to the CPU to initiate or terminate an input output operation, such as data transfer with a hard disk.
Interrupts are produced by hardware or by software error-detection circuits that invoke error-handling routines within the operating system. An attempt by an instruction to divide by 0 is an example of software generated interrupt. A power-supply failure can generate an interrupt that requests interrupt handler to save critical data about the system's state.
The interrupt is initiated by a signal generated by an external devices or a signal generated externally by the CPU. When CPU receives an interrupt signal from peripherals it stops executing the current program, saves the content or statue of various registers in the stack and then CPU executes a sub routine in order to perform the specific task requested by the interrupt.
The interrupt generated by special instructions are called software interrupts and they are used to implement system services.
In general, interrupts can be classified in the following three ways:

- Hardware and Software interrupt
- Vectored and Non-vectored interrupts
- Maskable and Non-maskable interrupts.

Hardware interrupt is a type of interrupt generated either externally by the hardware devices such as input output ports, key board, and disk drives etc or internally by the microprocessor. External hardware interrupts are used by devices to request attention from CPU. Internal hardware interrupts are generated by the CPU to control events.
The software interrupts are program instructions. These instructions are inserted at desired location in a program. A program generated interrupt, also called trap, which stops current processing in order to request a service provided by the CPU. While running a program, if software interrupt instruction is encountered the CPU initiates

an interrupt. For example, a program might generate a software interrupt to read input from keyboard.

When interrupt signal is generated the CPU responds to the interrupt signal by storing the return address from program counter into memory stack and then control is transfer or branches to the service routine that processes the interrupt. The processor chooses the branch address of the service routine in two different ways. One is called vectored interrupt and the other is called non-vectored interrupt.

In non-vectored interrupt, the branch address is assigned to a fixed location in memory. In vectored interrupt, the source that initiated the interrupt supplies the branch information. This information is called the interrupt vector.

In certain situations it may be desired that some of the several interrupts should not occur while CPU is busy in performing some important task. This can be done by masking. The interrupt that can be masked off is called maskable interrupt. When interrupts are masked, they are not withdrawn. They remain pending. Once the masking is removed, interrupt takes place.

Certain interrupts have to be serviced without delay; else something serious damage may be caused to program, data or results. Thus the CPU must have means to distinguish between urgent and non-urgent interrupts. Such interrupts are known as non-maskable interrupts and CPU does not mask (ignore) them. For example, memory failure, that must be serviced immediately.

Q.52 Explain the significance of different fields of an instruction with an example. **(4)**

Ans.

An instruction is a command given to a computer to perform a specified operation on some given data and the format in which the instruction is specified is known as Instruction format. The most common fields found in the instruction are:-

(i) An operation code field that specifies the operation to be performed. It is known as opcode field.

(ii) An address field that designates the registers address and/or a memory addresses.

(iii) A mode field that specifies the way the operands or the effective address is determined.

For example,

ADD R1, R0, ADD is the opcode and R1, R0 are the address field. Operations specified by computer instructions are executed on some data stored in memory or some registers. Operands residing on memory are specified by register address. The instruction may be of several different lengths containing different number of addresses. The number of address fields in the instruction format of a computer system depends on the internal architecture/organization of registers.

Q.53 The 8-bit registers A, B, C & D are loaded with the value $(F2)_H$, $(FF)_H$, $(B9)_H$ and $(EA)_H$ respectively.

Determine the register content after the execution of the following sequence of micro-operations sequentially.

(i) $A \leftarrow A + B$, $C \leftarrow C + Shl(D)$

(ii) $C \leftarrow C \wedge D$, $B \leftarrow B + 1$.

(iii) $A \leftarrow A - C$.

(iv) $A \leftarrow Shr(B) \oplus Cir(D)$ **(8)**

Ans.

$A = (F2)_H = (11110010)_2$
$B = (FF)_H = (11111111)_2$
$C = (B9)_H = (10111001)_2$
$D = (EA)_H = (11101010)_2$

(i) $A \leftarrow A + B, C \leftarrow C + shl(D)$

A+B = 11110010
+11111111

11110001 $= (F1)_H$

shl(D) = shl(11101010) = 11010100 $= (D4)_H$

C + shl (D) = 10111001
+11010100

10001101 $= (8D)_H$

After these microoperations the content of A, B, C, and D are $(F1)_H$, $(FF)_H$, $(8D)_H$ and $(D4)_H$ respectively.

(ii) $C \leftarrow C \wedge D, B \leftarrow B+1$

C ^ D = 10001101
^11010100

10000100 $= (84)_H$

B+1 = 11111111
+ 1

00000000 $= (00)_H$

After these microoperations the content of A, B, C, and D are $(F1)_H$, $(00)_H$, $(84)_H$ and $(D4)_H$ respectively.

(iii) $A \leftarrow A - C$

A – C = 11110001
-10000101

01101100 $= (6D)_H$

After this microoperation, the content of A, B, C, and D are (6D)H, (00)H, (84)H and (D4)H respectively.

(iv) $A \leftarrow shr(B) \oplus Cir(D)$

shr(B) = shr(00000000) = 00000000 $= (00)_H$

Cir(D) = Cir (11010100) = (01101010) $= (6A)_H$

shr(B) ⊕ Cir(D) = 00000000
⊕ 01101010

01101010 $= (6A)_H$

After this microoperation, the content of A, B, C, and D are $(6A)_H$, $(00)_H$, $(84)_H$ and $(6A)_H$ respectively.

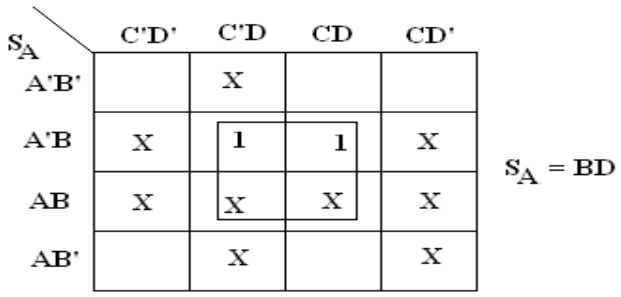
Q.54 Design a synchronous self starting counter using S-R flip flops for counter the sequence 0, 2, 3, 5, 8, 7, 15, 12, 11, 10 & repeat. **(8)**

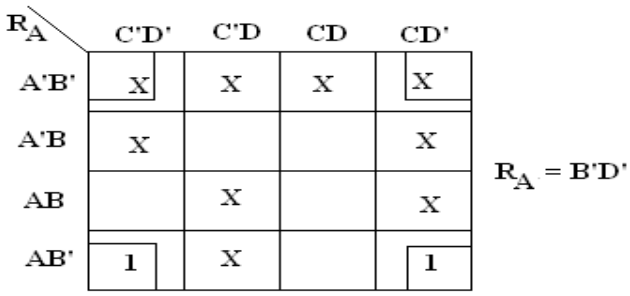
Ans.

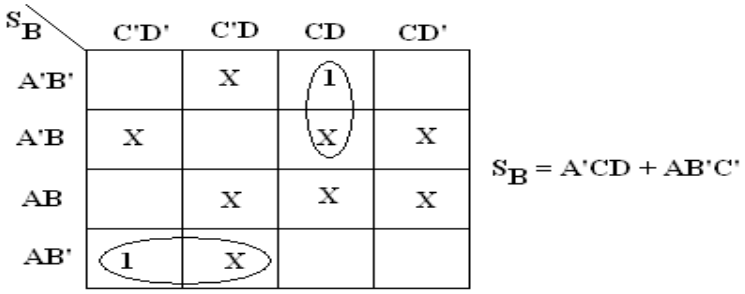
Excitation table for counter using S-R Flip Flop

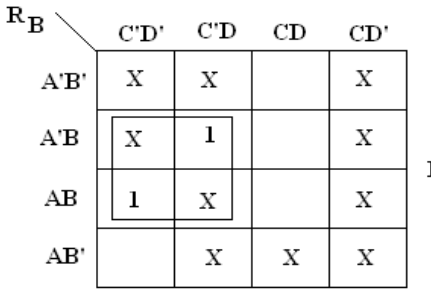
A	B	C	D	S_A	R_A	S_B	R_B	S_C	R_C	S_D	R_D
0	0	0	0	0	x	0	x	1	0	0	X
0	0	1	0	0	x	0	x	x	0	1	0
0	0	1	1	0	x	1	0	0	1	x	0
0	1	0	1	1	0	0	1	0	x	0	1
1	0	0	0	0	1	1	0	1	0	1	0
0	1	1	1	1	0	x	0	x	0	x	0
1	1	1	1	x	0	x	0	0	1	0	1
1	1	0	0	x	0	0	1	1	0	1	0
1	0	1	1	x	0	0	x	x	0	0	1
1	0	1	0	0	1	0	x	0	1	0	x

The K – maps are as follows:-



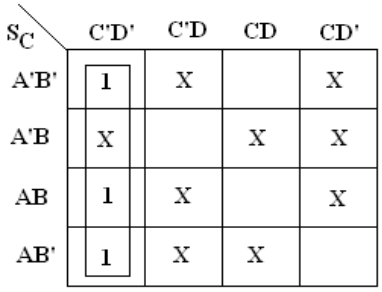






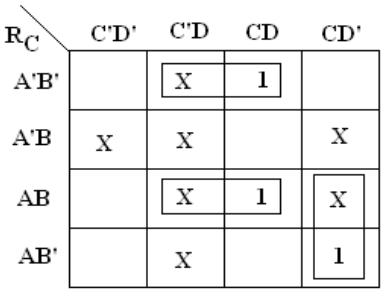
R_B	C'D'	C'D	CD	CD'
A'B'	X	X		X
A'B	X	1		X
AB	1	X		X
AB'		X	X	X

$R_B = BC'$



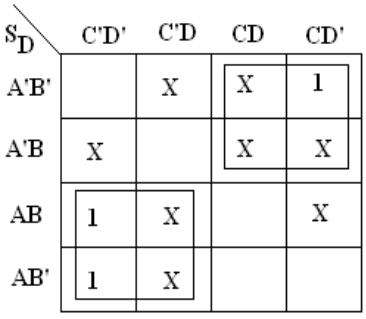
S_C	C'D'	C'D	CD	CD'
A'B'	1	X		X
A'B	X		X	X
AB	1	X		X
AB'	1	X	X	

$S_C = C'D'$



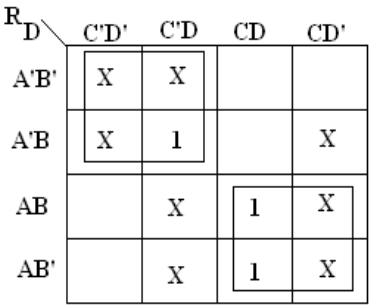
R_C	C'D'	C'D	CD	CD'
A'B'		X	1	
A'B	X	X		X
AB		X	1	X
AB'		X		1

$R_C = A'B'D$
$= + A B D + A C D'$
$= (A \odot B) D + A C D'$



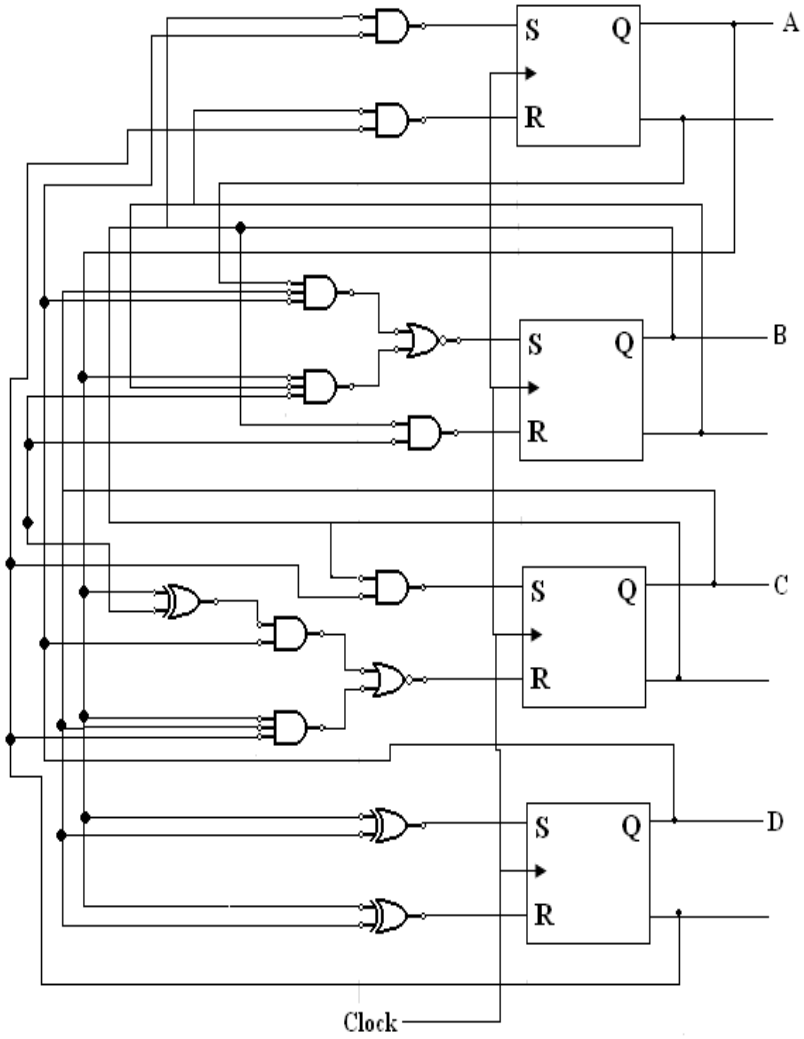
S_D	C'D'	C'D	CD	CD'
A'B'		X	X	1
A'B	X		X	X
AB	1	X		X
AB'	1	X		

$S D = AC' + A'C$
$= A \oplus C$



R_D	C'D'	C'D	CD	CD'
A'B'	X	X		
A'B	X	1		X
AB		X	1	X
AB'		X	1	X

$RB = A'C' + AC$
$= A \odot C$



Logic Circuit

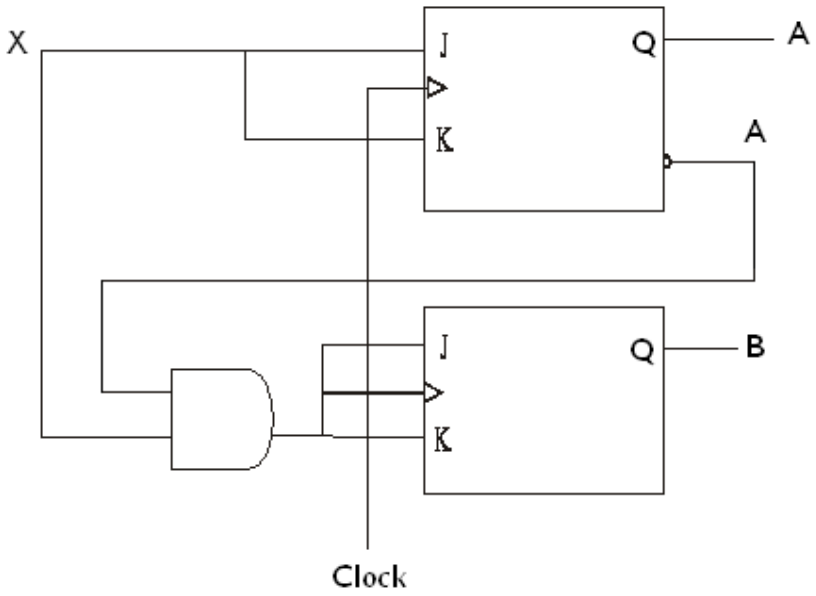
Given, f (b, a, c) = Σm (1,3,5,6,7,11,13,14) and don't care M_4, M_9, M_{10}

	a'c'	a'c	ac	ac'
d'b'		1	1	
d'b	X	1	1	1
db		1		1
db'		X	1	X

F = a'c + d'c + db'a + bac'

Q. 55 Design a two bit countdown counter with two flip-flop and one input x.
When x = 0, the state of the flip-flop does not change. When x = 1, the state sequence is 11, 10. 01. 00, 11 and repeats.

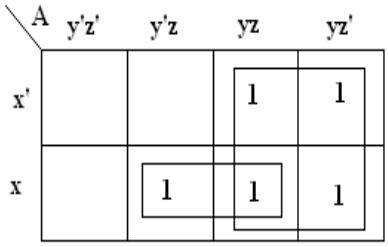
Ans.
$J_A = K_A = x$
$J_B = K_B = a'x$
Two bit countdown counter



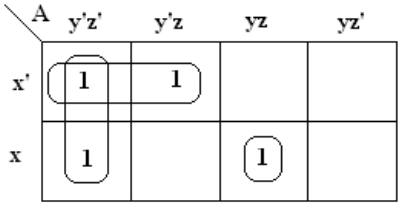
Q. 56 Design a combinational circuit with three inputs x, y, z and three outputs A, B, C. When the binary input is 0, 1, 2, or 3 the binary output is two greater than the output. When the binary input is 4, 5, 6 and 7, the binary output is one less than the input.

Ans.

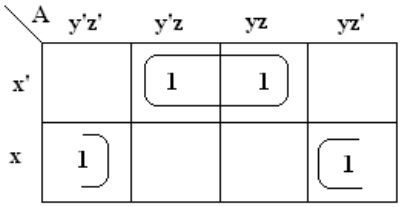
x	y	z	A	B	C
0	0	0	0	1	0
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	1	0	1
1	0	0	0	1	1
1	0	1	1	0	0
1	1	0	1	0	1
1	1	1	1	1	0



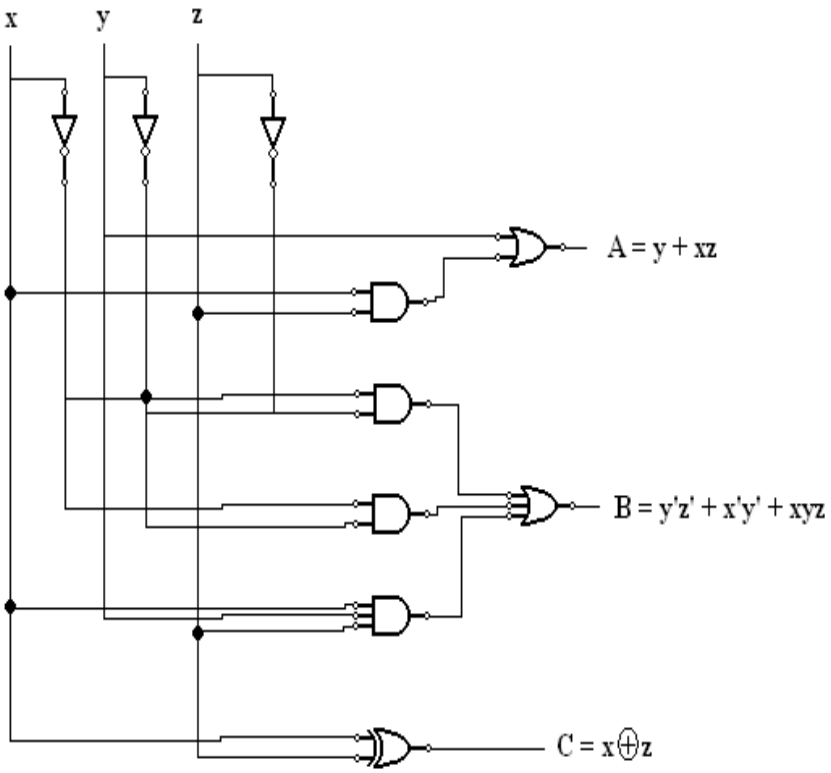
A = y + xz



B = y'z' + x'y' + xyz

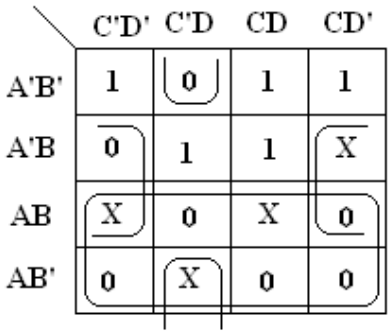


C = x'z + xz' = x⊕z



Q. 57 If $Y = (M_0, M_2, M_3, M_5, M_7) + (M_6, M_9, M_{12}, M_{15})$ where 'd' stands for don't care. Express the Boolean expression is product of sum form and also show the k-map for that product of sum form.

Ans.



F = A'(B'+D) (B + C + D')

Q. 58 A RAM chip 4096 x 8 bits has two enable lines. How many pains are needed for the integrated circuit package of Draw a block diagram and label all input and outputs pins of the RAM. What is the main feature of random access memory?

Ans.
RAM chip is of 4096 x 8. Hence, 12 bits for inputs. 2 for enable lines and 8 bits for outputs. Hence, total 22 pins are needed.

MAIN MEMORY:-

The main memory is the central storage unit in a computer system. It is a relatively large and fast memory used to store programs and data during the computer operation. The principal technology used for the main memory is based on semiconductor integrated circuits. Integrated circuit RAM chips are available in two possible operating modes, *static* and *dynamic.*

The static RAM consists essentially of internal flip-flop that store the binary information. The stored information remains valid as long as power is applied to be unit. The dynamic RAM stores the binary information in the form of electric charges that are applied to capacitors. The capacitors are provided inside the chip by MOS transistors. The stored charge on the capacitors tend to discharge with time and the capacitors must be periodically recharged by refreshing the dynamic memory. Refreshing is done by cycling through the words every few milliseconds to store the delaying charge. The dynamic RAM offers reduced power consumption and large storage capacity in a single memory chip.

The static RAM is easier to use and has shorter read and write cycles.

Most of the main memory in a general-purpose computer is made up of RAM integrated circuit chips, but a portion of the memory may be constructed with ROM chips. Originally, RAM was used to refer to a random-access memory, but now it is used to designate a read/write memory to distinguish it from a ready-only memory, although ROM is also random access. RAM is used for storing the bulk of the programs and data that are subject to change. ROM is used for storing programs

that are permanently resident in the computer and for tables of constants that do not change in value once the production of the computer is completed.

Q.59 The RAM IC as described above is used in a microprocessor system, having 16 bit address line and 8-bit data line. It's enable-1 input is active when A_{15} and A_{14} bits are 0 & 1 and enable-2 input is active when A_{13}, A_{12} bits are 'X' and 'O'. What shall be the range of addresses that is being used by the RAM. **(4)**

Ans.

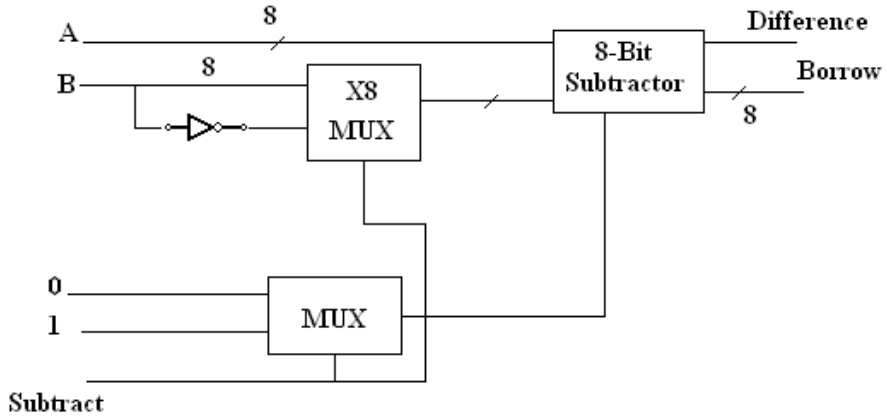
A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
0	**1**	**x**	**0**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**

Thus, its range of address being used by RAM is 4000 - 6PFF.

Q. 60 Implement a full subtractor logic by using multiplexer.

Ans.

Full subtractor circuit

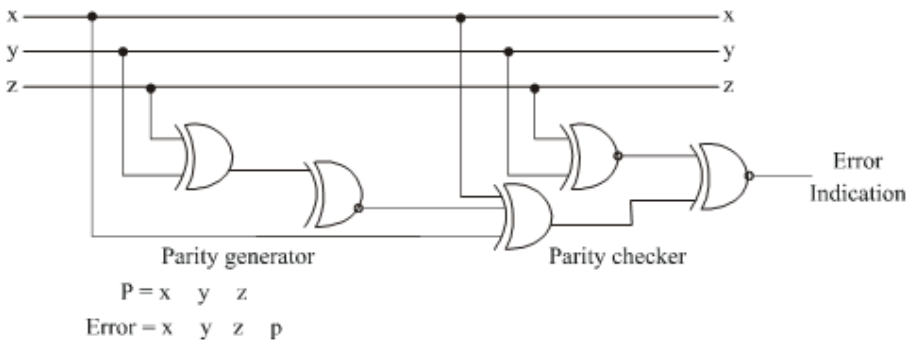


$$\text{Difference} = A \oplus B \oplus P$$

$$\text{Borrow} = B.P + A'.(B + P)$$

A	B	P	Difference	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Q. 61 Derive the circuit for a 3-bit parity generator and 4-bit parity checker using an even parity bit.



Q. 62 What is microoperation? Give suitable examples of some four types of microoperations.

Ans.

A microoperation is an elementary operation performed with the data stored in registers.

1) Register transfer microoperation transfer binary information from one register to another.
2) Arithmetic microoperations perform arithmetic operation on numeric data stored in registers.
3) Logic micro operation performs bit manipulation operation on numeric data stored in register.
4) Shift microoperation performs shit operation on data stored in registers.

Example:-

Arithmetic microoperation

$R_3 \longleftarrow R_1 + R_2$

Subtract microoperation

$R_3 \leftarrow R_1 + \overline{R_2} + 1$

Logic microoperation

$P : R_1 \leftarrow R_2 \oplus R_3$

$R_4 \leftarrow R_5 \vee R_6$

Shift Microoperation

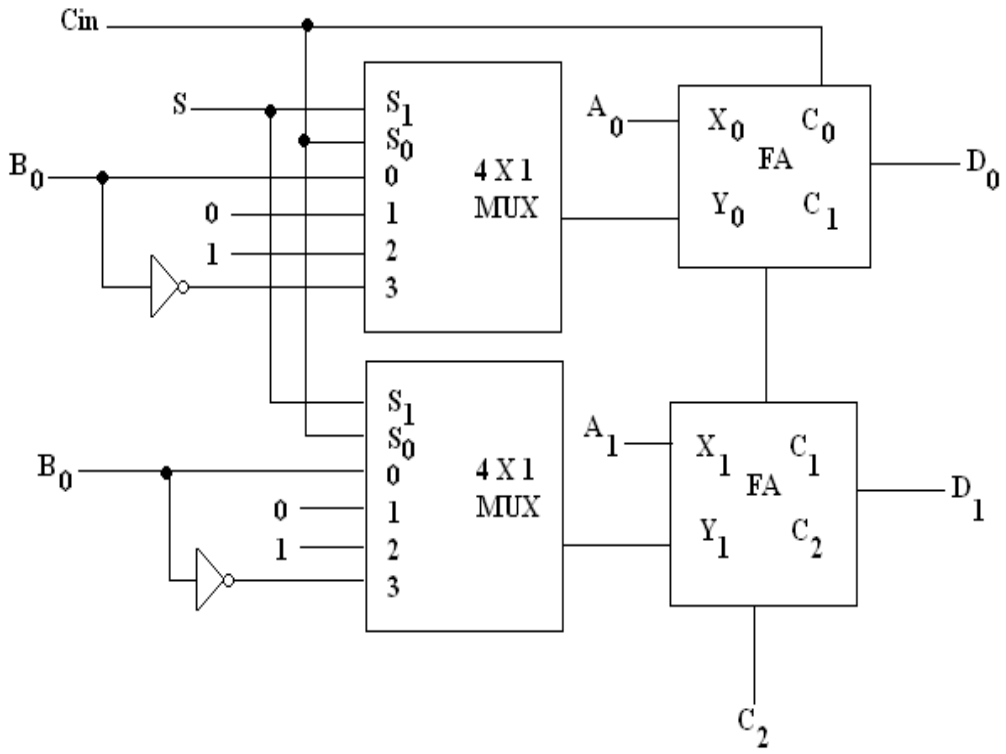
$R_1 \leftarrow ShlR_1$

$R_2 \leftarrow ShrR_2$

Q. 63 Give the hardware realization of 4-bit arithmetic circuit capable of doing addition, subtraction, increment, decrement etc. Give the function table and explain its operation.

Ans.
Arithmetic Circuit

S	Cin	X	Y	
0	0	A	B	(A+B)
0	1	A	0	(A+1)
1	0	A	1	(A-1)
1	1	A	B	(A-B)



Q. 64 Give the comparison between & examples of hardwired control unit and micro programmed control unit.

Ans.
Comparison between Hardwired and microprogrammed control unit

Characteristics	Hardwired Control	Microprogrammed Control
(1) Speed	Fast	Slow
(2) Implementation	Hardware	Software
(3) Ability to handle large/complex instruction sets	Somewhat difficult	Easier
(4) Design process	Difficult for more operation	Easy
(5) memory	No memory used	Control memory used.
(6) Flexibility	No flexibility	More flexibility

Q.65 What do you mean by Fetch cycle, instruction cycle, machine cycle, interrupt acknowledgement cycle.

Ans.
The execution of an instruction may itself involve a number of steps. The two stages of fetch and execution as follows.
The instruction fetch is a common fraction instruction from location is memory. The instruction execution may involve several operations and depends on the nature of the instructions. The instruction cycle is referred to as the fetch cycle and execute cycle. Interrupt acknowledgement cycle that I_1 regardless of the values of the other two lower-priority inputs. The output for I_2 is generated only if higher-priority inputs are 0_1 and on down the priority level.

Inputs				Outputs			
I_0	I_1	I_2	I_3	x	y	IST	Boolean Function
1	x	x	x	0	0	1	
0	1	x	x	0	1	1	$x = I'_0 I'_1$
0	0	1	x	1	0	1	$y = I'_0 I_1 + I'_0 I_2$
0	0	0	1	1	1	1	$(IST) = I'_0 + I_1 + I'_2 I_3$

The interrupt status IST is set only when one or more inputs are equal to 1. If all inputs are I_0 IST is cleared to 0 and the other outputs of the encoder are not used, so they are marked with don't care conditions. This is because the vector address is not transferred to the CPU when IST = 0.
The output of the priority encoder is used to form part of the vector, address for each interrupt source. The other bits of the vector address can be assigned any value.

Interrupt Cycle:-
The interrupt enable flip-flop *IEN* shown in Fig. 11-14 can be set or cleared by program instructions. When *IEN* is cleared, the interrupt request coming from IST is neglected by the CPU. The program-controlled *IEN* bit allows the programme to choose whether to use the interrupt facility. If an instruction to clear *IEN* has been inserted in the program, it means that the user does not want his program to be interrupted. An instruction to set *IEN* indicates that the interrupt facility will be used while the current program is running. Most computers include internal hardware that clears *IEN* to 0 every time an interrupt is acknowledged by the processor.
At the end of each instruction cycle the CPU checks *IEN* and the interrupt signal from *IST*. If either is equal to 0, control continues with the next instruction. It both *IEN* and *IST* are equal to 1; the CPU goes to an interrupt cycle. During the interrupt cycle the CPU performs the following sequence of microoperations:

SP ← SP – 1	Decrement stack pointer
M [SP] ← PC	Push PC into stack
INTACK ← 1	Enable interrupt acknowledge
PC ← VAD	Transfer Vector address to PC
IEN ← 0	Disable further interrupts

The CPU pushes the return address from PC into the stack. It then acknowledges the interrupt by enabling the *INTACK* line. The priority interrupt unit responds by placing a unique interrupt vector into the CPU data bus. The CPU transfers the vector address into *PC* and clears *IEN* prior to going to the next fetch phase. The instruction read from memory during the next fetch phase will be the one located at the vector address.

Q. 66 Explain in brief how a digital computer system works in a interrupt driven input-output programming.

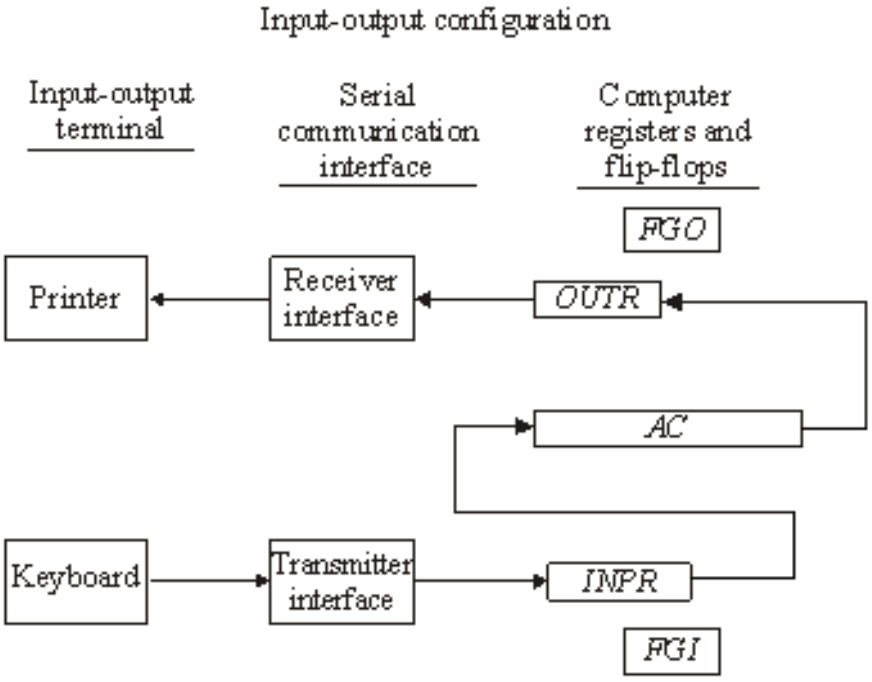
Ans.

A computer can serve no useful purpose unless it communicates with the external environment. Instructions and data stored in memory must come from some input device. Computational results must be transmitted to the user through some output device. Commercial computers include many types of input and output devices. To demonstrate the most basic requirements for input and output communication, we will use as an illustration a terminal unit with a keyboard and printer.

Input-Output Configuration:-

The terminal sends and receives serial information. Each quantity of information has eight bits of an alphanumeric code. The serial information from the keyboard is shifted into the input register *INPR*. These two registers communicate with a communication interface serially and with the AC in parallel. The input-output configuration is shown in. The transmitter interface receives serial information from the keyboard and transmits it to *INPR*. The receiver interface receives information from OUTR and sends it to the printer serially. The input register INPR consists of eight bits and holds alphanumeric input information. The 1-bit input flag *FGI* is a control flip-flop. The flag bit is set to 1 when new information is available in the input device and is cleared to 0 when the information is accepted by the computer.

Output register:-



The output register *OUTR* works similarly but the direction of information flow is reversed. Initially, the output flag *FGO* is set to 1. The computer checks the flag bit; if it is 1, the information from *AC* is transferred in parallel to *OUTR* and *FGO* is cleared to 0. The output device accepts the coded information, prints the corresponding character, and when the operation is completed, it sets *FGO* to 1. The computer does not load a new character into *OUTR* when *FGO* is 0 because this condition indicates that the output device is in the process of printing the character.

Input-Output Instructions. Input and output instructions are needed for transferring information to and from *AC* register, for checking the flag bits, and for controlling the interrupt facility. Input-output instructions have an operation code 1111 and are recognized by the control when $D_7 = 1$ and I = 1. The remaining bits of the instruction specify the particular operation. The control functions and microoperations for the input-output instructions are listed. These instructions are executed with the clock transition associated with timing signal T_3. Each control function needs a Boolean relation D_7IT_3, which we designate for convenience by the symbol *p*. The control function is distinguished by one of the bits in *IR* (6-11). By assigning the symbol B_i to bit i of *IR,* all control functions can be denoted by pBi for i=6 through 11. The sequence counter SC is cleared to O when $p=D_7IT_3 = 1$

Input-Output Instructions

$D_7IT_3 = p$ (common to all input-output instructions)			
IR (i) = B_i [bit in *IR*(6-11) that specifies the instructions]			
	p:	SC ← 0	Clear SC 0
	INP	pB_{11}: AC(07) ← INPR, FGI ← 0	Input character
	OUT	pB_{10}: OUTR ← AC(0-7), FGO ← 0	Output character
	SKI	pB_9: If (FGI=1) then (PC ← PC+1)	Skip on input flag
	SKO	pB_8: If(FGO=1) then (PC ← PC+1)	Skip on output flag
	ION	pB_7: IEN ← 1	interrupt enable on
	IOF	pB_6: IEN ← 0	interrupt enable off

Q.67 Design a CPU that meets the following specifications:
It can access 64 words of memory, each word being 8-bit long. The CPU does this by outputing a 6-bit address on its output pins A [5.....0] and reading in the 8-bit value from memory on inputs D [7.....0]. It has one 8-bit accumulator, 8-bit address register, 6-bit program counter, 2-bit instruction register, 8-bit data register. The CPU must realize the following instruction set.

Instruction	**Instruction code**	**operation**		
AND	00AAAAAA	AC ←AC+M(AAAAAA)		
JMP	01AAAAAA	Go to AAAAAA		
ADD	10AAAAAA	AC ←AC + M(AAAAAA)		
INC	11xxxxxx	AC ←AC+1		
Label	**Microoperation**	**CD**	**BR**	**AD**
AND	ORG 16			

	MOP	I	CALL	INDRCT
	READ	U	JMP	NEXT
ANDOP	AND	U	JMP	FETCH
ADD	ORGO			
	MOP	I	CALL	INDRCT
	READ	U	JMP	NEXT
	ADD	U	JMP	FETCH
STORE	ORG8			
	NOP	I	CALL	INDRCT
	ACTRD	U	JMP	NEXT
	WRITE	U	JMP	FETCH
COMPLEMENT	NOP	I	CALL	INDRCT
	READ	U	JMP	NEXT
	COM	U	JMP	FETCH

Q.68 What do you mean by software of hardware interrupts? How there are used in a microprocessor system?

Ans.

Software and hardware interrupt:

The software interrupts are program instructions. These instructions are inserted at desired location in a program. A program generated interrupt also called trap, which stops current processing in order to request a service provided by the CPU. While running a program, if software interrupt instruction is encountered the CPU initiates an interrupt. For example a program might generate a software interrupt to read input from keyboard. Hardware interrupt is a type of interrupt generated either externally by the hardware devices such as input/ output ports, keyboard and disk drive etc or internally by the microprocessor. External hardware interrupts are used by device to request attention from CPU. Internal hardware interrupts are generated by the CPU to control events.

Q.69 What are the reasons of Pipe-Line conflicts is a Pipe Lined processor? How are they resolved?

Ans.

There are three major difficulties that cause the instruction pipeline to deviate form its normal operation.

1) Resource conflicts caused by access to memory by two segments at the same time. Most of these conflicts can be resolved by using separate instruction and data memories.

2) Data dependency conflicts arise when an instruction depends on the result of a previous instruction, but this result is not available.

3) Branch difficulties arise from branch and other instructions that change the value of PC. In computer, for solving conflicts problems to the compiler that translates the high level programming language into a machine language program. The compiler for such computer is designed to detect a data conflict and re order the instructions, to delay the loading of the conflicting data by inserting no-operation instructions. This method is referred to as delayed load.

Q. 70 Explain the difference between a subroutine & macro.

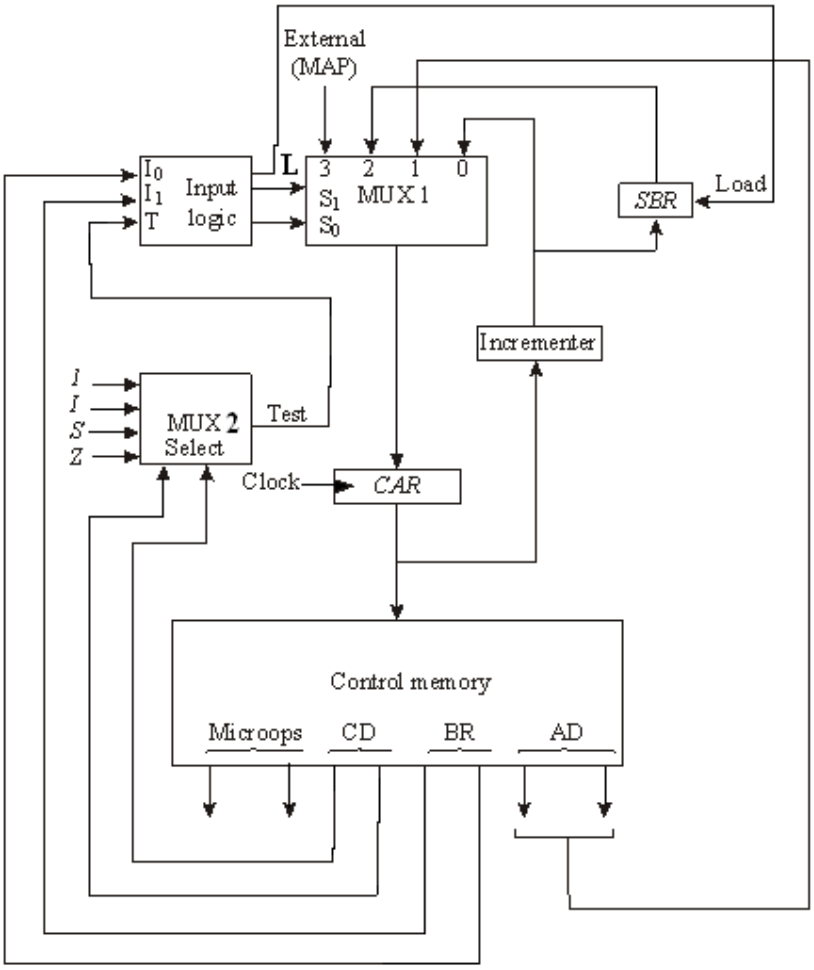
Ans.

It is inefficient to have to write code for standard routines. For example reading a character form the keyboard or saving a block of data to disk. Standard routines are available called library routines. They may be called up macros and many of the most useful routines are available as operating system calls. A call to a macro is a single command, which can be replaced by many commands that set put into the programme where the macro name is encountered. A set of common instructions that can be used in a program in called a subroutine.

Q. 71 With neat block diagram explain the working of a microprogram sequencer for control memory.

Ans.

Block Diagram of Micro program sequencer



The input logic circuit has three inputs I_0, I_1, and T, and three outputs, S_0, S_1, and L. Variables S_0, and S_1, select one of the source addresses for CAR. Variable L enables the load input in SBR. The binary values of the two selection variables

determine the path in the multiplexer. For example, with S_1 S_0 = 0, multiplexer input number 2 is selected and establishes a transfer path from *SBR* to *CAR*. Note that each of the four inputs as well as the output of MUX 1 contains a *7*-bit address. The truth table for the input logic circuit is shown. Inputs I_1 and I_0 are identical to the bit values in the BR field. The bit values for $S_1$and S_0 are determined from the stated function and the path in the multiplexer that establishes the required transfer. The subroutine register is loaded with the incremented value of *CAR* during a call microinstruction (BR = 01) provided that the status bit condition is satisfied (*T* = 1).

$$S_1 = I_1$$
$$S_0 = I_1 I_0 + I'_1 T$$
$$L = I'_1 I_0 T$$

Input Logic Truth Table for Microprogram Sequence

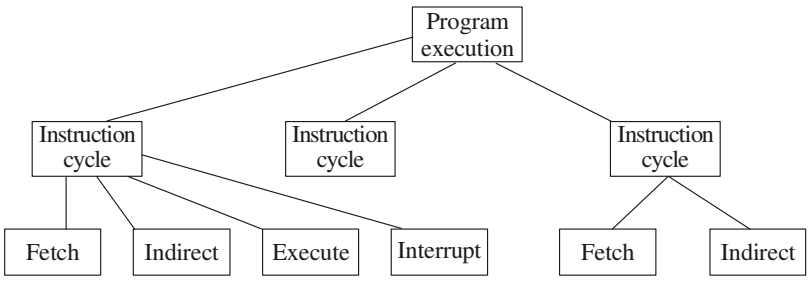
BR Field		Input I_1	I_0	T	Mux 1 S_1	S_0	Load *SBR* L
0	0	0	0	0	0	0	0
0	0	0	0	1	0	1	0
0	1	0	1	0	0	0	0
0	1	0	1	1	0	1	1
1	0	1	0	x	1	0	0
1	1	1	1	x	1	1	0

Q. 72 Discuss different method used for specifying micro operation in microoperation field of microcode. State their merits and demerits.

Ans.

The function of a computer is to execute program. We have the operation of a computer, in executing a program consists of a sequence of instruction cycles with one machine instruction per cycle. When we are referring to here is the execution time sequence of instructions. We have further seen that each instruction cycle can be considered to be made up a number of smaller writs.

Micro-operation - The prefix micro refers to the fact that each step is very simple and accomplish very little. The execution of program consists of the sequential execution of instructions. The performance



of each sub cycle involves one or more shorter operations, that is micro-operations. Micro-operation are the functional, or atomic, operation of a CPU.

The fetch cycle - The fetch cycle, which occurs at the beginning of each instruction cycle and causes an instruction to be fetched from memory.

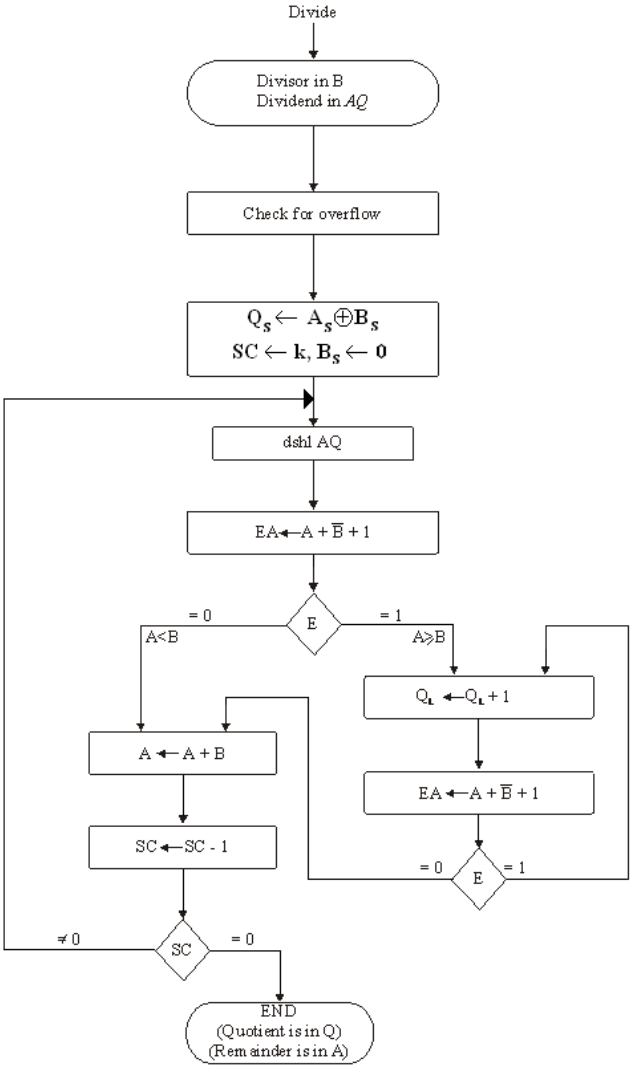
Instruction cycle:- The instruction cycle can be decomposed into a sequence of elementary micro-operations. There is one sequence each for the fetch, indirect and interrupt cycle and for execute cycle, there is one sequence of micro-operations for each opcode.

Micro - operations :-

- A computer executes a program
- Fetch/ execute cycle
- Each cycle has a number of steps are pipelining
- Called micro-operation
- Each step does very little

Q. 73 With neat flow chart, explain the procedure for division of floating point numbers carried out in a computer.

Ans:



Flowchart for decimal division

Decimal division is similar to binary division except of course that the quotient digits may have any of the 10 values from 0 to 9. In the restoring division method, the divisor is subtracted from the dividend or partial remainder as many times as necessary until a negative reminder results. The correct remainder is then restored by adding the divisor. The digit in the quotient reflects the number of subtractions up to but excluding the one that caused the negative difference. The decimal division algorithm is shown. It is similar to the algorithm with binary data except for the way the quotient bits are formed. The dividend (or partial remainder) is shifted to the left, with its most significant digit placed in A_e .The divisor is then subtracted by adding its 10's complement value. Since B_e is initially cleared, its complement value is 9 as required. The carry in E determines the relative magnitude of A and B, If E=0, it signifies that A<B. In this case the divisor is added to restore the partial remainder and Q_L stays at 0 (inserted there during the shift). If E=1, it signifies that A≥B. The quotient digit in Q_L is incremented once and the divisor subtracted again. This process is repeated until the subtraction results in a negative difference which is recognized by E being 0. When this occurs, the quotient digit is not incremented but the divisor is added to restore the positive remainder. In this way, the quotient digit is made equal to the number of times that the partial remainder "goes" into the divisor. The partial remainder and the quotient bits are shifted once to the left and the process is repeated k times to form k quotient digits. The remainder is then found in register A and the quotient is to register Q. The value of E is neglected.

Q.74 Give the flow table for register contents used in implementing booth's algorithm for the multiplier = - 6 and multiplicant = + 5.

Ans.

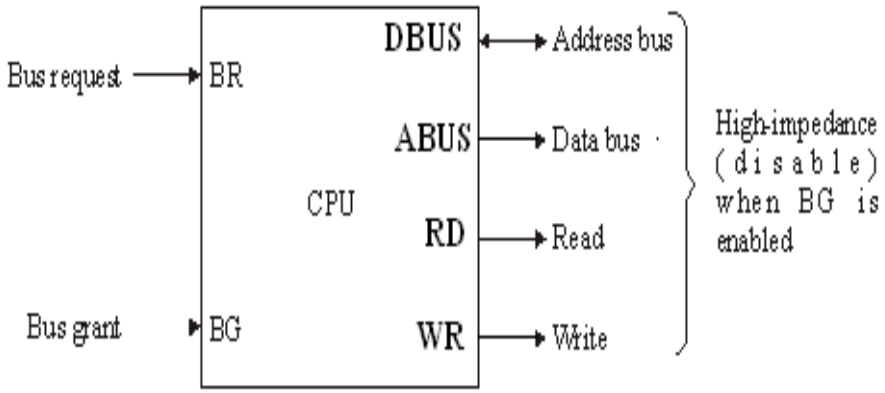
		BR = 0101				
Q_n	Q_{n+1}	$\overline{BR}+1$ = 1011	AC	QR	Q_{n+1}	SC
		Initial	0000	1010	0	100
0	0	Ashr	0000	0101	0	011
1	0	Subtract BR	$\frac{1011}{1011}$			
		ashr	1101	1010	1	010
0	1	Add BR	$\frac{0101}{0010}$			
		ashr	0001	0101	0	001
1	0	Subtract BR	$\frac{1011}{1100}$			
		ashr	1110	0010	1	000

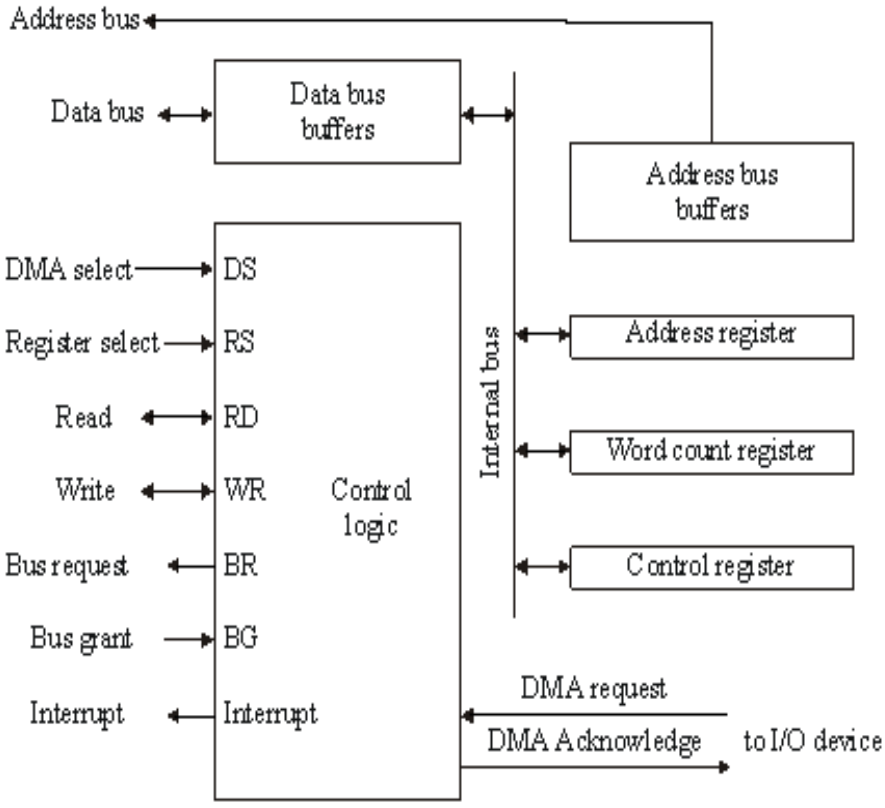
Final product is 11100010

Q.75 What do you mean by initialization of DMA controller? How DMA controller works? Explain with suitable block diagram.

Ans.
Figure below shows two control signals in the CPU that facilitate the DMA transfer. The bus request (BR) input is used by the DMA controller to request the CPU to

relinquish control of the buses. The CPU activates the bus grant (BG) output to inform the external DMA that the buses are in the high-impedance state. The DMA that originated the bus request can now take control of the buses to conduct memory transfers without processor intervention. When the DMA terminates the transfer, it disables the bus request line. The CPU disables the bus grant.





When the DMA takes control of the bus system, it communicates directly with the memory. Figure shows the block diagram of a typical DMA controller. The unit communicates with the CPU via the data bus and control lines. The registers in the DMA are selected by the CPU through the address bus by enabling the *DS* (DMA

select) and *RS* (register select) inputs. The *RD* (read) and *WR* (write) inputs are bidirectional. When the *BG* (bus grant) input is 0, the CPU can communicate with the DMA registers through the data bus to read from or write to the DMA registers. When *BG* = 1, the CPU has relinquished the buses and the DMA can communicate directly with the memory by specifying an address in the address bus and activating the *RD* or *WR* control.

The DMA controller has three registers: an address register, a word count register, and a control register. For each word that is transferred, the DMA increments its address registers and decrements its word count register. If the word count does not reach zero, the DMA checks the request line coming from the peripheral. For a high-speed device, the line will be active as soon as the previous transfer is completed. A second transfer is then initiated, and the process continues until the entire block is transferred. If the peripheral speed is slower, the DMA request line may come somewhat later. In this case the DMA disables the bus request line so that the CPU can continue to execute its program. When the peripheral requests a transfer, the DMA requests the buses again.

Q.76 The access time of a cache memory is 120 ns and that of main memory 900 ns. It is estimated that 80% of the memory requests are for read and remaining 20% for write. The hit ratio for read access only is 0.9. A write-through procedure is used.

(i) What is the average access time of the system considering only memory real cycles?
(ii) What is the hit ratio taking in to consideration the write cycle?
(iii) What is the average access time of the system for both read and write requests.

Ans.

i) 0.9 x 120 + 0.1 x 11000 = 108 + 110 = 218 nsec.
cache access memory access

ii) 0.2 x 900 + 0.8 x 200 = 180 + 100 = 280 nsec.
write access

iii) Hit ratio = 0.8 x 0.9 = 0.72

Q.77 Write short notes an any two of the followings.

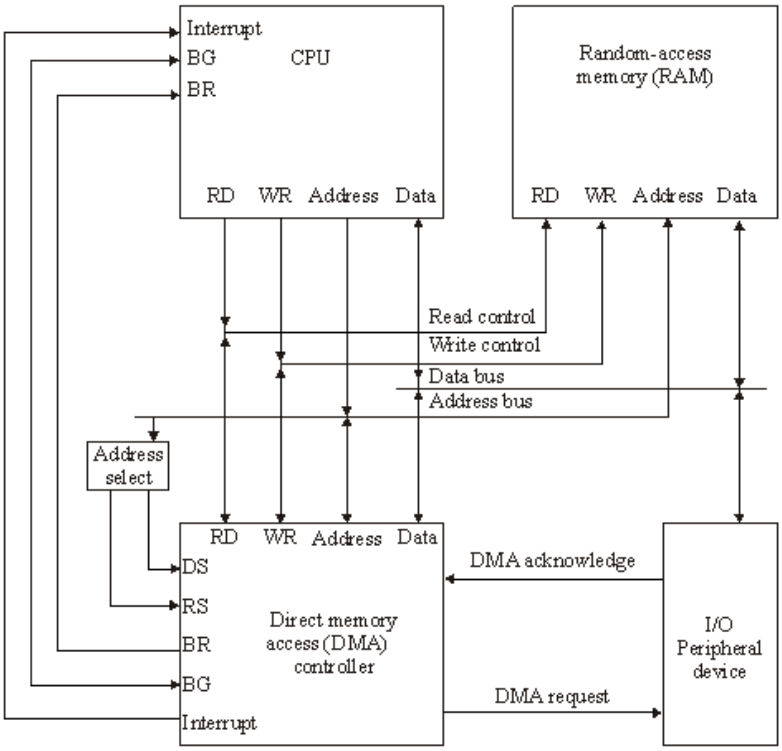
(i) DMA data transfer.
(ii) Handshaking method of data transfer
(iii) Isolated Vs memory mapped I/O.
(iv) RISC architecture.

Ans.

i) **DMA data transfer**:-

The position of the DMA controller among the other components in a computer system is shown in figure. The CPU communicates with the DMA through the address and data buses as with any interface unit. The DMA has its own address, which activates the *DS* and *RS* lines. The CPU initializes the DMA through the data bus. Once the DMA receives the start control command, it can start the transfer between the peripheral device and the memory. When the peripheral device receives a DMA acknowledge, it puts a word in the data bus (for write) or receives a word

from the data bus (for read). Thus the DMA controls the read or write operations and supplies the address for the memory. The peripheral unit can

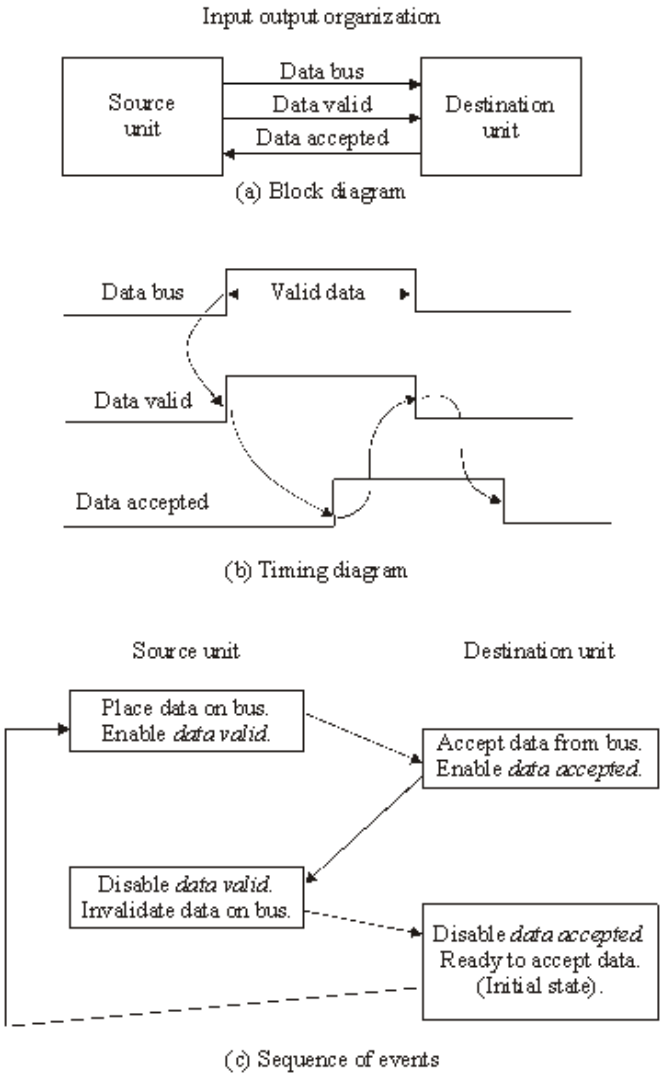


then communicate with memory through the data bus for direct transfer between the two units while the CPU is momentarily disabled. For each word that is transferred, the DMA increments its address registers and decrements its word count register. If the word count does not reach zero, the DMA checks the request line coming from the peripheral. For a high-speed device, the line will be active as soon as the previous transfer is completed. A second transfer is then initiated, and the process continues until the entire block is transferred. If the peripheral speed is slower, the DMA request line may come somewhat later. In this case the DMA disables the bus request line so that the CPU can continue to execute its program. When the peripheral requests a transfer, the DMA can continue to execute its program. When the peripheral requests a transfer, the DMA requests the buses again.

DMA transfer is very useful in many applications. It is used for fast transfer of information between magnetic disks and memory. It is also useful for updating the display in an interactive terminal. Typically, an image of the screen display of the terminal is kept in memory which can be updated under program control. The contents of the memory can be transferred to the screen periodically by means of DMA transfer.

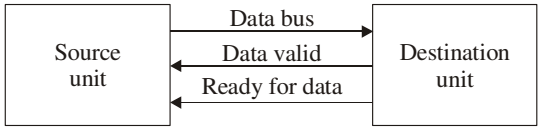
ii) **Handshaking method of data transfer**:-

The disadvantage of the strobe method is that the source unit that initiates the transfer has no way of knowing whether the destination unit has actually received the data item that was placed in the bus. Similarly, a destination unit that initiates the transfer has no way of knowing whether the source unit has actually placed the data on the bus.

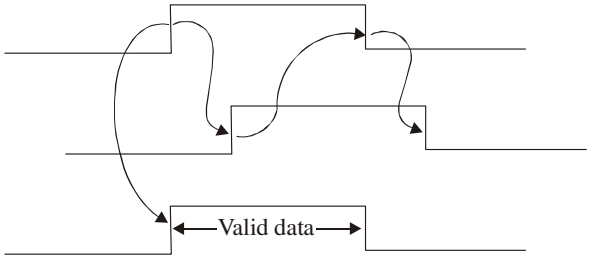


The two handshaking lines are *data valid*, which is generated by the source unit, and *data accepted,* generated by the destination unit. The timing diagram shows the exchange of signals between the two units. The sequence of events listed in part (C) shows the four possible states that the system can be at any given time. The source unit initiates the transfer by placing the data on the bus and enabling its *data valid* signal. The *data accepted* signal is activated by the destination unit after it accepts the data from the bus. The source unit then disables its *data valid* signal, which invalidates the data on the bus. The destination unit then disables its *data accepted* signal and the system goes into its initial state. The source does not send the next data item until after the destination unit shows its readiness to accept new data by disabling its *data accepted* signal. This scheme allows arbitrary delays from one state to the next and permits each unit to respond at its own data transfer rate. The rate of transfer is determined by the slowest unit.
The destination-initiated transfer using handshaking lines is shown. Notes that the name of the signal generated by the destination unit

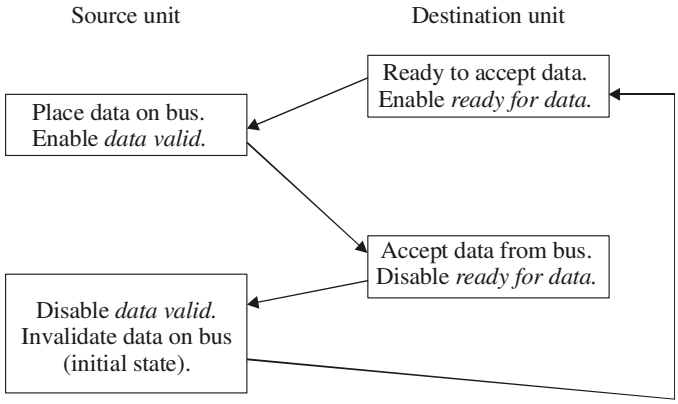
Destination-initiated transfer using handshaking.



(a) Block diagram



(b) Timing diagram



(c) Sequence of events

has been changed to *ready for data* to reflect its new meaning. The source unit in this case does not place data on the bus until after it receives the *ready for data* signal from the destination unit. From there on, the handshaking procedure follows the same pattern as in the source-initiated case. Note that the sequence of events in both cases would be identical if we consider the *ready for data* signal as the complement of *data accepted*. In fact, the only difference between the source-initiated and the destination-initiated transfer is in their choice of initial state.

iii) **Isolated vs memory mapped I/O:-**

In the isolated I/O configuration, the CPU has distinct input and output instructions, and each of these instructions is associated with the address of an interface register. When the CPU fetches and decodes the operation code of an input or output instruction, it places the address associated with the instruction into the common address lines. The isolated I/O method isolates memory and I/O addresses so that memory address values are not affected by interface address assignment since each has its own address space. The other alternative is to use the same address space for

both memory and I/O. This configuration is referred to as *memory-mapped I/O*. In a memory-mapped I/O organization there is no specific input or output instructions. The CPU can manipulate I/O data residing in interface registers with the same instructions that are used to manipulate memory words. Computers with memory-mapped I/O can use memory-type instructions to access I/O data. The advantage is that the load and store instructions used for reading and writing from memory can be used to input and output data from I/O registers. In a typical computer, there are more memory-reference instructions than I/O instructions. With memory-mapped I/O all instructions that refer to memory are also available for I/O.

iv) **RISC architecture**:-

The concept of RISC architecture involves an attempt to reduce execution time by simplifying the instruction set of the computer. The major characteristics of a RISC processor are:

1. Relatively few instructions.
2. Relatively few addressing modes.
3. Memory access limited to load and store instructions.
4. All operations done within the registers of the CPU.
5. Fixed-length, easily decoded instruction format.
6. Single-cycle instruction execution.
7. Hardwired rather than microprogrammed control.
8. Faster execution.

Other characteristics attributed to RISC architecture are:

1. A relatively large number of registers in the processor unit.
2. Use of overlapped register windows to speed-up procedure call and return.
3. Efficient instruction pipeline.
4. Compiler support for efficient translation of high-level language programs into machine language programs.

A large number of registers is useful for storing intermediate results and for optimizing operand references. The advantage of register storage as opposed to memory storage is that registers can transfer information to other registers much faster than the transfer of information to and from memory. Thus register-to-memory operations can be minimized by keeping the most frequent accessed operands in registers. Studies that show improved performance for RISC architecture do not differentiate between the effects of the reduced instruction set and the effects of a large register file.

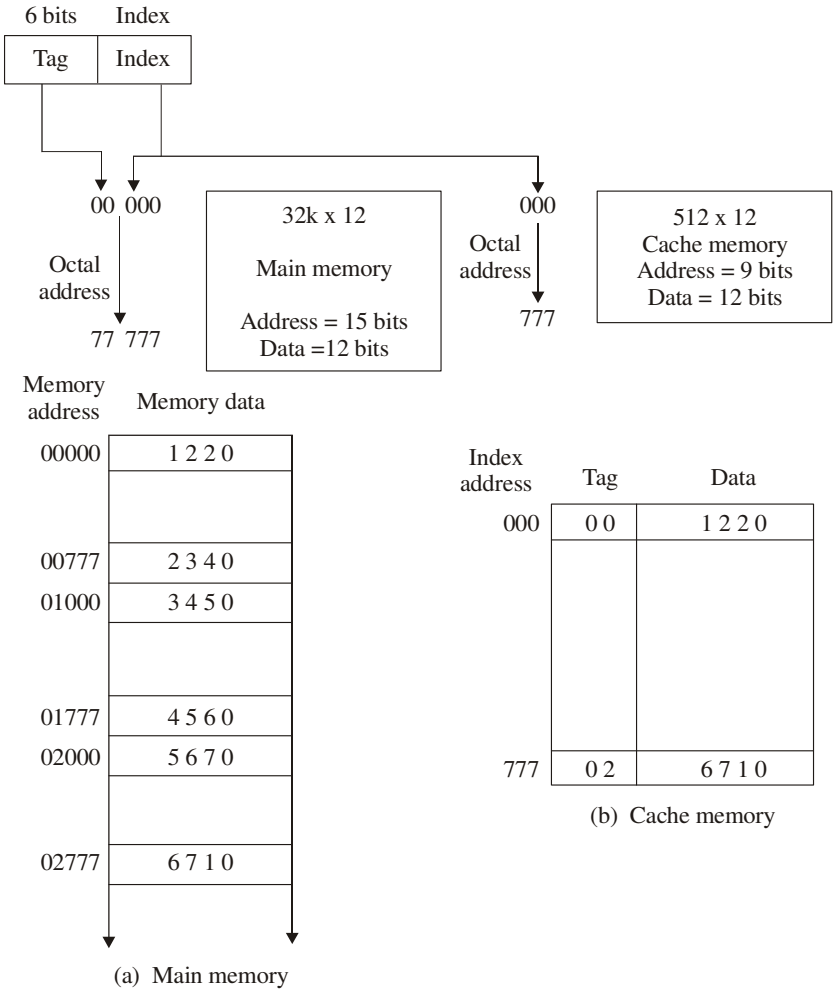
Q. 78 Explain direct mapping of cache memory system.

Ans.

Associative memories are expensive compared to random-access memories because of the added logic associated with each cell. The possibility of using a random-access memory for the cache is investigated. The CPU address of 15 bits is divided into two fields. The nine least significant bits constitute the *index* field and the remaining six bits form the *tag* field. The figure shows that main memory needs an address that includes both the tag and the index bits. The number of bits in the index field is equal to the number of address bits required to access the cache memory. In the general case, there are 2^k words in cache memory and 2^n words in

main memory. The n bit memory address is divided into two fields: k bits for the index field and the *n*-k bits for the tag field. The direct mapping cache organization uses the *n*-bit address to access the main memory and the k-bit index to access the cache. The internal organization of the words in the cache memory is as shown. Each word in cache consists of the data word and its associated tag. When a new word is first brought into the cache the tag bits are stored alongside the data bits. When the CPU generates a memory request, the index field is used for the address to access that cache. The tag field of the CPU address is compared with the tag in the word read from the cache. If the two tags match, there is a hit and the desired data word is in cache. If there is no match, there is a miss and the required word is read from main memory. It is then stored in the cache together with the new tag, replacing the previous value. The disadvantage of direct mapping is that the hit ratio can drop considerably if two or more words whose addresses have the same index but different tags are accessed repeatedly. However, this possibility is minimized by the fact that such words are

Addressing relationships between main and cache memories



Direct mapping cache organization

Relatively far apart in the address range. To see how the direct-mapping organization operates, consider the numerical example shown. The word at address

zero is presently stored in the cache (index = 000, tag = 00, data = 1220). Suppose that the CPU now wants to access the word at address 02000. The index address is 000, so it is used to access the cache. The two tags are then compared. The cache tag is 00 but the address tag is 02, which does not produce a match. Therefore, the main memory is accessed and the ata word 5670 is transferred to the CPU. The cache word at index address 000 is then replaced with a tag of 02 and data of 5670.

Q.79 What do you mean by locality of reference?

Ans.
The references to memory at any given internal of time tend to be confined within a few localized areas in memory. This phenomenon is known as the property of locality of reference. The locality of reference property, that over a short internal of time, the addresses generated by a typical program refer to a few localized area of memory repeatedly, while the remainder of memory is accessed relatively infrequently.

Q.80 A virtual memory system has an address space of 8k words, memory space of 4k words and Page & Block size of 1k words. The following page reference changes occur during a given time interval.

4, 2, 0, 1, 2, 6, 1, 4, 0, 1, 0, 2, 3, 5, 7

Determine the four pages that are resident in main memory after each Page reference change if the replacement algorithm used is (i)FIFO (ii) LRU.

Ans.
An address space of 8K and a memory of 4K words and page Block size of 1K words. Four pages of address space may reside in main memory in any one of the four blocks.

Page 0	
Page 1	
Page 2	Block 0
Page 3	Block 1
Page 4	Block 2
Page 5	Block 3
Page 6	
Page 7	

Address Space
$N = 8K = 2^{13}$

Memory Space
$M = 4K = 2^{12}$

(1) **FIFO**
String 4, 2, 0, 1, 2, 6, 1, 4, 0, 1, 0, 2, 3, 5, 7

4	2	0	1	2	6	1	4	0	1	0	2	3	5	7
4	4	4	4		6		6				6	6	5	5
	2	2	2		2		4				4	4	4	7
		0	0		0		0				2	2	2	2
			1		1		1				1	3	3	3

Page fault by FIFO = 10

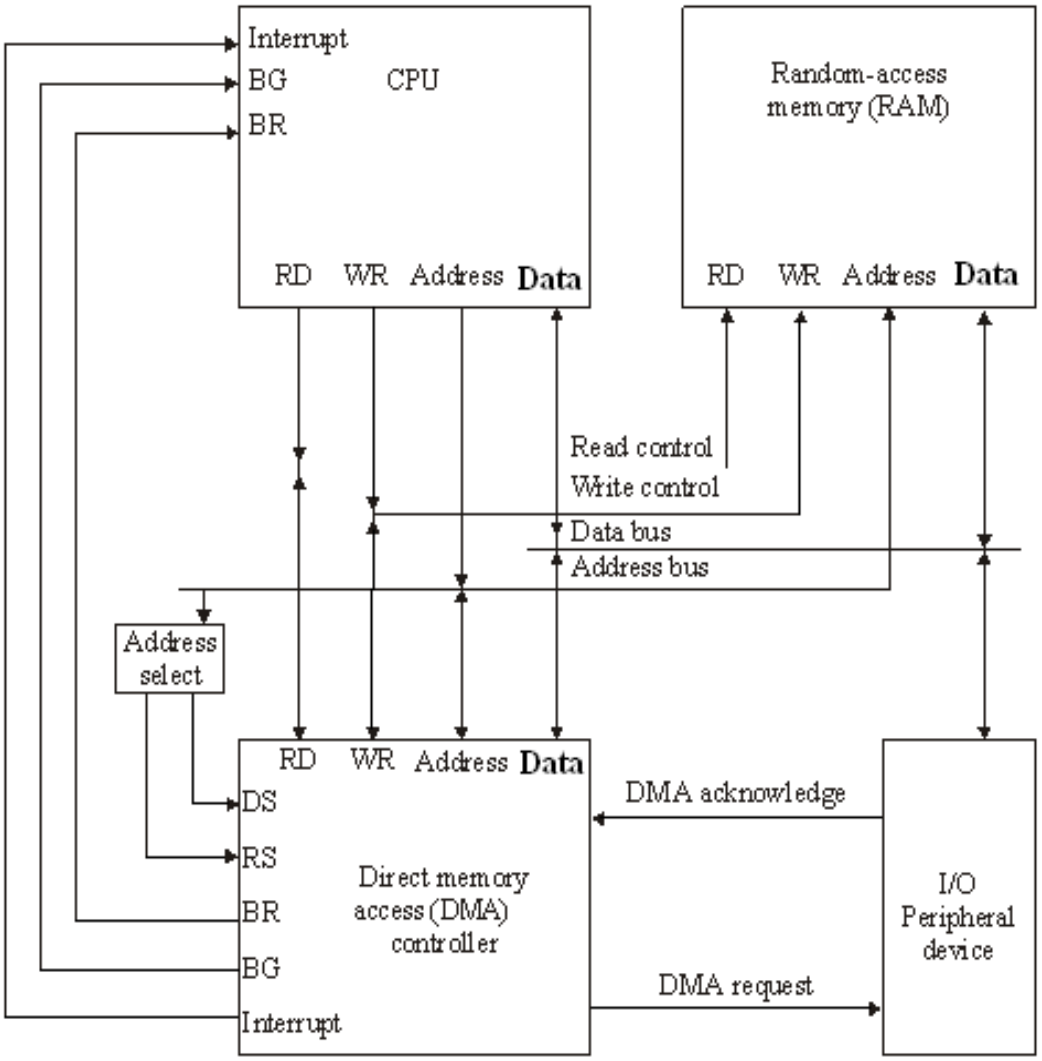
(2) **LRU**

4	2	0	1	2	6	1	4	0	1	0	2	3	5	7
4	4	4	4		6		6	6			2	2	2	2
	2	2	2		2		2	0			0	0	0	7
		0	0		0		4	4			4	3	3	3
			1		1		1	1			1	1	5	5

Page fault by LRU = 11

Q.81 With neat block diagram, explain how DMA controller is initialized for DMA data transfer.

Ans.



The position of the DMA controller among the other components in a computer system is illustrated in Fig. The CPU communicates with the DMA through the address and data buses as with any interface unit. The DMA has its own address, which activates the DS and RS lines. The CPU initializes the DMA through the data bus. Once the DMA receives the start control command, it can start the transfer between the peripheral device and the memory. When the peripheral device sends a DMA request, the DMA controller activates the BR line, informing the CPU to relinquish the buses. The CPU responds with its BG line, informing the DMA that its buses are disabled. The DMA then puts the current value of its address register into the address bus, initiates the RD or WR signal, and sends a DMA acknowledge to the peripheral device. Note that the RD and WR lines in the DMA controller are bi-directional. The direction of transfer depends on the status of the BG line. When BG = 0, the RD and WR are input lines allowing the CPU to communicate with the internal DMA controller to the random-access memory to specify the read or write operation for the data. When the peripheral device receives a DMA acknowledge, it puts a word in the data bus (for write) or receives a word from the data bus (for read). Thus the DMA controls the read or write operations and supplies the address for the memory. The peripheral unit can then communicate with memory through the data bus for direct transfer between the two units while the CPU is momentarily disabled.

Q.82 How data is transmitted in synchronous serial communication system?

Ans.

Synchronous transmission does not use start-stop bits to frame characters and therefore makes more efficient. In synchronous transmission, where an entire block of character is transmitted, each character has a parity bit for the receiver to check. After the entire block is sent the transmitter sends one more character that constitutes a parity over the length of the message. This character is called a longitudinal redundancy check (LRC) and is the accumulation of the exclusive OR of all transmitted character. The receiving station calculates the LRC as it receives characters and compares it with the transmitted LRC. The calculated and received LRC should be equal for error-free messages. If the receiver finds an error in the transmitted block, it inform the sender to retransmit the same block once again.

Q.83 How many characters per second can be transmitted over a 1200 baud line in asynchronous serial transmission in following modes – assume a character code is of eight bits?

(i) Synchronous Serial transfer
(ii) Asynchronous Serial Transfer with 2 stop bits
(iii) Asynchronous Serial Transfer with one stop bit.

Ans.

Baud Rate = 1200
Character Code = 8 bits
(i) Transmitted Characters per second in Synchronous Serial transmission
Transfer = 1200/8=150 Character per second
(ii) Asynchronous Serial Transfer with 2 stop bits
total number of bits = 1 start bit + 8 information bits + 2 stop bits

= 1+8+2 = 11 bits
baud rate = 1200
Transmitted Character per second = 1200/11
= 109 Character per second
(iii) Asynchronous Serial Transfer with one stop bit.
Total no. of bits = 1 start bit + 8 information bits + 1 stop bit
= 10 bits
band rate = 1200
►Transmitted Character per second = 1200/10 = 120 Character per second

Q.84 A Computer uses a memory unit with 256K words of 32 bits each. A binary instruction code is stored in one word of memory. The instruction has four parts: an indirect bit, an operation code, a register code part to specify one of 64 registers and an address part.

(i) How many bits are there in the operation code, the register code part and the address part?
(ii) Draw the instruction word format and indicate the number of bits in each part.
(iii) How many bits are there in the data and address inputs of the memory?

Ans.

(i) For a memory unit with 256K words of 32 bits each we need 18 bits to specify an address.
Operation Code = 7 bits
32-25 = 7 bits for opcode
Register Code = 6 bits
> $2^6 = 64$

(ii) Instruction Word Format

	1	7	6	18 = 32 bits
	I	Opcode	Register Code	Address

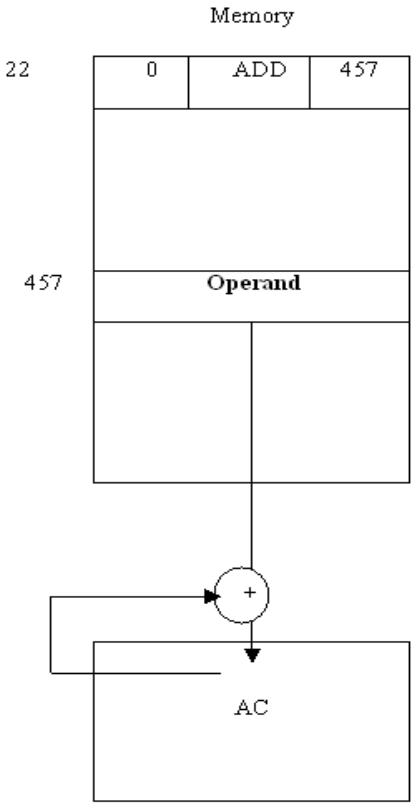
(iii) data and address inputs of the memory
data = 32 bits
address = 18 bits

Q.85 What is difference between a direct and an indirect address instruction? How many references to memory are needed for each type of instruction to bring an operand into a processor register?

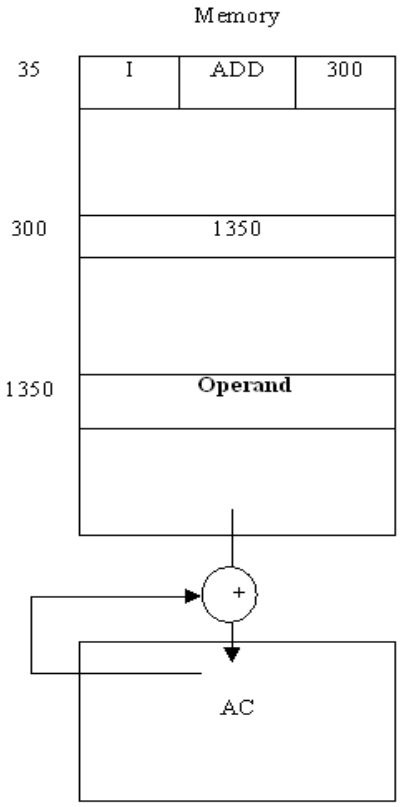
Ans.

15	14	12	11
1	OP Code	Address	

Instruction format fig. a



Direct Address
Fig.b



Indirect address
Fig.c

Direct address:-

Instruction code format shown in fig. a. It consists of a 3-bit operation code. A 12 bit address, and an indirect address mode bit designated by I. The mode bit is 0 for a direct address and I for an indirect address. A direct address instruction is shown in fig. b. It is placed in address 22 in memory. The I bit is 0, so the instructors is recognized as a direct address instruction. The operation code specifies an ADD instruction, and the address part is the binary equivalent of 457. The control finds the operand in memory at address 457 and adds is to the content of AC.

Indirect address:-

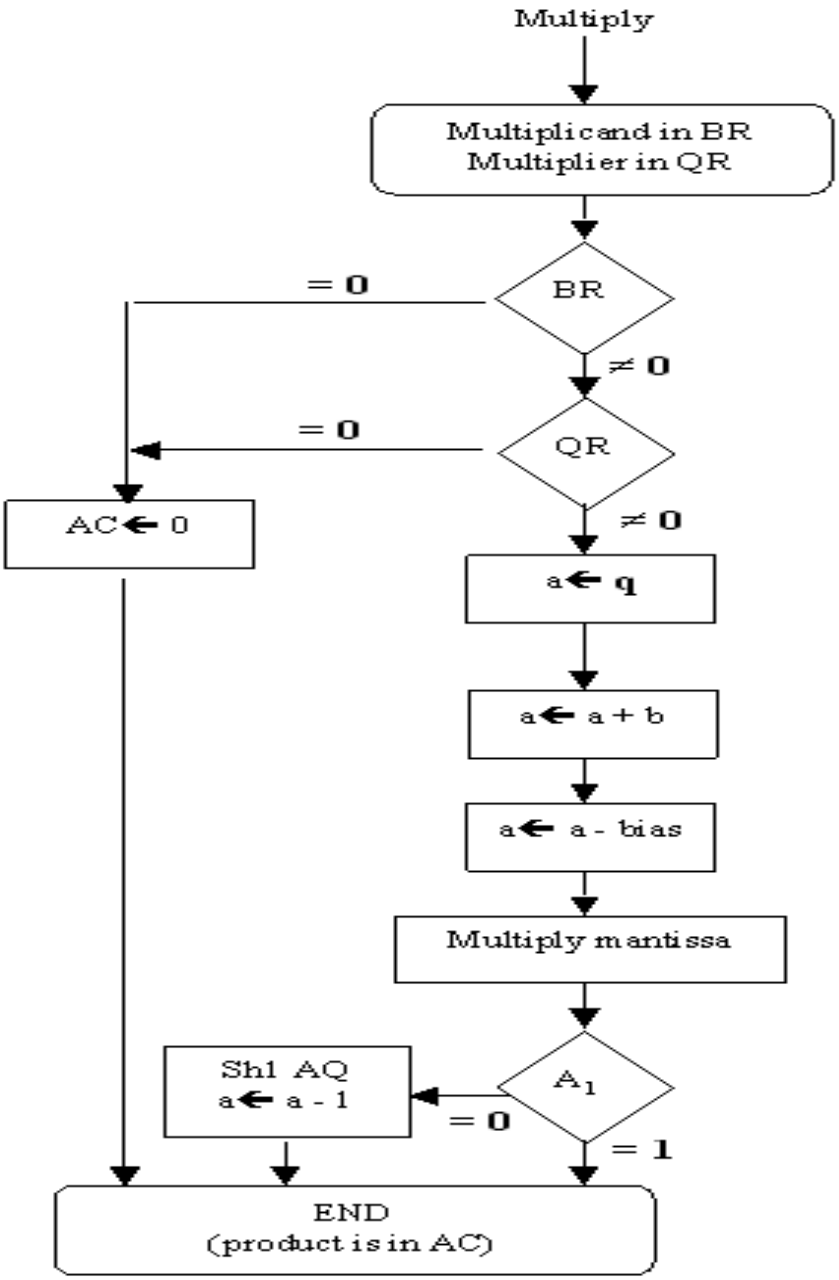
The instruction in address 35 shown in fig. c has a mode bit I=1. Therefore, it is recognized as an indirect address instruction. The address part is the binary equivalent of 300. The control goes to address 300 to find the address of the operand. The address of the operand in this case is 1350. The operand found in address 1350 is then added to the content of AC. The indirect address instruction needs two references to memory to fetch and operand. The first reference is needed to read the address of the operand the second is for the operand itself. The effective address to be the address of the operand in a computation type instruction or the target address in a branch-type instruction.

(1) ADD to AC
(2) ADD to AC
(3) LDA: Load to AC
(4) STA: Store AC

(5) BUN: Branch Unconditionally
(6) BSA: Branch and Save Return Address
(7) ISZ: Increment and Skip to Zero

Q.86 With neat flow chart discuss the procedure for floating point multiplication. Explain with the help of an example.

Ans.



The multiplication of two floating-point numbers requires that we multiply the mantissas and add the exponents.
The multiplication algorithm can be subdivided into four parts:

1. Check for zeros.
2. Add the exponents.
3. Multiply the mantissas.
4. Normalize the product.

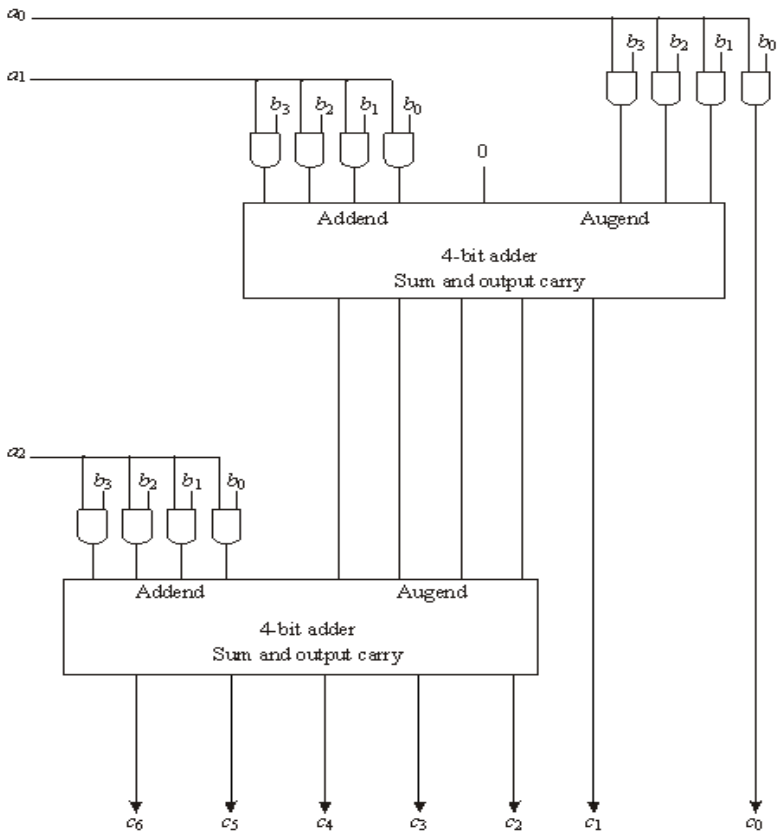
Steps 2 and 3 can be done simultaneously if separate adders are available for the mantissas the exponents.

The flowchart for floating-point multiplication is shown in Fig. The two operands are checked to determine if they contain a zero. If either operand is equal to zero, the product in the AC is set to zero and the operation is terminated. If neither of the operands is equal to zero, the process continues with the exponent addition.

The exponent of the multiplier is in q and the adder is between exponents a and b. It is necessary to transfer the exponents from q to a, add the two exponents, and transfer the sum into a. Since both exponents are biased by the addition of a constant, the exponent sum will have double this bias. The correct biased exponent for the product is obtained by subtracting the bias number from the sum. The mantissas are then multiplied as in the fixed – point case with the product residing in A and Q. Overflow cannot occur during multiplication, the product may have an underflow, so the most significant bit in A is checked. If it is a 1, the product is already normalized. If it is a 0, the mantissa in AQ is shifted left and the exponent decremented.

Q. 87 Design a Three-bit array multiplier. Use AND gates and binary address.

Ans.



Q.88 Write a program to evaluate the arithmetic statement.

$$P=\frac{(X-Y+Z)*(M*n-o)}{Q+R*S}$$

By using

(i) Two address instructions
(ii) One address instructions
(iii) Zero address instructions

Ans.

(i) Two Address Instruction

MOV	R1 X	R1 ← M [X]
ADD	R1, Z	R1 ← R1 + M[Z]
SUB	R1, Y	R1 ←R1 - M[Y]
MOV	R2, m	R2 ← M[m]
MUL	R2, n	R2 ← R2 * M[n]
SUB	R2, o	R2 ←R2 – M[o]
MUL	R1, R2	R1 ← R1 x R2
MOV	R3, R	R3 ← R3 * M[R]
MUL	R3, S	R3 ← R3 * M[S]
ADD	R3, Q	R3 ← R3 + M[Q]
DIV	R1, R3	R1← R1 ¸ R3
MOV	P, R1	M[P] ← R1

(ii) One address Instruction

$$P=\frac{(X-Y+Z)*(M*n-o)}{Q+R*S}$$

LOAD	X	AC←M [X]
ADDZ		AC←AC + M[Z]
SUB	Y	AC← AC - M[Y]
STORE	T	M [T] ←AC
LOAD	M	AC←M[m]
MUL	N	AC←AC x M[n]
SUB	O	AC←AC – M[o)
MUL	T	AC←AC x M[T]
STORE	U	M[U] ← AC
LOAD	R	AC ← M [R]
MUL	S	AC← AC x M[S)
ADD	Q	AC ← AC + M [Q]
DIV	V	AC ← M[v]/ AC
STORE	P	M[P] ← AC

(iii) Zero address Instruction

$$P=\frac{(X-Y+Z)*(M*n-o)}{Q+R*S}$$

PUSH	X	TOS←X
PUSH	Y	TOS←Y

SUB		TOS ← X-Y
PUSH	Z	TOS ← Z
ADD		TOS ← (X-Y+Z)
PUSH	M	TOS ← M
PUSH	N	TOS ← N
MUL		TOS ← M*N
PUSH	O	TOS ← O
SUB		TOS ← (M*N-O)
MUL		TOS ← (X-Y+Z) x (M*N-O)
PUSH	R	TOS ← R
PUSH	S	TOS ← S
MUL		TOS ← R*S
PUSH	Q	TOS ← Q
ADD		TOS ← Q-R*S
DIV	O	
POP	P	M [P] ← TOS

Q.89 Convert the following arithmetic expression from infix notation to RPN.
A*B + B* (B*D+C*E)

Ans.
B*D+C*E
In RPN
BD*CE*+
Infix A*B+B*(BxD+CxE)
RPN AB*BBD*CE*+*+

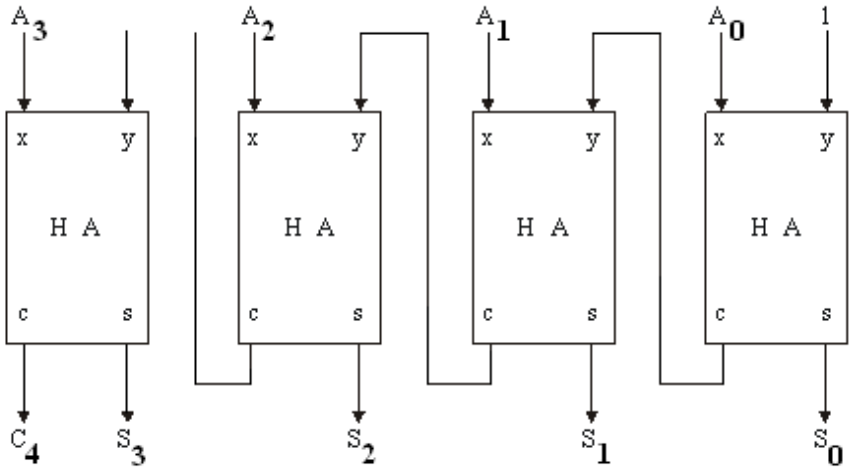
Q.90 What is subroutine? How it is executed by the processor? What is the importance of subroutine parameters and data linkage? How is it established?

Ans.
Subroutines:-
Frequently, the same piece of code must written over again in many different parts of a program. Instead of repeating the code every time it is needed, there is an obvious advantage if the common instructions are written only once. A set of common instructions that can be used in a program many times is called a *subroutine.* Each time that a subroutine is used in the main part of the program, a branch is executed to the beginning of the subroutine. After the subroutine has been executed, a branch it is made back to the main program.
Subroutine Parameters and Data Linkage:-
When a subroutine is called, the main program must transfer the data. The subroutine shifted the number and left it there to be accepted by the main program. It is necessary for the subroutine to have access to data from the calling program and to return results to that program. The accumulator can be used for a single input parameter and a single output parameter.
Consider a subroutine that performs the logic OR operation. *Two* operands must be transferred to the subroutine and the subroutine must return the result of the operation. The accumulator can be used to transfer one operand and to receive the result. The other operand is inserted in the location following the BSA instruction.

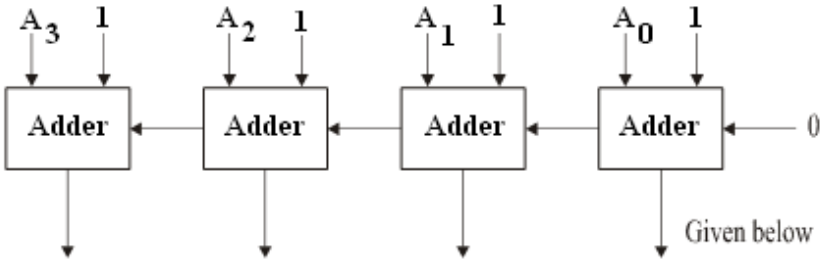
The subroutine must increment the return address stored in its first location for each operand that it extracts from the calling program. Moreover, the calling program can reserve one or more locations for the subroutine to return results that are computed. The first location in the subroutine must be incremented for these locations as well, before the return. If there is a large amount of data to be transferred , the data can be placed in a block of storage and the address of the first item in the block is then used as the linking parameter. A subroutine that moves a block of data starting at address into a block starting with address. The length of the block is 16 words. The first introduction is a branch to subroutine MVE. The items are retrieved from their blocks by the use of two pointers. The counter ensures that only 16 items are moved. When the subroutine completes its operation, the data required is in the block starting address. The return to the main program is to the HLT instruction.

Q.91 Design a 4-bit combinational incrementer and decrementer circuit.

Ans.



Incrementer Circuit



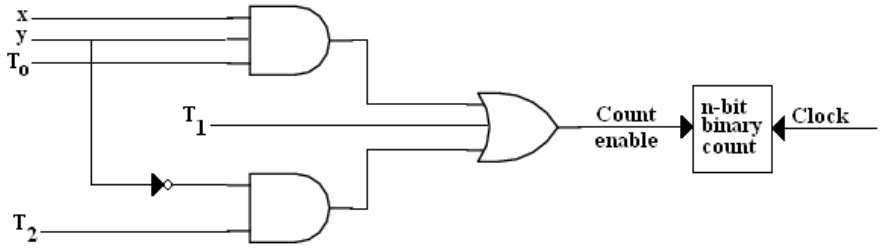
Decrementer Circuit

Q.92 Show the hardware implementation of following statement:

$xyT_0+T_1+yT_2: AR \leftarrow AR+1$

Where x, y are control functions and T_0, T_1, T_2 are T - State

Ans.

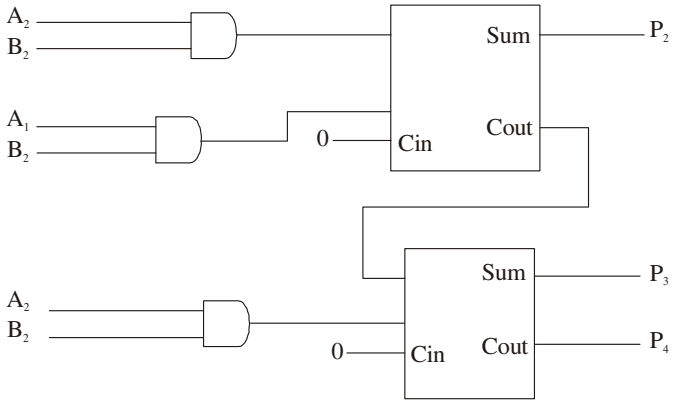


Q.93 Represent the given conditional control statement by two register transfer Statements with Control functions.
If (P = 1), Than $R_1 \leftarrow R_2$ else if (Q = 1) Than $R_1 \leftarrow R_3$. **(2)**

Ans: The two resistor transfer statements are :
P: $R_1 \leftarrow R_2$
P' Q: $R_1 \leftarrow R_3$

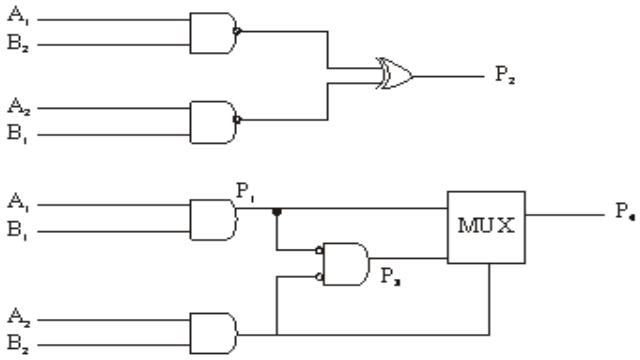
Q.94 Implement a 2 - bit multiplier circuit by using multipliers.

Ans.



2 bit multiplier circuit

By using multiplier



Q. 95 If $P = \sum(m_0, m_3, m_4, m_8, m_{12}, m_{13}, m_{15})$ and $Q = \Pi(M_1, M_4, M_5, M_{12}, M_{14})$, Then find the expression for $X = P \oplus Q$ in SOP & POS. **(8)**

Ans.

$P = \sum(m_0, m_3, m_4, m_8, m_{12}, m_{13}, m_{15})$

$Q = \pi (M_1, M_4, M_5, M_{12}, M_{14})$

$M_o = 0000$, $M_3 = 0011$, $M_4 = 0100$

$M_8 = 1000$, $M_{12} = 1100$, $M_{13} = 1101$, $M_{15} = 1111$

$M_1 = 0001$, $M_4 = 0100$, $M_5 = 0101$,

$M_{12} = 1100$, M14 = 1110

K - Map for SOP

CD \ AB	A'B'	A'B	AB	AB'
C'D'	1	0	1	X
C'D	1	0	X	X
CD	1	1	1	0
CD'	1	X	X	X

F = A'B' + CDB + C'D'A

F = C'D + C'A'B + DB'

= (C + 0')(C + A + B')(0' + B)

	A'B'	A'B	AB	AB'
C'D'	1	0	1	X
C'D	0	0	X	0
CD	0	1	1	0
CD'	1	X	X	X

F = (D' + B) (C + A + B')

Q. 96 What is excitation table? List the excitation table for SR-FF, JK-FF D-FF and T-FF.

Ans.

Flip-flop specifies the next state when the input and the present state are known. During the design of sequential circuits, the required transition form present state to next state and to find the FF input conditions that will cause the required transition. For this reason we need a table that lists the required input combinations for a given change of state. Such a table is called a flip-flop excitation table.

SR Flip-flop			
Q(t)	Q(t+1)	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

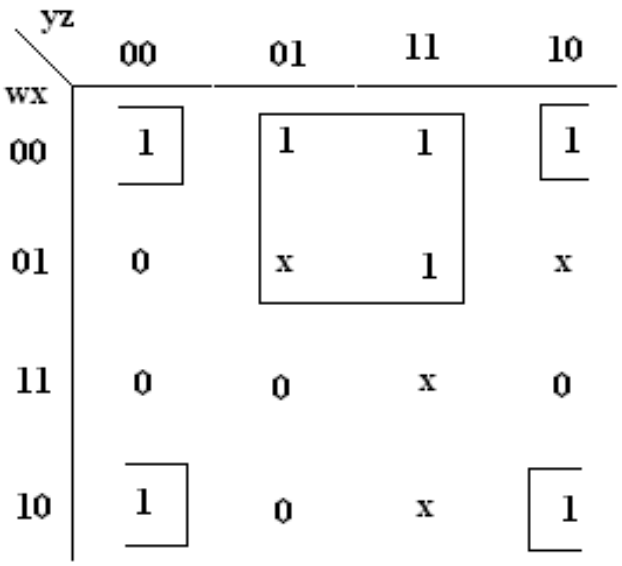
D Flip-flop		
Q(t)	Q(t+1)	DR
0	0	0
0	1	1
1	0	0
1	1	1

JK flip-flop			
Q(t)	Q(t+1)	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

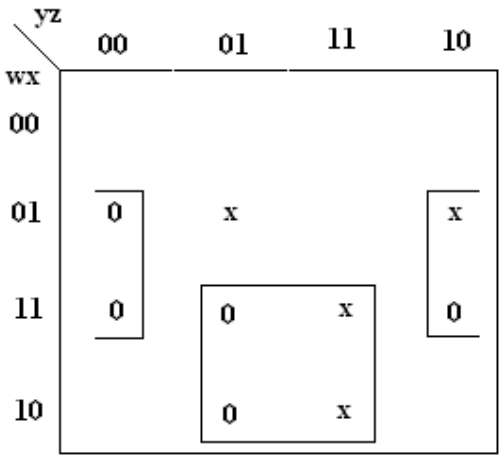
T flip-flop		
Q(t)	Q(t+1)	DR
0	0	0
0	1	1
1	0	1
1	1	0

Q. 97 Simplify the Boolean function F together with don't care condition D in
(i) Sum of Product (ii) Product of sums
F(w, x, y, z) = Σ (0, 1, 2,3, 7, 8, 10)
D(w, x, y, z) = Σ (5, 6, 11, 15)

Ans. (i)



.(ii)

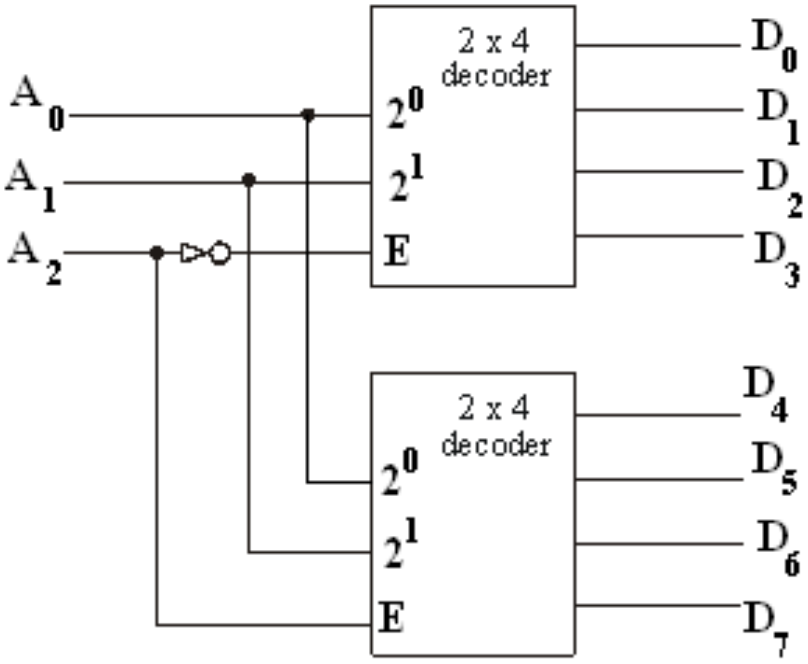


$F = (w' + z')\ (x' + z)$

Q. 98 Design a 3x8 decoder with the help of two 2x4 decoders.

Ans.

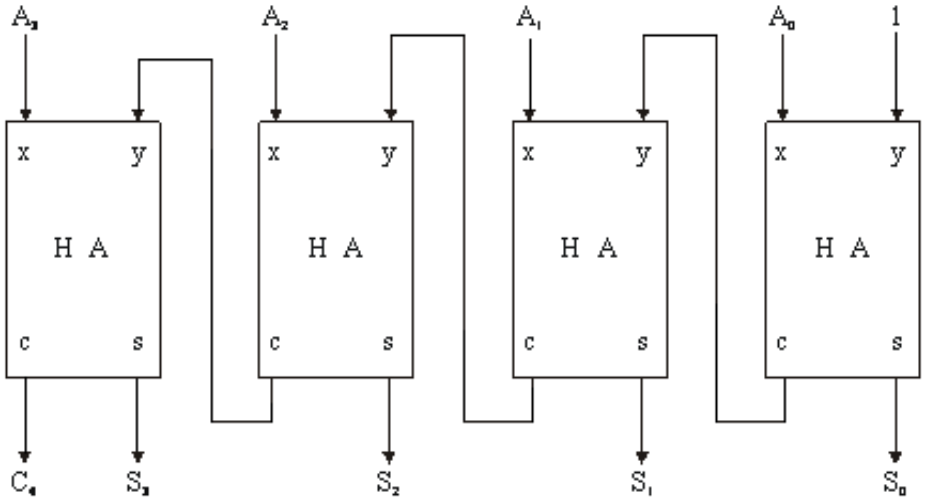
Decoder circuit



Q. 99 Design a binary Incrementer and binary Decrementer.

Ans.

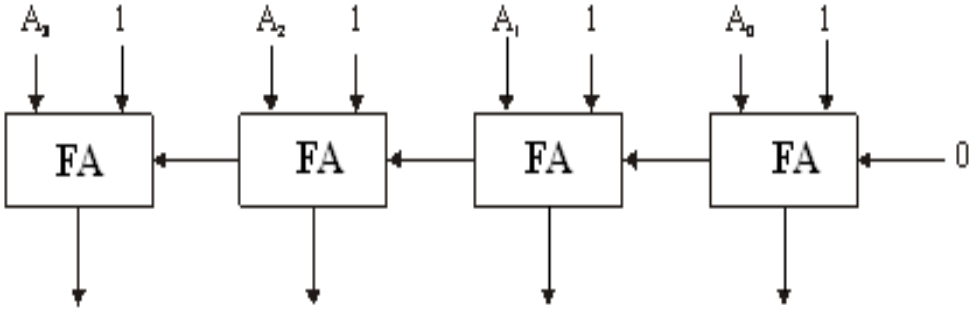
Incrementer circuit



4 bit binary incrementer

Decrementer circuit

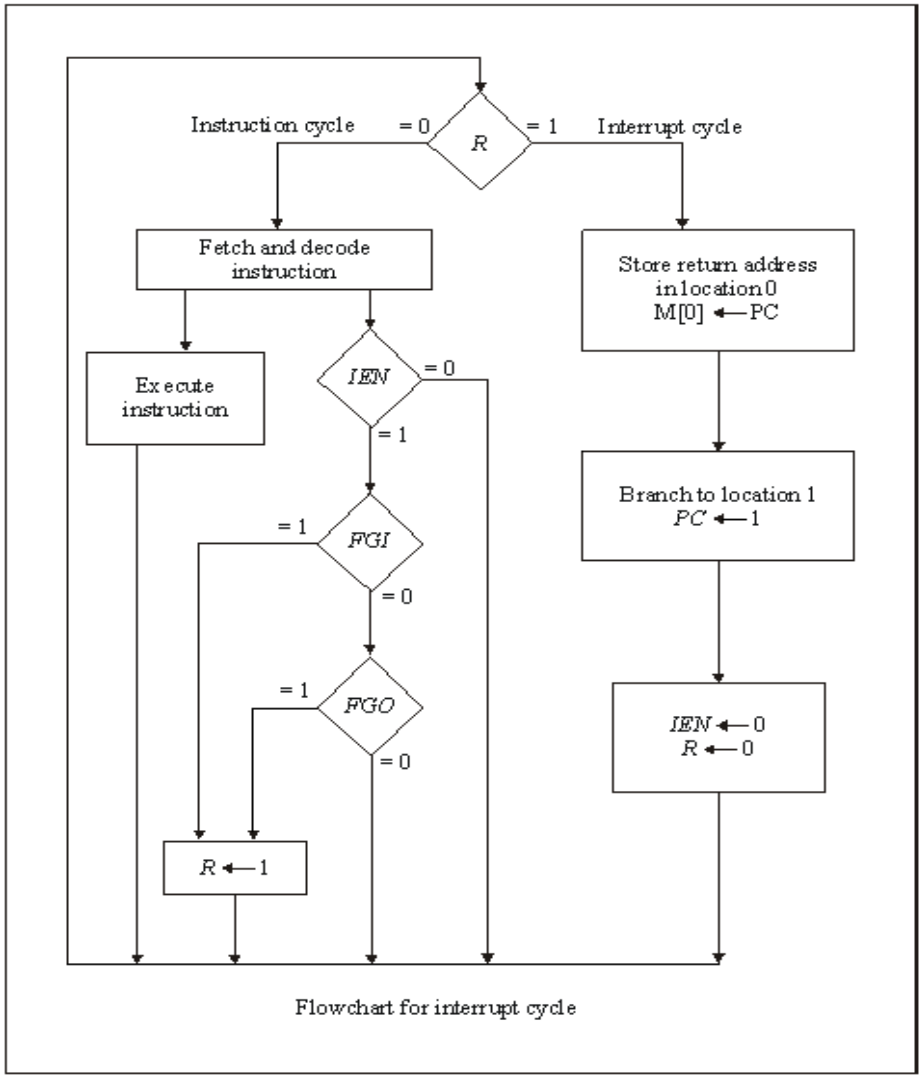
A – 1 = A + 2's complement of 1 = A + 1111



Given below

Q. 100 How does a basic computer handle an interrupt? Explain what happens during the interrupt with the help of an example. Also, give the register transfer statements.

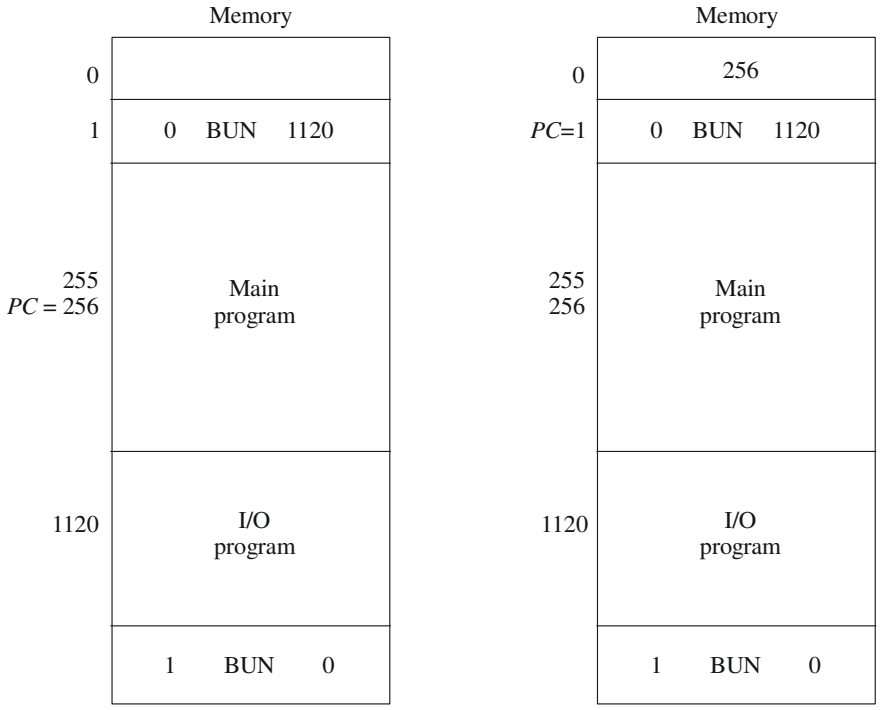
Ans.



Flowchart for interrupt cycle

The interrupt is handled by the computer can be explained by means of the flowchart. An interrupt flip-flop R is included in the computer. When R = 0, the computer goes through an instruction cycle. During the execute phase of the instruction cycle IEN in checked by the control. If it is 0, it indicates that the programmer does not want to use the interrupt, so control continues with the next instruction cycle. If IEN is 1, control checks the flag bits. If both flags are 0, it indicates that neither the input nor the output registers are ready for transfer of information. If either flag is set to 1 while IEN = 1, flip- flop R is set to 1. At the end of the execute phase, control checks the value of R, and if it is equal to 1, it goes to an interrupt cycle instead of an instruction cycle.

An example that shows, during the interrupt cycle is shown in fig. that an interrupt occurs and R is set to 1 while the control is executing the instruction at address 255. The return address 256 is in PC. The programmer has previously placed an input-output service program in memory starting from address 1120 and a BUN 1120 instruction at address 1.



Register transfer statement:-

The interrupt cycle is initiated after the last execute phase if the interrupt flip-flop is equal to 1. This flip-flop is set to 1 if IEN = 1 and either FG1 or FG0 are equal to 1. When timing signal T_0 T_1 or T_2 are active. This can be expressed with the following register transfer statements.

$$T_0'T_1'T_2'(IEN)(FGI+FGO):R \leftarrow 1$$

Q.101 Explain all the phases of instruction cycle.

Ans.

Instruction Cycle:-

A program residing in the memory unit of the computer consists of a sequence of instructions. The program is executed in the computer by going through a cycle for each instruction. Each instruction cycle in turn is subdivided into a sequence of sub cycles or phases. In the basic computer each instruction cycle consists of the following phases:

1. Fetch an instruction from memory.
2. Decode the instruction
3. Read the effective address from memory if the instruction has an indirect address.
4. Execute the instruction.

Fetch and Decode:-

The program counter PC is loaded with the address of the first instruction in the program. The sequence counter SC is cleared to 0, providing a decoded timing signal T_0. After each clock pulse, SC is incremented by one, so that the timing

signals go through a sequence T_0, T_1, and so on. The microoperation for the fetch and decode phases can be specified by the following register transfer statements.

T_0 : AR ← PC

T_1 : IR ← M[AR], PC ← PC + 1

T_2 : $D_0 \ldots D_7$ ← Decode IR(12-14), AR ← IR (011),
I ← IR(15)

To provide the data path for the transfer of PC to AR we must apply timing signal T_0 to achieve the following connection:

1. Place the content of PC onto the bus by making the bus selection inputs S_2 S_1 S_0 equal to 010.
2. Transfer the content of the bus to AR by enabling the LD input of AR.

T_1 : IR ← M[AR], PC ← PC + 1

The next check transition initiates the transfer from PC to AR since $T_0 = 1$. In order to implement the second statement T_1: IR ← M[R], PC ← PC +1
It is necessary to use timing signal T1, to provide the following connections in the bus system.

1. Enable the read input of memory.
2. Place the content of memory onto the bus by making
   S_2 S_1 S_0 = 111.
3. Transfer the content of the bus to IR by enabling the LD input of IR.
4. Increment PC by enabling the INR input of PC.

The three instruction types are subdivided into four separate paths. The selected operation is activated with the clock transition associated with timing signal T_3.

This can be symbolized as follows:

$D_7'I'T_3$: AR M[AR]

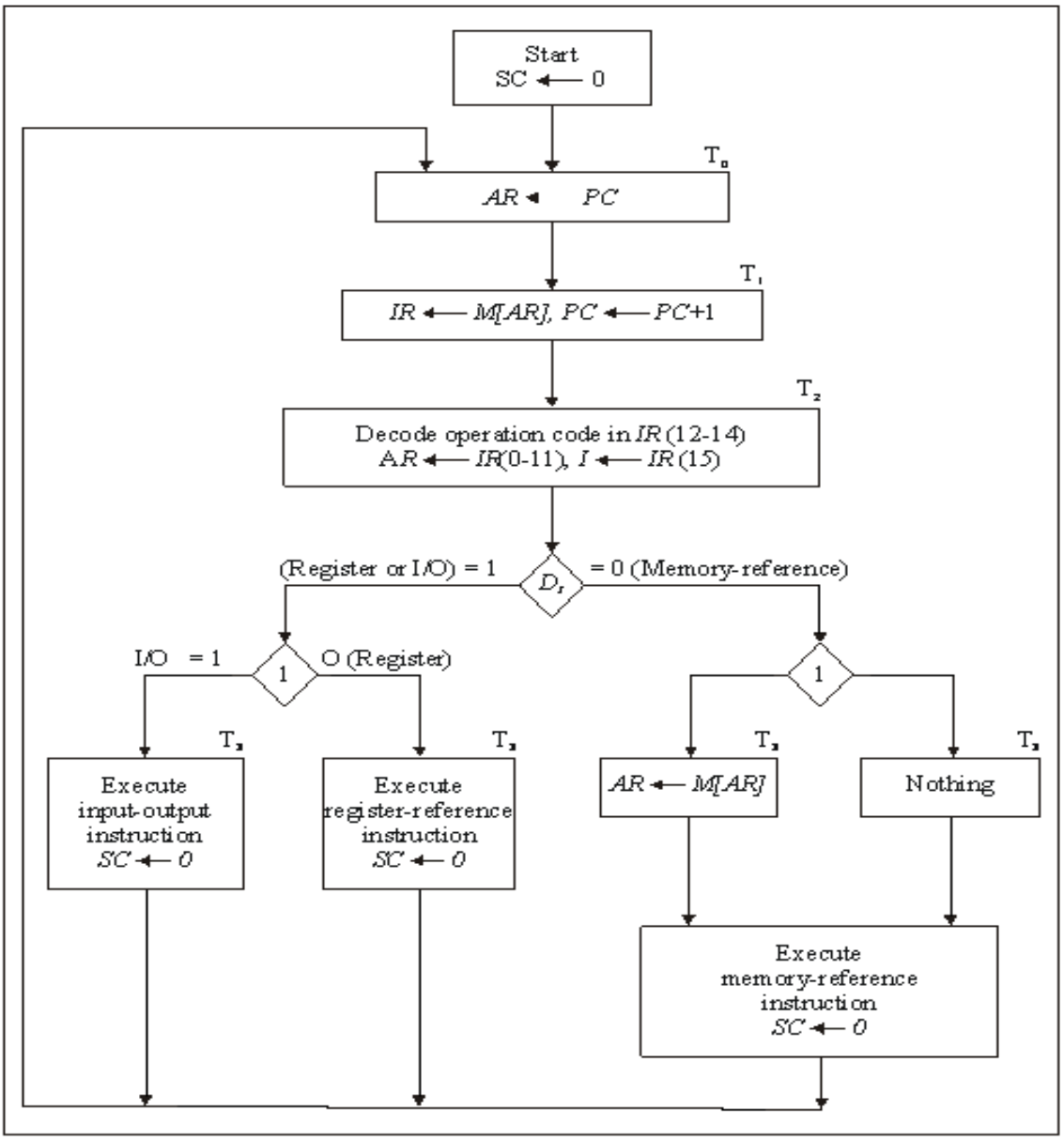
$D_7'I'T_3$: Nothing

$D_7'I'T_3$: Execute a register-reference instruction

$D_7'I'T_3$: Execute an input-output instruction

When a memory-reference instruction with I = 0 is encountered, it is not necessary to do anything since the effective address is already in AR. However, the sequence counter SC must be incremented with $D_7'I'T_3 = 1$, so that the execution of the memory-reference instruction can be continued with timing variable T_4. A register-reference or input-output instruction can be executed with the clock associated with timing signal T_3. After the instruction is executed, SC is cleared to 0 and control returns to the fetch phase with $T_o = 1$.

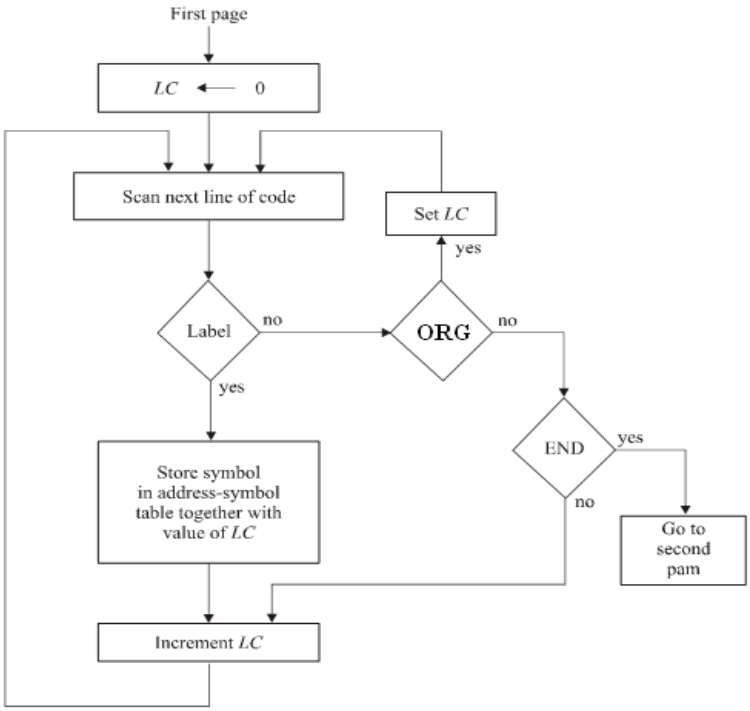
Note that the sequence counter SC is either incremented or cleared to 0 with every positive clock transition. We will adopt the convention that if SC is incremented, we will not write the statement SC ← SC + 1, but it will be implied that the control goes to the next timing signal is sequence. When SC is to be cleared, we will include the statement SC ← 0.

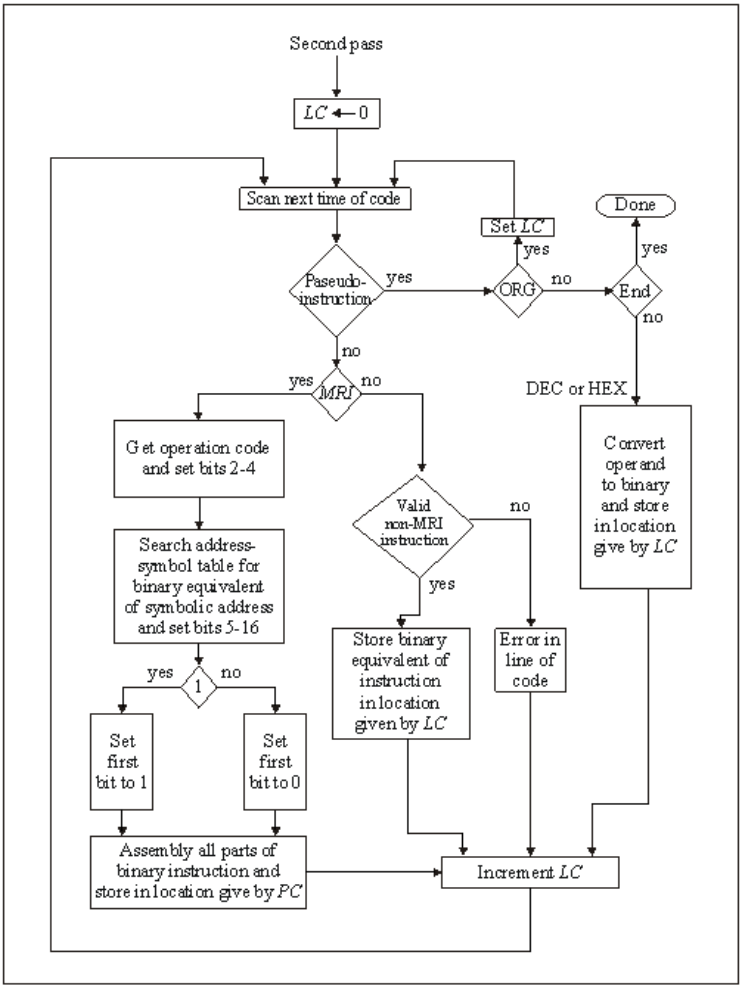


Q. 102 Explain working of two pass assembler. (Explain both pass 1 and pass 2 with flow chart).

Ans.

LC is initially set to 0. Lines of code are then analyzed one at a time. Labels are neglected during the second pass, so the assembler goes immediately to the instruction field and proceeds to check the first symbol encountered. It first checks the pseudo instruction table. A match with ORG sends the assembler to a subroutine that sets LC to an initial value. A match with END terminates the translation process. An operand pseudo instruction causes a conversion of the operand into binary. This operand is placed in the memory location specified by the content of LC. The location counter is then incremented by 1 and the assembler continues to analyze the next line of code.


First page
LC ← 0
Scan next line of code
Set LC
yes
Label
no
ORG
no
yes
END
yes
Store symbol in address-symbol table together with value of LC
no
Go to second pam
Increment LC


Second pass
LC ← 0
Scan next time of code
Set LC
yes
Done
yes
Paseudo-instruction
yes
ORG
no
End
yes
no
no
yes
MRI
no
DEC or HEX
Get operation code and set bits 2-4
Valid non-MRI instruction
no
Convert operand to binary and store in location give by LC
Search address-symbol table for binary equivalent of symbolic address and set bits 5-16
yes
Store binary equivalent of instruction in location given by LC
Error in line of code
yes
I
no
Set first bit to 1
Set first bit to 0
Assembly all parts of binary instruction and store in location give by PC
Increment LC

Q. 103 Write an assembly language program to multiply two positive numbers by a repeated addition method. For example to multiply 7*4 the program evaluated the product by adding 7 four times.

Ans.
Assembly language programme to multiply two positive numbers.

```
        ORG 100
LOP     CLE             /Clear
        LDA Y           /Load multiplier
        CIR             /Transfer multiplier bit to E
        STAY            /Store shifted multiplier
        SZE             /Check if bit is zero
        BUN ONE         /Bit is one; go to ONE
        BUN ZRO         /Bit is one; go to ZRO
ONE,    LDAX            /Load multiplicand
        ADD P           /Add to partial product
        STA P           /Store partial product
        CLE             /Clear E
ZRO,    LDA X           /Load multiplicand
        CIL             /Shift left
        STA X           /Store shifted multiplicand
        ISZ CTR           /Increment counter
        BUN LOP           /Counter not zero; repeat loop
        HLT               /Counter is zero; halt
CTR,    DEC – 8           /This location serves as a counter
X,      HEX 000F          /Multiplicand stored here
Y,      HEX 000B          /Multiplier stored here
P,      HEX 0             /Product formed here
```

Q. 104 Differentiate between the following:
(i) Autoincrement and Autodecrement addressing mode.
(ii) Program interrupt and subroutine call & return.

Ans.
Autoincrement and Autodecrement mode :-
The register is incremented or decremented after (or before) its value is used to access memory. The address stored in the register refers to a label of data in memory, it is necessary to increment or decrement the register after every access. This is achieved by using the increment or decrement instruction.

Subroutine call and Return :-
A subroutine call is a self contained sequence of instructions that performs a given computational task. During the execution of program, a subroutine may be called to perform its function many times at various points in the main program.
The BSA instruction performs the function usually referred to as a subroutine call. The indirect BUN instruction at the end of the subroutine performs the function referred to as a subroutine return. In most commercial computers, the return address associated with a subroutine is stored in either a processor register or in a portion of memory called a stack. When a subroutine is called, the main program must transfer the data. The subroutine shifted the number and left it there to be accepted by the

main program. It is necessary for the subroutine to have access to data from the calling program and to return results to that program. The accumulator can be used for a single input parameter and a single output parameter consider a subroutine that performs the logic OR operation. Two operands must be transferred to the subroutine and the subroutine must return the result of the operation.

Q. 105 What is a microinstruction? Write a micro instruction code format and explain all the fields in it.

Ans.
Each word in control memory contains within it a microinstruction. The microinstruction specifies one or more micro-operations for the system. A sequence of microinstruction constitutes a microprogram. It is an instruction stored in control memory.

OP code								
Computer Instruction		0	1	1	1			address
Mapping Bits	0	x	x	x	x	0	0	
Microinstruction address	0	1	0	1	1	0	0	

Instruction code to microinstruction address

A special type of branch exist when a microinstruction specifies a branch to the first word in control memory where a microprogram routine for an instruction is located. The status bits for this type of branch are the bits in the operation code part of the instruction. Micro instruction format is 20 bits in length. It divided into four functional parts. The three fields F_1, F_2 and F_3 specify micro operations for the computer. The CD field selects status bits condition:
* the BR field specifies the type of Branch to be used.
* the AD field contains a branch address. The address field is even bits wide, since the control memory has $128 = 2^7$ words.

F_1, F_2, F_3 : Micro operation fields

3	3	3	2	2	7
F1	F2	F3	CD	BR	AD

CD : Condition for branching
BR : Branch field
AD : Address field

* Micro operations are sub-divided into three fields of three bits each. These bits in each field are encoded to specify seven distinct micro operations. This gives 21 micro operations. If fewer than three micro operations are used, one or more of the fields will use the binary code 000 for no operation.

Symbols and Binary Code for Micro Instruction Fields

F_1	Micro operation	Symbol
000	None NOP	
001	AC ← AC + DR	ADD
010	AC ← 0	CLRAC
011	AC ← AC + 1	INCAC
100	AC ← DR	DRTAC
101	AR ← DR (0-10)	DRTAR
110	AR ← PC	DCTAR
111	M [AR] ← DR	WRITE

F_2	Micro Operation	Symbol
000	None NOP	
001	AC ← AC-DR	SUB
010	AC ← AC ∨ DR	OR
011	AC ← AC ∧ DR	AND
100	DR ← M[AR]	READ
101	DR ← AC	ACTDR
110	DR ← DR+1	INCDR
111	DR(0-10) ← PC	PCTDR

F_3	Micro Operation	Symbol
000	None NOP	
001	AC ← AC ⊕ DR	XOR
010	AC ← AC COM	
011	AC ← shl AC	SHL
100	AC ← shr AC	SHR
101	PC ← PC+1	INCPC
110	PC ← AR	ARTPC
111	Reserved	

CD	Condition	Symbol	Comments
00	Always=1	U	Unconditioned branch
01	DR(15)	I	Indirect address bit
10	AC(15)	S	Sign bit of AC
11	AC=0	Z	Zero value in AC

BR	Symbol	Function
00	JMP	CAR ← AD, if condition=1 CAR ← CAR+1, if condition=0
01	CALL	CAR ← AD, SBR ← CAR+1, if condition=1 CAR ← CAR+1, if condition=0
10	RET	CAR ← SBR (return from subroutine)
11	MAP	CAR(2-5) ← DR(11-14), CAR(0, 1, 6) ← 0

Q. 106 What is a microprogram? Write a microprogram for the fetch routine.

Ans.

A sequence of microinstructions constitutes a microprogram. The use of a micro program involves placing all control variables in words of ROM for use by the control unit through successive record operation.

Binary Microprogram for Control Memory (Partial)

Micro Routine	Address Decimal	Address Binary	Binary Microinstruction F1	F2	F3	CD	BR	AD
	0	0000000	000	000	000	01	01	1000011
	1	0000001	000	100	000	00	00	0000010
	2	0000010	001	000	000	00	00	1000000
	3	0000011	000	000	000	00	00	1000000
BRANCH	4	0000100	000	000	000	10	00	0000110
	5	0000101	000	000	000	00	00	1000000
	6	0000110	000	000	000	01	01	1000011
	7	0000111	000	000	110	00	00	1000000
STORE	8	0001000	000	000	000	01	01	1000011
	9	0001001	000	101	000	00	00	0001010
	10	0001010	111	000	000	00	00	1000000
	11	0001011	000	000	000	01	01	1000011
EXCHANGE	12	0001100	000	000	000	01	01	1000011
	13	0001101	001	000	000	00	00	0001110
	14	0001110	100	101	000	00	00	0001111
	15	0001111	111	000	000	00	00	1000000
FETCH	64	1000000	110	000	000	00	00	1000000
	65	1000001	000	100	101	00	00	1000010
	66	1000010	101	000	000	00	11	0000000
INDRCT	67	1000011	000	010	000	00	00	1000100
	68	1000100	101	000	000	00	10	0000000

Q. 107 Formulate a four segment instruction pipeline for a computer. Specify the operation to be performed in each segment.

Ans.

Instruction pipelines :- An instruction pipeline reads consecutive instructions from memory while previous instructions are being executed in other segments. This causes the instruction fetch and execute phases to overlap and perform simultaneous operations. The instruction fetch segment can be implemented by means of a first-in, first-out (FIFO) buffer. This is a type of unit that forms a queue rather than a stack. The execution unit is not using memory, the control increments the program counter and uses its address value to read consecutive instructions from memory. An instruction stream can be placed in a queue, waiting for decoding and processing by the execution segment. The instruction stream queuing mechanism provides an efficient way for reducing the average access time to memory for reading instructions. The computer needs to process each instruction with the following sequence of steps.

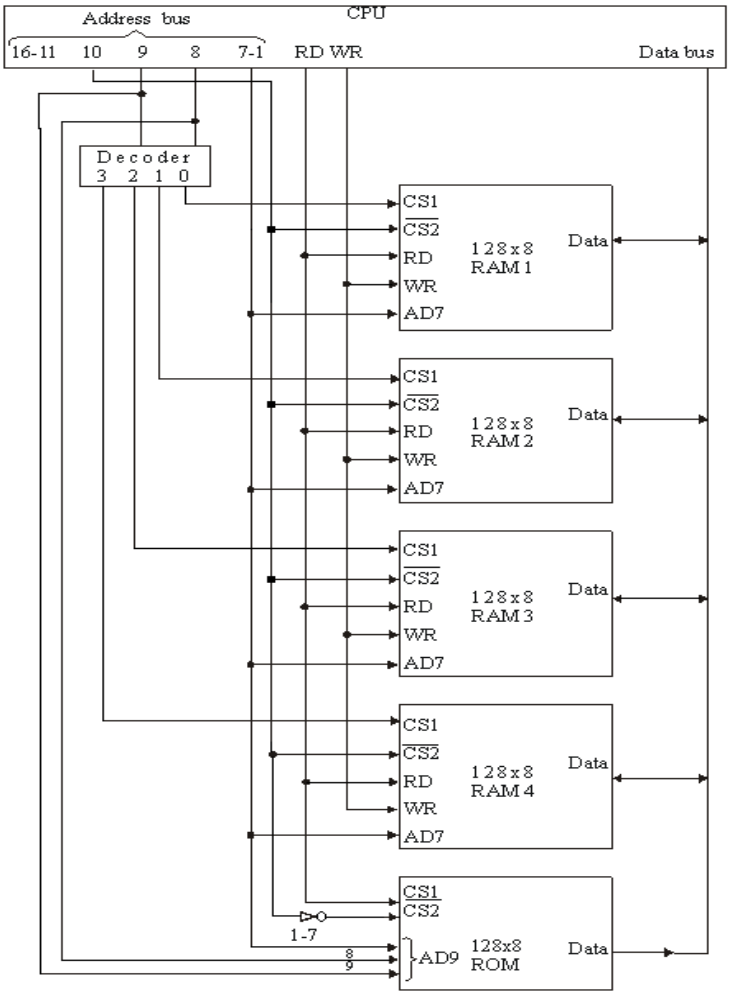
1. Fetch the instruction from memory.
2. Decode the instruction.

3. Calculate the effective address.
4. Fetch the operands from memory.
5. Execute the instruction.
6. Store the result in the proper place.

There are certain difficulties that will prevent the instruction pipeline from operating at its maximum rate. Different segments may take different times to operate on the incoming information. Some segments are skipped for certain operations. For example, a register mode instruction does not need an effective address calculation. Two or more segments may require memory access at the same time, causing one segment to wait until another is finished with the memory.
The design of an instruction pipeline will be most efficient if the instruction cycle is divided into segments of equal duration. The time that each step takes to fulfill its function depends on the instruction and the way it is executed.

Q. 108 Show the memory organization (1024 bytes) of a computer with four 128x8 RAM Chips and 512x8 ROM Chip. How many address lines are required to access memory.

Ans.
Memory organization:-



Address lines:-

128 = 128 = 2^7

512 = 2^9

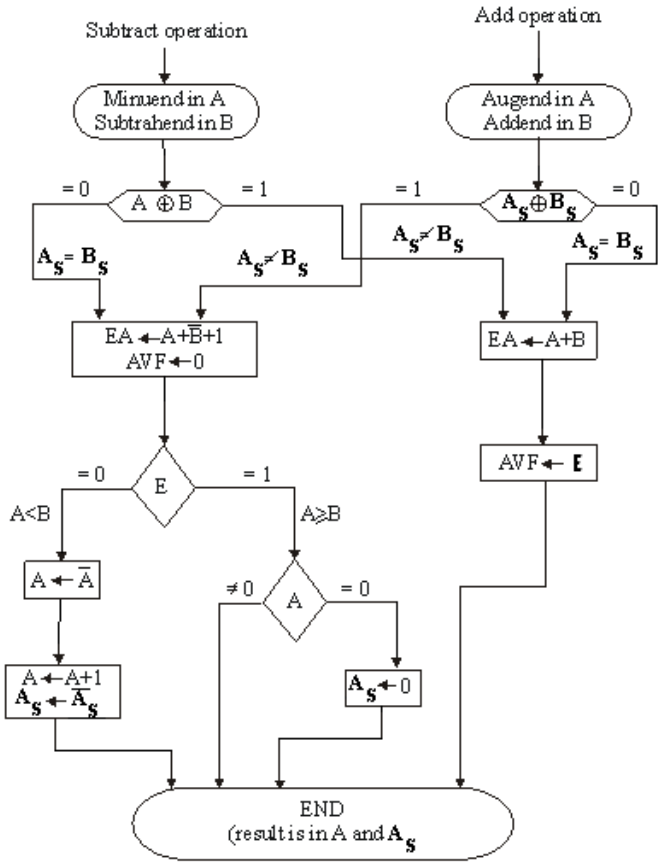
Address lines = 16

Q. 109 Write a general algorithm and flow chart for addition and subtraction of two signed magnitude Numbers.

Ans.

Hardware Algorithm:-

The flowchart for the hardware algorithm is presented. The two signs A_s, and B_s are compared by an exclusive-OR gate. If the output of the gate is 0, the signs are identical; if it is 1, the signs are different. For an add operation, identical signs dictate that the



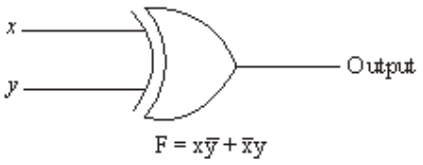
magnitudes are added. For a subtract operation, different signs dictate that the magnitudes be added. The magnitudes are added with a microoperation EA A + B, where EA is a register that combines E and A. The carry in E after the addition constitutes an overflow if it is equal to 1. The value of E is transferred into the add-overflow flip-flop AVF.

Q.110 Ram wants to purchase a bicycle. The bicycle must have brakes. The bicycle which has either a hand brake or foot brake, No bicycle has both type of brakes. Implement the same using basic gates.

Ans.

Suppose hand break = 0 = x
foot break = 1 = y
Truth table obtain by EX - OR gate

x	y	Output
0	0	0
1	0	1
0	1	1
1	1	0

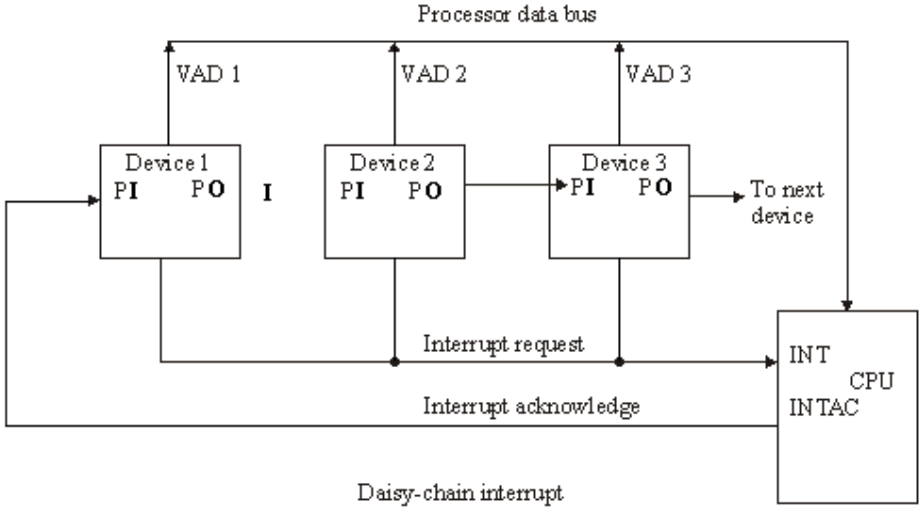


Q. 111 Write short notes on followings
(i) Daisy chaining priority.
(ii) Direct Memory Access.
(iii) Handshaking method for data transfer.
(iv) Associative Memory

Ans.

(i) **Daisy chaining priority**:-

The daisy-chaining method of establishing priority consists of a serial connection of all devices that request an interrupt.



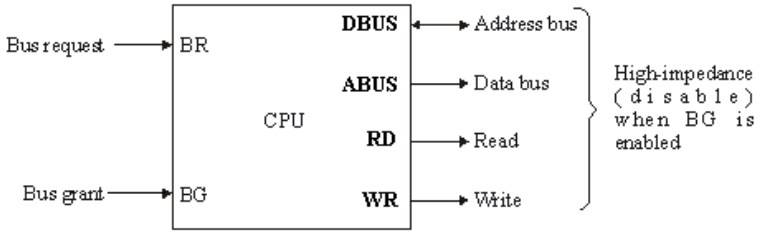
The device with the highest priority is placed in the first position, followed by lower-priority devices up to the device with the lowest priority, which is placed last in the chain. This method of connection between three devices and the CPU is shown. The interrupt request lines is common to all devices and forms a wired logic connection. If any device has its interrupt signal in the low-level state, the interrupt line goes to the low-level state and enables the interrupt input in the CPU. When no interrupts are pending, the interrupt line stays in the high-level state and no interrupts are recognized

by the CPU. This is equivalent to a negative logic OR operation. The CPU responds to an interrupt request by enabling the interrupt acknowledge line.

(ii) **Direct Memory Access**:-

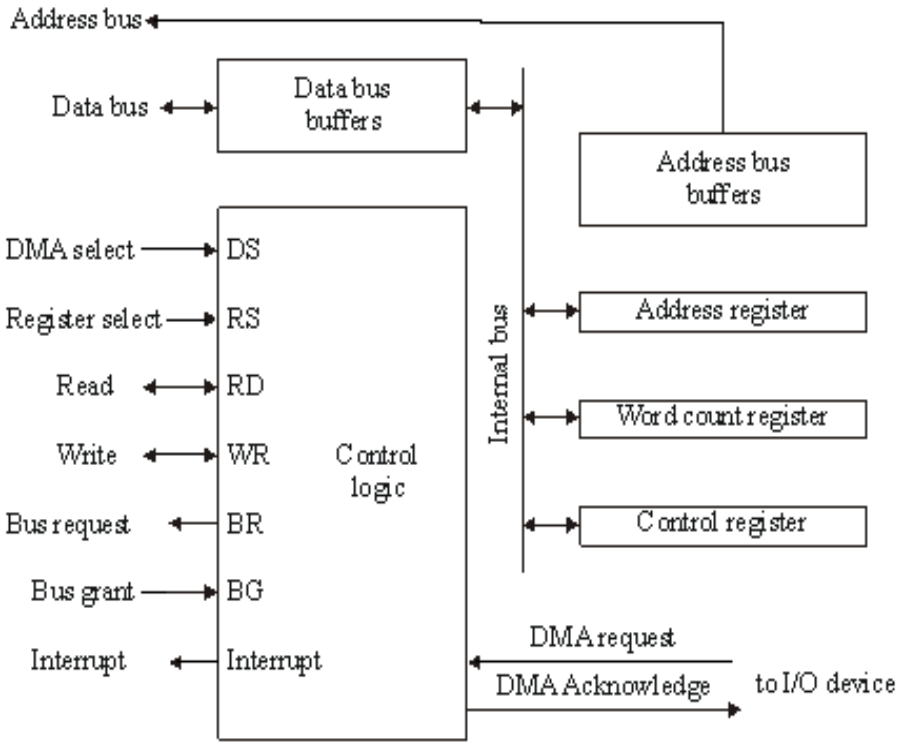
The transfer of data between a fast storage device such as magnetic disk and memory is often limited by the speed of the CPU. Removing the CPU from the path and letting the peripheral device manage the memory buses directly would improve the speed of transfer. This transfer technique is called direct memory access (DMA). During DMA transfer, the CPU is idle and has no control of the memory buses.

CPU bus signals for DMA transfer



The figure shows two control signals in the CPU that facilitate the DMA transfer. The bus request (BR) input is used by the DMA controller to request the CPU to relinquish control of the buses. The CPU activates the bus grant (BG) output to inform the external DMA that the buses are in the high-impedance state. The DMA that originated the bus request can now take control of the buses

Block diagram of DMA controller

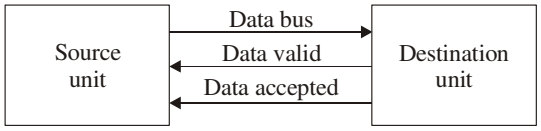


to conduct memory transfers without processor intervention. When the DMA terminates the transfer, it disables the bus request line. The CPU disables the bus grant. When the DMA takes control of the bus system, it communicates directly with the memory.

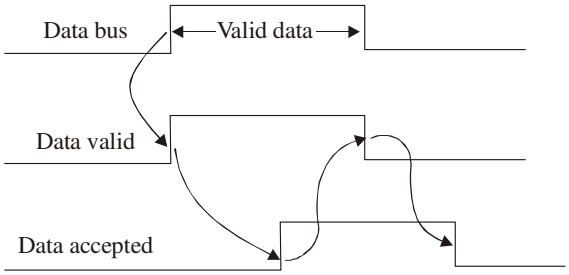
(iii) Handshaking method for data transfer:-

Source-initiated data transfer:- The disadvantage of the strobe method is that the source unit that initiates the transfer has no way of knowing whether the destination unit has actually received the data item that was placed in the bus. Similarly, a destination unit

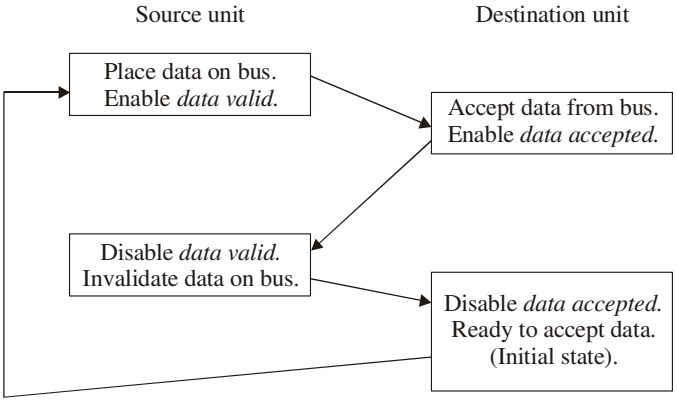
Input output organization



(a) Block diagram



(b) Timing diagram



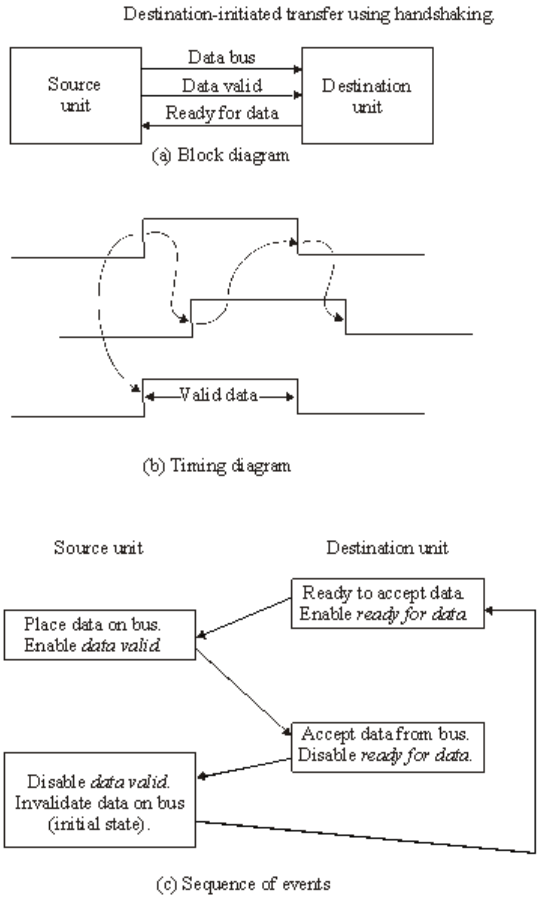
(c) Sequence of events

that initiates the transfer has no way of knowing whether the source unit has actually placed the data on the bus. The two handshaking lines are *data valid,* which is generated by the source unit, and *data accepted,* generated by the destination unit. The timing diagram shows the exchange of signals between the two units. The sequence of events listed in part (c) shows the four possible states that the system can be at any given time. The source unit initiates the transfer by placing the data on the bus and enabling its *data valid* signal. The *data accepted* signal is activated by the destination unit after it accepts the data from the bus. The source unit then

disables its *data valid* signal, which invalidates the data on the bus. The destination unit then disables its *data accepted* signal and the system goes into its initial state. The source does not send the next data item unit after the destination unit shows its readiness to accept new data by disabling its *data accepted* signal. This scheme allows arbitrary delays from one state to the next and permits each unit to respond at its own data transfer rate. The rate of transfer is determined by the slowed unit.

Destination initiated data transfer:-

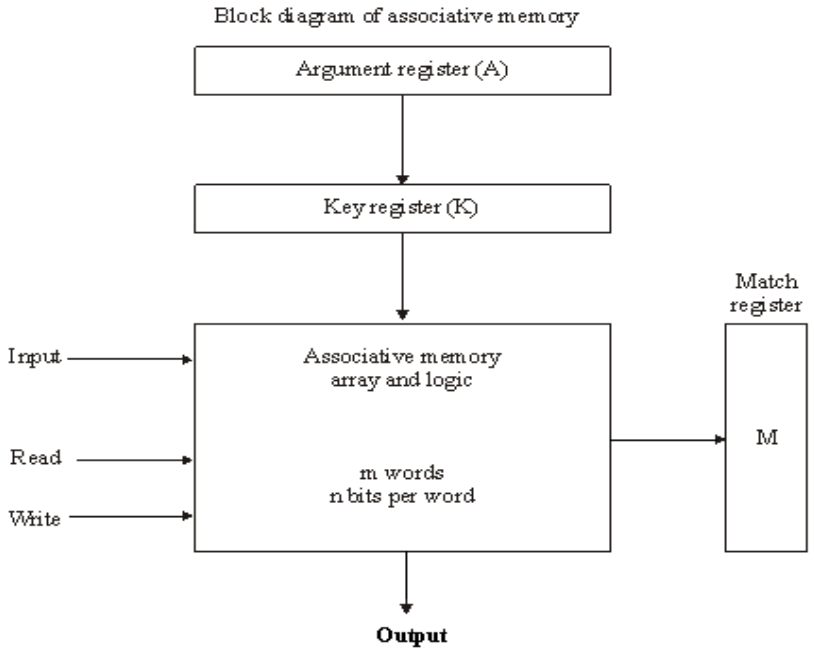
The destination-initiated transfer using handshaking lines. Note that the name of the signal generated by the destination unit has been changed to *ready for data* to reflect its new meaning. The source unit in this case does not place data on the bus until after it receives the *ready for data* signal from the destination unit. From there on, the handshaking procedure follows the same pattern as in the source-initiated case. Note that the sequence of events in both cases would be identical if we consider the *ready for data* signal as the complement of *data accepted*. In fact, the only difference between the source-initiated and the destination-initiated transfer is in their choice of initial state.



This disadvantage of the strobe method is that the source unit that initiates the transfer has no way of knowing whether the destination unit has actually received the data item that was placed in the bus. Similarly, a destination unit that initiates the transfer has no way of knowing whether the source unit has actually placed the data on the bus.

(iv) **Associative Memory**:-

The time required to find an item stored in memory can be reduced considerably if stored data can be identified for access by the content of the data itself rather than by an address. A memory unit access by the content of the data itself rather than by an address. A memory unit accessed by content is called an associative memory or content addressable memory (CAM) . This type of memory is accessed simultaneously and in parallel on the basis of data content rather than by specific address or location.



Q.112 a. Show the Truth Table's for the Following functions:-

i) $f(w, x, y, z) = w + x + y + z$

ii) $f(w, x, y, z) = wx + xz + \bar{y}$

Ans. (i) $f(w, x, y, z) = w + x + y + z$

w	x	y	z	output
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1

1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

(ii) $f(w, x, y, z) = wx + xz + \bar{y}$

w	x	y	z	Output= $wx + xz + \bar{y}$
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

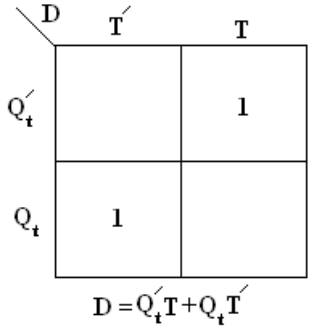
Q. 113 Construct a T flip flop using a
i) D - FF ii) J-K FF

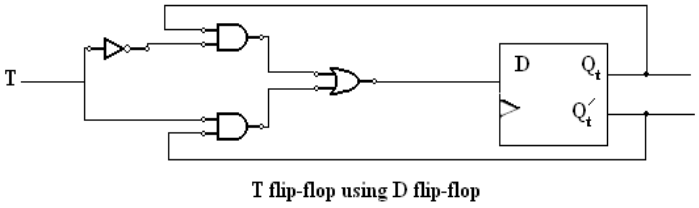
Ans. (i)
Conversion of D flip-flop to T flip-flop

Truth table of T		
Q_t	**T**	**Q_{t+1}**
0	**0**	**0**
0	**1**	**1**
1	**0**	**1**
1	**1**	**0**

Excitation table of D		
Q_t	**Q_{t+1}**	**D**
0	**0**	**0**
0	**1**	**1**
1	**0**	**0**
1	**1**	**1**

Conversion table			
Q_t	**T**	**Q_{t+1}**	**D**
0	**0**	**0**	**0**
0	**1**	**1**	**1**
1	**0**	**1**	**1**
1	**1**	**0**	**0**





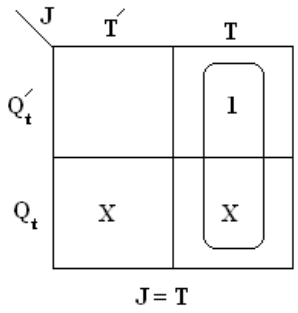
T flip-flop using D flip-flop

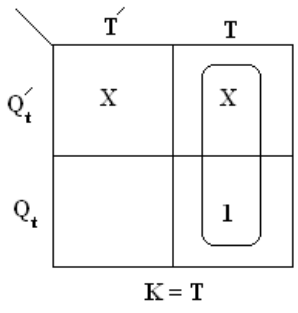
(ii)
Conversion of JK flip- flop to T flip-flop

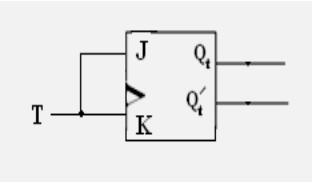
Truth table of T		
Q_t	T	Q_{t+1}
0	0	0
0	1	1
1	0	1
1	1	0

Excitation table of JK			
Q_t	Q_{t+1}	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

Conversion table				
Q_t	T	Q_{t+1}	J	K
0	0	0	0	x
0	1	1	1	x
1	0	1	x	0
1	1	0	x	1



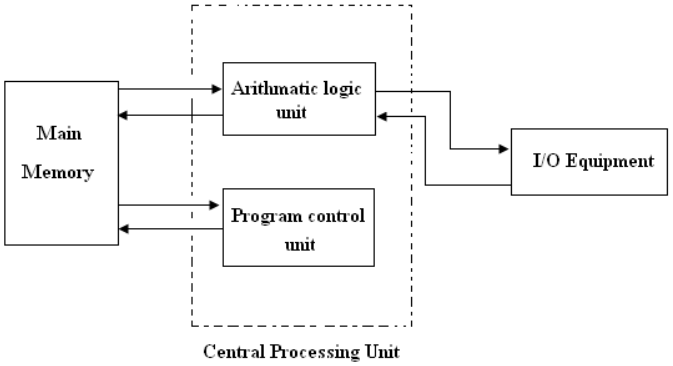




Q. 114 Explain Ven Neumann architector and stored program concept.

Ans.

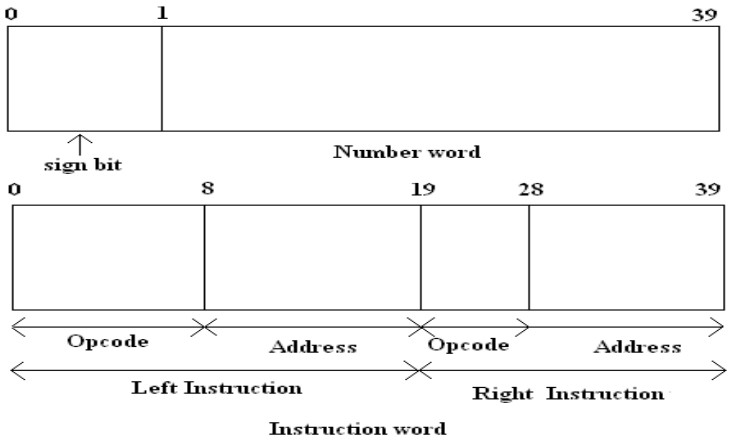
In 1946, John Von Neumann and his colleagues began the design of a new stored – program computer referred to as IAS computer. The general structure of IAS computer is



It consists of the following:

(a) A main memory, which stores both data and instructions.
(b) An arithmetic and logic unit (ALU) capable of operating on binary data.
(c) A control unit, which interprets the instructions in memory and causes them to be executed.
(d) Input and output (I/O) equipment operated by the control unit.

The memory of the IAS consists of 1000 storage locations, called words of 40 binary digits (bits) each. Both data and instructions are stored there. Thus the numbers must be represented in binary form, and each instruction also has to be in a binary code, in their respective formats as below.



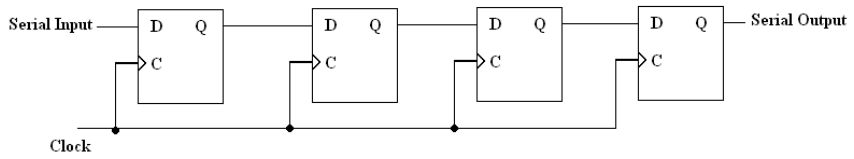
Each number is represented by a sign bit and a 39- bit value. A word may also contain two 20- bit instruction, with each instruction of an 8-bit operation code (opcode) specifying the operation to be performed and a 12-bit address designated one of the words in memory. John Von Neumann gave the idea of storing 'Programme' and 'Data' in the same memory. Storing of programs in memory helps in executing series of instructions repetitively. It makes the operation of computer automatic.

Q. 115 Show the hardware to implement the following micro-operations.

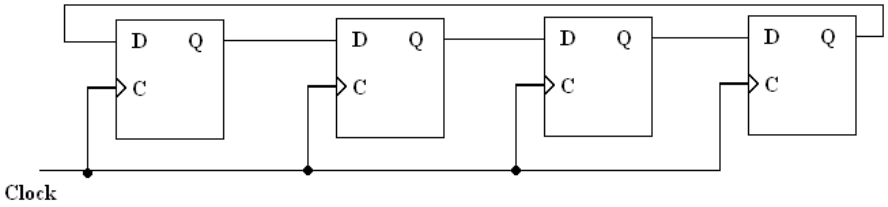
(i) L : α shl (x) (ii) α : cir (x)

'x' consist of four D-FFs.

Ans. (i) L : α shl (x)



(ii) α : cir (x)



Q. 116 Discuss the properties of an ideal instruction set computer.

Ans.

(i) A computer has a set of instruction so that the user can construct machine language programs to evaluate any function.

(ii) Input and output instructions are needed for communication between the computer and the user.

(iii) Programs and data transferred into memory and results of computations transferred back to the user.

Q. 117 Explain instruction cycle. Implement the RTLs of fetch phase.

Ans.

A program residing in the memory unit of the computer consists of a sequence of instructions. The program is executed in the computer by going through a cycle for each instruction. In the basic computer each instruction cycle consists of the following phases:

(1) Fetch an instruction from memory

(2) Decode the instruction

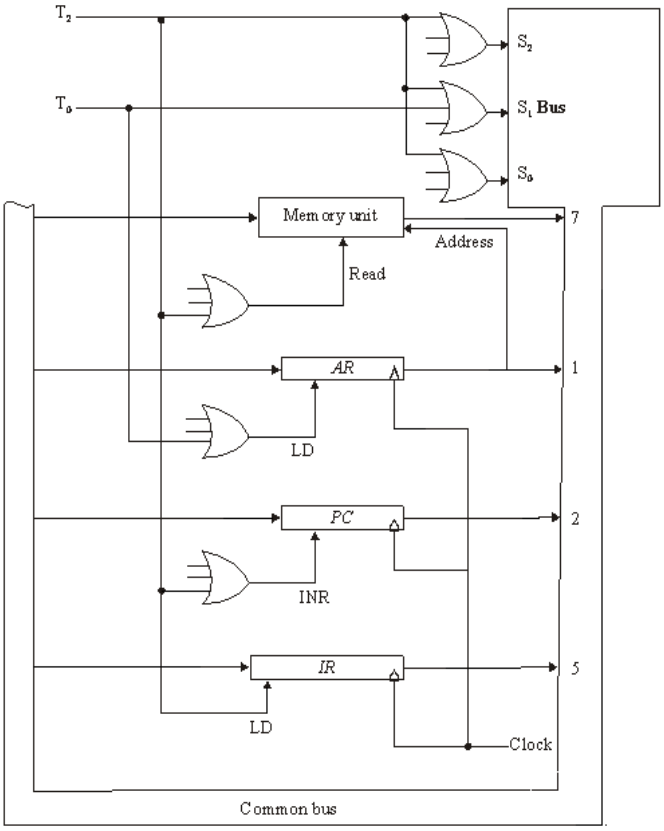
(3) Execute the instruction

Fetch

The program counter PC is loaded with the address of the first instruction in the program. The sequence counter SC is cleared to 0, providing a decoded tining signal T_o. After each clock pulse, SC is incremented by one.

T_o : AR ← PC

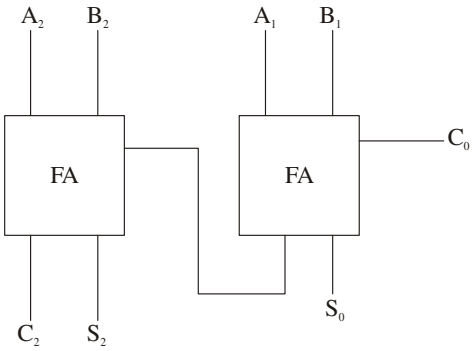
T_1 : IR ← M[AR], PC ← PC+1



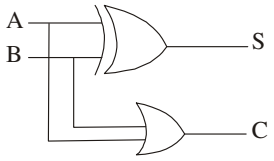
Q. 118 Design a 2-bit adder and logic circuit capable of perfering AND ADD, complement and shift left operation.

Ans.

2 bit adder

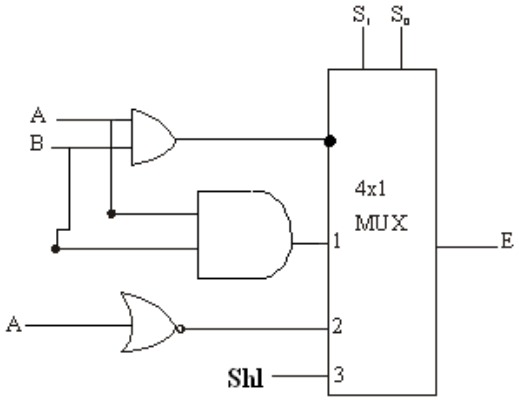


Logic Circuit



A	B	S	C
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

S1	S0	Output	Operation
0	0	E=A^B	AND
0	1	E=A ∨ B	ADD
1	0	E=A	Complement
1	1	Shl	



Q. 119 Discuss the different addressing modes of an instruction.

Ans.

In this mode the operands are specified implicitly in the definition of the instruction.

Immediate Mode:- In this mode the operand is specified in the instruction itself. In other words, an immediate-mode instruction has an operand field rather than an address field. The operand field contains the actual operand to be used in conjunction with the operation specified in the instruction.

Register mode:- In this mode the operands are in registers that reside within in CPU. The particular register is selected from a register field in the instruction. A k-bit field can specify any one of 2^k registers.

Register Indirect mode:- In this mode the instruction specifies register in CPU whose contents give the address of the operand in memory. Before using a register indirect mode instruction, the programmer must ensure that the memory address of the operand is placed in the processor register with a previous instruction.

Autoincrement or Autodecrement Mode:- This is similar to the register indirect mode except that the register is incremented or decremented after for before) its value is used to access memory. When the address stored in the register refers to a table of data is memory. It is necessary to increment or decrement the register after every access to the table.

Direct Address Mode:- In this mode the effective address is equal to the address part of the instruction.

Indirect Address Mode:- In this mode the address field of the instruction gives the address where the effective address is stored in memory. Control fetches the instruction from memory and uses its address part to access memory again to read the effective address.

effective address = address part of instruction + content of CPU register

Relative Address Mode:- In this mode the content of the program counter is added to the address part of the instruction in order to obtain the effective address. When this number is added to the content of the program counter, the result produces an

effective address whose position in memory is relative to the address of the next instruction.

Indexed Addressing Mode:- In this mode the content of an index register is added to the address paid of the instruction to obtain the effective address.

Base Register Addressing Mode:- In this mode the content of a base register is added to the address part of the instruction to obtain the effective address.

Q.120 What is the significance of program status word.

Ans.

The collection of all status bit condition in the CPU is called a program status word or PSW. The PSW is stored in a separate hardware register and contains the status information that characterizes the state of the CPU. The CPU does not respond to an interrupt until the end of an instruction execution. Before going to the next fetch phase, control checks for any interrupt signals.

If an interrupt is pending, control goes to hardware interrupt cycle. Desiring this cycle, the contents of PC and PSW are pushed onto the stack. The PSW is transferred to the status register and the return address to the program counts. The CPU state is restored and the original program continues executing.

Q.121 What is software interrupt? State it use.

Ans.

A software interrupt is initiated by executing an instruction. Software interrupt is a special call instruction that behaves like an interrupt rather than a subjective call. The most common use of software interrupt is associated with supervision call instruction. This instruction provides means for switching from a CPU user mode to the supervision mode. A software interrupt that steers the old CPU state and brings is new PSW that belongs to the supervisor mode.

Q. 122 What is the difference between 1's complement subtraction and 2's complement subtraction of binary numbers? Show it by example.

Ans.

One's complement representation:-

In a binary number, each 1 is replaced by 0 and 0 by 1, the resulting number is known as the one's complement of the first number. If one of these numbers is positive then the other number will be negative with the some magnitude and vice-versa.

Two's complement Representation :-

If 1 added to 1's complement of a binary number, the resulting number is known as the two's complement of the binary number. For example, 2's complement of 0101 is 1011. In this representations, if the MSB is 0 the number is positive, whereas if the MSB is 1 the number is negative. For an n bit number the maximum positive number is $(2^{n-1}-1)$ and the maximum negative number is -2^{n-1}

Q. 123 Show the step-by-step multiplication process using booth's algorithm, when +14 is multiplied by -14 Assume 5-bit registers that hold signed numbers.

Ans.

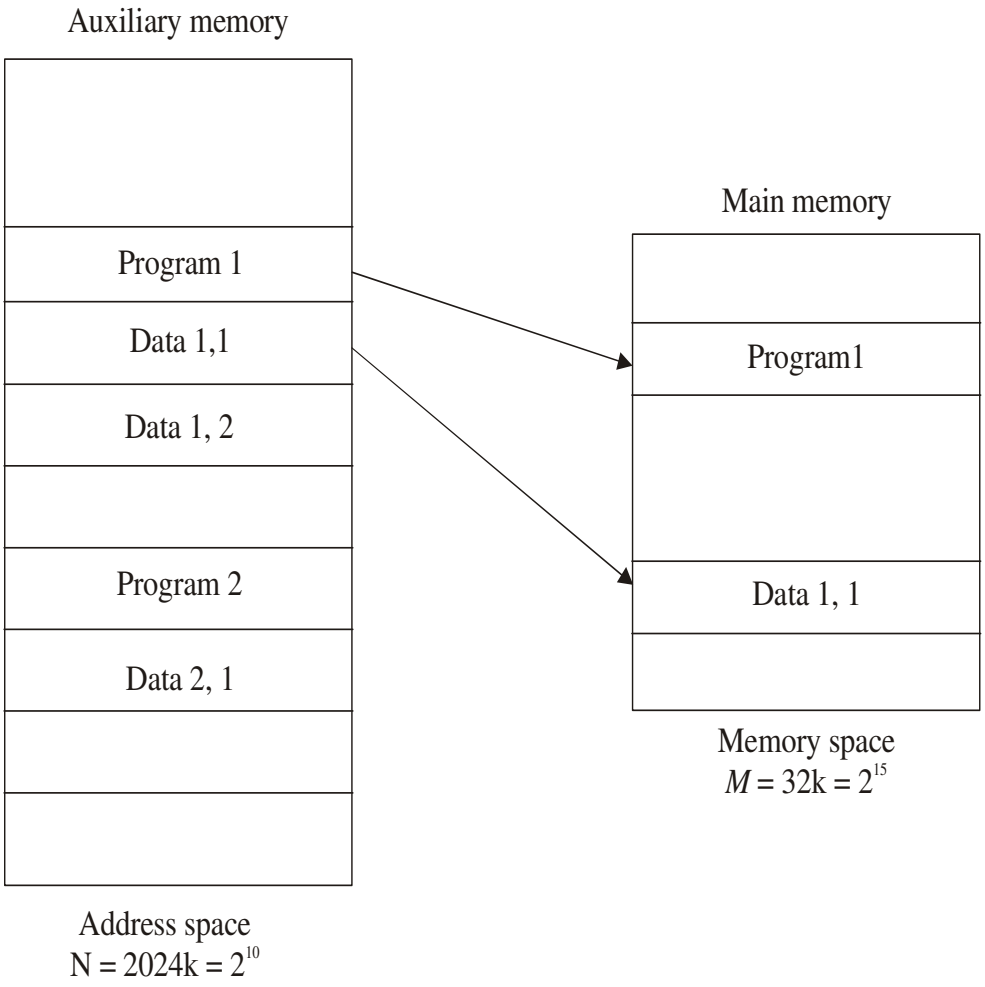
Q_n	Q_{n+1}	BR=01110 $\overline{BR}+1=10010$	AC	QR	Q_{n+1}	SC
		Initial	00000	10010	0	101
0	0	ashr	00000	01001	0	100
1	0	Subtract BR	10010 10010			
		ashr	11001	00100	1	011
0	1	Add BR	01110 00111			
		ashr	00011	10010	0	010
0	0	ashr	00001	11001	0	001
1	0	Subtract BR	10010 10011			
		ashr	11001	11100	1	000
Final Product = 1100111100						

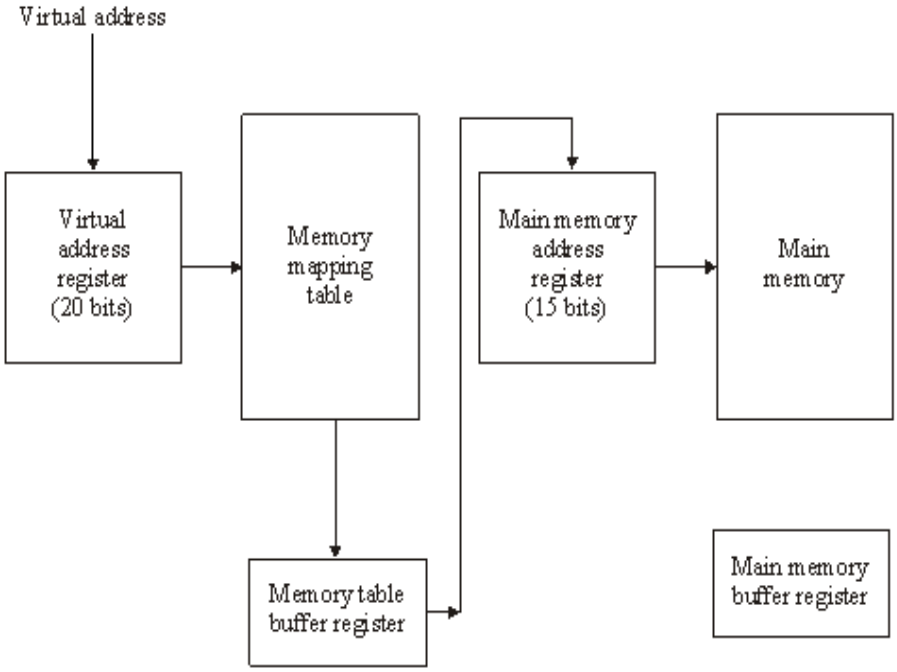
Q.124 Explain the use of time out mechanism in handshaking data transfer scheme.

Ans.
The handshaking scheme provides a hight degree of flexibility and reliability because the successful completion of a data transfer relies on active participation by both units. If one unit is faulty, the data transfer will not be completed. An error can be detected by means of a timeout mechanism, which produces an alarm if the data transfer is not completed within a predetermined time. The timeout signal used for interrupt the processor and hence executes a service routine that takes appropriate error recover action.

Q.125 Explain virtual memory & its mapping scheme.

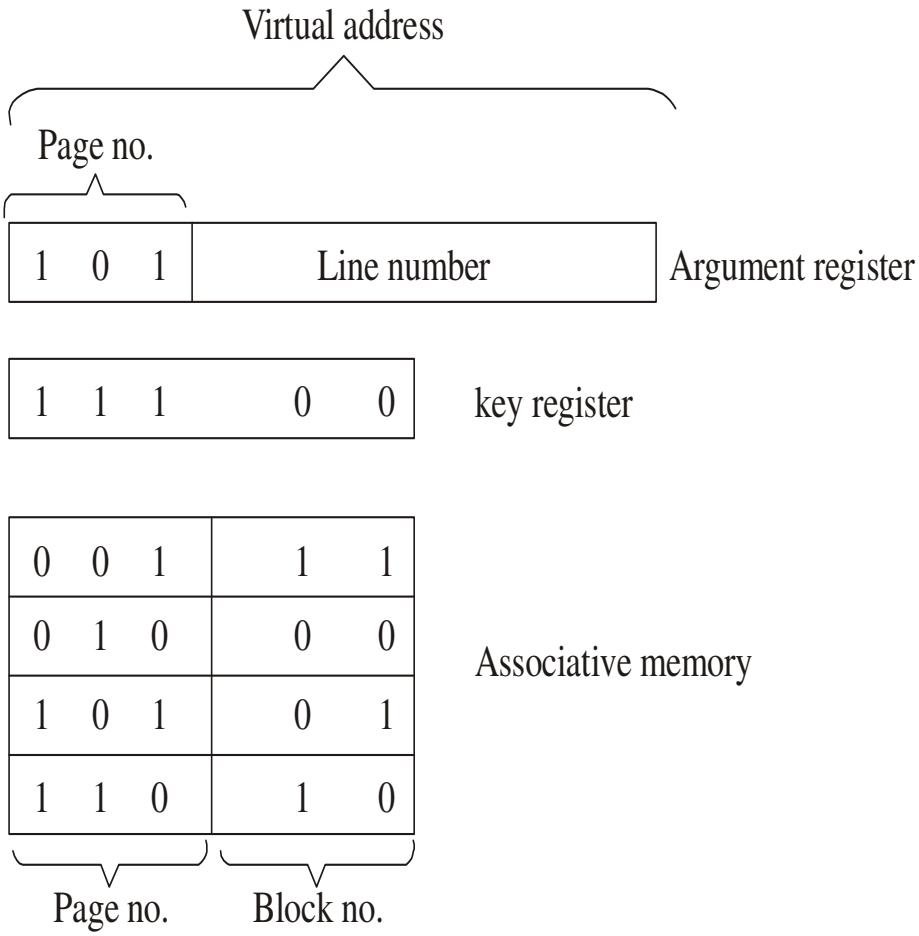
Ans.
Virtual memory is a concept used in some large computer systems that permit the user to construct programs as though a large memory space were available, equal to the totality of auxiliary memory. Virtual memory is used to give programmers the illusion that they have a very large memory at their disposal, even though the computer actually has a relatively small main memory. A virtual memory system provides a mechanism for translating program-generated addresses into correct main memory locations. In a virtual memory system, programmers are told that they have the total address space at their disposal. Moreover, the address field of the instruction code has a sufficient number of bits to specify all virtual addresses. In our example, the address field of an instruction code will consist of 20 bits but physical memory addresses must be specified with only 15 bits. Thus CPU will reference instructions and data with a 20-bit address, but the information. at this address must be taken from physical memory because access to auxiliary storage for individual words will be prohibitively long.


Auxiliary memory
Program 1
Data 1,1
Data 1, 2
Program 2
Data 2, 1
Address space
N = 2024k = 2^10
Main memory
Program1
Data 1, 1
Memory space
M = 32k = 2^15


Virtual address
Virtual address register (20 bits)
Memory mapping table
Main memory address register (15 bits)
Main memory
Memory table buffer register
Main memory buffer register

to map a virtual address of 20 bits to a physical address of 15 bits. The mapping is a dynamic operation, which means that every address is translated immediately as a word is reference by CPU an address space of 8k and a memory space of 4k. If we split each into groups of 1k words we obtain eight pages and four blocks as shown in four pages of address space may reside in main memory in any one of the four blocks. A virtual address has 13 bits. Since each page consists of 2^{10} = 1024 words, the high-order three bits of a virtual address will specify one of the eight pages and the low-order 10 bits give the line address within the page. A system with *n* pages and *m* blocks will be marked with block numbers and all others will be empty. A more efficient way to organize the page table would be to construct it with a number of words equal to the number of blocks in main memory. In this way the size of the memory is reduced and each location is fully utilized. The method can be implemented by means of an associative memory with each word in memory

An associative memory page table



containing a page number together with its corresponding block number. The page field in each word is compared with the page number in the virtual address. If a match occurs, the word is read from memory and its corresponding block. The page field in each word is compared with the page number in the virtual address. If a match occurs, the word is read from memory and its corresponding block number is

extracted. Each entry in the associative memory array consists of two fields. The first three bits specify a field for storing the page number. The last two bits constitute a field for storing the block number. The virtual address is placed in the argument register. The page number bits in the argument register are compared with all page numbers in the page field of the associative memory. If the page number is found, the 5-bit word is read out from memory. The corresponding block number, being in the same word, is transferred to the main memory address register. If no match occurs, a call to the operating system is generated to bring the required page from auxiliary memory.

Q.126 State the advantages of cache memory.

Ans.

Advantages –

(1) The average memory access time of a computer system can be improved by use of a cache.
(2) The fast access time of cache memory.
(3) Very little or no time wasted when searching for words in the cache.
(4) Cache memory is a fast & small memory.
(5) Program segment and data frequently needed by CPU are stored in cache memory and hence fast processing.

Q.127 Compare memory mapped I/O vs I/o mapped I/O.

Ans.

1) Memory mapped I/O use memory type instructions to access I/O data. It allows the computer to use the same instructions for either input-output transfers.
2) The load and store instructions are for reading and writing from memory can be used to input and output data from I/O registers.
3) Memory mapped I/O all instructions that refer to memory are also available for I/O.
4) In I/O mapped I/O configuration, the CPU has distinct input and output instructions, and each of these instructions is associated with the address of an interface register. When the CPU fetches and decodes the operation code of an input or output instruction, it places the address associated with the instruction in the common address lines and memory address values are not affected by interface assignment.

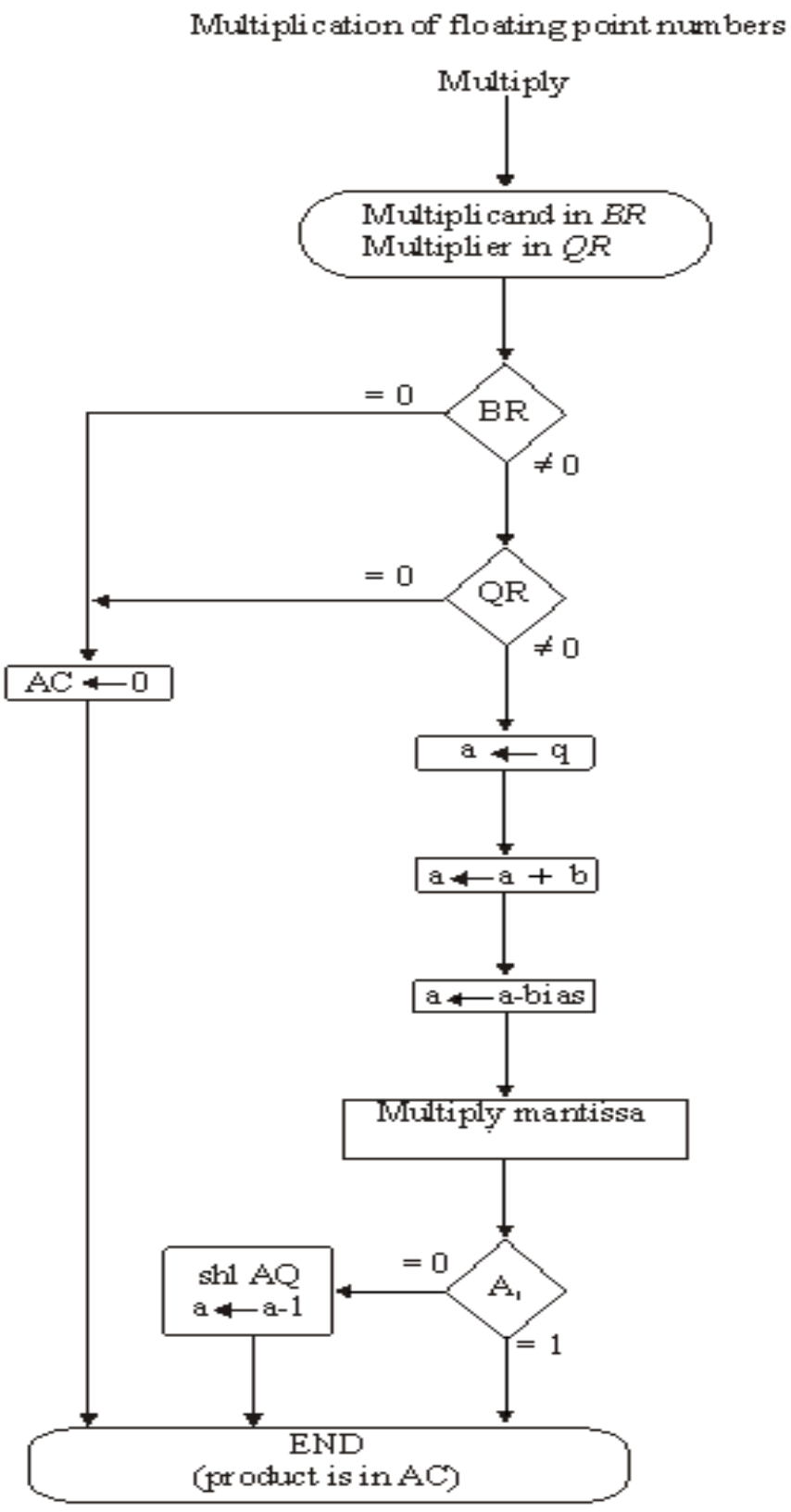
Q.128 Give the flow chart for multiplication of two floating-point numbers.

Ans.

Multiplication:-

The multiplication of two floating-point numbers requires that we multiply the mantissas and add the exponents. The multiplication algorithm can be subdivided into four parts.

1. Check for zeros.
2. Add the exponents.
3. Multiply the mantissas.
4. Normalize the product.


Multiplication of floating point numbers
Multiply
Multiplicand in BR
Multiplier in QR
BR
= 0
≠ 0
QR
= 0
≠ 0
AC ← 0
a ← q
a ← a + b
a ← a-bias
Multiply mantissa
A1
= 0
= 1
shl AQ
a ← a-1
END
(product is in AC)

TYPICAL QUESTIONS & ANSWERS

PART-I

OBJECTIVE TYPE QUESTIONS

Each Question carries 2 marks.

Choose the correct or best alternative in the following:

Q.1 In Reverse Polish notation, expression A*B+C*D is written as

(A) AB*CD*+ **(B)** A*BCD*+
(C) AB*CD+* **(D)** A*B*CD+

Ans: A

Q.2 SIMD represents an organization that _____.
(A) refers to a computer system capable of processing several programs at the same time.
(B) represents organization of single computer containing a control unit, processor unit and a memory unit.
(C) includes many processing units under the supervision of a common control unit
(D) none of the above.

Ans: C

Q.3 Floating point representation is used to store
(A) Boolean values **(B)** whole numbers
(C) real integers **(D)** integers

Ans: C

Q.4 Suppose that a bus has 16 data lines and requires 4 cycles of 250 nsecs each to transfer data. The bandwidth of this bus would be 2 Megabytes/sec. If the cycle time of the bus was reduced to 125 nsecs and the number of cycles required for transfer stayed the same what would the bandwidth of the bus?
(A) 1 Megabyte/sec **(B)** 4 Megabytes/sec
(C) 8 Megabytes/sec **(D)** 2 Megabytes/sec

Ans: D

Q.5 Assembly language

(A) uses alphabetic codes in place of binary numbers used in machine language
(B) is the easiest language to write programs
(C) need not be translated into machine language

(D) None of these

Ans: A

Q.6 In computers, subtraction is generally carried out by

(A) 9's complement **(B)** 10's complement
(C) 1's complement **(D)** 2's complement

Ans: D

Q.7 The amount of time required to read a block of data from a disk into memory is composed of seek time, rotational latency, and transfer time. Rotational latency refers to

(A) the time its takes for the platter to make a full rotation
(B) the time it takes for the read-write head to move into position over the appropriate track
(C) the time it takes for the platter to rotate the correct sector under the head
(D) none of the above

Ans: A

Q.8 What characteristic of RAM memory makes it not suitable for permanent storage?

(A) too slow **(B)** unreliable
(C) it is volatile **(D)** too bulky

Ans: C

Q.9 Computers use addressing mode techniques for _____.

(A) giving programming versatility to the user by providing facilities as pointers to memory counters for loop control
(B) to reduce no. of bits in the field of instruction
(C) specifying rules for modifying or interpreting address field of the instruction
(D) All the above

Ans: D

Q.10 The circuit used to store one bit of data is known as

(A) Register **(B)** Encoder
(C) Decoder **(D)** Flip Flop

Ans: D

Q. 11 $(2FAOC)_{16}$ is equivalent to

(A) $(195\ 084)_{10}$ **(B)** $(001011111010\ 0000\ 1100)_2$
(C) Both **(A)** and **(B)** **(D)** None of these
Ans: B

Q.12 The average time required to reach a storage location in memory and obtain its contents is called the
(A) seek time **(B)** turnaround time
(C) access time **(D)** transfer time

Ans: C

Q.13 Which of the following is not a weighted code?
(A) Decimal Number system **(B)** Excess 3-cod
(C) Binary number System **(D)** None of these

Ans: B

Q.14 The idea of cache memory is based
(A) on the property of locality of reference
(B) on the heuristic 90-10 rule
(C) on the fact that references generally tend to cluster
(D) all of the above

Ans: A

Q.15 _____ register keeps track of the instructions stored in program stored in memory.
(A) AR (Address Register) **(B)** XR (Index Register)
(C) PC (Program Counter) **(D)** AC (Accumulator)

Ans: C

Q.16 The addressing mode used in an instruction of the form ADD X Y, is
(A) Absolute **(B)** indirect
(C) index **(D)** none of these

Ans: C

Q.17 If memory access takes 20 ns with cache and 110 ns with out it, then the ratio (cache uses a 10 ns memory) is
(A) 93% **(B)** 90%
(C) 88% **(D)** 87%
Ans: B

Q.18 In a memory-mapped I/O system, which of the following will not be there?
(A) LDA **(B)** IN
(C) ADD **(D)** OUT

Ans: A

Q.19 In a vectored interrupt.
(A) the branch address is assigned to a fixed location in memory.
(B) the interrupting source supplies the branch information to the processor through an interrupt vector.
(C) the branch address is obtained from a register in the processor
(D) none of the above

Ans: B

Q.20 Von Neumann architecture is
(A) SISD **(B)** SIMD
(C) MIMD **(D)** MISD

Ans: A

Q. 21 The circuit used to store one bit of data is known as
(A) Encoder **(B)** OR gate
(C) Flip Flop **(D)** Decoder

Ans: C

Q.22 Cache memory acts between
(A) CPU and RAM **(B)** RAM and ROM
(C) CPU and Hard Disk **(D)** None of these

Ans: A

Q.23 Write Through technique is used in which memory for updating the data
(A) Virtual memory **(B)** Main memory
(C) Auxiliary memory **(D)** Cache memory

Ans: D

Q.24 Generally Dynamic RAM is used as main memory in a computer system as it
(A) Consumes less power **(B)** has higher speed
(C) has lower cell density **(D)** needs refreshing circuitary

Ans: B

Q.25 In signed-magnitude binary division, if the dividend is $(11100)_2$ and divisor is $(10011)_2$ then the result is

(A) $(00100)_2$ **(B)** $(10100)_2$
(C) $(11001)_2$ **(D)** $(01100)_2$

Ans: B

Q.26 Virtual memory consists of
(A) Static RAM **(B)** Dynamic RAM
(C) Magnetic memory **(D)** None of these

Ans: A

Q.27 In a program using subroutine call instruction, it is necessary
(A) initialise program counter
(B) Clear the accumulator
(C) Reset the microprocessor
(D) Clear the instruction register

Ans: D

Q.28 A Stack-organised Computer uses instruction of
(A) Indirect addressing **(B)** Two-addressing
(C) Zero addressing **(D)** Index addressing

Ans: C

Q.29 If the main memory is of 8K bytes and the cache memory is of 2K words. It uses associative mapping. Then each word of cache memory shall be
(A) 11 bits **(B)** 21 bits
(C) 16 bits **(D)** 20 bits

Ans: C

Q.30 A-Flip Flop can be converted into T-Flip Flop by using additional logic circuit
(A) $D = T \bullet Q_n$ **(B)** $D = \overline{T}$
(C) $D = T \cdot Q_n$ **(D)** $D = T \oplus Q_n$

Ans: D

Q.31 Logic X-OR operation of $(4ACO)_H$ & $(B53F)_H$ results
(A) AACB **(B)** 0000
(C) FFFF **(D)** ABCD

Ans: C

Q.32 When CPU is executing a Program that is part of the Operating System, it is said to be in
(A) Interrupt mode
(B) System mode
(C) Half mode
(D) Simplex mode

Ans: B

Q.33 An n-bit microprocessor has
(A) n-bit program counter
(B) n-bit address register
(C) n-bit ALU
(D) n-bit instruction register

Ans: D

Q.34 Cache memory works on the principle of
(A) Locality of data
(B) Locality of memory
(C) Locality of reference
(D) Locality of reference & memory

Ans: C

Q.35 The main memory in a Personal Computer (PC) is made of
(A) cache memory.
(B) static RAM
(C) Dynamic Ram
(D) both **(A)** and **(B)**.

Ans: D

Q.36 In computers, subtraction is carried out generally by
(A) 1's complement method
(B) 2's complement method
(C) signed magnitude method
(D) BCD subtraction method

Ans: B

Q.37 PSW is saved in stack when there is a
(A) interrupt recognised
(B) execution of RST instruction
(C) Execution of CALL instruction
(D) All of these

Ans: A

Q.38 The multiplicand register & multiplier register of a hardware circuit implementing booth's algorithm have (11101) & (1100). The result shall be
(A) $(812)_{10}$
(B) $(-12)_{10}$
(C) $(12)_{10}$
(D) $(-812)_{10}$

Ans: A

Q.39 The circuit converting binary data in to decimal is
(A) Encoder **(B)** Multiplexer
(C) Decoder **(D)** Code converter

Ans: D

Q.40 A three input NOR gate gives logic high output only when
(A) one input is high **(B)** one input is low
(C) two input are low **(D)** all input are high

Ans: D

Q.41 n bits in operation code imply that there are _____ possible distinct operators
(A) 2n **(B)** 2^n
(C) n/2 **(D)** n^2

Ans: B

Q.42 _____ register keeps tracks of the instructions stored in program stored in memory.
(A) AR (Address Register) **(B)** XR (Index Register)
(C) PC (Program Counter) **(D)** AC (Accumulator)

Ans: C

Q.43 Memory unit accessed by content is called
(A) Read only memory **(B)** Programmable Memory
(C) Virtual Memory **(D)** Associative Memory

Ans: D

Q.44 'Aging registers' are
(A) Counters which indicate how long ago their associated pages have been referenced.
(B) Registers which keep track of when the program was last accessed.
(C) Counters to keep track of last accessed instruction.
(D) Counters to keep track of the latest data structures referred.
Ans: A

Q.45 The instruction 'ORG O' is a
(A) Machine Instruction. **(B)** Pseudo instruction.
(C) High level instruction. **(D)** Memory instruction.
Ans: B

Q.46 Translation from symbolic program into Binary is done in
(A) Two passes. **(B)** Directly
(C) Three passes. **(D)** Four passes.

Ans: A

Q.47 A floating point number that has a O in the MSB of mantissa is said to have
(A) Overflow **(B)** Underflow
(C) Important number **(D)** Undefined

Ans: B

Q.48 The BSA instruction is
(A) Branch and store accumulator
(B) Branch and save return address
(C) Branch and shift address
(D) Branch and show accumulator

Ans: B

Q.49 State whether True or False.
(i) Arithmetic operations with fixed point numbers take longer time for execution as compared to with floating point numbers.
Ans: True.

(ii) An arithmetic shift left multiplies a signed binary number by 2.
Ans: False.

Q.50 Logic gates with a set of input and outputs is arrangement of
(A) Combinational circuit **(B)** Logic circuit
(C) Design circuits **(D)** Register

Ans: A

Q.51 MIMD stands for
(A) Multiple instruction multiple data
(B) Multiple instruction memory data
(C) Memory instruction multiple data
(D) Multiple information memory data

Ans: A

Q.52 A k-bit field can specify any one of
(A) 3^k registers **(B)** 2^k registers
(C) K^2 registers **(D)** K^3 registers

Ans: B

Q.53 The time interval between adjacent bits is called the
(A) Word-time **(B)** Bit-time
(C) Turn around time **(D)** Slice time

Ans: B

Q.54 A group of bits that tell the computer to perform a specific operation is known as
(A) Instruction code **(B)** Micro-operation
(C) Accumulator **(D)** Register

Ans: A

Q.55 The load instruction is mostly used to designate a transfer from memory to a processor register known as
(A) Accumulator **(B)** Instruction Register
(C) Program counter **(D)** Memory address Register

Ans: A

Q.56 The communication between the components in a microcomputer takes place via the address and
(A) I/O bus **(B)** Data bus
(C) Address bus **(D)** Control lines

Ans: B

Q.57 An instruction pipeline can be implemented by means of
(A) LIFO buffer **(B)** FIFO buffer
(C) Stack **(D)** None of the above

Ans: B

Q.58 Data input command is just the opposite of a
(A) Test command **(B)** Control command
(C) Data output **(D)** Data channel

Ans: C

Q.59 A microprogram sequencer
(A) generates the address of next micro instruction to be executed.
(B) generates the control signals to execute a microinstruction.
(C) sequentially averages all microinstructions in the control memory.
(D) enables the efficient handling of a micro program subroutine.

Ans: A

Q.60 A binary digit is called a
(A) Bit **(B)** Byte
(C) Number **(D)** Character

Ans: A

Q.61 A flip-flop is a binary cell capable of storing information of
(A) One bit **(B)** Byte
(C) Zero bit **(D)** Eight bit

Ans: A

Q.62 The operation executed on data stored in registers is called
(A) Macro-operation **(B)** Micro-operation
(C) Bit-operation **(D)** Byte-operation

Ans: B

Q.63 MRI indicates
(A) Memory Reference Information.
(B) Memory Reference Instruction.
(C) Memory Registers Instruction.
(D) Memory Register information

Ans: B

Q.64 Self-contained sequence of instructions that performs a given computational task is called
(A) Function **(B)** Procedure
(C) Subroutine **(D)** Routine

Ans: A

Q.65 Microinstructions are stored in control memory groups, with each group specifying a
(A) Routine **(B)** Subroutine
(C) Vector **(D)** Address

Ans: A

Q.66 An interface that provides a method for transferring binary information between internal storage and external devices is called
(A) I/O interface **(B)** Input interface
(C) Output interface **(D)** I/O bus

Ans: A

Q.67 Status bit is also called
(A) Binary bit **(B)** Flag bit
(C) Signed bit **(D)** Unsigned bit

Ans: B

Q.68 An address in main memory is called

(A) Physical address **(B)** Logical address
(C) Memory address **(D)** Word address

Ans: A

Q.69 If the value V(x) of the target operand is contained in the address field itself, the addressing mode is
(A) immediate. **(B)** direct.
(C) indirect. **(D)** implied.

Ans: B

Q.70 $(-27)_{10}$ can be represented in a signed magnitude format and in a 1's complement format as
(A) 111011 & 100100 **(B)** 100100 & 111011
(C) 011011 & 100100 **(D)** 100100 & 011011

Ans: A

Q.71 The instructions which copy information from one location to another either in the processor's internal register set or in the external main memory are called
(A) Data transfer instructions. **(B)** Program control instructions.
(C) Input-output instructions. **(D)** Logical instructions.

Ans: A

Q.72 A device/circuit that goes through a predefined sequence of states upon the application of input pulses is called
(A) register **(B)** flip-flop
(C) transistor. **(D)** counter.

Ans: D

Q.73 The performance of cache memory is frequently measured in terms of a quantity called
(A) Miss ratio. **(B)** Hit ratio.
(C) Latency ratio. **(D)** Read ratio.
Ans: C

Q.74 The information available in a state table may be represented graphically in a
(A) simple diagram. **(B)** state diagram.
(C) complex diagram. **(D)** data flow diagram.
Ans: B

Q.75 Content of the program counter is added to the address part of the instruction in order to obtain the effective address is called.
(A) relative address mode. **(B)** index addressing mode.
(C) register mode. **(D)** implied mode.

Ans: A

Q.76 An interface that provides I/O transfer of data directly to and form the memory unit and peripheral is termed as

(A) DDA. **(B)** Serial interface.
(C) BR. **(D)** DMA.

Ans: D

Q.77 The 2s compliment form (Use 6 bit word) of the number 1010 is

(A) 111100. **(B)** 110110.
(C) 110111. **(D)** 1011.

Ans: B

Q.78 A register capable of shifting its binary information either to the right or the left is called a

(A) parallel register. **(B)** serial register.
(C) shift register. **(D)** storage register.

Ans: C

Q.79 What is the content of Stack Pointer (SP)?

(A) Address of the current instruction
(B) Address of the next instruction
(C) Address of the top element of the stack
(D) Size of the stack.

Ans: C

Q.80 Which of the following interrupt is non maskable

(A) INTR. **(B)** RST 7.5.
(C) RST 6.5. **(D)** TRAP.

Ans: D

Q.81 Which of the following is a main memory

(A) Secondary memory. **(B)** Auxiliary memory.
(C) Cache memory. **(D)** Virtual memory.

Ans: C

Q.82 Which of the following are not a machine instructions

(A) MOV. **(B)** ORG.
(C) END. **(D) (B)** & **(C)**.

Ans: D

Q.83 In Assembly language programming, minimum number of operands required for an instruction is/are

(A) Zero. **(B)** One.
(C) Two. **(D)** Both **(B)** & **(C)**.

Ans: A

Q.84 The maximum addressing capacity of a micro processor which uses 16 bit database & 32 bit address base is

(A) 64 K. **(B)** 4 GB.
(C) both **(A)** & **(B)**. **(D)** None of these.

Ans: B

Q.85 The memory unit that communicates directly with the CPU is called the

(A) main memory **(B)** Secondary memory
(C) shared memory **(D)** auxiliary memory.

Ans: A

Q.86 The average time required to reach a storage location in memory and obtain its contents is called

(A) Latency time. **(B)** Access time.
(C) Turnaround time. **(D)** Response time.

Ans: B

State True or False

Q.87 A byte is a group of 16 bits.

Ans: False

Q.88 A nibble is a group of 16 bits.

Ans: False

Q.89 When a word is to be written in an associative memory, address has got to be given.

Ans: False

Q.90 When two equal numbers are subtracted, the result would be _____and not_____.

Ans: +ZERO, -ZERO.

Q.91 A _____development system and an _____are essential tools for writing large assembly language programs.

Ans: Microprocessor, assembler

Q.92 In an operation performed by the ALU, carry bit is set to 1 if the end carry C_8 is _____. It is cleared to 0 (zero) if the carry is _____ _____.

Ans: One, zero

Choose the correct alternative

Q.93 A successive A/D converter is

(A) a high-speed converter. **(B)** a low speed converter.
(C) a medium speed converter. **(D)** none of these.

Ans: C

Q.94 When necessary, the results are transferred from the CPU to main memory by

(A) I/O devices. **(B)** CPU.
(C) shift registers. **(D)** none of these.

Ans: B

Q.95 The gray code equivalent of $(1011)_2$ is

(A) 1101. **(B)** 1010.
(C) 1110. **(D)** 1111.

Ans: C

Q.96 A combinational logic circuit which sends data coming from a single source to two or more separate destinations is

(A) Decoder. **(B)** Encoder.
(C) Multiplexer. **(D)** Demultiplexer.

Ans: D

Q.97 In which addressing mode the operand is given explicitly in the instruction

(A) Absolute. **(B)** Immediate.
(C) Indirect. **(D)** Direct.

Ans: B

Q.98 A stack organized computer has

(A) Three-address Instruction. **(B)** Two-address Instruction.
(C) One-address Instruction. **(D)** Zero-address Instruction.

Ans: D

Q.99 A Program Counter contains a number 825 and address part of the instruction contains the number 24. The effective address in the relative address mode, when an instruction is read from the memory is

(A) 849. **(B)** 850.
(C) 801. **(D)** 802.

Ans: B

Q.100 A system program that translates and executes an instruction simultaneously is

(A) Compiler. **(B)** Interpreter.
(C) Assembler. **(D)** Operating system.

Ans: C

Q.101 The cache memory of 1K words uses direct mapping with a block size of 4 words. How many blocks can the cache accommodate.

(A) 256 words. **(B)** 512 words.
(C) 1024 words. **(D)** 128 words.

Ans: A

Q.102 A page fault

(A) Occurs when there is an error in a specific page.
(B) Occurs when a program accesses a page of main memory.
(C) Occurs when a program accesses a page not currently in main memory.
(D) Occurs when a program accesses a page belonging to another program.

Ans: C

DESCRIPTIVES

Q.1 Simplify the Boolean function F together with don't care conditions d in sum-of-products and product-of-sum forms

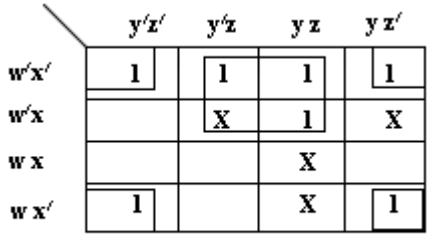
$$F(w,x,y,z) = \Sigma(0,1,2,3,7,8,10)$$
$$d(w,x,y,z) = \Sigma(5,6,11,15)$$ **(6)**

Ans:

Given function

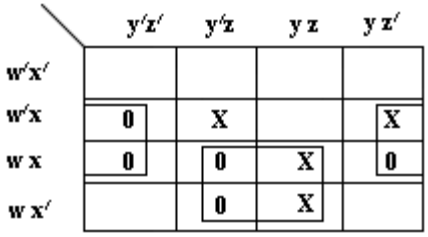
$$F(w,x,y,z) = \Sigma(0,1,2,3,7,8,10)$$
$$d(w,x,y,z) = \Sigma(5,6,11,15)$$

Sum of products



$F = w'z + x'z'$

Product of sums



$F = (w'+z')(x'+z)$

Q.2 Represent the condition control statement by two register transfer statements with control functions.

If (P=1) then $(R_1 \leftarrow R_2)$ else if (Q=1) then $(R_1 \leftarrow R_3)$ **(6)**

Ans:

The two register transfer statements are

P: $R_1 \leftarrow R_2$

P′Q: $R_1 \leftarrow R_3$

Q.3 Explain the use of subroutine with the help of suitable example. **(4)**

Ans:
A subroutine is a self-contained sequence of instructions that performs a given computational task. During the program execution, a subroutine can be called many times to perform its operations. Each time a subroutine is called, a branch is executed to start executing its set of instruction and a branch is made back to the main program when the subroutine has been executed. The instruction that causes the transfer of program control to a subroutine is known by different names. Some common names are called subroutine, jump to subroutine or branch and save address.

The call subroutine instruction consists of an operation code and the address that specifies the beginning of subroutine. The instruction is executed by performing two operations (i) the address of the next instruction i.e. the return address in the program counter is stored at some temporary location, (ii) the control is transferred at the beginning of the subroutine. The last instruction of every subroutine, called as return from subroutine, transfers the return address stored in temporary location into the program counter. This return address helps in a transfer of program control to the instruction whose address was stored in temporary location. This temporary location can be the first memory location of the subroutine, or some fixed memory location or a processor register or can be a memory stack. But most efficient way is to store the return address in a memory stack.

Q.4 Justify the statement "Stack computer consists of an operation code only with no address field". **(5)**

Ans:
Stack-oriented machines do not contain any accumulator or general-purpose registers. Computers with stack organization have PUSH and POP instructions which requires an address field. Thus the instruction

PUSH X TOP ← M[X]

will push the data at address X to the top of the stack. The SP is automatically updated. The operation instruction does not contain any address field because the operation is performed on two top most operands of the stack. For example,

ADD

The instruction ADD consists of only operation code with no address field. This instruction pops the top two operands from the stack, add the numbers and then PUSH the result into the stack.

Q.5 Given that following are all the instructions related to AR, determine the control functions load(LD), clear(CLR) and increment(INC) for AR.

$R'T_0 : AR \leftarrow PC$

$R'T_2 : AR \leftarrow IR(0-11)$

$$D_7' I T_3 : AR \leftarrow M[AR]$$
$$RT_0 : AR \leftarrow 0$$
$$D_5 T_4 : AR \leftarrow AR + 1 \qquad \textbf{(6)}$$

Ans:

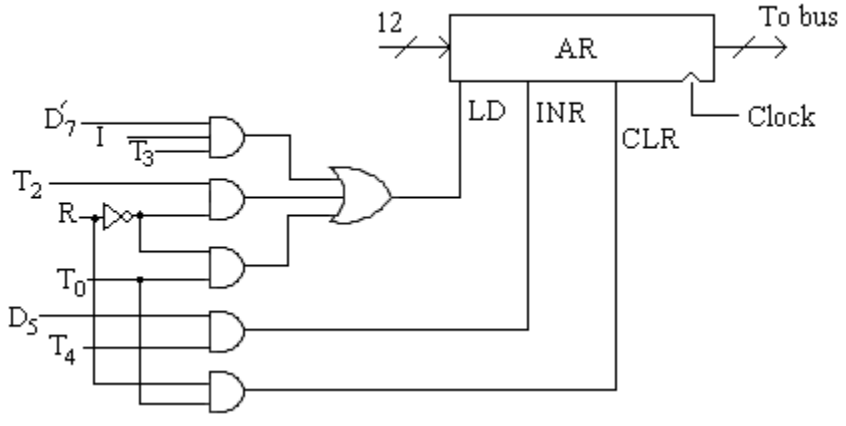
Given instructions are

$$R'T_0 : AR \leftarrow PC$$
$$R'T_2 : AR \leftarrow IR(0-11)$$
$$D_7' I T_3 : AR \leftarrow M[AR]$$
$$RT_0 : AR \leftarrow 0$$
$$D_5 T_4 : AR \leftarrow AR + 1$$

The first three statements specify transfer of information from a register or memory to AR. The content of the source register or memory is placed on the bus and the content of the bus is transferred into AR by enabling its LD control input. The fourth statement clears AR to 0. The last statement increments AR by 1. The control functions can be combined into three Boolean expressions as follows:

$$LD(AR) = R'T_0 + R'T_2 + D_7'IT_3$$
$$CLR(AR) = RT_0$$
$$INR(AR) = D_5T_4$$

Where LD(AR) is the load input of AR, CLR(AR) is the clear input of AR, and INR(AR) is the increment input of AR. The control gate logic associated with AR is shown in figure below.



Control gates associated with AR

Q.6 Explain indirect address mode and how the effective address is calculated in this case. **(5)**

Ans:

In indirect addressing mode the address field of the instruction gives the address where the operand is stored in the memory. Control fetches the instruction from memory and uses its address part to access memory again to read the effective address.

Opcode	Memory Address = Operand Address

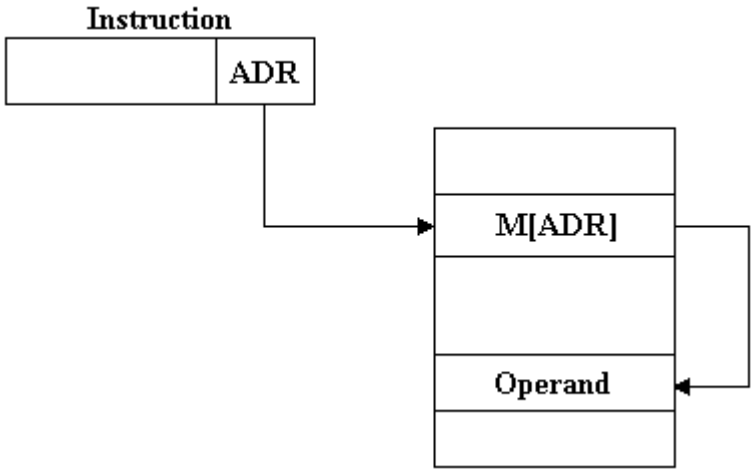
Indirect Addressing Mode

For Example,

Efffective address = M[ADR]

LD@ADR

AC ←M[M[ADR]]



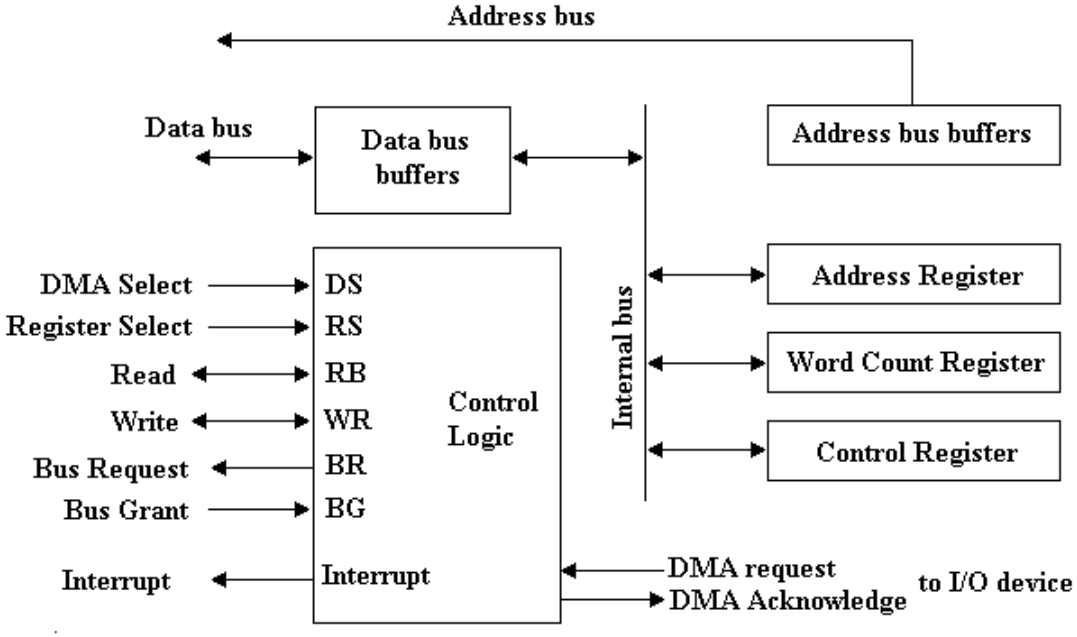
Q.7 Why is read and write control lines in a DMA controller bidirectional? Under what condition and for what purpose are they used as inputs? **(4)**

Ans:

The DMA controller consists of circuits of an interface to communicate with the CPU and I/O device. It also has an address register, a word count register, a set of address lines. The address register and address lines are used for direct communication with the memory. The word count register specifies the number of words that must be transferred. The figure shows the block diagram of DMA controller. The unit communicates with the CPU via the data bus and control lines. The registers in the DMA are selected by the CPU through the address bus by

enabling the DS (DMA select) and RS (register select) inputs. When the BG (bus grant) input is 0, the CPU can communicate with the DMA registers through the data bus to read from or write to the DMA registers. When BG=1, the CPU relinquished the buses and the DMA can communicate directly with the memory by specifying an address in the address bus and activating the RD or WR control.

The DMA controller has three registers. The address register contains an address to specify the desired location in memory and the address bits goes into the address bus through address bus buffers. The address register is incremented after each word that is transferred from/to memory. The word count register holds the number of words to be transferred to memory. This register is decremented by one after each word transferred from/to memory and internally tested for zero. The control register specifies the mode of transfer. All registers in DMA appears to the CPU as I/O interface registers and hence the CPU can read from or write into the DMA registers.



Q.8 Explain the different types of mapping procedures in the organization of cache memory with diagram. **(12)**

Ans:

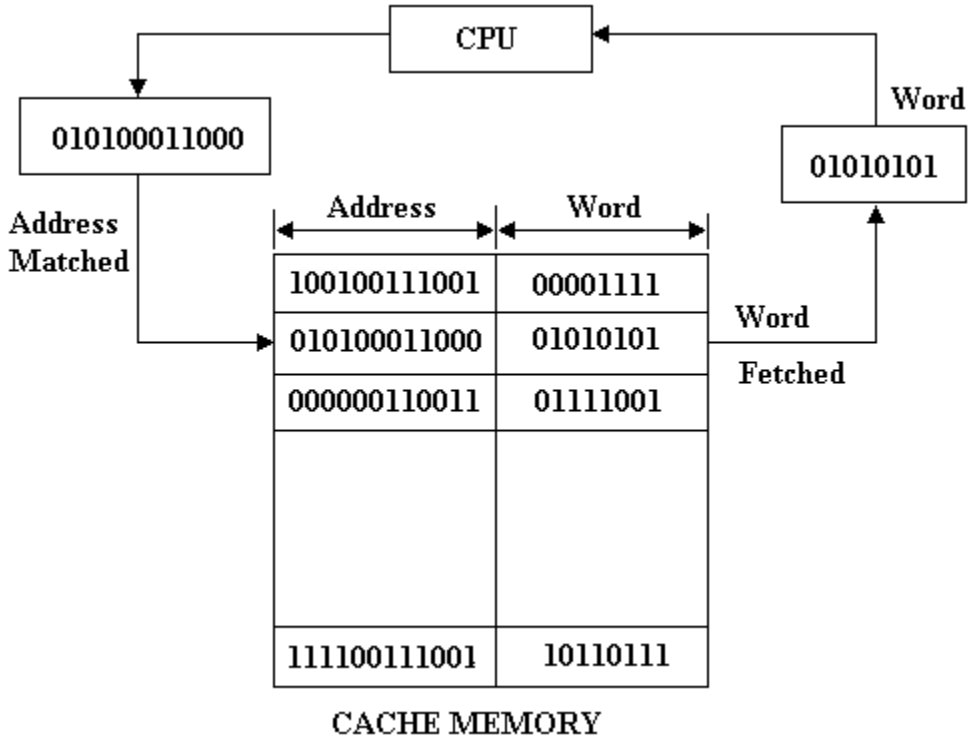
Associative Mapping

In case of associative mapping, the contents of cache memory are not associated with any address. Data stored in the cache memory are not accessed by specifying any address. Instead, data or part of data is searched by matching with the contents. In associative mapping method, both the word and the address of the word (in the main memory) are stored in the cache as shown in Figure. The address bits, sent by the CPU to search, are matched with addresses stored in the cache memory. If any address is matched, the corresponding word is fetched from the cache and sent to the CPU.

If not match is found in cache memory, the word is searched in the main memory. The word along with address is then copied from main memory into cache. If the cache is full, then the existing word along with its address must be removed to make room for the new word.

Associative mapping has the advantage that it is a very fast access method, but it has the disadvantage that it is very expensive and complicated because of complex logical circuits that ate required to implement data searching by content and not by address.

Due to the high cost associated with logic circuits required to implement associative mapping, other methods are used in which data is cache memory are accessed by address, like direct mapping and set associative mapping.



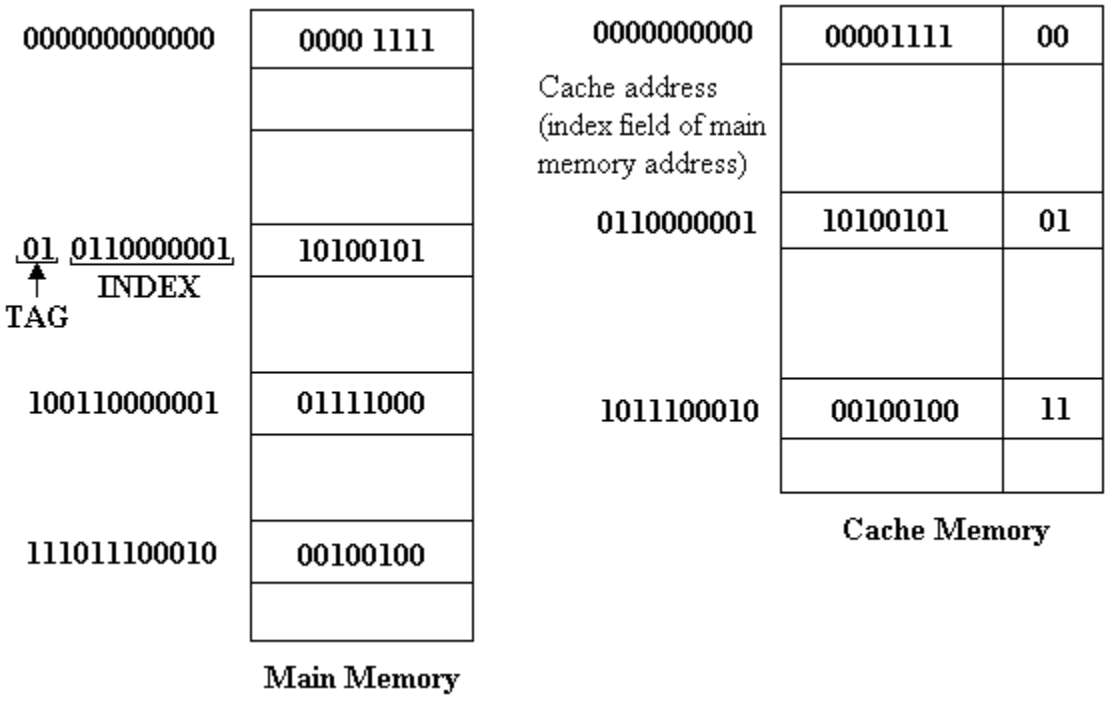
Direct Mapping

Suppose a computer has 4K main memory i.e. 4x1024 bytes and 1K cache memory. To address a word in the main memory, a 12 bit ($4K = 2^{12}$) address is required. Similarly, to address a word in the cache memory a 10-bit ($1K = 2^{10}$) address is required. Thus, a cache memory needs 10 bits to address a word and main memory requires 12 bits to address a word. In direct mapping method, the 12 bit address sent by CPU is divided into two parts called tag field and index field. The index field has the number of bits equal to the number of bits required to address a word in cache.

Thus, if a computer has main memory of capacity 2^m and cache memory of 2^n, then the address bits will be divided into n bits index field and (m-n) bits of tag field. The

tag field for above computer system will be of 2-bits and index field will have 10 bits.

In a direct mapping method, the cache memory stored the word as well as the tag field. The words will be stored at that location in the cache which is represented by the index fields of their addresses as shown in figure. When an address is sent by the CPU, the index part of the address is used to get a memory location in the cache. If the tag stored at that location matches the tag field of the requested address, the word is fetched. Else, if tag does not match, the word is searched in the main memory. When a word needs to be moved into cache memory from the main memory, its address I the main memory is divided into index and tag fields. The disadvantage of direct mapping is that the hit ratio can drop considerably if two or more words whose addresses have same index but different tags are accessed repeatedly.



Set Associative mapping

The disadvantage of direct mapping is that two words with same index but different tag cannot be stored into cache at the same time. As an improvement to this disadvantage of direct mapping, a third type of cache organization called set-associative mapping is used. In this mapping process, each word of a cache can store two or more words of main memory under the same index address. Each data word is stored along with its tag and the number of tag data pair in one word of cache is said to form a set. An example of set associative cache organization with set size of two is shown in figure.

INDEX	TAG	DATA	TAG	DATA
0000110010	00	10101010	01	10010110
0000111011	01	11000011	10	11000011

The words stored at address 000000110010 and 010000110010 of main memory are stored in cache memory at index address 0000110010. Similarly, the words stored at address 010000111011 and 100000111011 of main memory are stored in cache memory at index address 0000111011. When CPU generates a memory request, the word is searched into cache with the help of index addresses.

Q.9 What does the instruction field in the assembly language program specify? **(3)**

Ans:
The instruction field specifies a machine instruction or pseudo instruction. The instruction field in an assembly program may specify one of the following:
(i) A memory-reference instruction (MRI)
(ii) A register-reference or input-output instruction (non-MRI)
(iii) A pseudo instruction with or without an operand

A memory-reference instruction occupies two or three symbols separated by spaces. The first must be three letter symbol defining an MRI operation code such as AND, ADD, LDA, STA etc. the second is a symbolic address. The third symbol, which may or may not be present, is the letter I. If I is missing, the line denotes a direct address instruction. The presence of the symbol I denotes an indirect address instruction.

A non-MRI is defined as an instruction that does not have an address part. A non-MRI is recognized in the instruction field of a program by any one of the three-letter symbols for the register-reference and input-output instructions. A symbol address in the instruction field specifies the memory location of an operand. This location must be defined somewhere in the program by appearing again as a label in the first column.

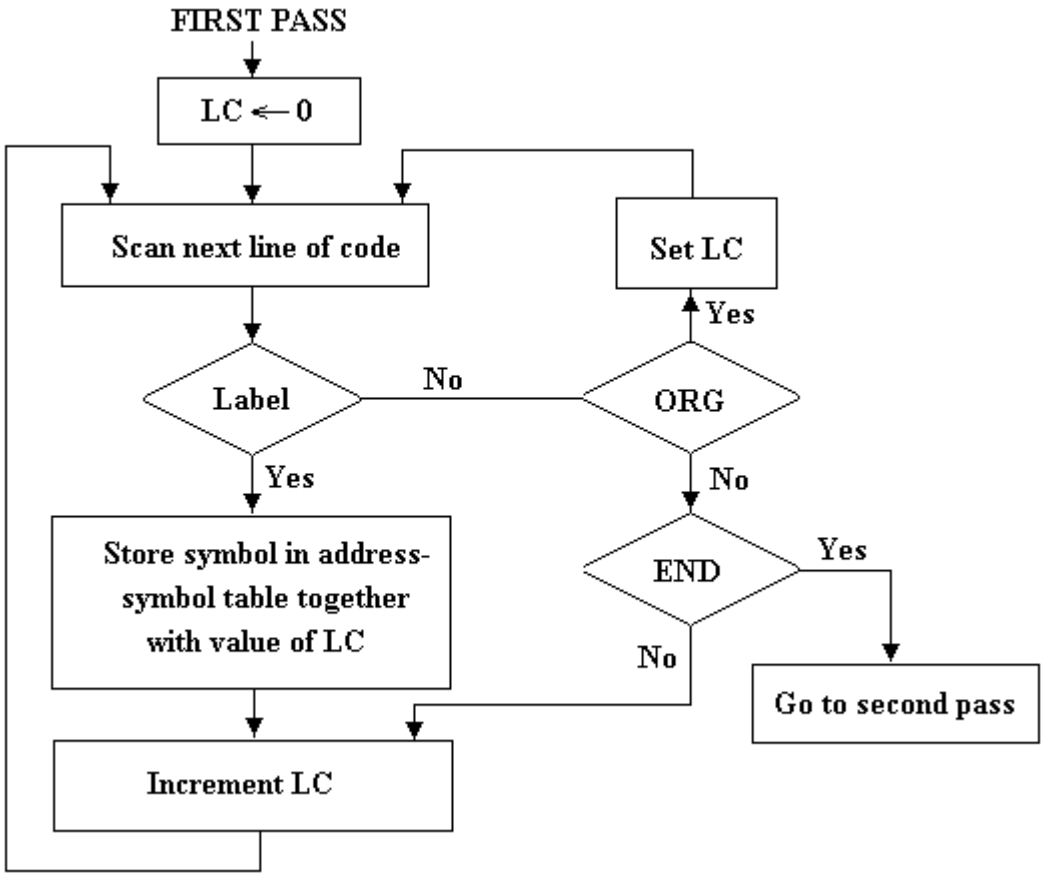
A pseudo instruction is not a machine instruction but rather an instruction to the assembler giving information about some phase of the translation. The ORG (origin) pseudo instruction informs the assembler that the instruction or operand is to be placed in the memory location specified by the number next to ORG. the END

symbol is placed at the end of the program to inform the assembler that the program is terminated.

Q. 10 Explain the working of first pass of assembler using flow chart. **(5)**

Ans:

The tasks performed by the assembler during first pass are explained in the flowchart. LC is initially set to 0. A line of symbolic code is analysed to determine if it has a label. If the line of code has no label, the assembler checks the symbol in the instruction field. If it contains an ORG pseudo instruction, the assembler sets LC to the number that follows ORG and goes back to process the next line. If the line has an END pseudo instruction, the assembler terminates the first pass and goes to the second pass. If the line of code contains a label, it is stored I the address symbol table together with its binary equivalent number specified by the content of LC. Nothing is stored in the table if no label is encountered. LC is then incremented by 1 and a new line of code is processed.



Flowchart for first pass of assembler

Q 11 Write a program in assembly language to subtract two integer numbers. **(8)**

Ans:

Assembly Language Program to Subtract Two Numbers

```
        ORG 100         /Origin of program is location 100
        LDA SUB         /Load subtrahend to AC
        CMA             /Complement AC
        INC             /Increment AC
        ADD MIN         /Add minuend to AC
        STA DIF         /Store difference
        HLT             /Halt computer
MIN,    DEC 83          /Minuend
SUB,    DEC -23         /Subtrahend
DIF,    HEX 0           /Difference stored here
        END             /End of symbolic program
```

Q.12 Explain the difference between vectored and non-vectored interrupt. Explain stating examples of each. **(8)**

Ans:
When interrupt signal is generated the CPU responds to the interrupt signal by storing the return address from program counter into memory stack and then control is transfers to or branches to the service routine that processes the interrupt. The processor chooses the branch address of the service routine in two different ways. One is called *vectored interrupt* and the other is called as *non-vectored interrupt*. In *non-vectored* interrupt, the branch address is assigned to a fixed location in memory. In vectored interrupt, the source that initiated the interrupt supplies the branch information to the computer. This information is called the interrupt vector. This can be first address of the I/O service routine or this address is an address that points to a location in memory where the beginning address of the I/O service routine is stored.

Q.13 Explain the various input output modes of data transfer. **(8)**

Ans:
There are three different ways by which the data can be transferred from input/output devices to the memory. They are
(i) Programmed I/O
(ii) Interrupt Initiated I/O
(iii) Direct Memory Access (DMA)

Programmed I/O
In this mode of data transfer the input/output instructions are written in the form of computer program. The instruction written in the program basically initiates the data transfer to and from a CPU register and Input/Output peripheral devices.

Programmed input/output requires that all input/output operations executed under the direct control of the CPU i.e. every data transfer operation involving an input/output devices requires the execution of an instruction by the CPU. Thus, in programmed input/output mode of transfer, the CPU stays in program loop until the input/output indicates that it is ready for data transfer. This is time-consuming process and keeps the processor busy. This mode of transfer is useful in small, low speed systems where hardware costs must be minimized.

Interrupt Initiated I/O

In interrupt-initiated I/O mode the transfer the CPU concentrate on some other program. This method uses interrupt signal to inform the CPU that the data are available from the peripheral device and the input-output flag is set to 1. When the flat is set the CPU deactivates its task to take care of the input-output transfer. After the transfer has been completed, the CPU returns to continue the previous program.

When the interrupt signal is generated, the CPU responds to it by storing the return address from program counter into the memory stack and the control branches to a service routine that processes the required the input/output transfer. There are two methods to choose the branch address of the service routine. One is called vectored interrupt and the other is called non-vectored interrupt. In vectored interrupt the source that interrupts, gives the branch information to the computer and this information is known as interrupt vector. In non-vectored interrupt, the branch address is assigned to a fixed location in memory.

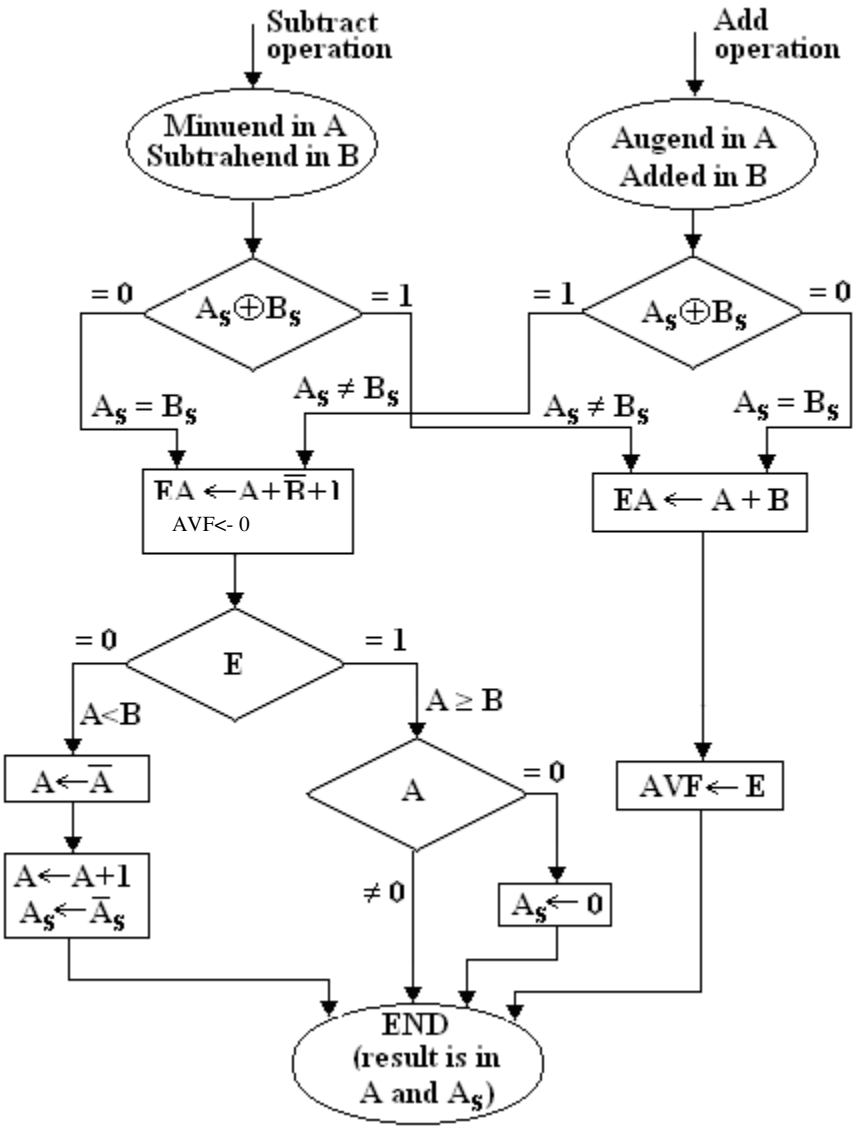
Direct Memory Access (DMA)

In DMA mode of transfer, the interface transfers data into and out of the memory unit through memory bus. The CPU initiates the transfer. It supplies the starting address and number of words to be transferred to the interface unit and then busy with other tasks. When the transfer is made, the DMA disables the bus request line. The transfer of data can be made in several ways. A block sequence consisting of a number of memory words can be transferred into a continuous burst. Alternative way

allows transferring one data word at a time, after which it must return control of the buses to the CPU.

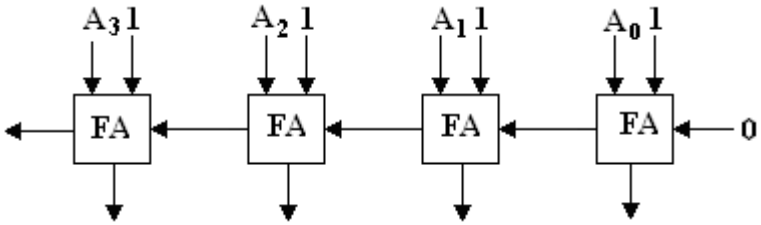
Q.14 Draw a flowchart for adding and subtracting two fixed point binary numbers where negative numbers are signed 1's complement presentation. **(8)**

Ans:



Q.15 Design a 4-bit combinational circuit decrementer using four full adders. **(8)**

Ans:
A – 1 = A + 2's complement of 1



Q.16 Differentiate between virtual and cache memory **(6)**

Ans:

Cache Memory is defined as a very high speed memory that is used in computer system to compensate the seed differential between the main memory access time and processor logic. A very high speed memory called a cache is used to increase the speed of processing by making current programs and data available to the CPU at a rapid rate. It is place between the CPU and the main memory. The cache memory access time is less than the access time of the main memory by a factor of 5 to 10. The cache is used for storing program segments currently being executed in the CPU and the data frequently used in the present calculations. By making programs and data available at a rapid rate, it is possible to increase the performance rate of a computer.

Virtual Memory is a concept used in some large computer system that permit the user to construct programs as though a large space were available, equal to the totality of auxiliary memory. Each address that is referenced by the CPU goes through an address mapping from the so-called virtual address to a physical address in main memory.

A virtual memory system provides a mechanism for translating program generated addresses into correct main memory locations. This is done dynamically, while programs are being executed in the CPU.

Q.17 Discuss Flynn's classification of Computer **(4)**

Ans:

The classification based on the multiplicity of instruction streams and data streams in a computer system is known a Flynn's Classification. Parallel processing can be classified in variety of ways. It can be considered from the internal organization of the processors, from the interconnection structure between processors, or from the flow of information through the system. One classification was introduced by M.J. Flynn, as "How do instructions and data flow in the system?" and this classification is known as Flynn's Classification. Flynn's classified parallel computers into four categories based on how instructions process data. These categories are:

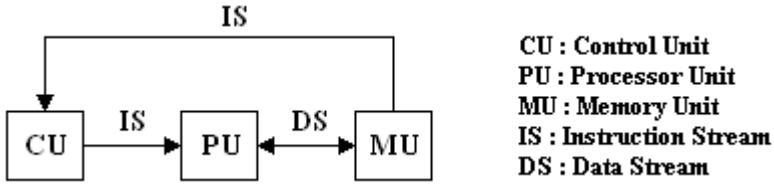
(i) Single Instruction Stream, Single Data Stream (SISD) Computer.
(ii) Single Instruction Stream, Multiple Data Stream (SIMD) Computer.
(iii) Multiple Instruction Stream, Single Data Stream (MISD) Computer.
(iv) Multiple Instruction Stream, Multiple Data Stream (MIMD) Computer.

The normal operation of a computer is to fetch instructions from memory and execute them in a processor. The sequence of instructions read from the memory constitutes an instruction stream. The operations performed on the data in the processor constitute a data stream. Parallel processing may occur in the instruction stream, in the data stream or in both.

Single Instruction Stream, Single Data Stream (SISD)

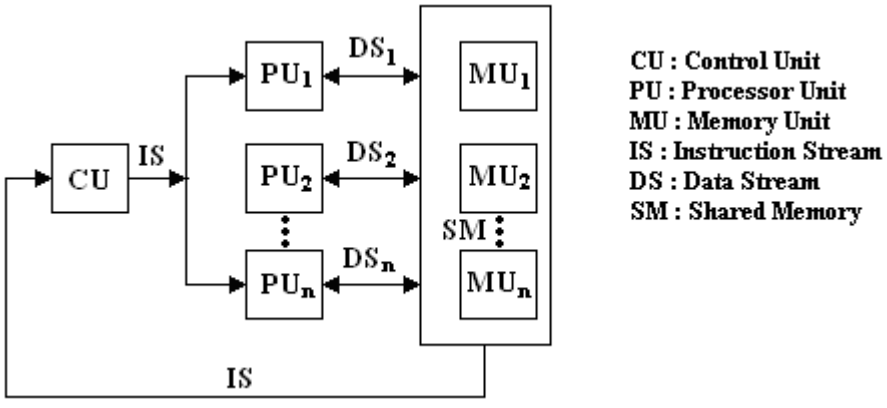
A computer with a single processor is called a Single Instruction Stream, Single Data Stream (SISD) Computer. It represents the organization of a single computer containing a control unit, a processor unit, and a memory unit. Instructions are executed sequentially and the system may or may not have internal parallel processing. Parallel processing may be achieved by means of a pipeline processing.

In such a computer a single stream of instructions and a single stream of data are accessed by the processing elements from the main memory, processed and the results are stored back in the main memory. SISD computer organization is shown in figure below.



Single Instruction Stream, Multiple Data Stream (SIMD)

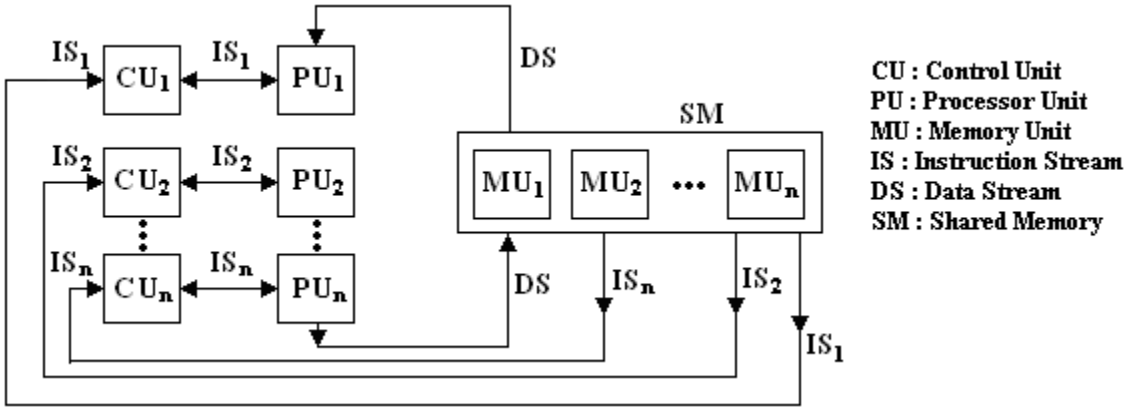
It represents an organization of computer which has multiple processors under the supervision of a common control unit. All processors receive the same instruction from the control unit but operate on different items of the data. SIMD computers are used to solve many problems in science which require identical operations to be applied to different data set synchronously. Examples are added a set of matrices simultaneously, such as $\sum_i \sum_k (a_{ik} + b_{ik})$. Such computers are known as array processors. SIMD computer organization is shown in figure below.



Multiple Instruction Stream, Single Data Stream (MISD)

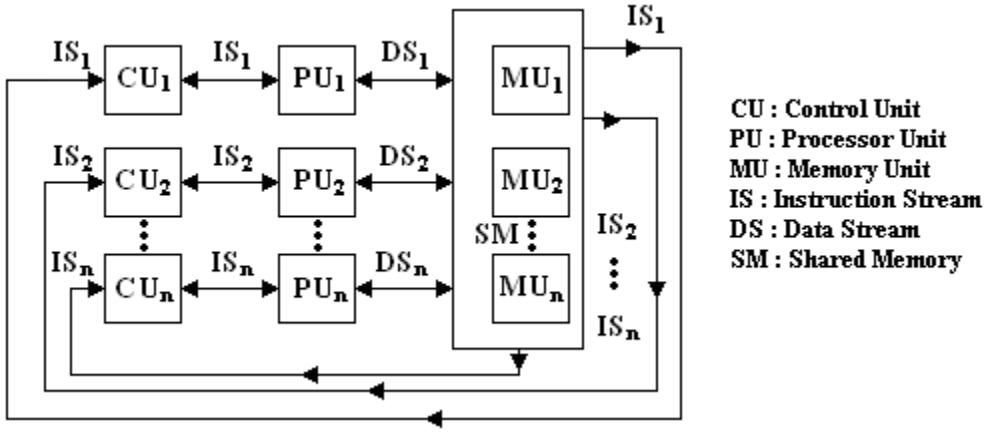
It refers to the computer in which several instructions manipulate the same data stream concurrently. In the structure different processing element run different

programs on the same data. This type of processor may be generalized using a 2-dimensional arrangement of processing elements. Such a structure is known as systolic processor. MISD computer organization is shown in figure below.



Multiple Instruction Stream, Multiple Data Stream (MIMD)

MIMD computers are the general purpose parallel computers. Its organization refers to a computer system capable of processing several programs at a same time. MIMD systems include all multiprocessing systems. MIMD computer organization is shown in figure below.



Q.18 A digital computer has a common bus system for 16 registers of 32 bits each. The bus is constructed with multiplexers.
(i) How many selection inputs are there in each multiplexer?
(ii) How many multiplexers are there in the bus? **(6)**

Ans:

Given, the common bus system consists of 16 registers and each register is of 32 bits.

(i) Since there are 16 registers i.e. 2^4 registers in the common bus system. Hence, 4 selection input lines are there to select one of 16 registers in each multiplexer.

(ii) Each register is of 32 bits, hence 32 multiplexers, one of each bit of the registers are there in the bus.

Q.19 Write a program to evaluate the arithmetic statement

$$Y = \frac{A - B + C}{G + H}$$

(i) Using an accumulator type computer with one address instruction.
(ii)Using a stack organized computer with zero-address instructions. **(8)**

Ans:

(i) Using one address instruction

Load A	AC ← M[A]
Sub B	AC ← AC – M[B]
Add C	AC ← AC + M[C]
Store T	M[T] ← AC
Load G	AC ← M[G]
Add H	AC ← AC + M[H]
Store Y	M[Y] ← AC
Load T	AC ← M[T]
Div Y	AC ← AC/M[Y]
Store Y	M[Y] ← AC

(ii) Using zero address instruction

Reverse polish notation of $Y = \frac{A - B + C}{G + H}$ is Y= AB–C+GH+/

Push A	TOS ← A
Push B	TOS ← B
Sub	TOS ← A – B
Push C	TOS ← C
Add	TOS ← A – B + C
Push G	TOS ← G
Push H	TOS ← H
Add	AC ← G + H
Div	TOS ← (A – B + C) / (G + H)
Pop Y	M[Y] ← TOS

Q.20 Explain the sequence that takes place when an interrupt occurs. **(8)**

Ans:
When an interrupt occur the three distinct actions as shown in Figure done automatically by the CPU is as follows:
(a) Saves the content and status of various CPU registers.

(b) Finding the cause of an interrupt.
(c) Branch the control to the Interrupt Service Routine (ISR).

Saving Content of CPU Registers

The interruption should not affect the result or logic of the current program. Hence, the CPU stores the content of various registers and the address of the next instruction to be executed so that once the interrupt has been serviced; the current program resumes its execution having the same condition. For saving the content and status of CPU registers the two operations can be:

(a) Certain registers in the CPU can be reserved for saving CPU status.
(b) Some location in the main memory can be reserved for saving CPU status.

If the internal registers of the CPU are used to save the status and content of different registers, the time taken for storing is less as compared to storing the CPU status at some locations in main memory. Generally, a part of memory is used as a stack for this purpose.

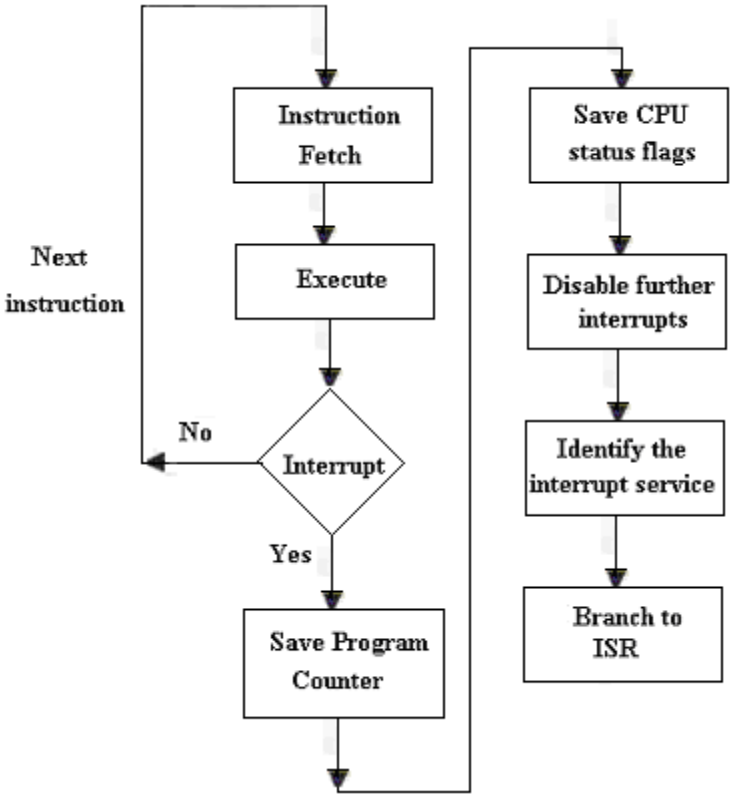
Identifying the interrupt source

For identifying the cause of interrupt or the source of interrupt, the CPU transmits interrupt acknowledgement (INTA) signal to the interrupt controller. The INTA signal conveys the following two messages to the interrupt controller:

(a) The CPU has sensed the interrupt signal issued by the interrupt controller.
(b) The interrupt controller should inform the CPU about the interrupt request should be serviced.

Branching to Interrupt Service Routine (ISR)

Once the source of the interrupt is found, the CPU should branch or transfer the control to the corresponding interrupt service routine (ISR) for taking necessary action. For this purpose, the CPU should have a mechanism to find the start addresses of the different ISRs. A common method is to store the ISRs of different interrupt in memory at different locations. But the start addresses of ISRs are stored at fixed location in memory one after another in form of a table. The CPU fetches the start address from the table and then loads the start address in program counter. And hence, the execution of the ISR is handled by CPU.



Interrupt Service Sequence

Q.21 Karnaugh Maps are useful for finding minimal implementations of Boolean expressions with only a few variables. Using the following K-Map:

(i) Find the minimal sum of products expression. Show your groupings.

(ii) Find the minimal sum of products expression. Show your groupings.

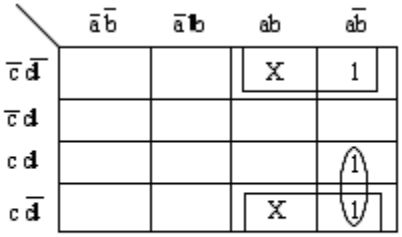
(iii) Are your solutions unique? If not, list and show the other minimal expression?

(iv) Does the minimal product of sums expression equal to minimal sum of products expression?

cd \ ab	00	01	11	10
00	0	0	X	1
01	0	0	0	0
11	0	0	0	1
10	0	0	X	1

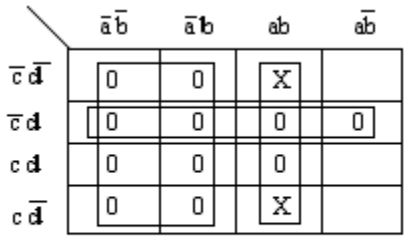
Ans:

i) Sum of Product



$F = \bar{d}a + c\,a\bar{b}$

ii) Product of Sum



$F' = a' + a\,b + c'\,d\,a$

$F = (a)(a'+b')(c+d'+a')$

iii) Our solution is unique. No other minimal expression.

iv) The minimal product of sums expression never equal to minimal sum of product expression

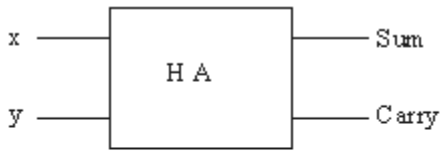
Q.22 A half-adder is a combinational logic circuit that has two inputs, x and y and two outputs, s and c that are the sum and carry-out, respectively, resulting from binary addition of x and y.

(i) Design a half-adder as a two-level AND-OR circuit.

(ii) Show how to implement a full adder, by using half adders.

Ans:

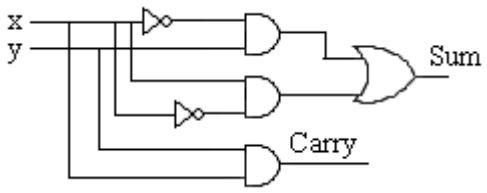
(i) Half-adder:-



Equations:-

$S = x'y + x\,y'$

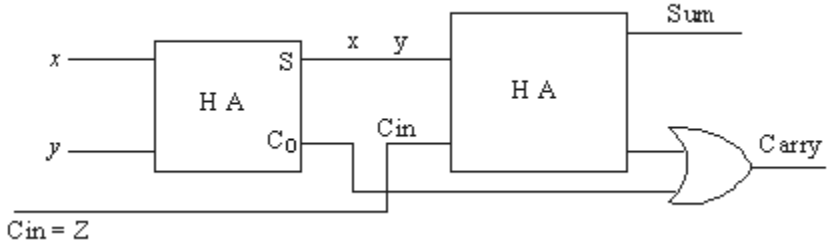
$C = x\,y$



Truth Table

INPUT		OUTPUT	
X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

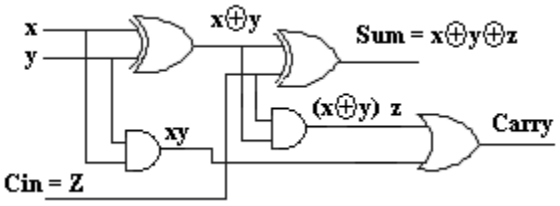
(ii) Full-Adder Circuit



Q.23 Show how to implement a full adder, by using half adders.

Ans:

Full-adder circuit by using two half-adder



Equations:

$S = \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz$

$C = xy + xz + yz$

Truth-Table

X	**Y**	**Z**	**C**	**S**
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1

0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Q.24 Simplify the following Boolean functions using four-variable maps.

(i) F (W, X, Y, Z) = ∑ (1, 4, 5, 6, 12, 14, 15)

(ii) F (A, B, C, D) = ∑ (0, 1, 2, 4, 5, 7, 11, 15)

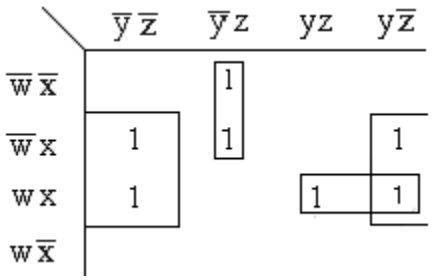
(iii) F (W, X, Y, Z) = ∑ (2, 3, 10, 11, 12, 13, 14, 15)

(iv) F (A, B, C, D) = ∑ (0, 2, 4, 5, 6, 7, 8, 10, 13, 15)

Ans:

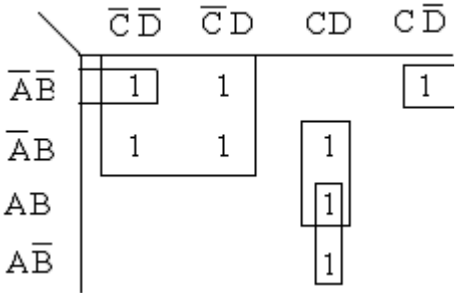
(i) F (W, X, Y, Z) = ∑(1, 4, 5, 6, 12, 14, 15)

$F = x\bar{z} + \bar{w}\bar{y}z + wxy$

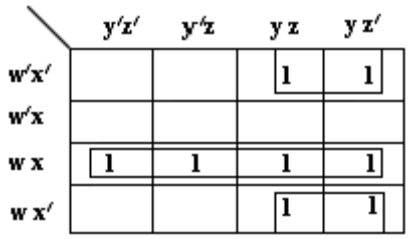


(ii) F (A, B, C, D) = ∑(0, 1, 2, 4, 5, 7, 11, 15)

$F = \bar{A}B\bar{C}D + \bar{A}BCD + A\bar{B}CD + ABCD$
$\bar{A}\bar{C} + ACD + BCD$
$\bar{A}\bar{B}\bar{D}$

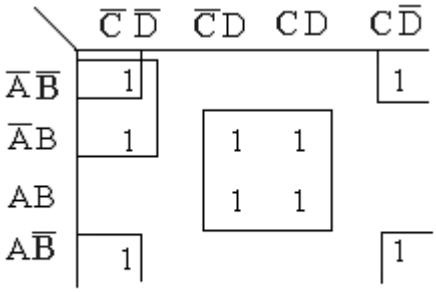


(iii) F (W, X, Y, Z) = ∑ (2, 3, 10, 11, 12, 13, 14, 15)



$F = wx + x'y$

(iv) $F(A, B, C, D) = \sum (0, 2, 4, 5, 6, 7, 8, 10, 13, 15)$



$F = BD + \bar{A}\bar{B} + \bar{B}\bar{D}$

Q.25 What is an instruction? With example explain three, two, one, zero address instructions.

Ans:

Instruction:-

A computer has a variety of instruction code format. It is the function of the control unit within the CPU to interpret each instruction code and provide the necessary control function needed to process the instruction.

Explanation:-

The format of an instruction is appear in memory words or in a control register. The bits of the instruction are divided into groups called fields. The most common fields found in instruction formats are:

1. An operation code field that specifies the operation to be performed.
2. An address field that designates a memory address or a processor register.
3. A mode field that specifies the way the operand or the effective address is determined.

Most computer fall into one of three types of CPU organizations:

1. Single accumulator organization.
2. General register organization.
3. Stack organization.

Accumulator-type organization is the basic computer presented. All operations are performed with an implied accumulator register. The instruction format in this type of computer uses one address field.

ADD X

The instruction format in general register type organization type of computer needs three register address fields.

ADD R1, R2, R3

In the stack-organized of CPU was presented. Would have PUSH and POP instructions.

PUSH x

Three-Address Instructions

Computers with three-address instruction formats can use each address field to specify either a processor register or a memory operand. The program in assembly language that evaluates X = evaluates X = (A+B) * (C+D) is shown below, together with comments that explain the register transfer operation of each instruction.

ADD R1, A, B	R1 ← M[A] + M[B]
ADD R2, C, D	R2 ← M[C] + M[D]
MUL X, R1, R2	M[X] ← R1 * R2.

It is assumed that the computer has two processor registers, *R*1and *R*2. The symbol *M*[A] denotes the operand at memory address symbolized by *A*.

Two-Address Instructions

Two-address instructions are the most common in commercial computers. Here again each address field can specify either a processor register or a memory word. The program to evaluate X = (A +B) * (C + D) is as follows:

MOV R1, A	R1 ← M[A]
ADD R1, B	R1 ← R1 + M[B]
MOV R2, C	R2 ← M[C]
ADD R2, D	R2 ← R2 + M[D]
MUL R1, R2	R1 ← R1 * R2
MOV X, R1	M[X] ← R1

The MOV instruction moves or transfers the operands to and form memory and processor registers.

One-Address Instructions

One-address instructions use an implied accumulator (*AC*) register for all data manipulation. For multiplication and division there is a need for a second register. However, here we will neglect the second register and assume that the *AC* contains the result of all operations. The program to evaluate X = (A + B) * (C + D) is

LOAD	A	AC ← M[A]
ADD	B	AC ← A[C] + M[B]
STORE	T	M[T] ← AC
LOAD	C	AC ← M[C]
ADD	D	AC ← AC+M[D]

MUL	T	AC ← AC * M[T]
STORE	X	M[X] ← AC

Zero-Addressing Instructions

A stack-organized computer does not use an address field for the instructions ADD and MUL. The PUSH and POP instructions, need an address field to specify the operand that communicates with the stack.

PUSH	A	TOS ← A
PUSH	B	TOS ← B
ADD		TOS ← (A + B)
PUSH	C	TOS ← C
PUSH	D	TOS ← D
ADD		TOS ← (C + D)
MUL		TOS ← (C + D) * (A + B)
POP	X	M[X] ← TOS

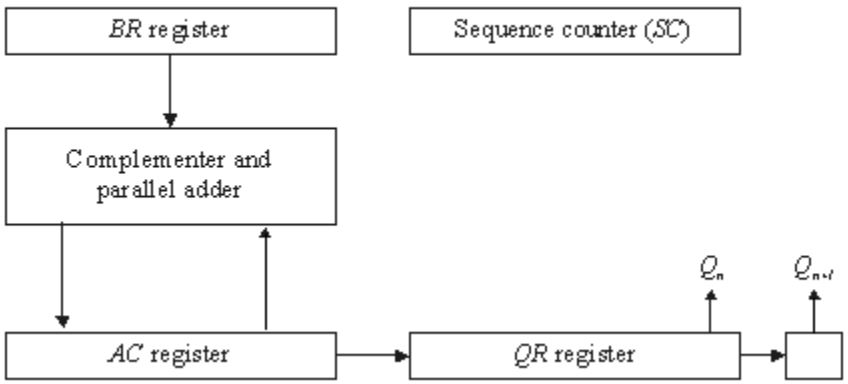
Q.26 Explain Booths algorithm for multiplying two signed binary numbers.

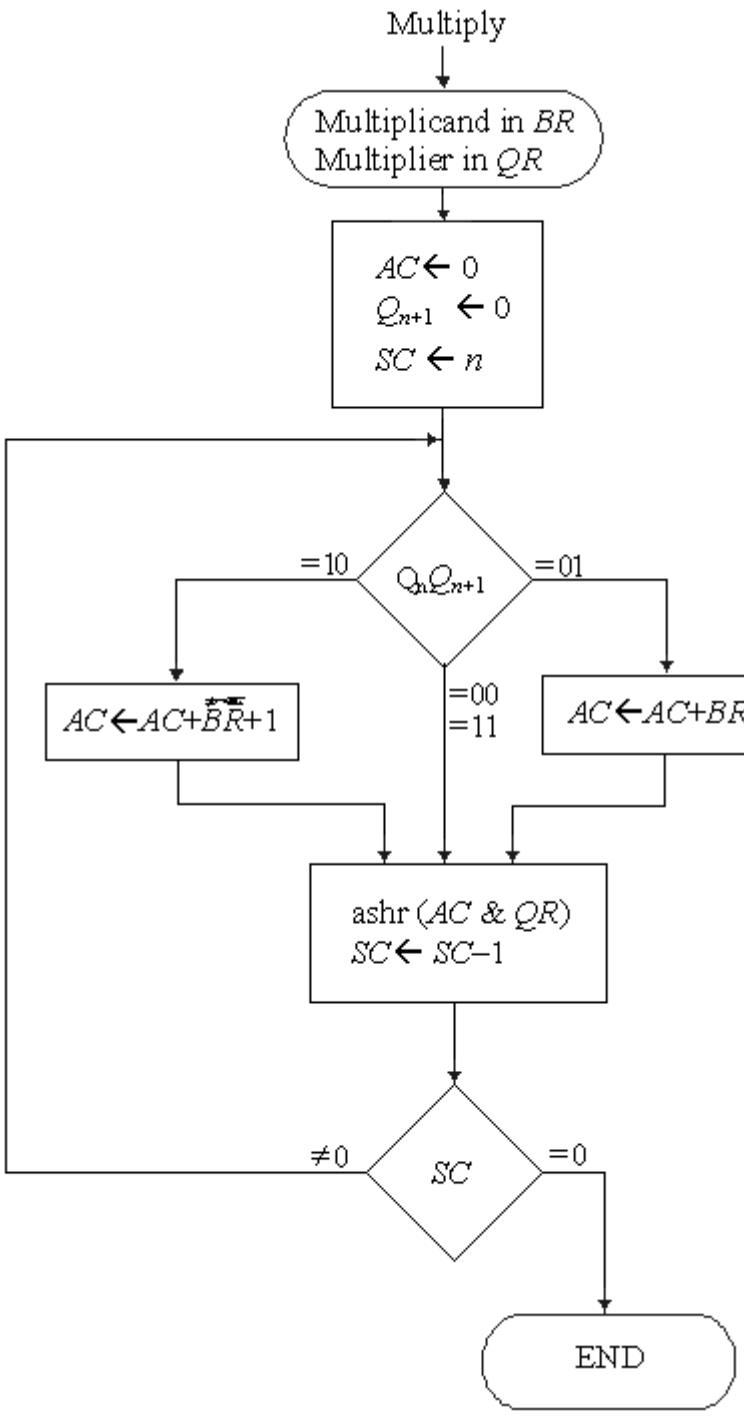
Ans:

Explanation:-

The hardware implementation of Booth algorithm requires the register configuration shown. The sign bits are not separted from the rest of the registers. Registers *A, B,* and *Q,* are named as *AC*, *BR,* and *QR,* respectively. Q_n designates the least significant bit of the multiplier in register QR. An extra flip-flop Q_{n+1} is appended to *QR* to facilitate a double bit inspection of the multiplier. The flowchart for Booth algorithm is shown.

Hardware for Booth algorithm





Example

Q_nQ_{n+1}	$\underline{BR}$ = 10111 $BR + 1$ = 01001	AC	QR	Q_{n+1}	SC
	Initial	00000	10011	0	101
1 0	Subtract BR	$\underline{01001}$ 01001			
	ashr	00100	110001	1	100
1 1	ashr	00010	01100	1	011

0 1	Add *BR*	10111			
		11001			
	ashr	11100	10110	0	010
0 0	ashr	11110	01011	0	001
1 0	Subtract *BR*	01001			
		00111			
	ashr	00011	10101	1	000

Q.27 A Computer uses a memory unit with 256 K words of 32 bits each. A binary instruction code is stored in one word of memory. The instruction has four parts: an indirect bit, an operation code, a register code part to specify one of 64 registers, and an address part.

(i) How many bits are there in the operation code, the register code part, and the address part?

(ii) Draw the instruction word format and indicate the number of bits in each part.

(iii) How many bits are there in the data and address inputs of the memory?

$256\ K = 2^8 \times 2^{10} = 2^{18}$

$64 = 2^6$

Ans:

(i) Memory unit = 256 K words = 2^{18} words

Hence, 18 bits is required to address one word of memory.

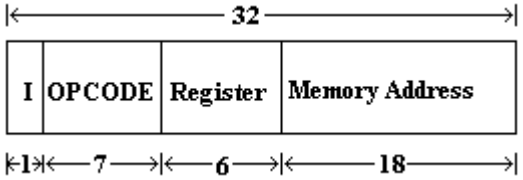
There are 64 registers i.e. 2^6 register

Hence, 6 bits are required to address one register.

1 bit is required for indirect bit.

Hence, 32 – (18+6+1) = 7 bits are required for operational code.

(ii)



(iii) Data : 32 bits;
address = 18 bits

Q.28 What is a micro-operation of list and explain the four categories of the most common micro-operations.

Ans:

A micro-operation is an elementary operation performed on the information stored in one or more registers. The result of the operation may replace the previous binary information of a register or may be transferred to another register.

Common micro-operations

1) Arithmetic Microoperation
2) Logic Microoperation
3) Shift Microoperaztion
4) Arithmetic Logic Shift Unit

Explanation

1) Arithmetic microoperation - The basic arithmetic micro-operations are addition, subtraction, increment, decrement and shift. The arithmetic microoperation defined by the statement.

$$R_3 \leftarrow R_1 + R_2$$

2) Logic Micro-operation - Logic microoperation specify binary operations for strings of bits stored in registers. These operations consider each bit of the register separately and treat them as binary variables.

Example:- P : $R_1 \leftarrow R_1 \ R_2$

3) Shift Micro-operation - Shift microoperations are used for serial transfer of data. They are also used in conjunction with arithmetic, logic and other data - processing operations.

Example :- $R_1 \leftarrow \text{Shl } R_1$

$R_1 \leftarrow \text{Shl } R_1$

4) Arithmetic logic shift unit - To perform a microoperation the contents of specified registers are placed in the inputs of the common ALU. The ALU performs an operation and the result of the operation is then transferred to a destination register.

Q.29 The following transfer statements specify a memory explain the memory operation in each case.

(i) $R_2 \leftarrow M[AR]$ (ii) $M[AR] \leftarrow R_3$ (iii) $R_5 \leftarrow M[R_5]$

Ans:

i) Read memory word specified by the address in AR into register R_2.

ii) Write content of register R3 into the memory word specified by the address in AR.

iii) Read memory word specified by the address in R5 and Transfer content to R_5 (over writing the previous value).

Q.30 Discuss Direct Memory Access is detail.

Ans:

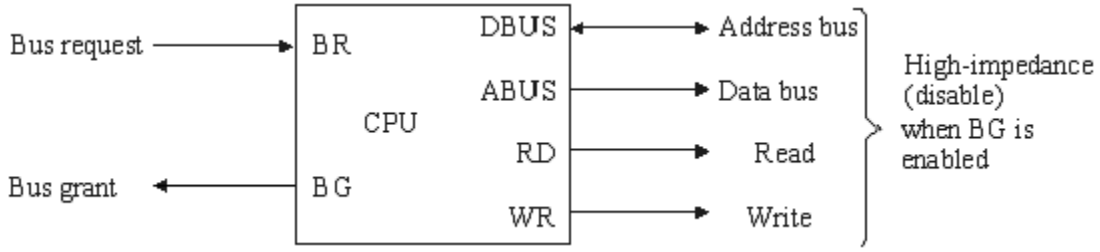
The transfer of data between a fast storage device such as magnetic disk and memory is often limited by the speed of the CPU. Removing the CPU from the path and letting the peripheral device manage the memory buses directly would improve the speed of

transfer. This transfer technique is called direct memory access (DMA). During DMA transfer, the CPU is idle and has no control of the memory buses.

Figure below shows two control signals in the CPU that facilitate the DMA transfer. The bus request (BR) input is used by the DMA controller to request the CPU to relinquish control of the buses.

The CPU activates the bus grant (BG) output to inform the external DMA that the buses are in the high-impedance state. The DMA that originated the bus request can now take control of the buses to conduct memory transfers without processor intervention. When the DMA terminates the transfer, it disables the bus request line. The CPU disables the bus grant, takes control of the buses, and returns to its normal operation.
When the DMA takes control of the bus system, it communicates directly with the memory.

Figure shows the block diagram of a typical DMA controller. The unit communicates with the CPU via the data bus and control lines. The registers in the DMA are selected by the CPU through the



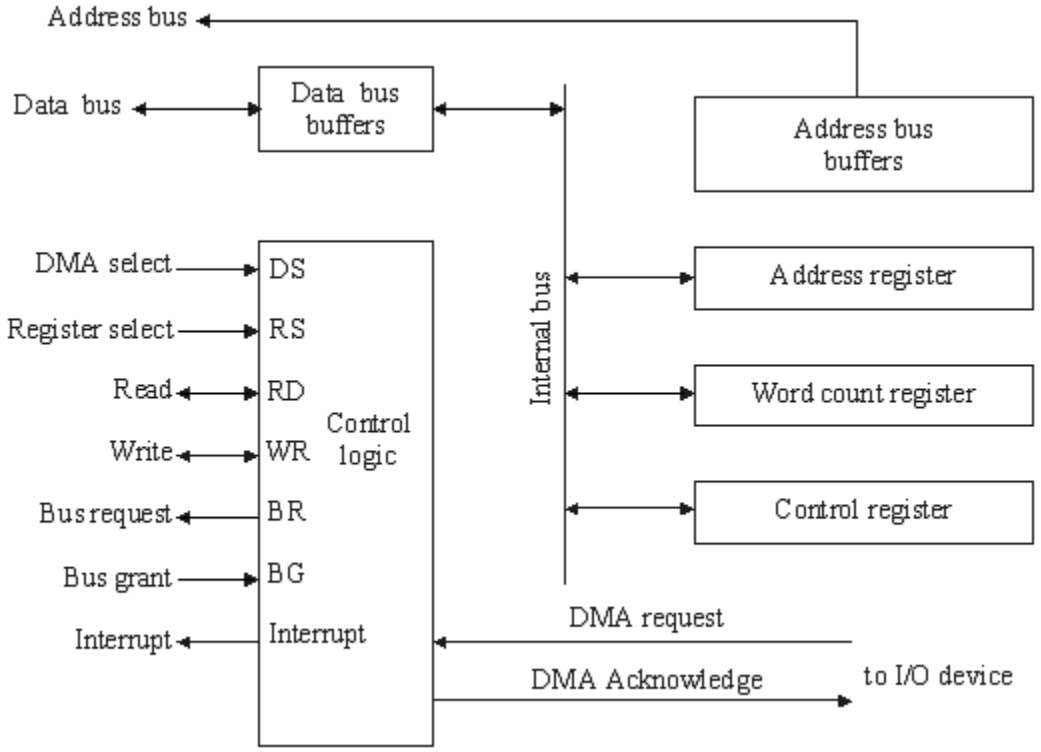
address bus by enabling the *DS* (DMA select) and *RS* (register select) inputs. The *RD* (read) and *WR* (write) inputs are bidirectional. When the *BG* (bus grant) input is 0, the CPU can communicate with the DMA registers through the data bus to read from or write to the DMA registers. When *BG* = 1, the CPU has relinquished the buses and the DMA can communicate directly with the memory by specifying an address in the address bus and activating the *RD* or *WR* control.

The DMA controller has three registers: an address register, a word count register, and a control register.

DMA Transfer:-

Explanation:- The CPU communicates with the DMA through the address and data buses as with any interface unit. The DMA has its own address, which activates the *DS* and *RS* lines. The CPU initializes the DMA through the data bus. Once the DMA

receives the start control command, it can start the transfer between the peripheral device and the memory.



When the peripheral device sends a *DMA* request, the DMA controller activates the *BR* line, informing the CPU to relinquish the buses. The CPU responds with its *BG* line, informing the DMA that its buses are disabled. The DMA then puts the current value of its address register into the address bus, initiates the *RD* or *WR* signal, and sends a DMA acknowledge to the peripheral device. Note that the *RD* and *WR* lines in the DMA controller are bidirectional. The direction of transfer depends on the status of the *BG* line. When *BG* = 0, the *RD* and *WR* are input lines allowing the CPU to communicate with the internal DMA registers. When *BG* = 1, the *RD* and *WR* are output lines from the DMA controller to the random-access memory to specify the read or write operation for the data.

When the peripheral device receives a DMA acknowledge, it puts a word in the data bus (for write) or receives a word from the data bus (for read). Thus the DMA controls the read or write operations and supplies the address for the memory. The peripheral unit can then communicate with memory through the data bus for direct transfer between the two units while the CPU is momentarily disabled.

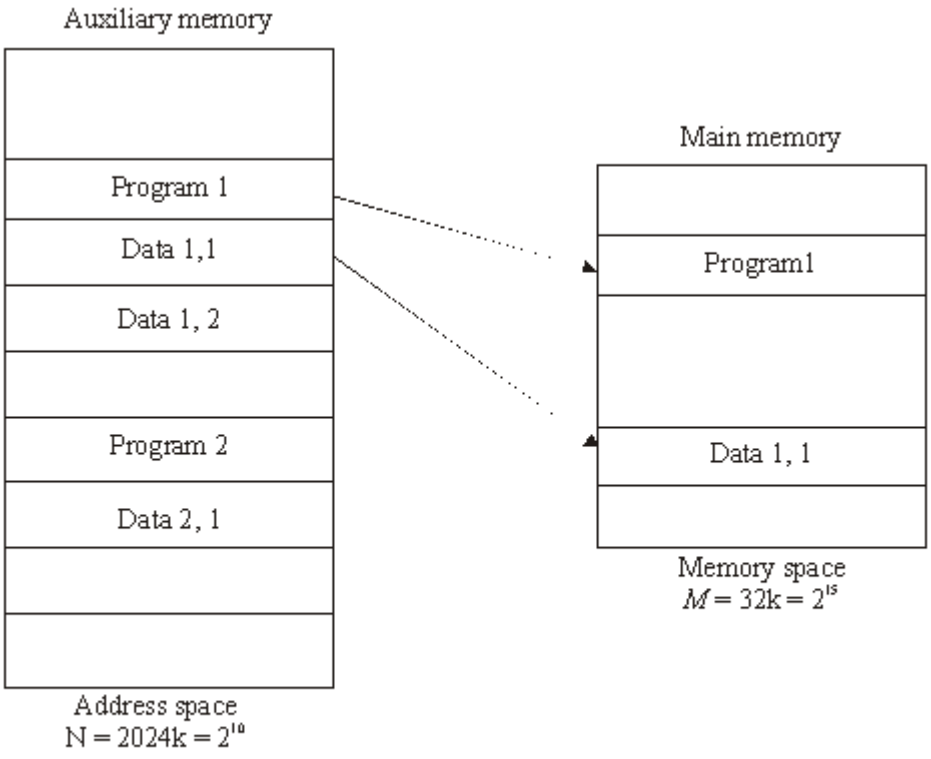
Q.31 Explain the concept of virtual memory with the help of diagram. Explain how virtual address in mapped to actual physical address.

Ans:

Virtual Memory:-

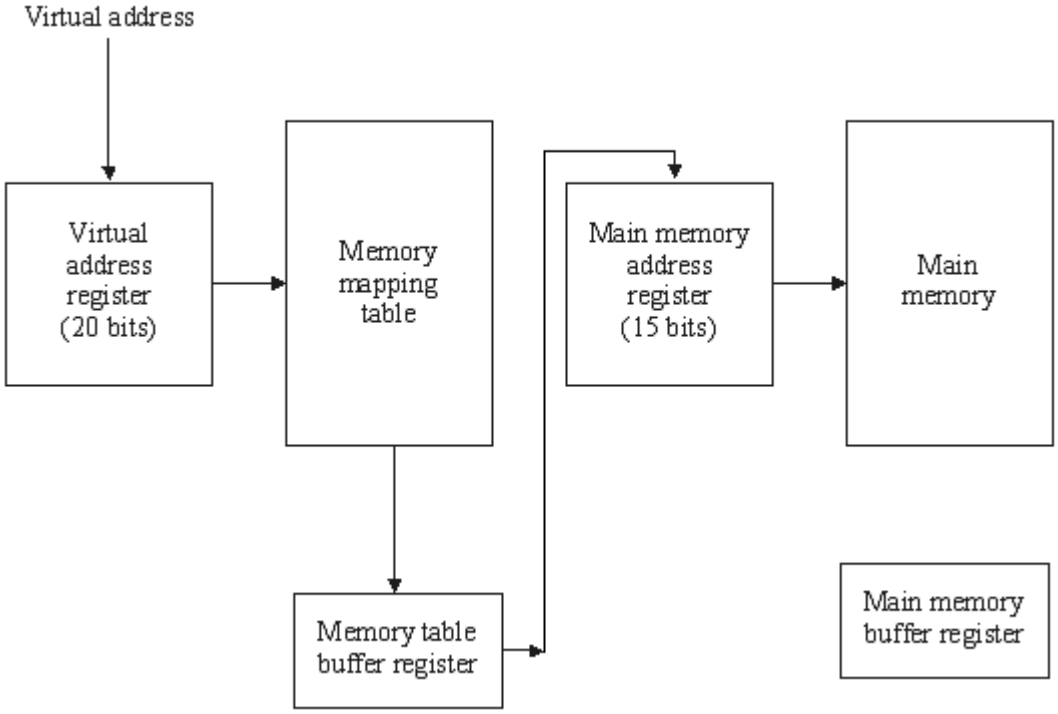
Virtual memory is a concept used in some large computer systems that permit the user to construct programs as though a large memory space were available, equal to the totality of auxiliary memory. Virtual memory is used to give programmers the illusion that they have a very large memory at their disposal, even though the computer actually has a relatively small main memory. A virtual memory system provides a mechanism for translating program generated address into correct main memory locations.

In a virtual memory system, programmers are told that they have the total address space at their disposal. Moreover, the address field of the instruction code has a sufficient number of bits to specify all virtual addresses. In our example, the address field of an instruction code will consist of 20 bits but physical memory addresses must be specified with only 15 bits. Thus CPU will



Relation between address and memory space in a virtual memory system.

reference instructions and data with a 20-bit address, but the information at this address must be taken from physical memory because access to auxiliary storage for individual words will be prohibitively long. To map a virtual address of 20 bits to a physical address of 15 bits we requires a table shown in figure below. The mapping is a dynamic operation, which means that every address is translated immediately as a word is referenced by CPU.



Q.32 Differentiate between:

(i) Isolated I/O and memory-mapped I/O

(ii) Source Initiated and Destination Initiated transfer using handshaking.

Ans:

(i) Isolated I/O and memory-mapped I/O.

Isolated I/O	Memory-Mapped I/O
(i) I/O transfer is made through separate read and write line.	(i) Memory transfer is made through separate read and write lines.
(ii) The I/O read and write control lines are enabled during the I/O are enabled during a memory transfer.	(ii) The memory read and memory write control lines are enabled during a memory transfer.
(iii) The isolated I/O configuration the CPU has distinct input and output instructions and each instruction is associated with the address of an interface register.	(iii) Computer with memory mapped I/O can use memory type instructions to access I/O data.

(iv)	The isolated I/O method isolates memory and I/O address so that memory address values are not affected by interface address assignment.	(iv)	In a memory mapped I/O organization theme are no specific input or output instructions.

(ii) Source Initiated and Destination Initiated transfer using handshaking.

Refer Page 395, 396, 397 from moris mano (3rd Edition)

Q. 33 Differentiate between hardwired control and micro programmed control. Draw the block diagram of a basic hardwired control organization with two decoders, a sequence counter and a number of control logic gates?

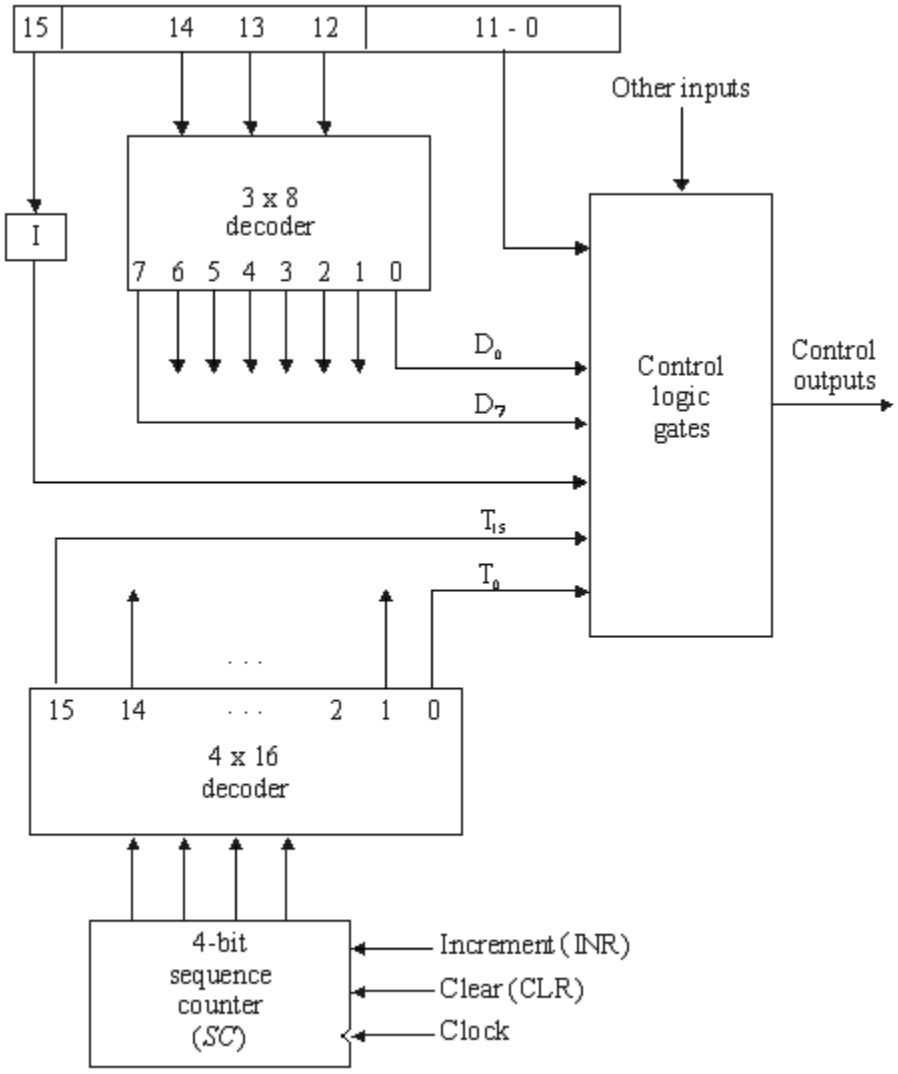
Ans:

The block diagram of the control unit is shown. It consists of two decoders, a sequence counter, and a number of control logic gates. An instruction read from memory is placed in the instruction register (IR). The position of this register in the common bus system is shown in figure. The instruction register is shown, where it is divided into three parts: the I bit, the operation code, and bit 0 through 11. The operation code in bits 12 through 14 are decoded with a 3 x 8 decoder. The eight outputs of the decoder are designated by the symbols D_0 through D_7. Bit 15 of the instruction is transferred to a flip-flop designated by the symbol I. Bits 0 through 11 are applied to the control logic gates. The 4-bit sequence counter can count in binary from 0 through 15. The outputs of the counter are decoded into 16 timing signals T_0 through T_{15}.

The sequence counter SC can be incremented or cleared synchronously, the timing diagram shows the time relationship of the control signals.

	Hardwired Control		**Microprogrammed control**
(i)	In hardwired organization, the control logic is implemented with gates, flip-flops, decoders and other digital circuits.	(i)	In microprogrammed organization, the control information is stored in a control memory. The control memory is programmed to initiate the required sequence of microoperations.

(ii) A hardwired control requires changes in the wired among the various components if the design has to be modified.	(ii) In microprogrammed control any required changes or modification can be done by updating the microprogram in control memory.



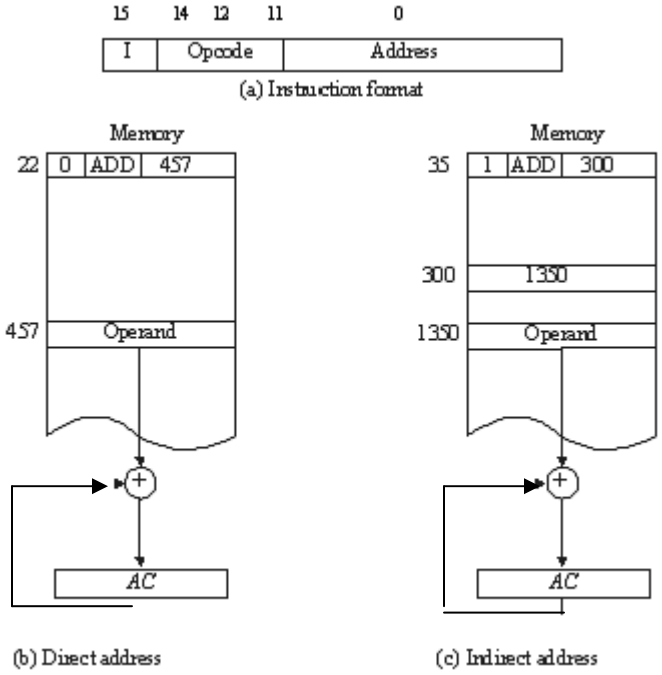
Control unit of basic computer

Q. 34 What is the difference between a direct and an indirect address instruction? How may references to memory are needed for each type of instruction to being an operand into a processor register.

Ans:

Consider the instruction code format shown in figure (a). It consists of a 3-bit operation code, a 12-bit address, and an indirect address mode bit designated by I. The mode bit is 0 for a direct address and 1 for an indirect address. A direct address instruction is shown in figure (b) . It is placed in address 22 in memory. The I bit is 0,

so the instruction is recognized as a direct address instruction. The opcode specifies an ADD instruction, and the address part is the binary equivalent of 457. The control finds the operand in memory at address 457 and adds it to the content of AC. The instruction in address 35 shown in figure (c) has a mode bit I = 1. Therefore, it is recognized as an indirect address instruction. The address part is the binary equivalent of 300. The control goes to address 300 to find the address of the operand. The address of the operand in this case is 1350. The operand found in address 1350 is then added to the content of AC.



A direct address instruction needs two reference to memory:

i) Read instruction

ii) Read operand

An indirect address instruction needs three reference to memory:

i) Read instruction

ii) Read effective address

iii) Read operand.

Q. 35 List any two

(i) Register reference instruction

(ii) Memory reference instruction

(iii) Input-output instruction

Ans.

(i)

Register-Reference Instructions

$D_7I'T_3 = r$ (common to all register-reference instructions)			
IR(i) = B (bit in IR (0-11) that specifies the operation)			
	r:	$SC \leftarrow 0$	Clear SC
CLA	rB_{11}:	$AC \leftarrow 0$	Clear AC
CLE	rB_{10}:	$E \leftarrow 0$	Clear E
CMA	rB_9:	$AC \leftarrow \overline{AC}$	Complement AC
CME	rB_8:	$E \leftarrow \overline{E}$	Complement E
CIR	rB_7:	$AC \leftarrow shr\ AC, AC(15) \leftarrow E, E \leftarrow AC(0)$	Complement E
CIL	rB_6:	$AC \leftarrow shl\ AC, AC(0) \leftarrow E, E \leftarrow AC(15)$	Circulate left
INC	rB_5:	$AC \leftarrow AC+1$	Increment AC
SPA	rB_4:	If (AC(15)=0) then ($PC \leftarrow PC+1$)	Skip if positive
SNA	rB_3:	If (AC(15)=1) then ($PC \leftarrow PC+1$)	Skip if negative
SZA	rB_2:	If (AC=0) then ($PC \leftarrow PC+1$)	Skip if AC zero
SZE	rB_1:	If (E=0) then ($PC \leftarrow PC+1$)	Skip if E zero
HLT	rB_0:	$S \leftarrow 0$ (S is a start-stop flip-flop)	Halt computer

Memory-Reference Instructions

Symbol	Operation decoder	Symbolic description
AND	D_0	$AC \leftarrow AC \wedge M[AR]$
ADD	D_1	$AC \leftarrow AC + M[AR], E \leftarrow Cout$
LDA	D_2	$AC \leftarrow M[AR]$
STA	D_3	$M[AR] \leftarrow AC$
BUN	D_4	$PC \leftarrow AR$
BSA	D_5	$M[AR] \leftarrow PC, PC \leftarrow AR+1$
ISZ	D_6	$M[AR] \leftarrow M[AR]+1$
		If $M[AR]+1=0$ then $PC \leftarrow PC+1$

Input-Output Instructions

D_7IT_3 = p (common to all input-output instructions)			
IR(i) = B_i [bit in IR (6-11) that specifies the instruction]			
	p:	SC ← 0	Clear SC
INP	pB_{11}:	AC(0-7) ← INPR, FGI ← 0	Input character
OUT	pB_{10}:	OUTAR ← AC(0-7), FGO ← 0	Output character
SKI	pB_9:	IF(FGI=1) then (PC ← PC+1)	Skip on input flag
SKO	pB_8:	IF(FGO=1) then (PC ← PC+1)	Skip on Output flag
ION	pB_7:	IEN ← 1	Interrupt enable on
IOF	pB_6:	IEN ← 0	Interrupt enable off

Q.36. What is the function of interrupt facility in a multiprogram environment? Explain how a source routine is initiated for the input or output transfer. Lists the tasks which this service routine is supported to perform. **(8)**

Ans: Refer Program Interrupt from page 281,282 from Morris Mano, (3rd Edition)

Q. 37 Write an assembly language program to:

(i) Input a character & store it in memory

(ii) Add two numbers

Ans:

(i) Input a character & store it in memory

```
CIF,    SKI         /check input flag
        BUN CIF     /Flag=0, branch to check again
        INP         /Flag=1, input character
    STA CHR         /Store character
        HLT
CHR,    M           /Store character in memory
```

(ii) Add two numbers

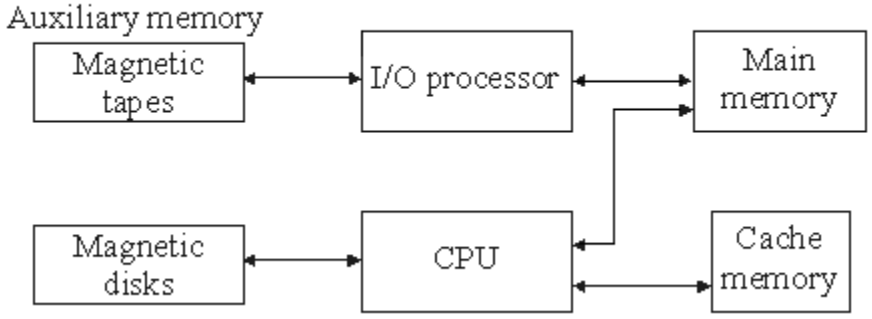
```
        ORG  0          /Origin of program is location O
        LDA  A          /Load operand from location A
        ADD  B          /Add operand from location B
        STA  C          /Store sum in location C
        HLT             /Halt Computer
A,      DEC  83         /Decimal operand
B,      DEC-23          /Decimal operand
C,      DEC  0          /Sum stored in location C
        END             /End of symbolic program
```

Q.38 Explain what is meant by a cache memory. What general principles are used to make effective use of cache memory?

Ans:

A special very-high-speed memory called a cache is sometimes used to increase the speed of processing by making current programs and data available to the CPU at a rapid rate. The cache memory is employed in computer systems to compensate for the speed differential between main memory access time and processor logic. CPU logic is usually faster than main memory access time, with the result that processing speed is limited primarily by the speed of main memory.

Memory hierarchy in a computer system



Cache Memory

The references to memory at any given interval of time tend to be confined within a few localized areas in memory. This phenomenon is known as the property of locality of reference.

If the active portions of the program and data are placed in a fast small memory, the average memory access time can be reduced, thus reducing the total execution time of the program. It is placed between the CPU and main memory as illustrated. The cache

memory access time is less than the access time of main memory by a factor of 5 to 10. The cache is the fastest component in the memory hierarchy and approaches the speed of CPU components.

The basic operation of the cache is as follows. When the CPU needs to access memory, the cache is examined. If the word is found in the cache, it is read from the fast memory. If the word addressed by the CPU is not found in the cache, the main memory is accessed to read the word.

The performance of cache memory is frequently measured in terms of a quantity called hit ratio. When the CPU refers to memory and finds the word in cache, it is said to produce a hit. If the word is not found in cache, it is in main memory and it counts as a miss. The ratio of the number of hits divided by the total CPU references to memory is the hit ratio.

The average memory access time of a computer system can be improved considerably by use of a cache. If the hit ratio is high enough so that most of the time the CPU accesses the cache instead of main memory, the average access time is closer to the access time of the fast cache memory.

The basic characteristic of cache memory is its fast access time. Therefore, very little or no time must be wasted when searching for words in the cache. The transformation of data from main memory to cache memory is referred to as a mapping process. Three types of mapping procedures are of practical interest when considering the organization of cache memory.

1. Associative mapping
2. Direct mapping
3. Set-associative mapping.

Q. 39 What is associative memory? Explain with the help of a block diagram. Also mention the situation in which associative memory can be effective utilized.

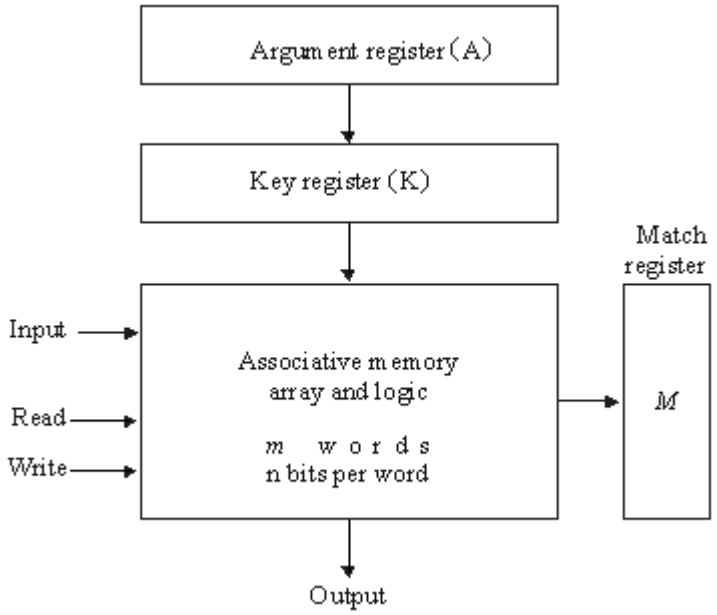
Ans:

Associative memory:-

The time required to find an item stored in memory can be reduced considerably if stored data can be identified for access by the content of the data itself rather than by an address. A memory unit accessed by content is called an associative memory or content addressable memory (CAM). This type of memory is accessed simultaneously and in parallel on the basis of data content rather than by specific address or location.

Hardwared Organization

The block diagram of an associative memory is shown. It consists of a memory array and logic for m words with *n* bits per word. The argument register A and key register K each have n bits, one for each bit of a word. Each word in memory is compared in parallel with the content of the argument register. The words that match the bits of the argument register set a corresponding bit in the match register. After the matching process, those bits in the match register that have been set indicate the fact that their corresponding, words have been match register.



Block diagram of associative memory

The key register provides a mask for choosing a particular field or key in the argument word. The entire argument is compared with each memory word if the key register contains all 1's. Otherwise, only those bits in the argument that have 1's in their corresponding position of the key register are compared.

A numerical example, suppose that the argument register A and the key register K have the bit configuration shown below. Only the three leftmost bits of A are compared with memory words because K has 1's these positons.

Word 2 matches the unmasked argument field because the three leftmost bits of the argument and the word are equal.

A	101 111100	
K	111 000000	
Word1	100 111100	no match
Word2	101 000001	match

Q. 40 With neat flow chart, explain binary division process.

Ans:

Hardware Algorithm:-

The flowchart is shown in the figure. The dividend is in *A* and *Q* and the divisor in *B*. The sign of the result is transferred into Q_s to be part of the quotient. A constant is set into the sequence counter *SC* to specify the number of bits in the quotient.

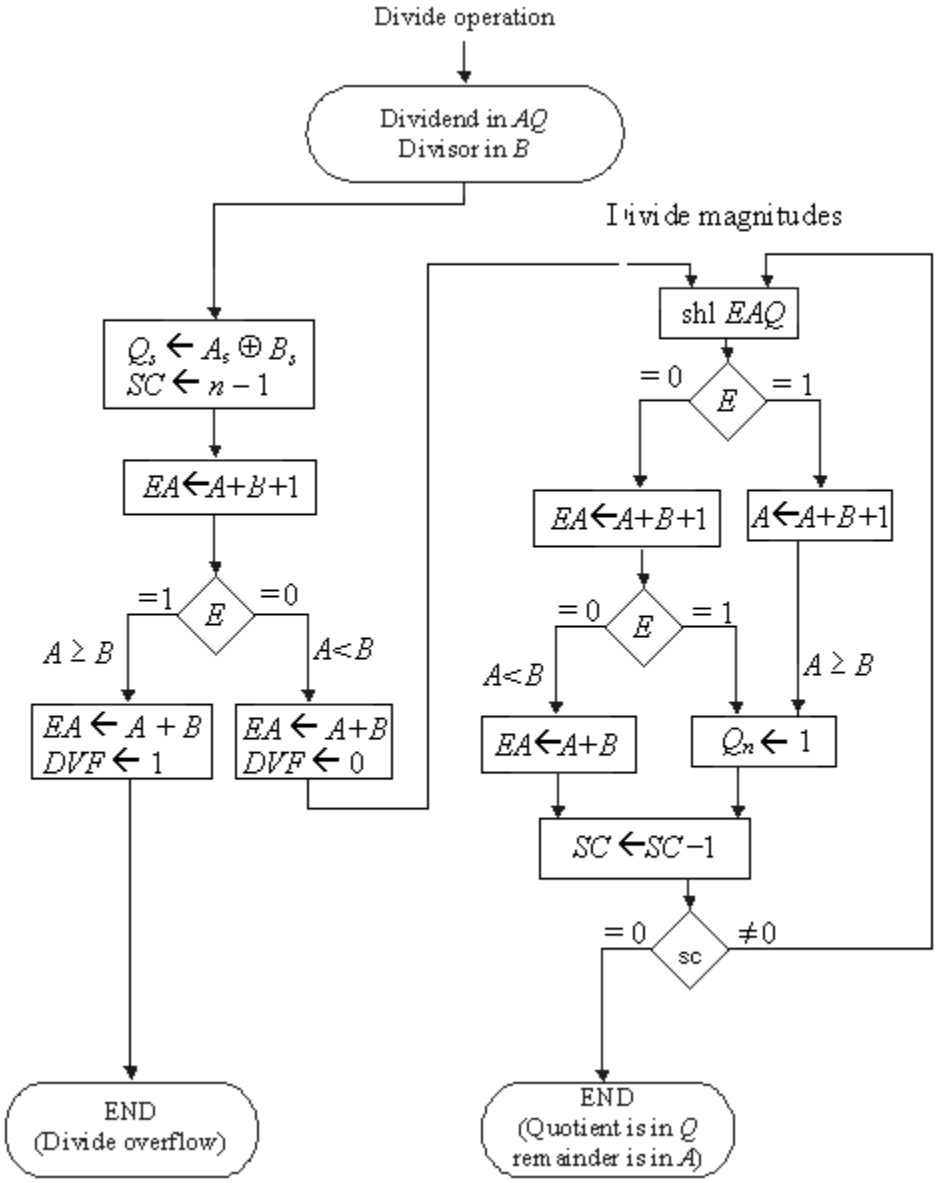
A divide-overflow condition is tested by subtracting the division in B from half of the bits of the dividend stored in A. If A $\geq$ B, the divide-overflow flip-flop *DVF* is set and

the operation is terminated prematurely. If A < B, no divide overflow occurs so the value of the dividend is restored by adding *B* to *A*.

The division of the magnitudes starts by shifting the dividend in AQ to the left with the high-order bit shifted into E. If the bit shifted into E is 1, we know that EA > B because EA consists of a 1 followed by n - 1 bits while B consists of only n - 1 bits.

If the shift-left operation inserts a 0 into *E,* the divisor is subtracted by adding its 2's complement value and the carry is transferred into *E.* If E = 1, it signifies that $A \geq B$; therefore, Q_n is set to 1. If $E = 0$, it signifies that $A < B$ and the original number is restored by adding *B* to *A*.

Flowchart for divide operation

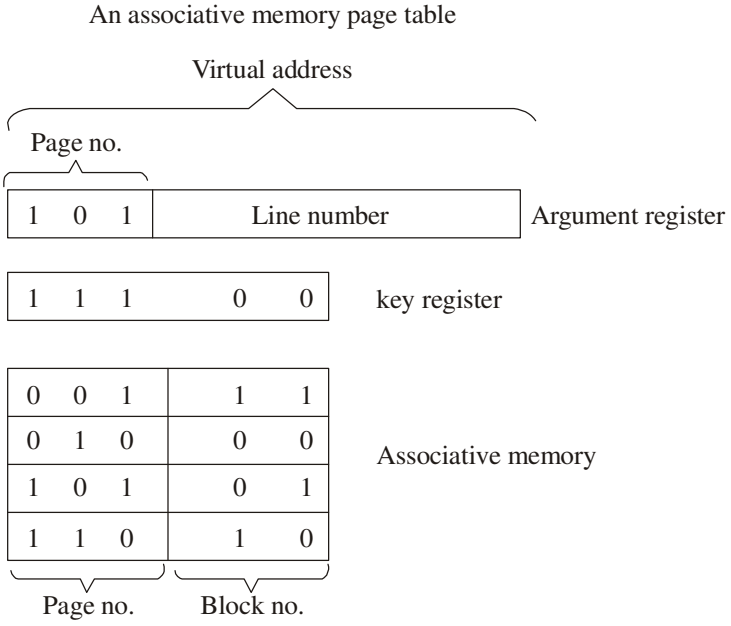


Q. 41 Describe through diagram how matched word can be read out from on associative memory.

Ans:
A system with *n* pages and *m* blocks would require a memory-page table of *n* locations of which up to *m* blocks will be marked with block numbers and all others will be empty.

A more efficient way to organize the page table would be to construct it with a number of words equal to the number of blocks in main memory. In this way the size of the memory is reduced and each location is fully utilized. This method can be implemented by means of an associative memory with each word in memory containing a page number together with its corresponding block number. The page field in each word is compared with the page number in the virtual address. If a match occurs, the word is read from memory and its corresponding block number is extracted.

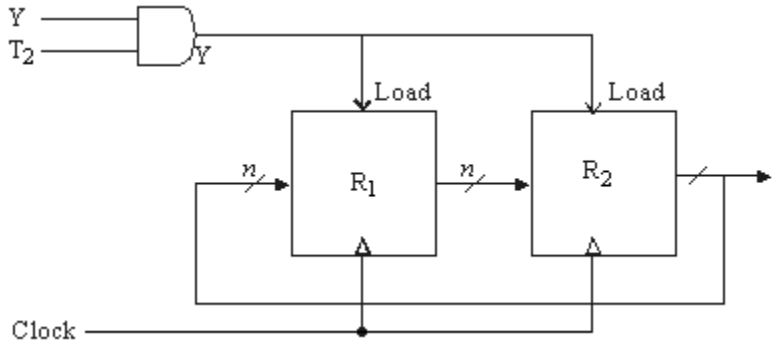
Each entry in the associative memory array consists of two fields. The first three bits specify a field for storing the page number. The last two bits constitute a field for storing the block number. The virtual address is placed in the argument register. The page number bits in the argument register are compared with all page numbers in the page field of the associative memory. If the page number is found, the 5-bit word is read out from memory. The corresponding block number, being in the same word, is transferred to the main memory address register. If no match occurs, a call to the operating system is generated to bring the required page from auxiliary memory.



Q. 42 Give the hardware implementation of flowing

$$YT_2 : R_2 \leftarrow R_1,\ R_1 \leftarrow R_2$$

Ans:

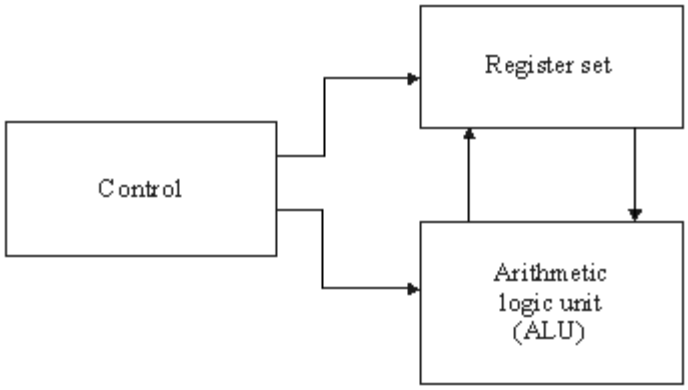


Q. 43 Briefly discuss the steps followed in designing a CPU.

Ans:

The part of the computer that performs the bulk of data processing operations is called the central processing unit and is referred to as the CPU. The register set stores intermediate data used during the execution of the instructions. The arithmetic logic unit (ALU) performs the required microoperations for executing the instructions. The control unit supervises the transfer of information among the registers and instructs the ALU as to which operation to perform.

The registers communicate with the ALU through buses and explain the operation of the memory stack.



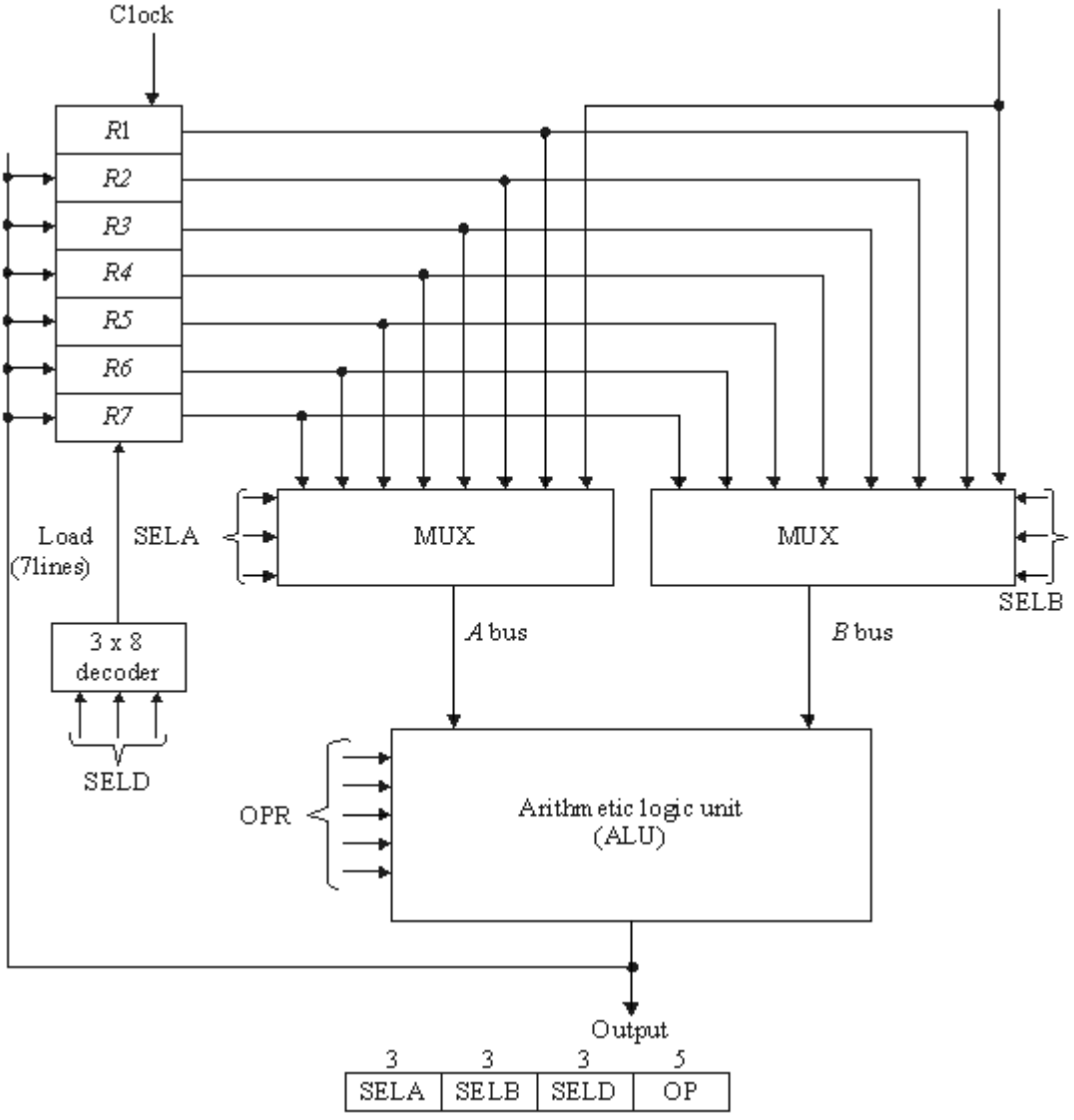
Major components of CPU

A large number of registers are included in the CPU, it is most efficient to connect them through a common bus system. The registers communicate with each other not only for direct data transfers, but also while performing various microoperations. Hence, it is necessary to provide a common unit that can perform all the arithmetic, logic, and shift microoperations in the processor.

A bus organization for seven CPU registers is shown. The output of each register is connected to two multiplexers (MUX) to form the two buses *A* and *B*.

There are 14 binary selection inputs in the unit, and their combined value specifies a *control word*. The 14-bit control word is defined. It consists of four fields. Three fields contain three bits each, and one field has five bits. The three bits of SELA select a source register for the *A* input of the ALU. The three bits of SELB select a register for the *B* input of the ALU. The three bits of SELD select a destination register using the decoder and its seven load outputs. The five bits of OPR select one of the operations in the ALU.

The ALU provides arithmetic and logic operations. In addition, the CPU must provide shift operations. The shifter may be placed in the input of the ALU to provide a preshift capability, or at the output of the ALU to provide postshifting capability. In some cases, the shift operations are included with the ALU.



(c) Control word

Register set with common ALU

Q. 44 What are the types of instructions that are included in an instruction set? Give two examples of each type.

Ans:
Refer Section 8.6 of Morris mano (3rd Edition)

Q. 45 The following program is stored in the memory unit of the basic computer. Show the contents of the AC, PC and IR (in hexa decimal), at the end, after each instruction is executed. All number listed below are in hexadecimal.

Location	Instruction
010	CLA
011	ADD 016
012	BUN 014
013	HLT
014	AND 017
015	BUN 013
016	C1A5
017	93C6

Ans:

		AC	PC	IR
010	CLA	0000	011	7800
017	ADD 016	C1A5	012	1016
012	BUN 014	C1A5	014	4014
013	HLT	8184	014	7001
014	AND 017	8184	015	0017
015	BUN 013	8184	013	4013
016	C1A5			
017	93C6			

$(C1A5)_{16}$	=	1100	0001	1010	0101	AND
$(93C6)_{16}$	=	1001	0011	1100	0110	
		1000	0001	1000	0100	
	=	$(8184)_{16}$				

Q. 46 What do you understand by subroutine call? What steps are carried out when a subroutine call is encountered? State the difference between subroutine call and interrupts.

Ans:

The procedure for branching to a subroutine and returning to the main program is referred to as a subroutine linkage. The BSA instruction performs an operation commonly called subroutine call. The procedure used in the basic computer for subroutine linkage is commonly found in computer with only one processor register.

Subroutine parameters and data linkage.

When a subroutine is called, the main program must transfer the data. The subroutine shifted the number and left is there to be accepted by the main program. It is necessary for the subroutine to have access to data from the calling program and to return results to that program. The accumulator can be used for a single input parameter and a single output parameter. Consider a subroutine that performs the logic OR operation. Two operands must be transferred to the subroutine and the subroutine must return the result of the operation. The accumulator can be used to transfer one operand and to receive the result. The other operand is inserted in the location following the BSA instruction.

The subroutine must increment the return address stored in its first location for each operand that it extracts from the calling program. Moreover, the calling program can reserve one or more locations for the subroutine to return results that are computed. The first location in the subroutine must be incremented for these locations as well, before the return. If there is a large amount of data to be transferred, the data can be placed to a block of storage and the address of the first item in the block is then used as the linking parameter.

A subroutine that moves a block of data starting at address into a block starting with address. The length of the block is 16 words. The first introduction is branch to subroutine MVE. The items are retrieved from their blocks by the use of two pointers. The counter ensures that only 16 items are moved. When the subroutine completes its operation, the data required is in the block starting address 200. The return to the main program is to the HLT instruction.

A subroutine call is a self contained sequence of instructions that performs a given computational task. During the execution of program, a subroutine may be called to perform its function many times at various points in the main program.

Similarity between subroutine and program interrupt. The interrupt procedure is in principle guite similar to a subroutine call except for three variations.

1) The interrupt is usually invited by an internal and external signal rather than execution of an instruction.

2) The address of the interrupt service program is determined by the hardware rather than address field of an instruction.

3) An interrupt procedure usually stores all the information necessary to define the state of CPU rather than storing only the program counter.

Q. 47 With neat diagram, explain the strobe control data transfer method, state its disadvantage.

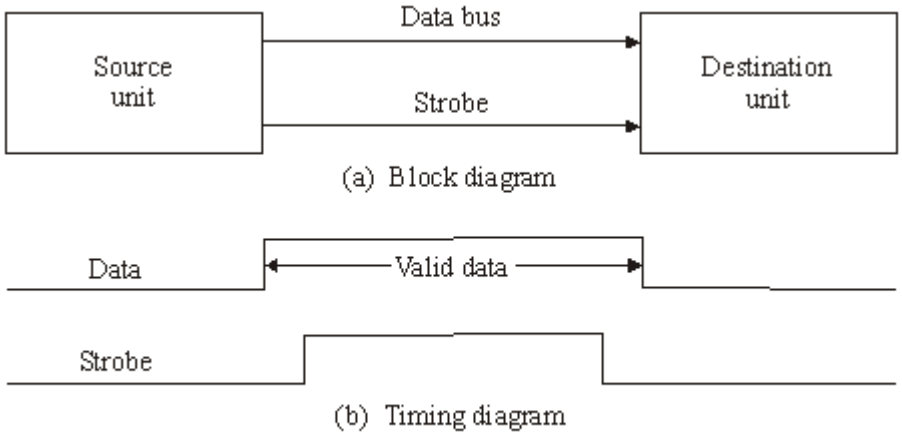
Ans:

Strobe Control:-

The strobe control method of asynchronous data transfer employs a single control line to time each transfer. The strobe may be activated by either the source or the destination unit.

The data bus carries the binary information from source unit to the destination unit. Typically, the bus has multiple lines to transfer an entire byte or word. The strobe is a single line that informs the destination unit when a valid data word is available in the bus.

In the timing diagram the source unit first places the data on the data bus. After a brief delay to ensure that the data settle to a steady value, the source activates the strobe pulse. The information on the data bus and the strobe signal remain in the active state for a sufficient time period to allow the destination unit to receive the data. The source removes the data from the bus a brief period after it disables its strobe pulse, which indicates that the data bus does not contain valid data. New valid data will be available only after the strobe is enabled again.

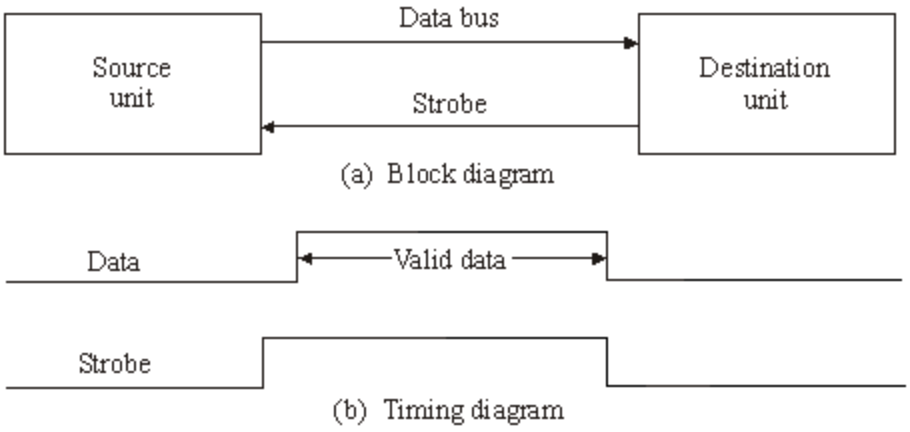


Source-initiated strobe for data transfer

Data transfer initiated by the desintation unit. The destination unit activates the strobe pulse, informing the source to provide the data. The source unit responds by placing the requested binary information on the data bus. The data must be valid and remain in the bus long enough for the destination unit to accept it.

Disadvantages:-

The disadvantage of the strobe method is that the source unit that initiates the transfer has no way of knowing whether the detination unit has actually received the data item that was placed in the bus. Similarly, a destination unit that initiates the transfer has no way of knowing whether the source unit has actually placed the data on the bus.



Destination-initiated strobe for data transfer

Q. 48 A virtual memory system has 6k words of address space and 3k words of memory space. Page references are made by CPU in following sequence:

3, 2, 0, 3, 4, 1, 2, 2, 0

Find out the pages that are available at the end if the replacement algorithm used is

(a) LRU (b) FIFO Assume the page and block size of 1k words.

Ans:

(a) LRU

3	2	0	3	4	1	2	2	0
3	3	3		3	3	2		2
	2	2		4	4	4		0
		0		0	1	1		1

(b) FIFO

3	2	0	3	4	1	2	2	0
3	3	3		4	4	4		0
	2	2		2	1	1		1
		0		0	0	2		2

Q. 49 What is the need of I/O interface?

Ans: Refer Section 11.2 of Morris Mano (3rd Edition)

Input-output interface provides a method for transferring information between internal storage and external I/O devices. Peripherals connected to a computer need special communication links for interfacing them with the central processing unit.

Q. 50 State the advantages of assembly language over machine language.

Ans:

Advantages

1) Assembly language programs are more understandable than machine language program.

2) Debugging is easy than machine language.

3) Assembly language is easy to write the program than machine language.

4) Assembly language uses assembler to generate object code.

5) Assembly language programs takes less computation time and computer's main memory.

Q. 51 State the advantages of memory mapped I/O over I/O mapped I/O.

Ans:

1) Memory mapped I/O use memory type instructions to access I/O data. It allows the computer to use the same instructions for either input-output transfers.

2) The load and store instructions and for reading and writing from memory can be used to input and output data from I/O registers.

3) Memory mapped I/O all instructions that refer to memory are also available for I/O.

Q. 52 Write short notes on any two.

(i) Instruction pipeline.

(ii) DMA based data transfer.

(iii) Shift operation of data in a register.

Ans:

(i) Instruction pipeline:-

An instruction pipeline reads consecutive instructions from memory while previous instructions are being executed in other segments. This causes the instruction fetch and execute phases to overlap and perform simultaneous operations.

The instruction fetch segment can be implemented by means of a first-in, first-out (FIFO) buffer. This is a type of unit that forms a queue rather than a stack. Whenever the execution unit is not using memory, the control increments the program counter and uses its address value to read consecutive instructions from memory. An instruction stream can be placed in a queue, waiting for decoding and processing by the execution segment. The instruction stream queuing mechanism provides an efficient way for reducing the average access time to memory for reading instructions.

The computer needs to process each instruction with the following sequence of steps.

1) Fetch the instruction from memory.
2) Decode the instruction.
3) Calculate the effective address.
4) Fetch the operands from memory.
5) Execute the instruction.
6) Store the result in the proper place.

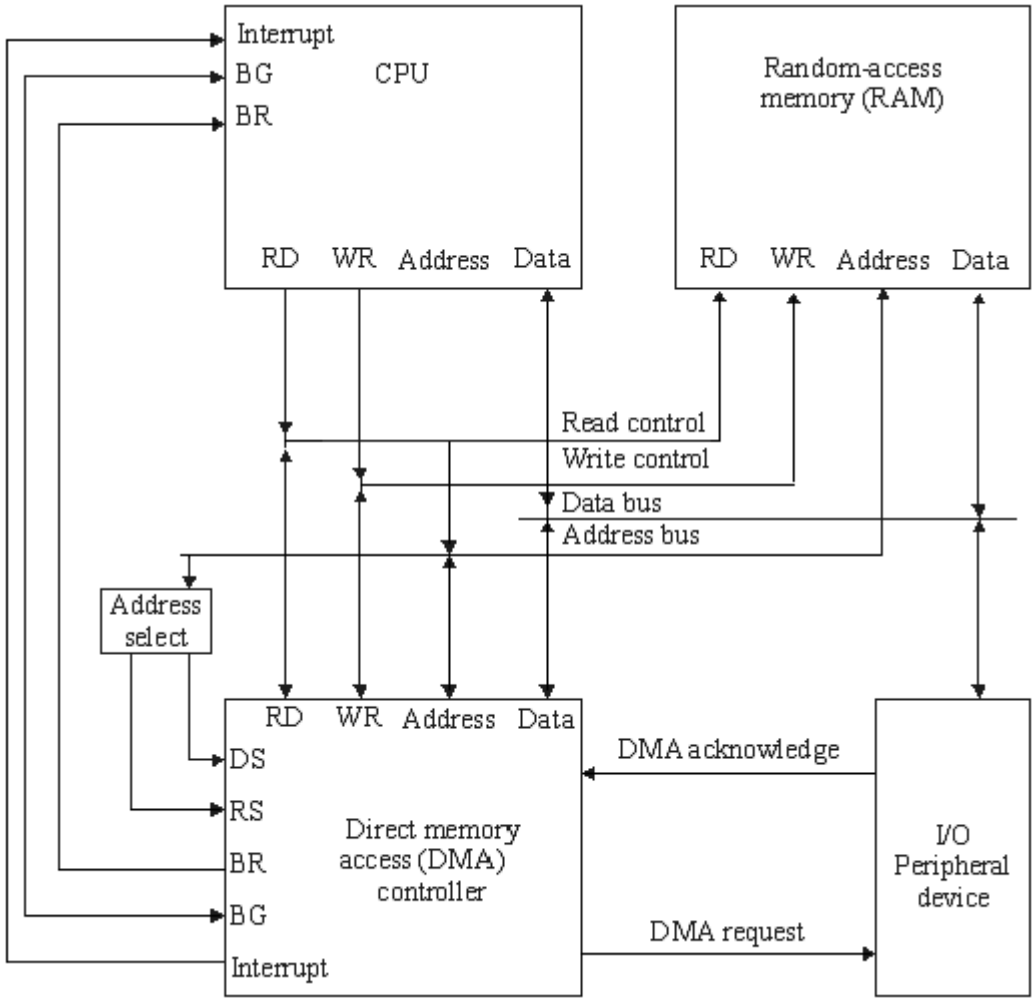
There are certain difficulties that will prevent the instruction pipeline from operating at the maximum rate. Different segments may take different time to operate on the incoming information. Some segments are skipped for certain operations. For example, a register mode instruction does not need an effective address calculation. Two or more segments may require memory access at the same time, causing our segment to wait until another is finished with the memory.

The design of an instruction pipeline will be most efficient if the instruction cycle is divided into segments of equal duration. The time that each step takes to fulfill its function depends on the instruction and the way it is executed.

(ii) DMA based data transfer:-

The transfer of data between a fast storage device such as magnetic disk and memory is often limited by the speed of the CPU. Removing the CPU from the path and letting the peripheral device manage the memory buses directly would improve the speed of transfer. This transfer technique is called direct memory access (DMA). During DMA transfer, the CPU is idle and has no control of the memory buses.

Two control signals in the CPU that facilitate the DMA transfer. The bus request (BR) input is used by the DMA controller to request the CPU to relinquish control of the buses. When this input is active, the CPU terminates the execution of the current instruction. The CPU activates the bus grant (BC) output to inform the external DMA that the buses are in the high-impedance state. The DMA that originated the bus request can now take control of the buses to conduct memory transfers without processor intervention. When the DMA terminates the transfer, it disables the bus request line. The CPU disables the bus grant, takes control



of the buses, and returns to its normal operation.

When the DMA takes control of the bus system, it communicates directly with the memory.

(iii) Shift operation of data in a register:-

The contents of a register can be shifted to the left or the right. There are three types of shifts: logical, circular, and arithmetic.

A logical shift is one that transfers 0 through the serial input. We will adopt the symbols shl and shr for logical shift-left and shift-right microoperations. For example:

$R1 \leftarrow \text{shl}\, R1$

$R2 \leftarrow \text{shr}\, R2$

The circular shift circulates the bits of the register around the two ends without loss of information. This is accomplished by connecting the serial output of the shift

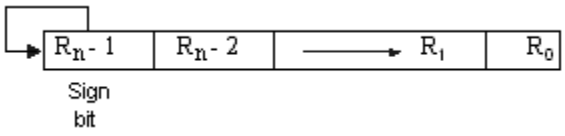
register to its serial input. We will use the symbols cil and cir for the circular shift left and right.

Shift Micro operations

Symbolic designation	Description
R ← shl R	Shift-left register R
R ← shr R	Shift-right register R
R ← cil R	Circular Shift-left register R
R ← cir R	Circular Shift-right register R
R ← ashl R	Arithmetic shift-left R
R ← ashr R	Arithmetic shift-right R

An arithmetic shift is a microoperation that shifts a signed binary number to the left or right. An arithmetic shift-left multiplies a signed binary number by 2. An arithmetic shift-right divides the number by 2. Arithmetic shifts must leave the sign bit unchanged because the sign of the number remains the same when it is multiplied or divided by 2. The leftmost bit in a register holds the sign bit, and the remaining bits hold the number. The sign bit is 0 for positive and 1 for negative. Negative numbers are in 2's complement form.

Arithmetic shift right

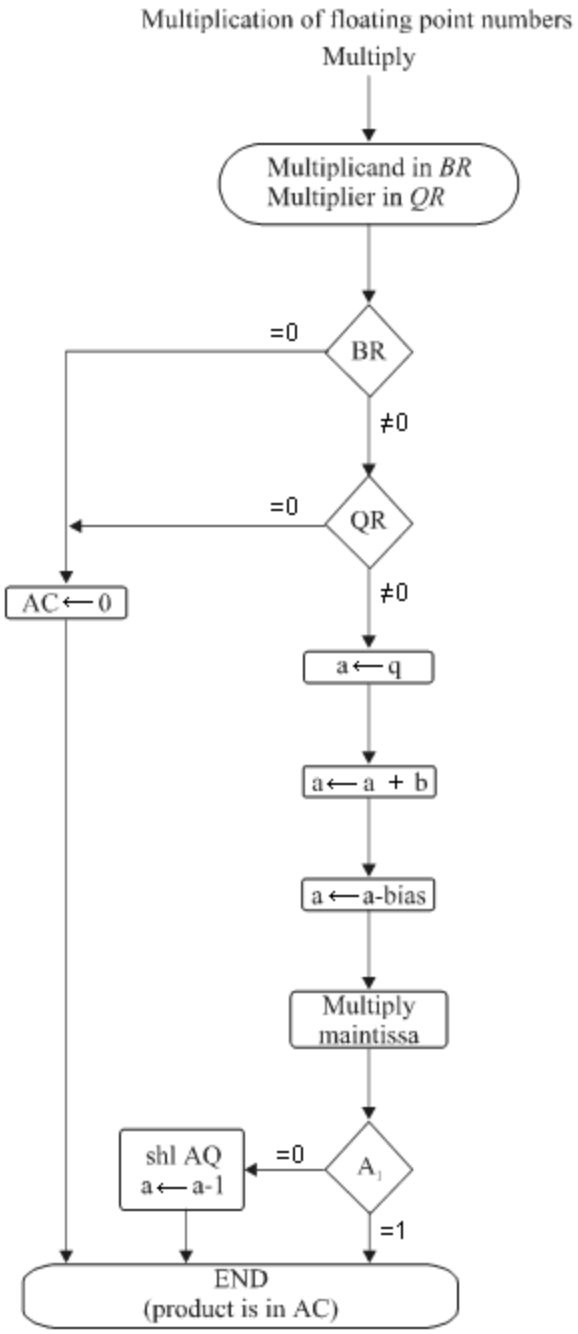


Q. 53 With neat flowchart, explain the process of multiplication of floating point numbers.

Ans:

Multiplication:-

The multiplication of two floating-point numbers requires that we multiply the mantissas and add the exponents.



The multiplication algorithm can be subdivided into four parts:

1) Check for zeros.
2) Add the exponents.
3) Multiply the mantissas.
4) Normalize the product.

Step 2 and 3 can be done simultaneously if separate adders are available for the mantissas and exponents.

The flowchart for floating-point multiplication is shown in figure. The two operands are checked to determine if they contain a zero. If either operand is equal to zero, the product in the AC is set to zero and the operation is terminated. If neither of the operands is equal to zero the process continues with the exponent addition.

The exponent of the multiplier is in *q* and the adder is between exponents a and b. It is necessary to transfer the exponents from *q* to *a*, add the two exponents, and transfer the sum into *a*. Since both exponents are biased by the addition of a constant, the exponent sum will have double this bias. The correct biased exponent for the product is obtained by subtracting the bias number from the sum.

The multiplication of the mantissa is done as in the fixed point case with the product residing in *A* and *Q*. Overflow cannot occur during multiplication, so there is no need to check for it.

The product may have an underflow, so the most significant bit in *A* is checked. If it is a 1, the product is already normalized. If it is a 0, the mantissa in *AQ* is shifted left and the exponent decremented. Note that only one normalization shift is necessary. The multiplier and multiplicand were originally normalized and contained fractions. The smallest normalized operand is 0.1, so the smallest possible product is 0.01. Therefore, only one leading zero may occur.

Although the low-order half of the mantissa is in *Q*, we do not use it for the floating point product. Only the value in the AC is taken as the product.

Q. 54 Divide dividend $(-53)_{10}$ by divisor $(-7)_{10}$ by using binary division algorithm Show all steps.

Ans:

Divisor B = 111011 B+1 = 000101

	E	A	Q	SC
Dividend		001011	000000	6
Shl EAQ	0	010110	000000	
add B+1		000101		
E = O	0	011011		
Set Q_0=1	0	011011	000001	5
Shl FAQ	0	110110	000010	
Add B+1		000101		
	0	111011		
E =);	0	111011		
Set Q_0=1	0	0111011	000011	4
Shl EAQ	1	110110	000110	
Add B+1		000101		
E = 1	1	111011	000111	3
Shl EAQ	1	110110	001110	
Add B+1		000101		
E = 1	1	111011		
leave Q_0=0	1	111011	001110	
Add B		111011		
E = O	0	110110		2
Shl EAQ	1	101100	011100	
Add B+1		000101		
E = 1	1	110001		
Set Q_0= 0	1	110001		
Shl EAQ	1	100011	11010	
Add B+1		000101		
E = 1	1	101000		
leave Q_0=0	1	101000	11010	
	1	111011		
Add B	0	100011	11010	0

Q. 55 What is wrong with the following register transfer statements?

(i) $xT : AR \leftarrow AR, AR \leftarrow 0$

(ii) $yT : R_1 \leftarrow R_2, R_1 \leftarrow 0$

(iii) $zT : PC \leftarrow AR, PC \leftarrow PC + 1$

Ans:

(i) Cannot complement and increment the same register at the same time.

(ii) Cannot transfer two different values (R_2 and R_3) to the same register (R_1) at the same time.

(iii) Cannot transfer a new value into a register (PC) and increment the original value by one at the same time.

Q.56 (i) How many 128 x 8 RAM chips are needed to provide a memory capacity of 2048 bytes?

(ii) How many lines of the address bus must be used to access 2048 bytes of memory? How many of there lines will be common to all chips?

(iii) How many lines must be decoded for chip select? Specify the size of the decoders.

Ans:

(i) Memory capacity = 2048 bytes

2^{11} = 2048 bytes

$$= \frac{2^{11}}{2^7} = \frac{2048}{128}$$

$= 2^4 =$ 16 chips RAM

(ii) Memory capacity = 2048 bytes

address bus = 11 (2^{11} = 2048)

$128 = 2^7$ 7 lines to address each chip and 4 lines to decoder for selecting 16 chips. Thus, 7 lines out of 11 will be common to all chips.

(iii) Remaining 4 lines must be decoded for chip select the size of decoder is 4 x 16.

Q. 57 Explain the following with example

(i) Memory-reference instructions.

(ii) Input-output instructions.

(iii) Program control instruction.

Ans:

(i) Memory reference instructions

Memory-Reference Instructions

Symbol	Operation decoder	Symbolic description
AND	D_0	$AC \leftarrow AC \wedge M[AR]$
ADD	D_1	$AC \leftarrow AC + M[AR], E \leftarrow Cout$
LDA	D_2	$AC \leftarrow M[AR]$
STA	D_3	$M[AR] \leftarrow AC$
BUN	D_4	$PC \leftarrow AR$
BSA	D_5	$M[AR] \leftarrow PC, PC \leftarrow AR+1$
ISZ	D_6	$M[AR] \leftarrow M[AR]+1$
		If $M[AR]+1=0$ then $PC \leftarrow PC+1$

(ii) Input-output instructions.

Input-Output Instructions

$D_7IT_3 = p$ (common to all input-output instructions)			
IR(i) = B_i [bit in IR (6-11) that specifies the instruction]			
	p:	$SC \leftarrow 0$	Clear SC
INP	pB_{11}:	$AC(0-7) \leftarrow INPR, FGI \leftarrow 0$	Input character
OUT	pB_{10}:	$OUTAR \leftarrow AC(0-7), FGO \leftarrow 0$	Output character
SKI	pB_9:	IF(FGI=1) then ($PC \leftarrow PC+1$)	Skip on input flag
SKO	pB_8:	IF(FGO=1) then ($PC \leftarrow PC+1$)	Skip on Output flag
ION	pB_7:	$IEN \leftarrow 1$	Interupt enable on
IOF	pB_6:	$IEN \leftarrow 0$	Interupt enable off

(iii) Program control instruction.

Name	Mnemonic
Branch	BR
Jump	JMP
Call	CALL
Return	RET
Compare (by subtraction)	CMP
Test (by ANDing)	

Q.58 What do you mean by effective address of data? List any four addressing modes. How is effective address calculated for them?

Ans:

Effective Address :- The effective address is defined to be the memory address obtained from the computation dictated by the given addressing mode. The effective address is the address of the operand in a computational type instruction

Addressing modes:-

(i) Direct mode

(ii) Indirect mode

(iii) Register direct & register indirect mode

(iv) Immediate mode

(v) Implicit mode

(i) Direct Mode:- In this mode. The instruction includes a memory address

Ex. LDAC 5,

Data from memory location 5 is stored in accumulator.

(ii) Indirect Mode:- The address specified in the instruction is not the address of the operand. It is the address of a memory location that contains the address of the operand.

Ex. LDAC @ 5.

First it retrieve the content of memory location 5 say 10. Then the CPU goes to location 10, reads the contents of that location and loads the data into the CPU.

(iii) Register Direct & Register indirect modes:-

Register modes work the same as direct & indirect modes discussed above, except they do not specify a memory address. Instead they specify a register.

Example: LDAC R $\Rightarrow$ Content of Reg. 'R' is copied in accumulator

LDAC (R) $\Rightarrow$ The content of Reg. 'R' gives the memory address whose content is to be copied in to accumulator.

(iv) Immediate Mode:- The operand specified in the instruction is not an address, it is the actual data to be used.

Example: LDAC # 5, It moves the data value 5 to accumulator.

(v) Implicit Mode:- It does not explicitly specify an operand. The instruction implicitly specify the operand because it always applies to a specific register.

Example: CLAC $\Rightarrow$ Clear accumulator

CMC $\Rightarrow$ Complement accumulator.

Q. 59 Write a program by using two - addressing & one-addressing format to evaluate.

$$\frac{A-B+C * (D-E)}{C + G * H}$$

Ans:

MOV	R_1, D	$R_1 \leftarrow$ M[D]
SUB	R_1, E	$R_1 \leftarrow R_1$ - M[E]
MOV	R_2 C	$R_2 \leftarrow$ M[C]
MUL	R_2, R_1	$R_2 \leftarrow R_1 * R_2$
MOV	R_3, B	$R_3 \leftarrow$ M[B]
ADD	R_3, R_2	$R_3 \leftarrow R_3 + R_2$
MOV	R_4, A	$R_4 \leftarrow$ M[A]
SUB	R_4, R_3	$R_4 \leftarrow R_4$ - R_3
MOV	R_5, G	$R_5 \leftarrow$ M[G]
MUL	R_5, H	$R_5 \leftarrow R_5 *$ M[H]
ADD	R_5, C	$R_5 \leftarrow R_5$+M[C]
DIV	R_4, R_5	$R_4 \leftarrow R_4 \div R_5$
MOV	Y, R_4	M[y] $\leftarrow R_4$

One-addressing

$$Y = \frac{A-B+C * (D-E)}{C+G * H}$$

Load	D	AC $\leftarrow$ M[D]
SUB	E	AC $\leftarrow$ AC-M[E]
MUL	C	AC $\leftarrow$ AC+M[C]
ADD	B	AC $\leftarrow$ AC+M[B]
SUB	A	AC $\leftarrow$ M[A]- AC
Store	T	M[T]- AC
Load	G	AC $\leftarrow$ M[G]
MUL	H	AC $\leftarrow$ AC* M[H]
ADD	C	AC $\leftarrow$ AC+M[C]
DIV	T	AC $\leftarrow$ M[T] $\div$ AC
Store	Y	M[y] $\leftarrow$ AC

Q.60 Define interrupt. Why priority of interrupt is required? How it is restored?

Ans:

Interrupt:- An interrupt is a signal sent by an I/O interface to CPU when it is ready to send information to the memory or receive information from the memory.

interrupt — A priority interrupt is a system that establishes a priority over the various sources to determine which condition is to be serviced first when two or more requests arrive simultaneously. Higher - priority interrupt levels are assigned to requests which, if delayed or interrupted. Devices with high speed transfers such as magnetic disks are given high priority, and slow devices such as keyboards receive low priority. When two devices interrupt the computer at the same time, the computer services the devices, with the higher priority first. The task of the interrupt system is to identify the source of the interrupt. There is also the possibility that several sources will request service simultaneously. In this case the system most also decide which device to service first, than use the priority interrupt.

Establishing the priority of simultaneous interrupts can be done by software or hardware. A polling procedure is used to identify the highest-priority source by software means. In this method there is one common branch address for all interrupts. The program that takes care of interrupts begins at the branch address and polls the interrupt sources in sequence. The order in which they are tested determines the priority of each interrupt. The highest-priority source is tested first, and if its interrupt signal is on, control branches to a service routine from this source. Otherwise, the next-lower-priority source is tested, and so on. Thus the initial service routine for all interrupts consists of a program that tests the interrupt sources in sequence and branches to one of many possible service routines. The particular service routine reached belongs to the highest-priority device among all devices that interrupted the computer. The disadvantage of the software method is that is there are many interrupts, the time required to poll them can exceed the time available to service the I/O device. In this situation a hardware priority-interrupt unit can be used to speed up the operation.

Daisy Chaining Priority

The daisy-chaining method of establishing priority consists of a serial connection of all devices that request an interrupt. The device with the highest priority is placed in the first position, followed by lower-priority devices up to the device with the lowest priority, which is placed last in the chain. This method of connection between three devices and the CPU is shown in figure. The interrupt request line is common to all devices and forms a wired logic connection. If any device has its interrupt signal in the low-level state, the interrupt line goes to the low-level state and enables the interrupt input in the CPU. When no interrupts are pending, the interrupt line stays in the high-level state and interrupts are pending, the interrupt line stays in the high-level state and no interrupts are recognized by the CPU. This is equivalent to a negative logic OR operation. The CPU responds to an interrupt request by enabling the interrupt acknowledge line. This signal is received by device 1 at its *PI* (priority in) input. The acknowledge signal passes on to the next device through the *PO* (priority out) output only if device 1 is not requesting an interrupt. If device 1 has a pending interrupt, it block the acknowledge signal from the next device by placing a 0 in the *PO* output. It

then proceeds to insert its own interrupt vector address (VAD) into the data bus for the CPU to use during the interrupt cycle.

A device with a 0 in its *PI* input generates a 0 in its *PO* output to inform the next-lower-priority device that the acknowledge signal has been blocked. A device that is requesting an interrupt and has 1 in its *PI* input will intercept the acknowledge signal by placing a 0 in its *PO* output. If the devices does not have pending interrupts, it transmits the acknowledge signal to the next device by placing a 1 in its *PO* output. Thus the device with *PI*=1 and *PO*=0 is the one with the highest priority that is requesting an interrupt, and this device places its VAD on the data bus. The daisy chain arrangement gives the highest priority to the device that receives the interrupt acknowledge signal from the CPU. The farther the device is from the first position, the lower is its priority.

Figure shows the internal logic that must be included with each device when connected in the daisy-chaining scheme. The device sets its RF flip-flop when it wants to interrupt the CPU. The output of the *RF* flip-flop goes through an open-collector inverter, a circuit that provides the wired logic for the common interrupt line. If *PI*=0, both *PO* and the enable line to VAD are equal to 0, irrespective of the value of *RF*. If *PI* =1 and *RF*=0, the *PO*=1 and the vector address is disabled. This condition passes the acknowledge signal to the next device through *PO*. The device is active when *PI*=1 and *RF*=1. This condition places a 0 in *PO* and enables the vector address for the data bus. It is assumed that each device has its own distinct vector address. The *RF* flip-flop is reset after a sufficient delay to ensure that the CPU has received the vector address.

Q.61 Differentiate between synchronous and asynchronous data transfer method. **(6)**

Ans:

Synchronous transmission

(i) The two circuits share a common clock frequency and bits are transmitted continously.

(ii) In long distant serial transmission, each unit is driven by a separate clock of the same frequency.

(iii) In synchronous transmission bits must be transmitted continously to keep the clock frequency in both units synchronized with each other.

Asynchronous transmission

(i) In synchronous transmission employs special bits that are inserted at both ends of the character code.

(ii) Each character consists of three parts: a start bit, the character bits, and stop bits.

(iii) Asynchrnous transmission sent only binary information.

Q. 62 Write on ALP to find square-root of a number.

Ans:

Example:- Suppose that X is the square root of number N. To find a suitable equation to be used by the computer for iteration the following manipulation is done:

$$x^2 = N$$
$$2X^2 = N + X^2$$
$$X^2 = \frac{N + X^2}{2}$$
$$\frac{N + X^2}{2}$$

Address	Machine codes	Label	Mnemonics	Operands	Comments
2600	3E, X		MVI	A,X	X is the first approximation
2602	57	BACK	MOV	D,A	
2603	2A,00,25		LHLD	2500	N in H-L.
2606	CD,06,24		CALL	2406	Division Sub-routine, N/X in L.
2609	7E		MOV	A,D	X in A.
260A	85		ADD	L	(X+N/X)
260B	6F		MOV	L,A	
260C	26,00		MVI	H,00	
260E	3E,02		MVI	H,02	
2610	CD,06,24		CLL	2406	(N/X+X)/2 in L=X_{new}
2613	7D		MOV	A,L	
2614	BA		CMP	D	Compare X_{new} with X.
2615	CA;1B,26		JZ	BELOW	Jump to BELOW, if $X_{new}=0$
2618	C3,02,26		JMP	BACK	Jump to Back, if $X_{new}=X$.
261B	32,50,25		BELOW	STA	2550
261A	76			HLT	

Q. 63 What is page fault and how page fault is handled by memory management software?

Ans:

Page fault: —

When a program starts execution, one or more pages are transferred into main memory and the page table is set to indicate their position. The program is executed from main memory until it attempts to reference a page that is still in auxiliary memory. This condition is called page fault.

Handle:—

Page fault occurs in a virtual memory system. It signifies that the page referenced by the CPU is not in main memory. A new page is than transferred from auxiliary memory to main memory. If main memory is full it would be necessary to remove a

page from a memory block to make room for the new page. So the replacement algorithm is used.

Two of the most common replacement algorithms used are the first - in - first - out and least recently used.

Performance:—

Want an algorithm which will result in minimum number of page faults. Same page may be brought into memory several times.

Q. 64 Discuss the different mapping techniques used for cache memory. What is the need of mapping techniques?

Ans:

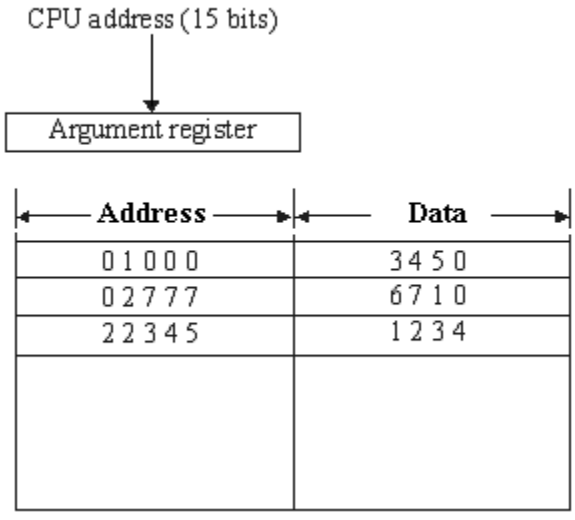
The basic characteristic of cache memory is its fast access time. The transformation of data from main memory to cache memory is referred to as a mapping process.

1) Associative mapping 2) Direct mapping
3) Set-associative mapping

Associative Mapping:-

The fastest and most flexible cache organization uses an associative memory. This organization is illustrated. The associative memory stores both the address and content (data) of the memory word.

This permits any location in cache to store any word from main memory. The diagram shows three words presently stored in the cache. The address value of 15 bits is shown as a five-digit octal

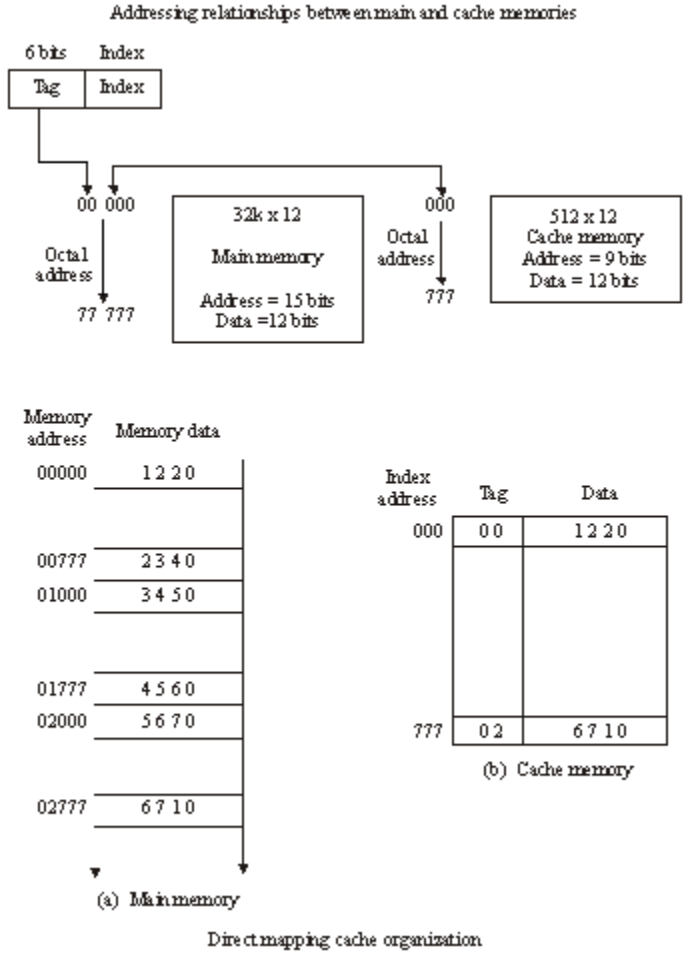


Associative mapping cache (all numbers in octal) number and its corresponding 12-bit word is shown as a four-digit octal number. A CPU address of 15 bits is placed in the argument register and the associative memory is searched for a matching address. If the address is found, the corresponding 12-bit data is read and sent to the CPU. If no match occurs, the main memory is accessed for the word. The address-data pair is then

transferred to the associative cache memory. If the cache is full, an address-data pair must be displaced to make room for a pair that is needed and not presently in the cache. The decision as to what pair is replaced is determined from the replacement algorithm that the designer chooses for the cache. A simple procedure is to replace cells of the cache in round-robin order whenever a new word is requested from main memory. This constitutes a first-in first-out (FIFO) replacement policy.

Direct Mapping:-

Associative memories are expensive compared to random-access memories because of the added logic associated with each cell. The possibility of using a random-access memory for the cache is investigated. The CPU address of 15 bits is divided into two fields.



Direct mapping cache organization

The nine least significant bits constitute the *index* field and the remaining six bits form the *tag* field. The figure shows that main memory needs an address that includes both the tag and the index bits. The number of bits in the index field is equal to the number of address bits required to access the cache memory.

In the general case, there are 2^k words in cache memory and 2^n words in main memory. The *n*-bit memory address is divided into two fields: *k* bits for the index field and n – k bits for the tag field. The direct mapping cache organization uses the *n*-bit address to

access the main memory and the *k*-bit index to access the cache. Each word in cache consists of the data word and its associated tag. When a new word is first brought into the cache, the tag bits are stored alongside the data bits. When the CPU generates a memory request, the index field is used for the address to access the cache. The tag field of the *CPU* address is compared with the tag in the word read from the cache. If the two tags match, there is a hit and the desired data word is in cache. If there is no match, there is a miss and the required word is read from main memory. It is then stored in the cache together with the new tag, replacing the previous value. The disadvantage of direct mapping is that the hit ratio can drop considerably if two or more words whose addresses have the same index but different tags are accessed repeatedly.

Set-Associative Mapping:-

The disadvantage of direct mapping is that two words with the same index in their address but with different tag values cannot reside in cache memory at the same time. A third type of cache organization, called set-associative mapping, is an improvement over the direct-mapping organization in that each word of cache can store two or more words of memory under the same index address. Each data word is stored together with its tag and the number of tag-data items in one word of cache is said to form a set. An example of a set-associative cache organization for a set size of two is shown. Each

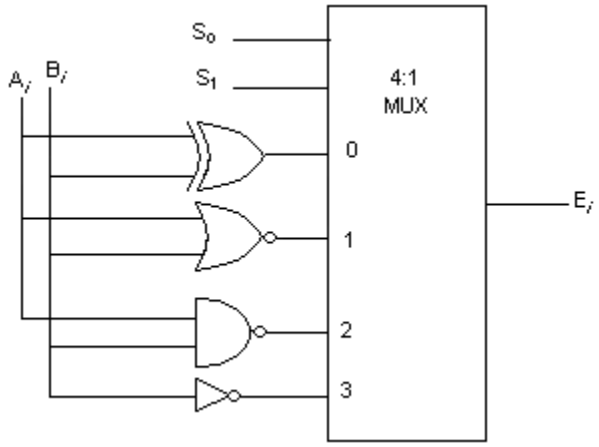
Index	Tag	Data	Tag	Data
000	0 1	3 4 5 0	0 2	5 6 7 0
777	0 2	6 7 1 0	0 0	2 3 4 0

index address refers to two data words and their associated tags. Each tag requires six bits and each data words has 12 bits, so the word length is 2(6+12) = 36 bits. An index address of nine bits can accommodate 512 words. Thus the size of cache memory is 512 x 36. It can accommodate 1024 words of main memory since each word of cache contains two data words. In general, a set-associative cache of set size K will accommodate *k* words of main memory in each word of cache.

When a miss occurs in a set-associative cache and the set is full, it is necessary to replace one of the tag-data items with new value. The most common replacement algorithms used are: FIFO and LRU. The FIFO procedure selects for replacement the item that has been in the set the longest. The LRU algorithm selects for replacement the items that have been least recently used by the CPU. Both FIFO and LRU can be implemented by adding a few extra bits in each word of cache.

Q. 65 Draw the logic diagram of a logic circuit capable of performing X - OR, NOR, NAND, complement operation of two bits A and B. During complement operation, the circuit should be capable of complementing B.

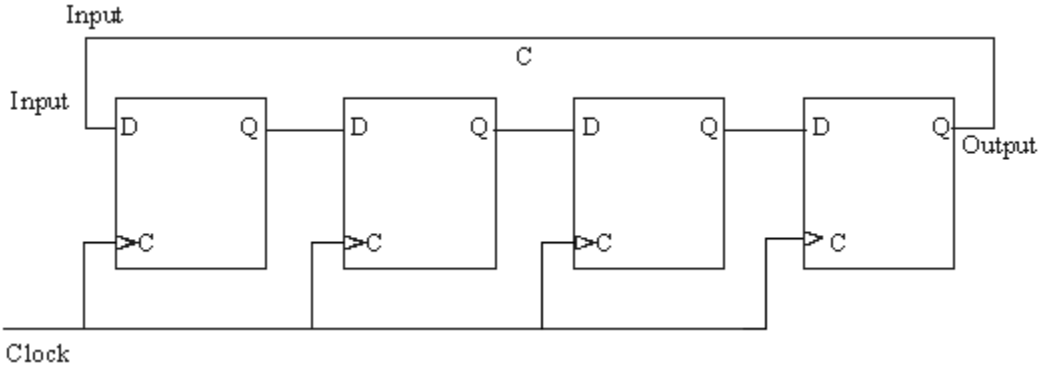
Ans:



S_1	S_0	output	operation
0	0	$E = A \oplus B$	XOR
0	1	E = (A+B)'	NOR
1	0	E = (AB)'	NAND
1	1	E = A'	Complement

Q. 66 Draw the logic circuit using D flip flop for implementing circular shift left operation and circular shift right operation order control input.

Ans.



Q. 67 Write short notes on any two

(i) Hardware polling

(ii) Arithmetic pipe lining

(iii) Common bus architecture

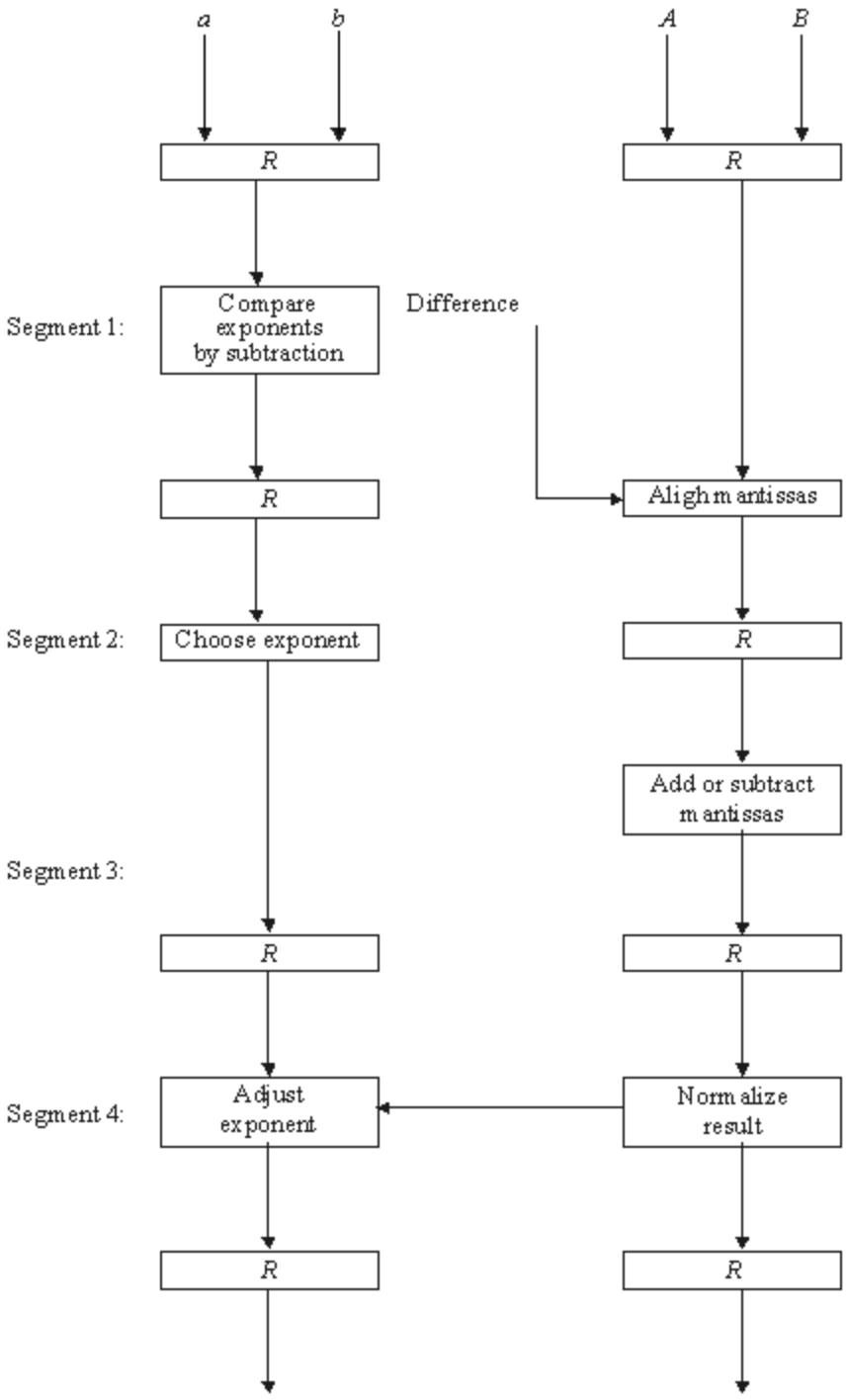
Ans:

(i) Hardware polling:-

A polling procedure is used to identify the highest-priority source by software means. In this method there is one common branch address for all interrupts. The program that takes care of interrupts begins at the branch address and polls the interrupt sources in sequence. The order in which they are tested determines the priority of each interrupt. The highest-priority source is tested first, and if its interrupt signal is on, control branches to a service routine for this source. Otherwise, the next-lower-priority source is tested, and so on. Thus the initial service routine for all interrupts consists of a program that tests the interrupt sources in sequence and branches to one of many possible service routines. The disadvantage of the software method is that if there are many interrupts, the time required to poll them can exceed the time available to service the I/O device. In this situation a hardware priority-interrupt unit can be used to speed up the operation.

A hardware priority-interrupt unit functions as an overall manager in an interrupt system environment. It accepts interrupt requests from many sources, determines which of the incoming requests has the highest priority, and issues an interrupt request to the computer based on this determination. To speed up the operation, each interrupt source has its own interrupt vector to access its own service routine directly. Thus no polling is required because all the decisions are established by the hardware priority-interrupt unit. The hardware priority function can be established by either a serial or a parallel connection of interrupt lines. The serial connection is also known as the daisy-chaining method.

(ii) Arithmetic Pipeline:-



Pipeline for floating-point addition and subtraction

Pipeline arithmetic units are usually found in very high speed computers. They are used to implement floating-point operations, multiplication of fixed-point numbers, and similar computations encountered in scientific problems.

The inputs to the floating-poing adder pipeline are two normalized floating-point binary numbers.

$X = A \times 2^a$

$Y = B \times 2^b$

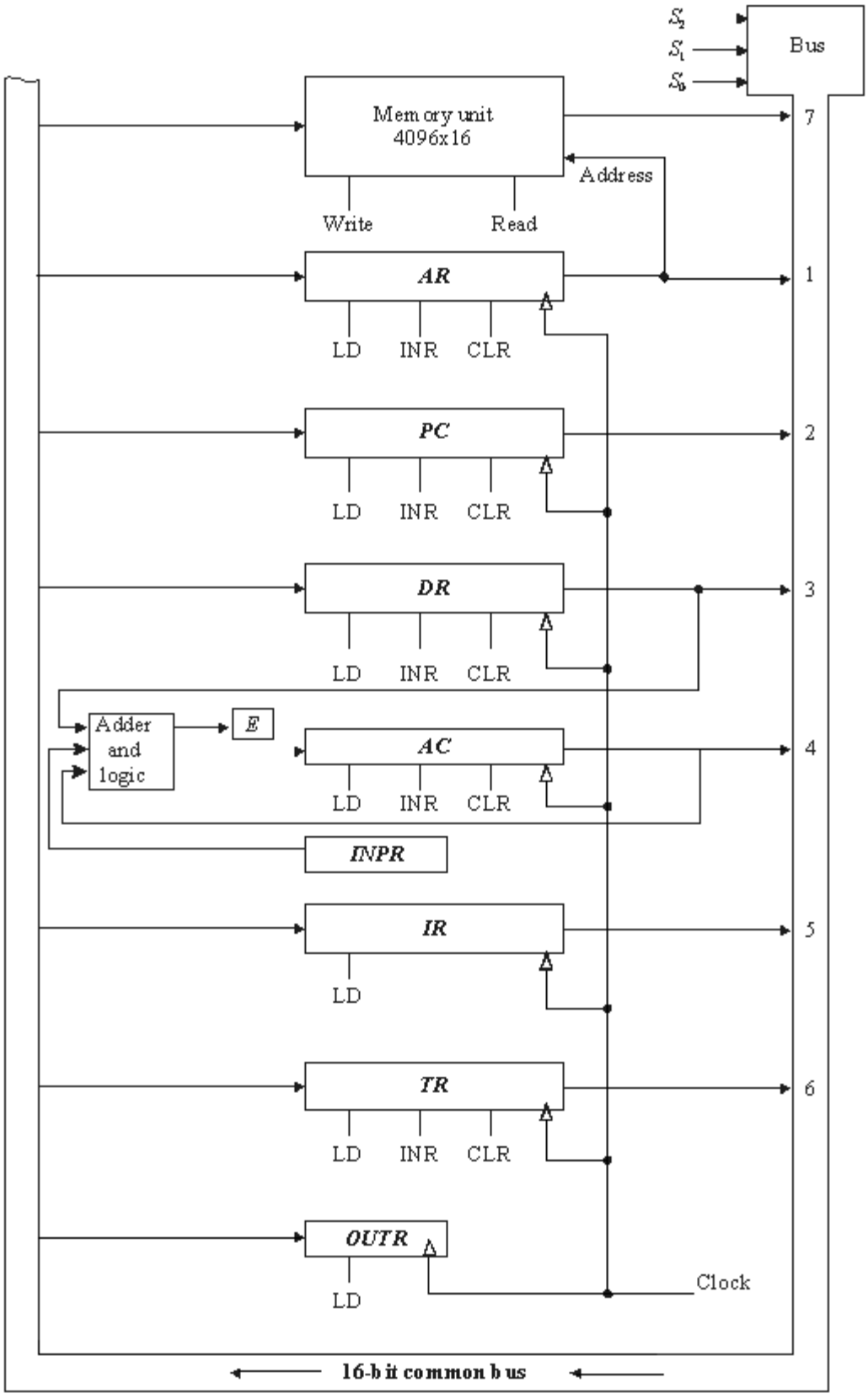
A and B are two fractions that represent the maintissas and a and b are the exponents. The floating-point addition and subtraction can be performed in four segments. The suboperations that are performed in the four segments are:

1) Compare the exponents.
2) Align the mantissas.
3) Add or subtract the mantissas.
4) The exponents are compared by subtracting them to determine their difference. The larger exponent is chosen as the exponent of the result. The exponent difference determines how many times the mantissa associated with the smaller exponent must be shifted to the right. This produces an alignment of the two mantissas. It should be noted that the shift must be designed as a combinational circuit to reduce the shift time. The two mantissas are added or subtracted in segment 3. The result is normalized in segment 4. When an overlow occurs, the mantissa of the sum or difference is shifted right and the exponent incremented by one. If an underflow occurs, the number of leading

zeros in the mantissa determines the number of left shifts in the mantissa and the number that must be subtracted from the exponent.

(iii) **Common Bus System:-**

The basic computer has eight registers, a memory unit, and a control unit .Paths must be provided to transfer information from one register to another and between memory and registers. The number of wires will be excessive if connections are made between the outputs of each register and the inputs of the other registers. A more efficient scheme for transferring information in a system with many registers is to use a common bus. The connection of the registers and memory of the basic computer to a common bus system is shown in figure.



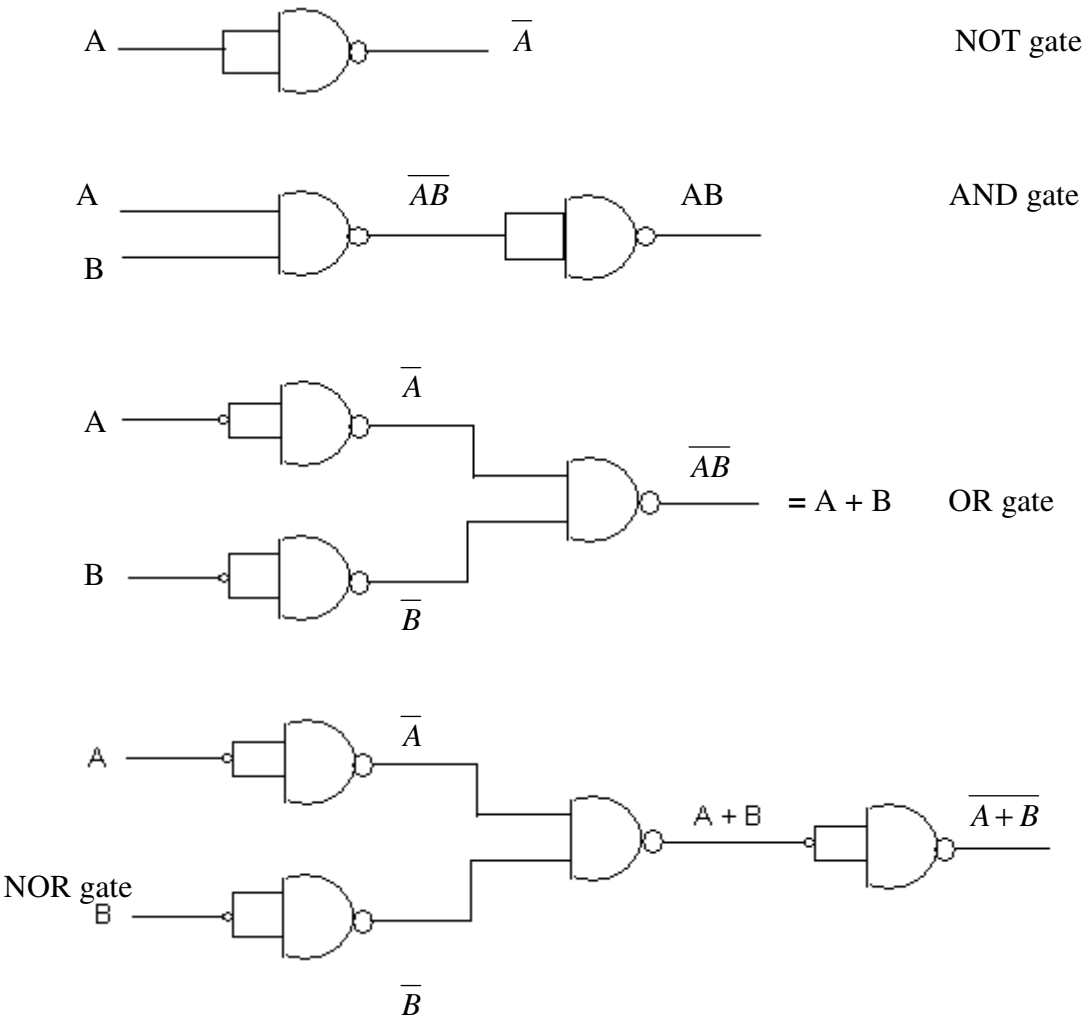
Basic computer registers connected to a common bus.

Q.68 What binary number does $5A34F_{16}$ represent? **(1)**

Ans: 0 1 01 1 0 1 0 0 0 1 1 0 1 0 0 1 1 1 1 $_2$

Q.69 Using NAND gate generate the NOT, AND, OR and NOR functions. **(2x4=8)**

Ans:

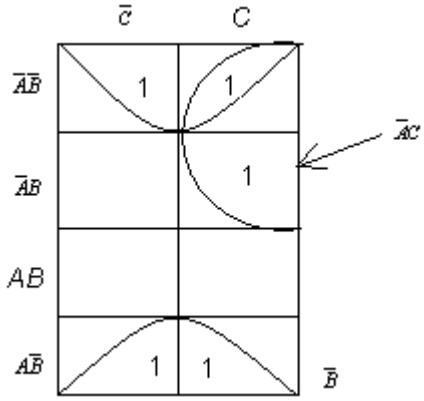


Q.70 Minimize the expression

$$X = A\overline{B}C + \overline{A}BC + \overline{A}\overline{B}C + \overline{A}\overline{B}\overline{C} + A\overline{B}\overline{C}$$

Using Karnaugh's map. **(4)**

Ans:



Four 1's is in adjacent group.
Remaining 1 is absorbed in overlapping gp – gp of four 1's produce a single variable terms $\overline{B}$. This is determined by observing that within the gp $\overline{B}$ is the only variable. That does not change from cell to cell. gp of 2 1's produce a 2 variable term $\overline{A}C$. To get minimized for the 2 terms are summed (OR ed) as

$$X = \overline{B} + \overline{A}C$$

Q.71 Subtract 1010100 – 1000011 using 2's complement. **(3)**

Ans:
X = 1 0 1 0 1 0 0
Y = 1 0 0 0 0 1 1
2's complement of Y = 0 1 1 1 1 0 1
X + Y = 1 0 0 1 0 0 0 1
Discard end carry 2 = - 10000000
X – Y = 0 0 1 0 0 0 1

Q.72 For the following memory units (specified by the number of words the number of bits per word), determine the number of address lines, input/output lines and the number of bytes that can be stored in the specified memory

(i) 64K x 8
(ii) 16M x 32
(iii) 4G x 64
(iv) 2K x 16 **(1 ½ x 4 = 6)**

Ans:
(i) 64K x 8
i/p, o/p lines = 8
Address lines = 16
Mem = 64K

(ii) 16M x 32
i/p, o/p lines = 32
Add = 24
Mem = 64M (16M x 4)

(iii) 4G x 64
i/p, o/p lines = 64
Add = 32
Mem = 32GB (4G x 8)

(iv) 2K x 16
i/p, o/p = 16
Add = 11
Mem = 4K

Q.73 What is a micro-operation? List and briefly explain the most commonly encountered arithmetic operations. **(1+4)**

Ans:
A micro-operation is an elementary operation performed with the data store in registers. The micro-operation most often encountered in digital computer are classified into four categories:

1. Register transfer micro-operations transfer binary information from register to another.
2. Arithmetic micro-operations perform arithmetic operation on numeric data stored in registers.
3. Logic micro-operations perform bit manipulation operation on numeric data stored in registers.
4. Shift micro-operations perform shift operations on data stored registers.

The basic arithmetic micro-operation are addition, subtraction increment, decrement, and shift. Arithmetic shifts are explained later in conjunction with the shift micro-operations. The arithmetic micro-operation defined by the statement

$$R3 \leftarrow R1 + R2$$

specifies an *add* micro-operation .It states that the contents of registerR1 added to the contents of register R3. Subtraction is most often implemented through complementation and addition. Instead of using the minimum operator, we can specify the subtraction by the following statement:

$$R3 \leftarrow R1 + \overline{R2} + 1$$

$\overline{R2}$ is the symbol for the 1's complement of R2. Adding 1 to the 1's complement produces the 2's complement. Adding the contents of R1to the 2's complement of R2 is equivalent to R1- R2.

Arithmetic Micro-operations

Symbolic designation	Description
$R3 \leftarrow R1 + R2$	Contents of R1 plus R2 transferred to R3
$R3 \leftarrow R1 - R2$	Contents of R1 minus R2 transferred to R3
$R2 \leftarrow \overline{R2}$	Complement the contents of R2 (1's complement)
$R2 \leftarrow \overline{R2} + 1$	2's complement the contents of R2 (negate)
$R3 \leftarrow R1 + \overline{R2} + 1$	R1 plus the 2's complement of R2 (subtraction)
$R1 \leftarrow R1 + 1$	Increment the contents of R1 by one
$R1 \leftarrow R1 - 1$	Decrement the contents of R1 by one

The increment and decrement micro-operations are symbolized by plus one and minus-one operation, respectively. These micro-operations are implemented with a combinational circuit or with or with a binary up-down counter.

Q.74 Discuss the different ways in which ROM can be programmed. **(5)**

Ans:

The required path in a ROM may be programmed in three different ways. The first, mask programming, is done by the semiconductor company during the last fabrication process of the unit. The procedure for fabricating a ROM requires that the customer fill out the truth table that he or she wishes the ROM to satisfy. The truth table may be submitted in a special form provided by the manufacturer makes the corresponding mask for the paths to produce the 1's and 0's according to the customer's truth table. This procedure is costly because the vendor charges the customer a special fee for custom masking the particular ROM. For this reason, mask programming is economical only if a large quantity of the same ROM configuration is to be ordered.

For small quantities it is more economical to use a second type of ROM called a programmable read-only memory or PROM. When ordered, PROM units contain all the fuses intact, giving all 1's in the bits of the stored words. The fuses in the PROM are blown by application of current pulses through the output terminals for each address. A blown fuse defines a binary 0 state, and an intact fuse gives a binary 1 state.

This allows users to program PROMs in their own laboratories to achieve the desired relationship between input addresses and words. Special instruments called PROM *programmers* are available commercially to facilitate this procedure. In any case, all procedures for programming ROMs are hardware procedures even through the word "programming" is used.

The hardware procedure for programming ROMs or PROMs is irreversible, and once programmed, the fixed pattern is permanent and cannot be altered. Once a bit pattern has been established, the unit must be discarded if the bit pattern is to be changed. A third type of ROM available is called *erasable* PROM or EPROM can be restructured to the initial value even through its fuses have been blown previously. When the EPROM is placed under a special ultraviolet light for a given period of time, the shortwave radiation discharges the internal gates that serve as fuses. After erasure, the EPROM returns to its initial state and can be reprogrammed to a new set of words. Certain PROMs can be erased with electrical signals instead of ultraviolet light. These PROMs are called *electrically erasable* PROM or EEPROM.

Q.75 Design a 4 bit arithmetic circuit which is capable of performing the following micro-operations:

Addition (with and without carry)

Subtraction (with and without borrow)

Increment

Decrement

and Transfer

Substantiate the circuit diagram with the help of an example. **(12)**

Ans:

The basic component of an arithmetic circuit is the parallel adder. By controlling the data inputs to the adder, it is possible to obtain different types of arithmetic operations.

The diagram of a 4-bit arithmetic circuit is shown in Fig 4. It has four full-adder circuits that constitute the 4-bit adder and four multiplexers for choosing different operations. There are two 4-bit inputs A and B and a 4-bit output D. The four inputs from A go

directly to the X inputs of the binary adder. Each of the four inputs from B are connected to the data inputs of the multiplexers. The multiplexers data inputs also receive the complement of B. The other two data inputs are connected to logic-0 and logic-1. Logic-0 is a fixed voltage value (0 volts for TTL integrated circuits) and the logic-1 signal can be generated through an inverter whose input is 0. The four multiplexers are controlled by two selection inputs, S_1 and S_0. The input carry C_{in} goes to carry input of the FA in the least significant position. The other carries are connected from one stage to the next.

The output of the binary adder is calculated from the following arithmetic sum:

$$D = A + Y + C_{in}$$

Where A is the 4-bit binary number at the X input and Y is the 4-bit binary number at the Y inputs of the binary adder. C_{in} is the input carry, which can be equal to 0 or 1. Note that the symbol + in the equation above denotes an arithmetic plus. By controlling the value of Y with the two selection inputs S and making C_{in} equal to 0 or 1, it is possible to generate the eight arithmetic micro-operations listed in Table 4.

TABLE 4: Arithmetic Circuit Function Table

Select		Input	Output	
S_1	S_0	C_{in}	Y	$D = A + Y + C_{in}$	Micro-operation
0	0	0	B	$D = A + B$	Add
0	0	1	B	$D = A + B + 1$	Add with carry
0	1	0	$\overline{B}$	$D = A + \overline{B}$	Subtractwith borrow
0	1	1	$\overline{B}$	$D = A + \overline{B} + 1$	Subtract
1	0	0	0	$D = A$	Transfer A
1	0	1	0	$D = A + 1$	Increment A
1	1	0	1	$D = A - 1$	Decrement A
1	1	1	1	$D = A$	Transfer A

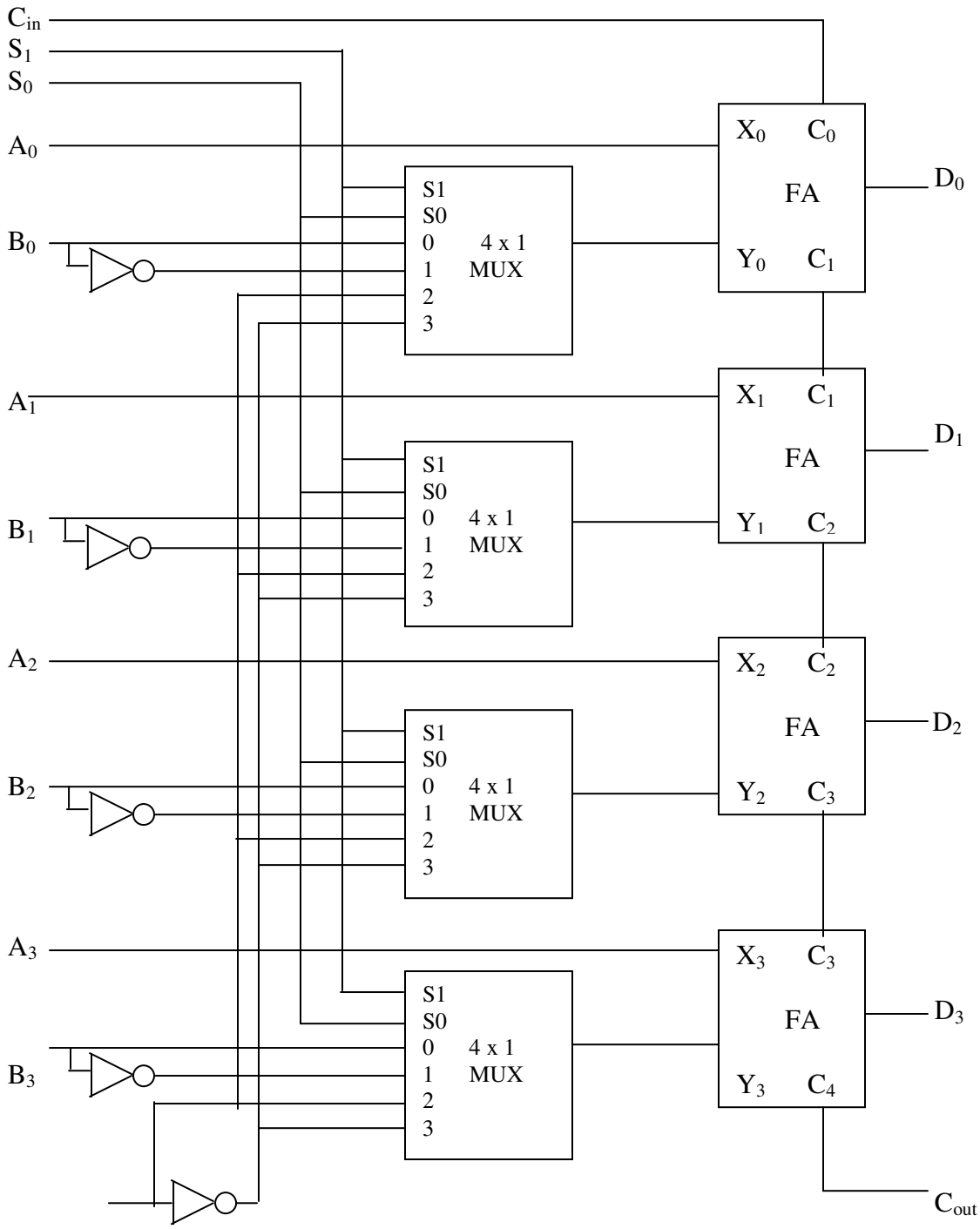


Fig 4 : 4 – bit arithmetic circuit

Q.76 Draw detailed flowchart of the instruction cycle. Indicate the conditions in which register-reference / memory-reference and input-output instructions are executed. Also include the interrupt cycle micro-operations in the flowchart. **(9)**

Ans:

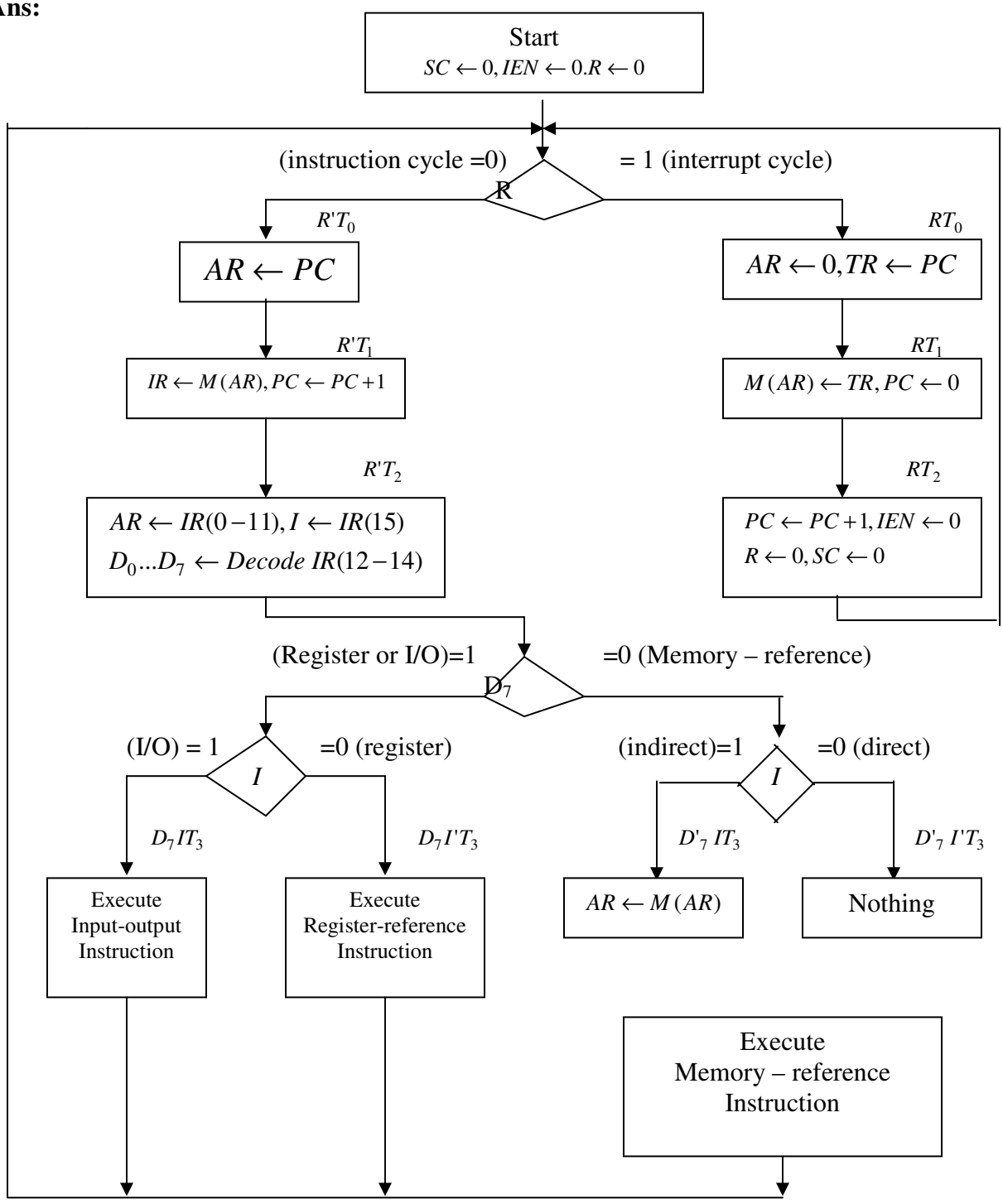


Fig 5 : Flowchart for computer operation.

Q.77 Give the cache access time as 10 ns memory access time as 100 ns and cache hit rate as 90%, calculate the effective memory access time. **(3)**

Ans: 0.90 * 10 + 0.10 * 100
= 19 ns Effective Mem Access Time.

Q.78 Explain the terms burst transfer and cycle stealing. **(5)**

Ans: Refer Section 11-6 page 418 from Morris mano (3rd Edition)
When the DMA takes control of the bus system, it communicates directly with the memory. The transfer can be made in several ways. In DMA burst transfer, a block sequence consisting of a number of memory words is transferred in a continuous burst while the DMA controller is master of the memory buses. This mode of transfer is needed for fast devices such as magnetic disks, where data transmission cannot be stopped or slowed down until an entire block is transferred. An alternative technique called cycle stealing allows the DMA controller to transfer one data word at a time, after which it must return control of the buses to the CPU. The CPU merely delays its operation for one memory cycle to allow the direct memory I/O transfer to "steal" one memory cycle.

Q.79 Explain how DMA controller communicates and transfers data between the peripherals devices and RAM. **(8)**

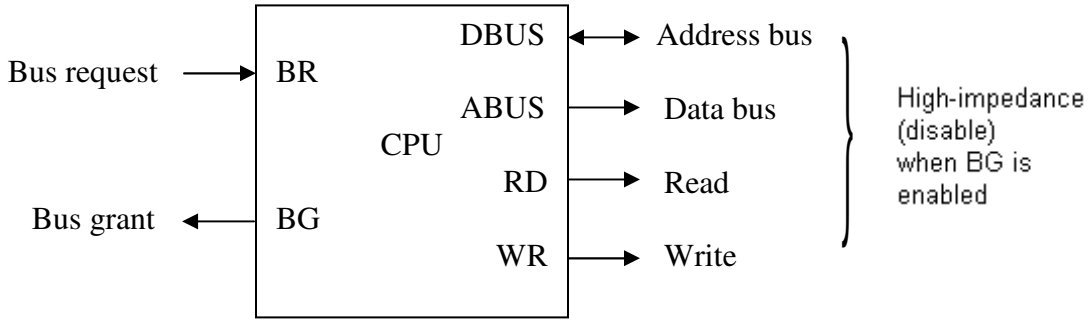
Ans:
During DMA transfer, the CPU is idle and has no control of the memory buses .A DMA controller takes over the buses to manage the transfer directly between the I/O device and memory.

The CPU may be placed in an idle state in variety of ways. One common method extensively used in microprocessor is to disable the buses through special control signals. Figure shows two control signals in the CPU that facilitate the DMA transfer. The bus request (BR) input is used by the DMA controller to request the CPU to relinquish control of the buses. When this input is active, the CPU terminates the execution of the current instruction and places the address bus, the data bus, and the read and writes lines into a high- impedance state. The high-impedance state behaves like an open circuit, which means that the output is disconnect and does not have logic significance. The CPU activates the bus grant (BG) output to inform the external DMA that the buses are in the high-impedance state. The DMA that originated the bus request can now take control of the buses to conduct memory transfer without processor intervention. The CPU disables the bus grant takes control of the buses, and returns to its normal operation.

When the DMA takes control of the bus system, it communicates directly with the memory. The transfers can be made in several ways. In DMA burst transfer, a block sequence consisting of a number of memory words in transferred in a continuous burst while the DMA controller is master of the memory buses. This mode of transfer is needed for fast devices such as magnetic disk, where data transmission cannot be stopped or slowed down until an entire block is transferred.

The DMA controller needs the usual circuits of an interface to communicate with the CPU and I/O device. In addition, it needs an address register, a word count register, and a set of address lines. The address register and address lines.

Figure 7.1 CPU bus signals for DMA transfer.



The DMA is first initialized by the CPU. After that, the DMA starts and continues to transfer data between memory and peripheral unit until an entire block is transferred. The initialization process is essentially a program consisting of I/O instructions that include the address for selecting particular DMA registers. The CPU initializes the DMA by sending the following information through the data bus:

1. The starting address of the memory block where data are available (for read) or where data are to be stored (for write)
2. The word count, which is the number of words in the memory block.
3. Control to specify the mode of transfer such as read or write.
4. A control to start the DMA transfer.

The starting address is stored in the address register. The word count is stored in the word count register, and the control information in the control register. Once the DMA is initialized, the CPU stops communicating with the DMA unless it receives an interrupt signal or if it wants to check how many words have been transferred.

DMA Transfer

The position of the DMA controller among the other components in a computer system is illustrated in Fig 7.2. The CPU communicates with the DMA through the address and data buses as with any interface unit. The DMA has its own address, which activates the DS and RS lines. The CPU initializes the DMA through the data bus. Once the DMA receives the start control command, it can start the transfer between the peripheral device and the memory

.

When the peripheral device sends a DMA request, the DMA controller activates the BR line, informing the CPU to relinquish the buses. The CPU responds with its BG line, informing the DMA that its buses are disabled. The DMA then puts the current value of its address register into the address bus initiates the RD and WR signal, and sends a DMA acknowledge to the peripheral device. Note that the RD and WR lines in the DMA controller are bidirectional .The direction of transfer depends on the status of the BG line.

When BG=0 the RD and WR are input lines allowing the CPU to communicate with the internal DMA registers. When BG =1, the RD and WR are output lines from the

DMA controller to the random–access memory to specify the read or write operation for the data.

When the peripheral device receives a DMA acknowledge, it puts a word in the data bus (for write) or receives a word from the data bus (for read). Thus the DMA controls the read or write operations and supplies the address for the memory. The peripheral unit can then communicate with memory through the data bus for direct transfer between the two units while the CPU is momentarily disabled.

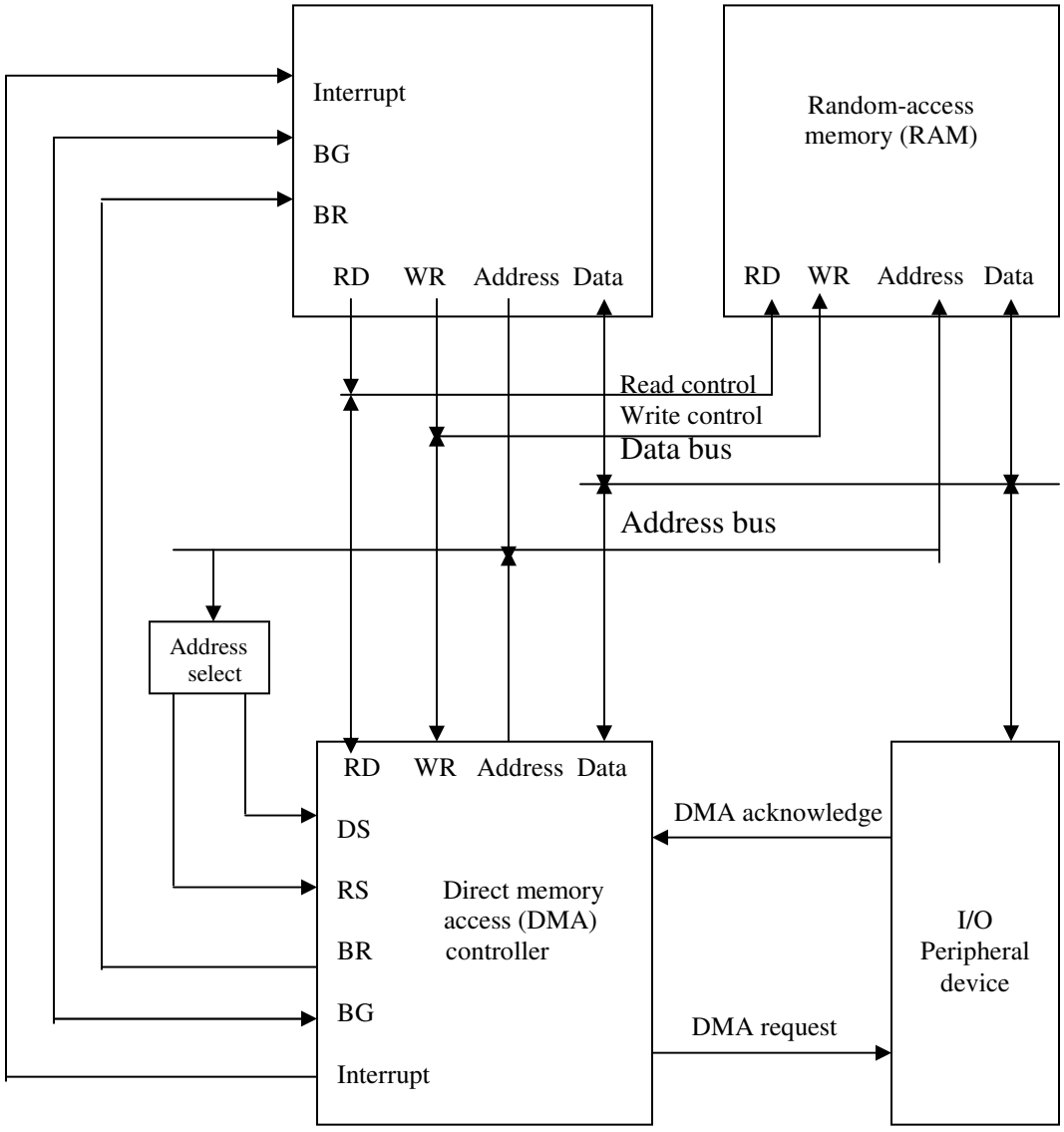


Fig 7.2 DMA transfer in a computer system.

For each word that is transferred, the DMA increment its address register and decrements its word count register. If the word count does not reach zero, the DMA checks the request line coming from the peripheral. For a high-speed device, the line will be active as soon as the previous transfer is completed. A second transfer is then initiated, and the process continues until the entire block is transferred. If the peripheral

speed is slower, the DMA request line may come somewhat later. In this case the DMA disables the bus request line so that the CPU can continue to execute its program. When the peripheral requests a transfer, the DMA requests the buses again.

If the word count register reaches zero, the DMA stops any further transfer and removes its bus request. It also informs the CPU of the termination by means of an interrupt. When the CPU responds to the interrupts, it reads the content of the words were transferred successfully. The CPU can read this register at any time to check the number of words already transferred.

A DMA controller may have more than one channel. In this case, each channel has a request and acknowledges pair of control signals which are connected to separate peripheral devices. Each channel also has its own address register and word count register within the DMA controller. A priority among the channels may be established so that channels with high priority are serviced before channels with lower priority.

Q.80 Differentiate between
(i) Isolated and Memory-Mapped I/O
(ii) Associated Mapping and Direct Mapping. **(8)**

Ans:

(i) Isolated versus Memory-Mapped I/O

Many computers use one common bus to transfer information between memory or I/O and the CPU. The distinction between a memory transfer and I/O transfer is made through separate read and write lines. The CPU specifies whether the address on the address line is for a memory word or for an interface register by enabling one of two possible read or write lines. The I/O read and memory write control lines are enabled during a memory write control lines are enabled during a memory transfer. This configuration isolates all I/O interface addresses from the addresses assigned to memory and is referred to as the isolated I/O method for assigning addresses in a common bus.

In the isolated I/O configuration, the CPU has distinct input and output instructions, and each of these instructions is associated with the address of an interface register. When the CPU fetches and decodes the operation code of an input or output instruction, it places the address associated with the instruction into the common address lines. At the same time, it enables the I/O read (for input) or I/O write (for output) control line. This informs the external components that are attached to the common bus that the address in the address lines is for an interface register and not for a memory word .On the other hand, when the CPU is fetching an instruction or an operand from memory, it places the memory address on the address lines and enables the memory read or memory write control line. This informs the external components that the address is for a memory word and not for an I/O interface.

The isolated I/O method isolates memory and I/O addresses so that memory address values are not affected by interface address assignment since each has its own address

space. The other alternative is to use the same address space for both memory and I/O. This is the case in computers that employ only one set of read and write signals and do not distinguish between memory and I/O addresses. This configuration is referred to as memory-mapped I/O. The computer treats an interface register as being part of the memory system. The assigned addresses for interface registers cannot be used for memory words which reduce the memory address range available.

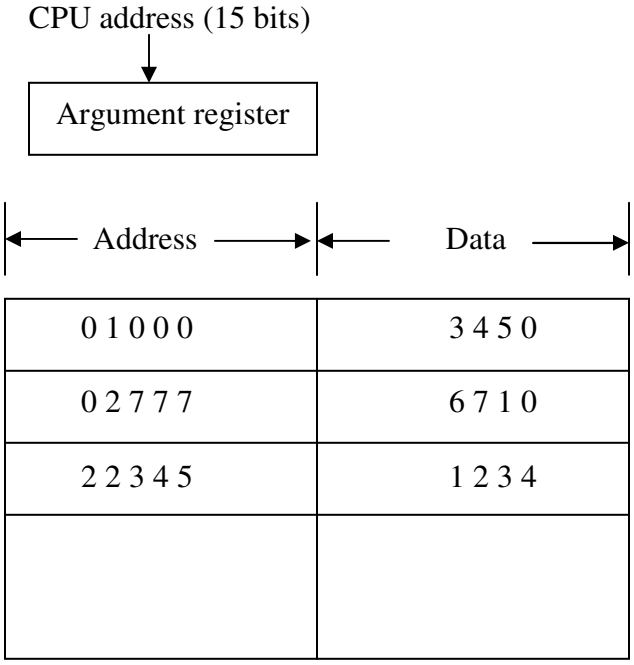
In a memory-mapped I/O organization there is no specific input or output instructions. The CPU can manipulate I/O data residing in interface registers with the same instructions that are used to manipulate memory words. Each interface is organized as a set of registers that respond to read and write requests in the normal address space. Typically, a segment of the total address space is reserved for interface registers, but in general, they can be located at any address as long as there is not also a memory word that responds to the same address.

Computers with memory-mapped I/O can use memory-type instructions to access I/O data. It allows the computer to use the same instructions for either input-output transfers or for memory transfers, The advantage is that the load and store instructions used for reading and writing from memory can be used to input and output data from I/O registers. In a typical computer, there are more memory-reference instructions than I/O instructions. With memory-mapped I/O all instructions that refer to memory are also available for I/O.

(ii) **Associative Mapping**

The fastest and most flexible cache organization uses an associative memory. As in fig. 7.1 the associative memory stores both the address and content (data) of the memory word. This permits any location in cache to store any word from main memory. The diagram shows three words presently stored in the cache. The address value of 15-bites is shown as a five-digit octal number. A CPU address of 15 bits is placed in the argument register and the associative memory is searched for a matching address.

Fig 7.1 Associate mapping cache (all numbers in octal).



If the address is found, the corresponding 12-bit data is read and sent to the CPU. If no match occurs, the main memory is accessed for the word. The address-data pair is then transferred to the associative cache memory. If the cache is full, an address-data pair must be displaced to make room for a pair that is needed and not presently in the cache. The decision as to what pair is replaced is determined from the replacement algorithm that the designer chooses for the cache. A simple procedure is to replace cell of the cache in round-robin order whenever a new word is requested from main memory. This constitutes a first-in first-out (FIFO) replacement policy.

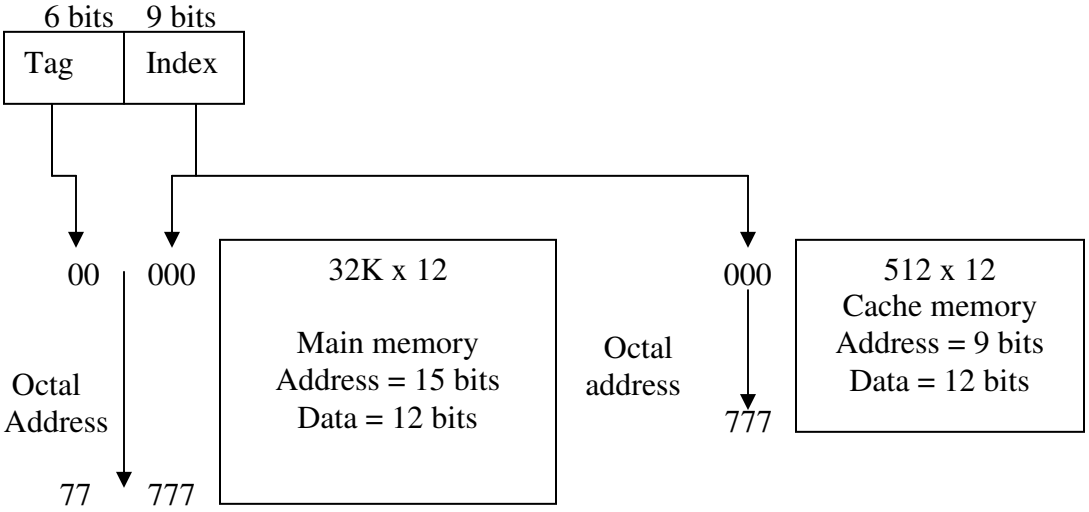
Direct Mapping

Associative memories are expensive compared to random- access memories because of the added logic associated with each cell. The possibility of using a random-access memory for the cache is investigated in Fig 7.2. The CPU address of 15 bits is divided into two fields. The nine least significant bits in the *index* field is equal to the number of address bits required to access the cache memory.

In the general case, there are 2^k words in cache memory and 2^n words in main memory. The n-bit memory address is divided into two fields: k bits for the index field and $n - k$

bits for the tag field. The direct mapping cache organization uses the n-bit address to access the main memory and the *k*-bit index to access the cache. Each word in cache consists of the data word and its associated tag. When a new word is first brought into the cache, the tag bits are stored alongside the data bits. When the CPU generates a memory request, the index field is used for the address to access the cache.

Fig 7.2 Addressing relationships between main and cache memories.



Q.81 What is a page fault? What does a page fault signify? Explain the different page replacement algorithms which determine the page to be removed in case of full memory. **(6)**

Ans:

A virtual memory system is a combination of hardware and software techniques.

The memory management software system handles all the software operation for the efficient utilization of memory space. It must decide (1) Which page in main memory ought to be removed to make room for a new page, (2) When a new page is to be transferred from auxiliary memory to main memory, and (3) Where the page is to be placed in main memory. The hardware mapping mechanism and the memory management software together constitute the architecture of a virtual memory.

When a program starts execution, one or more pages are transferred into main memory and the page table is set to indicate their position. The program is executed from main memory until it attempts to reference a page that is still in auxiliary memory. This condition is called page fault. When page fault occurs, the execution of the present program is suspended until the required page is brought into main memory. Since loading a page from auxiliary memory to main memory is basically an I/O operation, the operating system assigns this task to the I/O processor. In the meantime, control is transferred to the next program in memory that is waiting to be processed in the CPU.

Later, when the memory block has been assigned and the transfer completed, the original program can resume its operation.

When a page fault occurs in a virtual memory system, it signifies that the page referenced by the CPU is not in main memory. A new page is then transferred from auxiliary memory to main memory. If main memory is full, it would be necessary to remove a page from a memory block to make room for the new page. The policy for choosing pages to remove is determined from the replacement algorithm that is used. The goal of a replacement policy is to try to remove the page least likely to be referenced in the immediate future.

Two of the most common replacement algorithms used are the first-in, first-out (FIFO) and the least recently used (LRU). The FIFO algorithm selects for replacement the page that has been in memory the longest time. Each time page is loaded into memory, its identification number is pushed into a FIFO stack. FIFO will be full whenever memory has no more empty blocks. When a new page must be loaded , the page least recently brought in is removed is easily determined because its identification number is at the top of the FIFO stack The FIFO replacement policy has the advantage of being easy to implement. It has the disadvantage that under certain circumstances pages are removed and loaded from memory too frequently.

The LRU policy is more difficult to implement but has been more attractive on the assumption that the least recently used page is a better candidate for removal than the least recently loaded page as in FIFO. The LRU algorithm can be implemented by associating a counter with every page that is in main memory. When a page is referenced, its associated with all pages presently in memory are incremented by 1. The least recently used page is the page with the highest count. The counters are often called aging registers, as their count indicates their age, that is, how long ago their associated pages have been referenced.

Q.82 What is the significance of initialising cache? How is it done? **(4)**

Ans:

One more aspect of cache organization that must be taken into consideration is the problem of initialization. The cache is initialized when power is applied to the computer or when the main memory is loaded with a complete set of programs from auxiliary memory. After initialization the cache is considered to be empty, but in effect it contains some non valid data. It is customary to include with each word in cache a valid bit to indicate whether or not the word contains valid data.

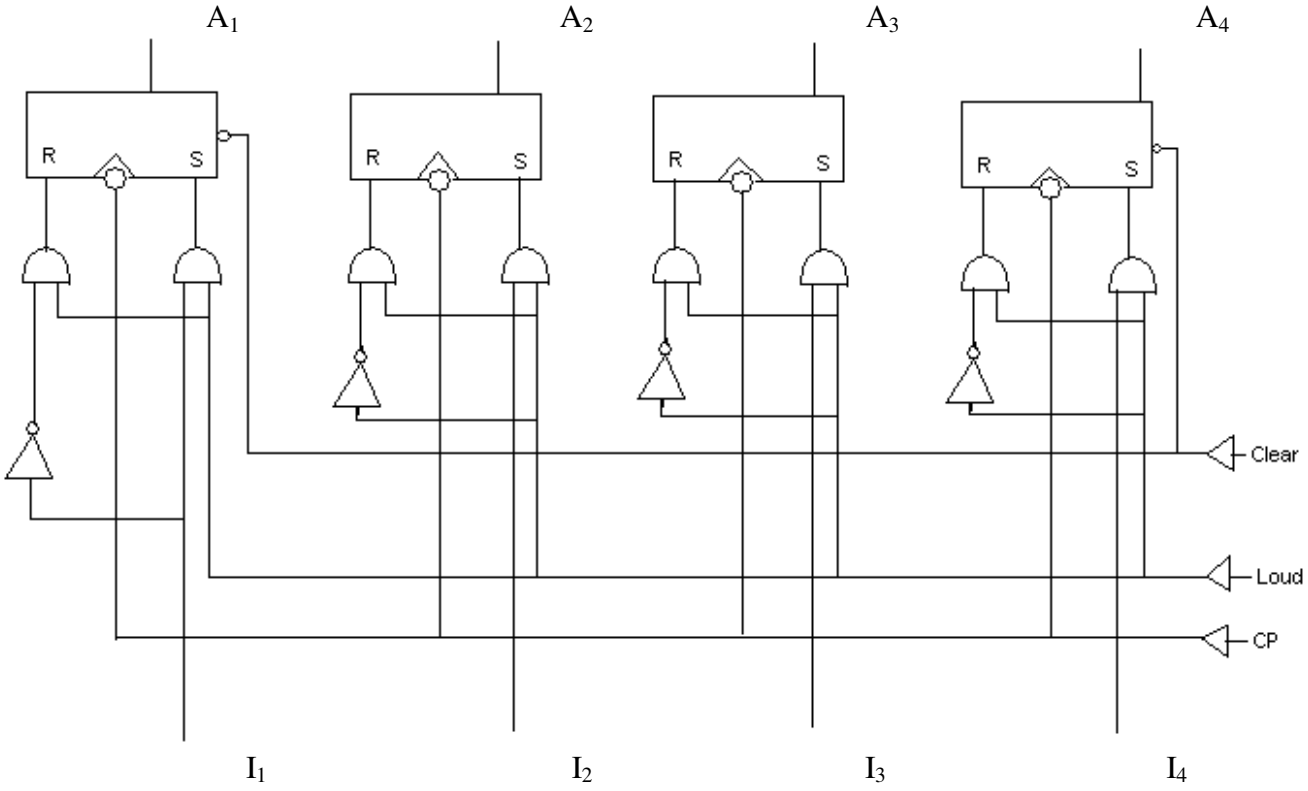
The cache is initialized by clearing all the valid bits to 0. The valid bit of a particular cache word is set to 1 the first time this word is loaded from main memory and stays set unless the cache has to be initialized again. The introduction of the valid bit means that

a word in cache is not replaced by another word unless the valid bit is set to 1 and a mismatch of tags occurs. `If the valid bit happens to be 0, the new word automatically replaces the invalid data. Thus the initialization condition has the effect of forcing misses from the cache until it fills with valid data.

Q.83 Draw the block diagram of four-bit register with parallel load. **(8)**

Ans:

Four – bit register with parallel load



Q.84 Define the following:

(i) Decoder

(ii) Multiplexers

(iii) Demultiplexers

Draw the block diagrams also. **(8)**

Ans:

(i) Decoder: A decoder is a digital function that converts binary information from one coded form to another.

(ii) Multiplexers: A Multiplexer is a digital function that receives binary information from 2^n lines and transmits information on single output line.

(iii)**Demultiplexers:** A demultiplexer is a digital function that receives information on a single line being selected is determined from the bit ambition of n selected lines.

Q.85 What is the sequence of external operations needed to store a word into memory? **(4)**

Ans: The sequence of entered operations needed to store a word into memory are:

1. Transfer the address bits of the required word into MAR.
2. Transfer the data bits of the word into the MBR.
3. Activate the write signal.

Q.86 Briefly explain fixed-point representation of numbers. What is the signed magnitude, 1's complement and 2's complement of -9? **(8)**

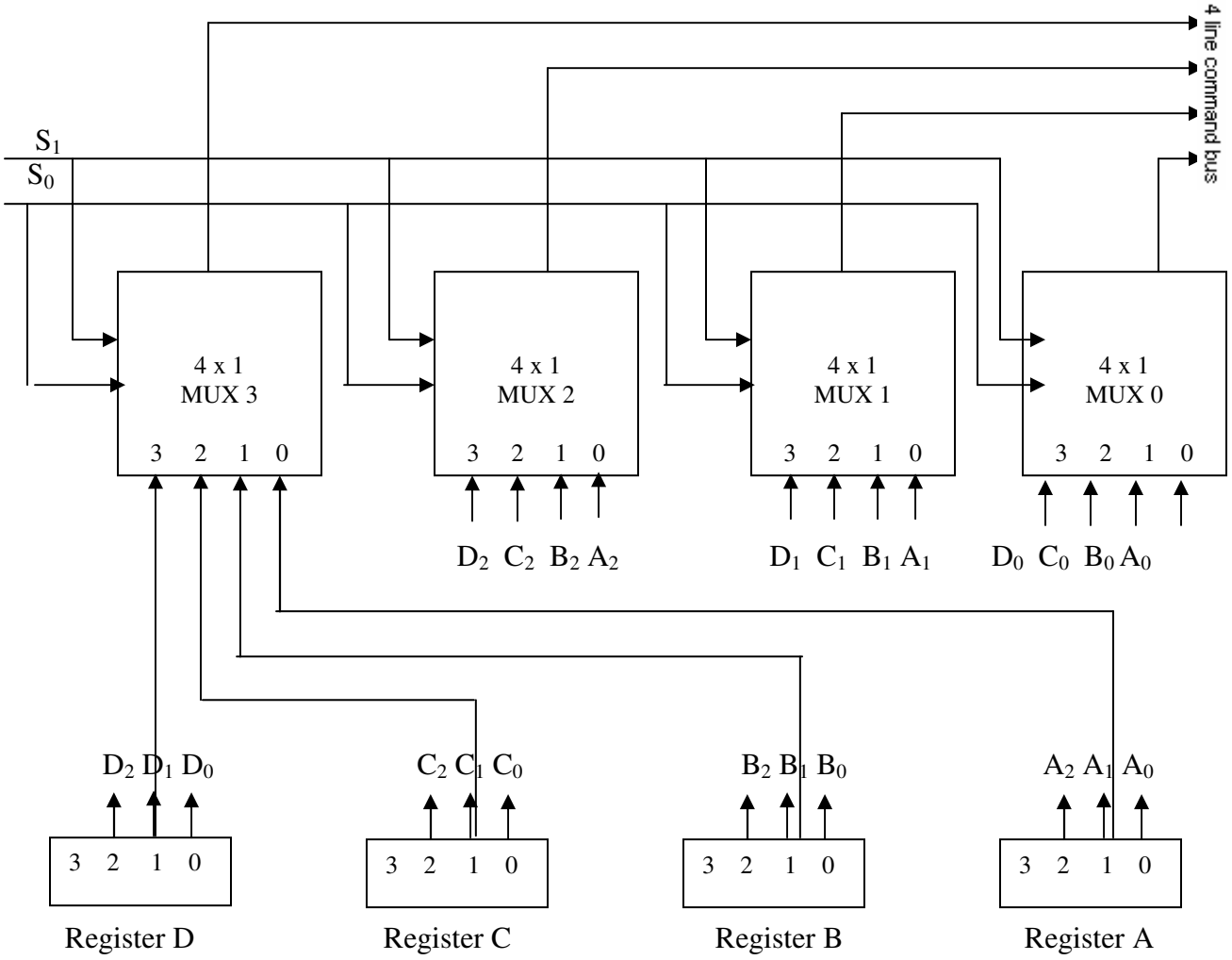
Ans: The fixed point method assumes that the decimal point is always fixed in one position. The two positions most widely used as (1) decimal point in the extreme left of the register to make the stored number a fraction & (2) a decimal point in the extreme right of the register to make the stored number an integer.

The signed magnitude representation of – 9 is obtained from +9 (0001001) by complementing only the sign bit. The signal -1's complement representation of -9 is obtained by complementing all the bits of 0001001 (+9), including the sign bit. The signed -2's complement designation is obtained by taking the 2's complement of the positive number, including its sign bit.

Q.87 Draw a diagram of bus system for four registers of 4-bits each. The bus is to be constructed with multiplexers. **(8)**

Ans:

Fig 4: Bus system for four registers.



Q.88 Explain the following:

(i) Memory-reference instruction.
(ii) Register-reference instruction
(iii) Input-output instruction. **(8)**

Ans:
(i) Refer Section 5-6 page 147 from Morris mano (3rd Edition)
(ii) Refer Section 5-5 page 145 from Morris mano (3rd Edition)
(iii) Refer Section 5-7 page 154 from Morris mano (3rd Edition)

Q.89 Write the instructions for multiplying two unsigned numbers in assembly language. **(8)**

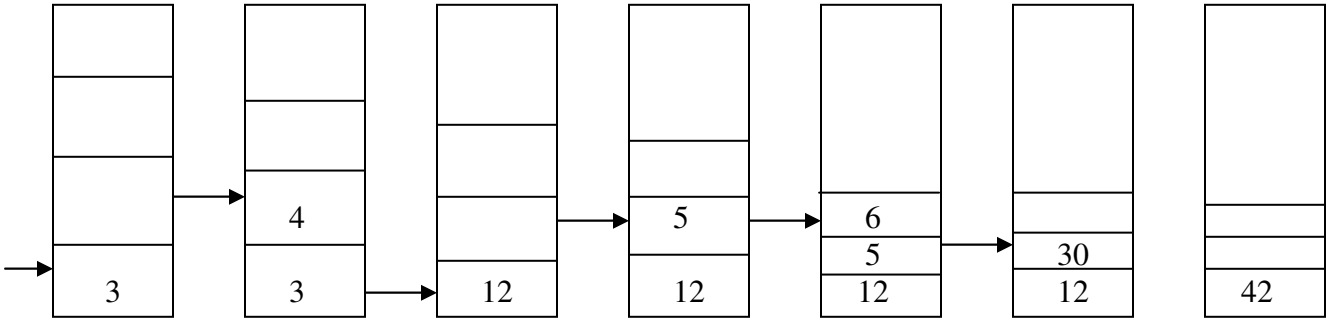
Ans:

```
L0P,    ORG 100
        CLE
```

```
                LDA Y
                CIR
                STA, Y
                SZE
                BUN ONE
                BUN ZR0

ONE,            LDA X
                ADD P
                STA P
                CLE

ZR0,            LDA X
                CIL
                STA X
                ISZ CTR
                BUN L0P
                HLT

CTR,            DEC -8
X,              HEX 000F
Y,              HEX 000B
P,              HEX 0
                END
```

Q.90 Give stack operations to evaluate 3*4+5*6. **(4)**

Ans:



Stack operations to evaluate 3 * 4 + 5 * 6

Q.91 What is the main difference between implied and immediate modes of addressing. **(4)**

Ans:

Implied mode: In this mode the operands are specified implicitly in the definition of the instruction. For example, the instruction "complement accumulator" is an implied-

mode instruction because the operand in the accumulator register is implied in the definition of the instruction.

Immediate mode: In this mode the operand is specified in the instruction itself. In other words, an immediate-mode instruction has an operand field rather than an address field. The operand field contains the actual operand to be used in conjunction with the operation specified in the instruction.

Q.92 Discuss the Base Register Addressing Mode in detail. **(4)**

Ans:
Base Register Addressing Mode: In this mode the content of a base register is added to the address part of the instruction to obtain the effective address. This is similar to the indexed addressing mode except that the register is now called a base register instead of an index register. A base register is assumed to hold a base address and the address field of the instruction gives a displacement relative to this base address.Thus,
Effective address = address part of instruction + content of base register

Q.93 List in detail typical program control instructions of basic computer. **(6)**

Ans:

Name	Mnemonic
Branch	BR
Jump	JMP
Skip	SKP
Call	CALL
Return	RET
Compare (by subtraction)	CMP
Test (by ANDing)	TST

Refer Section 8-7 page 275-276 from Morris mano (3rd Edition)

Q.94 What is a cache memory? How is the performance of cache memory measured? **(6)**

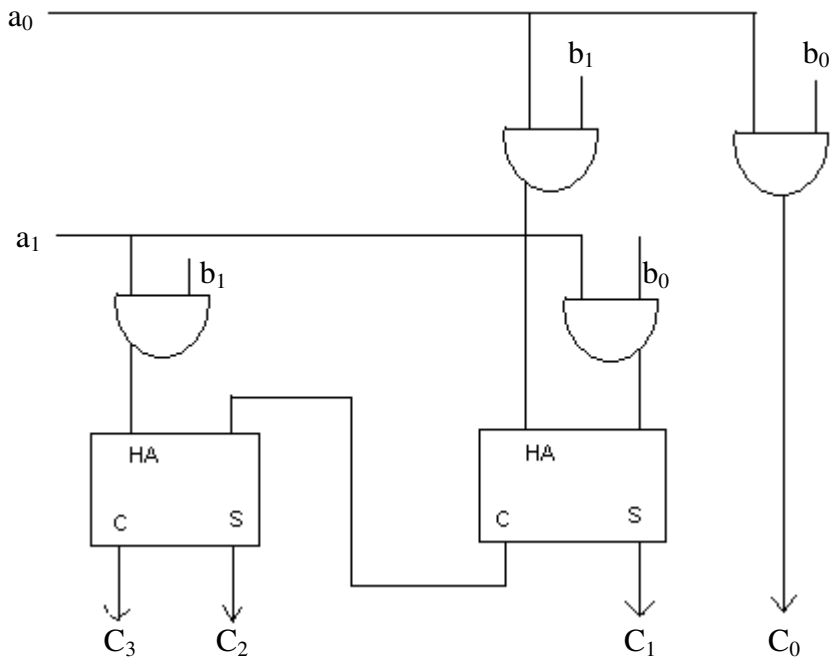
Ans:
When a program loop is executed, the CPU repeatedly refers to the set of instructions in memory that constitute the loop. Every time a given subroutine is called, its set of instructions are fetched from memory. Thus loops and subroutines tend to localize the references to memory for fetching instructions. To a lesser degree, memory references to data also tend to be localized. Table-lookup procedures repeatedly refer to that portion in memory where the table is stored. Iterative procedures refer to common memory locations and array of numbers are confined within a local portion of memory. The result of all these observations is the locality of reference property, which states that over a short interval of time, the addresses generated by a typical program refer to a few localized areas of memory repeatedly, while the remainder of memory is accessed relatively infrequently.

If the active portions of the program and data are placed in a fast small memory, the average memory access time can be reduced, thus reducing the total execution time of the program. Such a fast small memory is referred to as a *cache memory*. It is placed between the CPU and main memory. The cache memory access time is less than the access time of main memory by a factor of 5 to 10. The cache is the fastest component in the memory hierarchy and approaches the speed of CPU components.

The fundamental idea of cache organization is that by keeping the most frequently accessed instructions and data in the fast cache memory, the average memory access time of a computer system can be improved considerably by use of cache.
The performance of cache memory is frequently measured in terms of a quantity called *hit ratio.* When the CPU refers to memory and finds the word in cache, it is said to produce a *hit.* If the word is not found in cache, it is in main memory and it counts as a *miss.* The ratio of the number of hits divided by the total CPU references to memory (hits plus misses) is the hit ratio. The hit ratio is best measured experimentally by running representative programs in the computer and measuring the number of hits and misses during a given interval of time. Hit ratios of 0.9 and higher have been reported. This high ratio verifies the validity of the locality of reference property.

Q.95 Design an array multiplier that multiplies two 2-bit numbers. Use AND gates and half adders. **(8)**

Ans: Refer page 349 from Morris mano (3rd Edition)



2 by 2 binary array multiplier

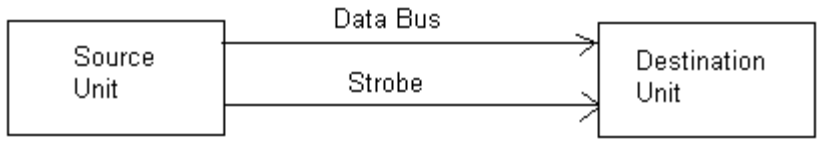
Q.96 Derive an algorithm in flowchart form to multiply two binary numbers in signed-2's complement representation. **(8)**

Ans: Refer fig 10-8 from page 347 from Morris mano (3rd Edition)

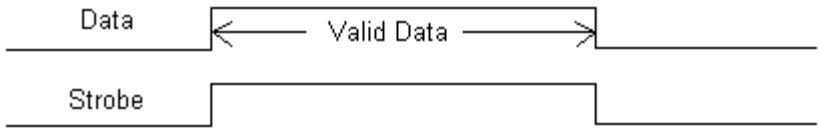
Q.97 What is strobe control? Draw a block diagram for source-initiated strobe for data transfer. **(8)**

Ans:
The strobe control method of asynchronous data transfer employs a single control line to time each transfer. The strobe may be activated by either the source or the destination unit.

Source-initiated strobe for data transfer



(a) Block Diagram



(b) Timing Diagram

Q.98 Write short notes on UART. **(4)**

Ans:
The transmitter accepts an 8-bit character from the computer 4 proceeds to send a serial 11-bit message into the printer line. The receiver accepts a serial 11-bit message from the keyboard line and forwards the 8-bit character code to computer. Integrated circuits are available which are specifically designed to provide the interface between computer and teletype or any other similar interactive terminal. Such circuit is called asynchronous communication interface or universal asynchronous receiver-transmitter (UART).

Q.99 State and verify the two DeMorgan's theorems by means of truth tables. **(7)**

Ans: DeMorgan's Law:

1) $\overline{AB} = \overline{A} + \overline{B}$

2) $\overline{A + B} = \overline{A} + \overline{B}$

1)

AB	A.B	$\overline{A.B}$	$\overline{A}$	$\overline{B}$	$\overline{A} + \overline{B}$
0 0	0	1	1	1	1
0 1	0	1	1	0	1
1 0	0	1	0	1	1
1 0	1	0	0	0	0

2)

AB	A+B	$\overline{A + B}$	$\overline{A}$	$\overline{B}$	$\overline{A}.\overline{B}$
0 0	0	1	1	1	1
0 1	1	0	1	0	0
1 0	1	0	0	1	0
1 1	1	0	0	0	0

Q.100 Obtain the truth table for the function F = AB' + B'C + A'C **(7)**

Ans:

A B C	$\overline{B}$	$A\overline{B}$	$\overline{B}C$	$\overline{A}$	$\overline{A}C$	$A\overline{B} + \overline{B}C$	$A\overline{B} + \overline{B}C + \overline{A}C$
0 0 0	1	0	0	1	0	0	0
0 0 1	1	0	1	1	1	1	1
0 1 0	0	0	0	1	0	0	0
0 1 1	0	0	0	1	1	0	1
1 0 0	1	1	0	0	0	1	1
1 0 1	1	1	1	0	0	1	1
1 1 0	0	0	0	0	0	0	0
1 1 1	0	0	0	0	0	0	0

Q.101 Obtain the simplified Boolean functions of the full-adder in sum of products form and draw the logic diagram using NAND gates. **(7)**

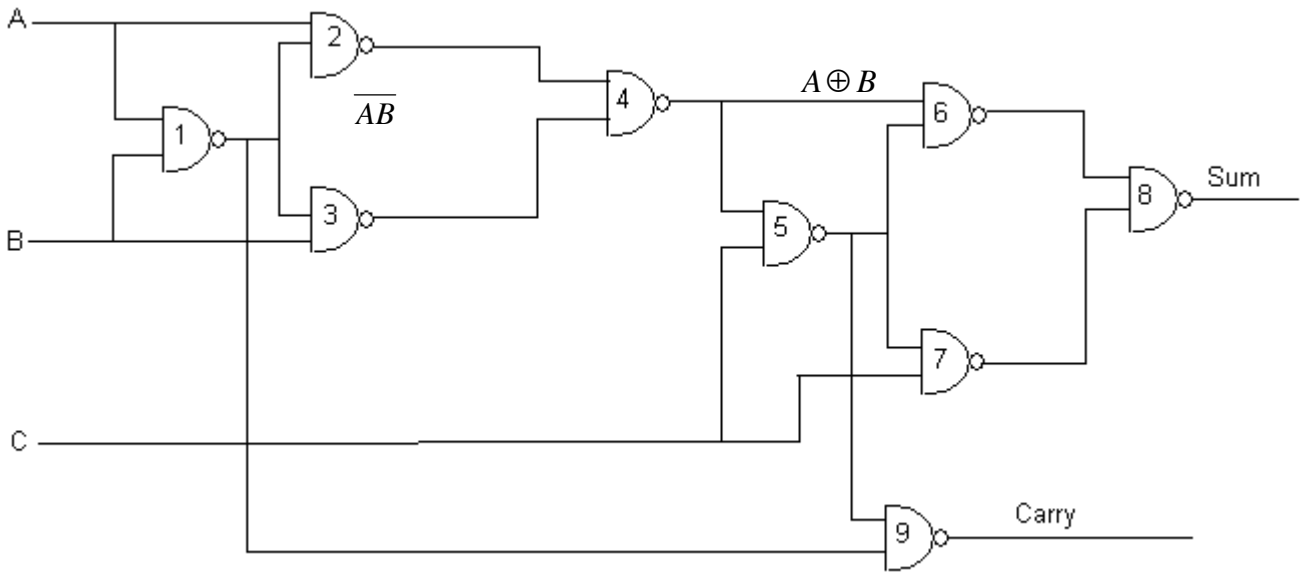
Ans:

$$\begin{aligned} A \oplus B &= \overline{A}B + A\overline{B} \\ &= \overline{\overline{A}B + A\overline{B}} \\ &= \overline{\overline{\overline{A}B}.\overline{A\overline{B}}} \\ &= \overline{A.\overline{\overline{A}B}.\overline{A\overline{B}}.B} = X \end{aligned}$$

This is realized by NAND gates 1, 2, 3 and 4 in Fig.3. Similarly, gates 5, 6, 7, 8 realize $X \oplus C = Sum$. Such a realization of exclusive OR function requires minimum number of NAND gates.

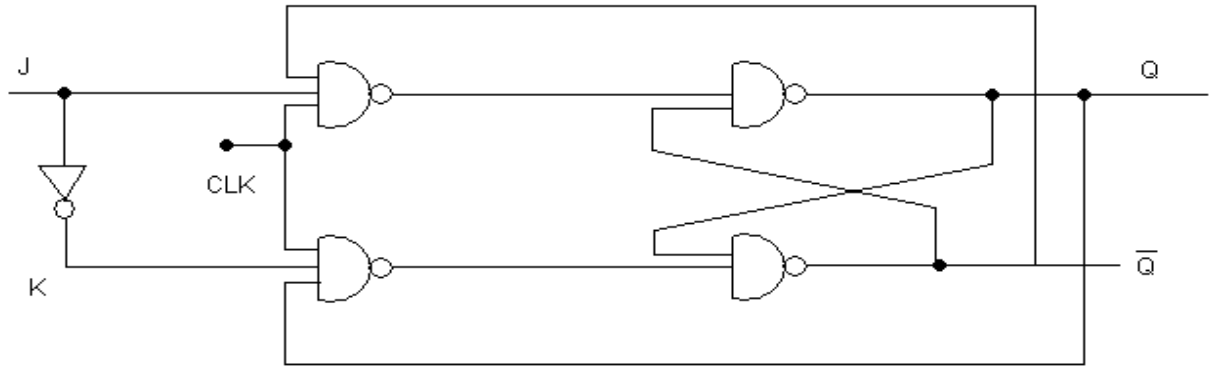
$$\begin{aligned} \text{Carry} &= AB + BC + AC \\ &= AB + BC + AB + AC \\ &= B(A + C) + A(B + C) \\ &= B(A + \overline{A}C) + A(B + \overline{B}C) \\ &= AB + \overline{A}BC + A\overline{B}C \\ &= \overline{AB + (A \oplus B).C} \\ &= \overline{\overline{AB + (A \oplus B).C}} \\ &= \overline{\overline{AB}.\overline{(A \oplus B).C}} \end{aligned}$$

Since $\overline{AB}$ and $\overline{(A \oplus B).C}$ are already available, carry can be implemented by an additional NAND gate no.9.



Q.102 Show that a JK flip-flop can be converted to a D flip-flop with an inverter between the J & K inputs. **(7)**

Ans:



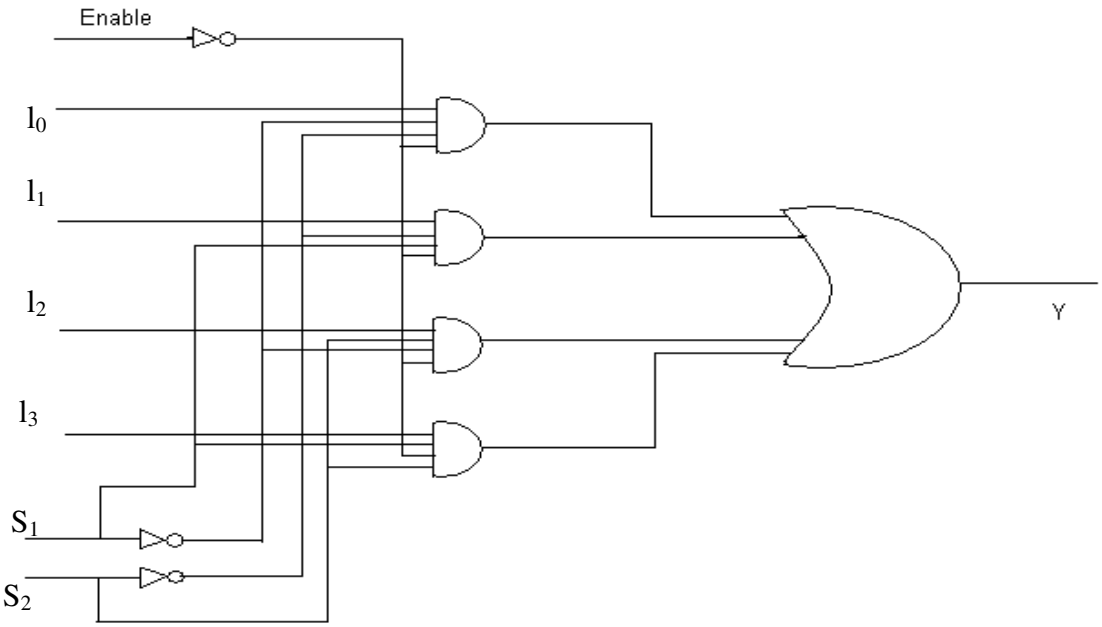
Initial Q = 1

$\overline{Q}$ = 0

CLK	J	Q	$\overline{Q}$
1	0	0	1
1	1	1	0

Q.103 Draw the logic diagram of a 4 by 1 multiplexer with an enable input. **(7)**

Ans: 4 : 1 MUX



Q.104 What are the two instructions needed in the basic computer to set the E register to 1? **(2)**

Ans:

1. Branch Instruction
2. Executing instruction

Q.105 What is the difference between an immediate, a direct, and an indirect address instruction? How many references to memory are needed for each type of instruction to bring an operand into a processor register? **(7)**

Ans:

Immediate → directly supplying the operand value eg. MVI A 05 : 05H is stored in register A → 0 Mem Ref.

Direct → directly giving memory address of operand. Eg. LDA (addr) : contents of memory location (addr) are stored in register A. 1 Mem Ref.

Indirect → giving the address of location where address of operand is stored 2 Mem Ref.

8085 has "register indirect" instructions which fetches data from memory location whose address is presented in (BC). (DE) or (HL) register pairs. It requires one memory reference.

Q.106 List the differences between a subroutine call and an interrupt. **(5)**

Ans:

Subroutine call: A call subroutine instruction consists of an operation code together with an address that specifies the beginning of the subroutine. The instruction is executed by performing two operations:

(1) the address of the next instruction available in the program counter (the return address) is stored in a temporary location so the subroutine knows where to return, and

(2) control is transferred to the beginning of the subroutine. The last instruction of every subroutine, commonly called return from subroutine**,** transfers the return address from the temporary location into the program counter. This results in a transfer of program control to the instruction whose address was originally stored in the temporary location.

Program Interrupt: The concept of program interrupt is used to handle a variety of problems which arise out of normal program sequence. Program interrupt refers to the transfer of program control from a currently running program to another service program as a result of an external or internal generated request. Control returns to the original program after the service program is executed.

The interrupt procedure is, in principle, quite similar to a subroutine call except for three variations:

1. The interrupt is usually initiated by an internal or external signal rather than from the execution of an instruction
2. The address of the interrupt service program is determined by the hardware rather than from the address field of an instruction.

An interrupt procedure usually stores all the information necessary to define the state of the CPU rather than storing only the program counter.

Q.107 Convert the following arithmetic expressions into reverse polish notation:
(i) A + B + C
(ii) A * B/C + D.
Show the intermediate steps.

Ans:
i) A + B + C → AB+ + C
A B+ C +

ii) A*B/C + D = (A*B/C) + D →
(ABC/*) + D →
ABC / * D +

Q.108 What is the condition that must exist in an index-mode instruction to make it the same as a register indirect-mode instruction? **(2)**

Ans:
Index – Mode Instruction: In this mode the content of an index register is added to the address part of the instruction to obtain the effective address. The index register is special CPU register that contains an index value. The address field of the instruction defines the beginning address. The distance between the beginning address and the address of the operand is the index value stored in the index register. Any operand in the array can be accessed with the same instruction provided the index register contains the correct index value.

Register Indirect – Mode: In this mode the instruction specifies a register in the CPU whose contents give the address of the operand in memory. In other words, the selected register contents the address of the operand rather than the operand itself. If the memory address is obtained by adding address part of instruction to the contents of the index register (as in index-mode instruction) and is stored in a CPU register, then a register-indirect-mode instruction can be used to access the operand. A reference to the register is then equivalent to specifying a memory address. The advantage of a register – indirect mode instruction is that the address field of the instruction uses fewer bits to select a register than would have been required to specify a memory address directly. This approach combines the register – indirect and index-mode instructions.

Q.109 Describe the different categories of instructions available in the instruction set of the basic computer. **(6)**

Ans:

The basic computer has three different instruction code formats. The operation part of the instruction contains three bits; the meaning of the remaining thirteen bits depends on the operation code encountered.

A memory – reference instruction uses the last 12 bits to specify an address and the first bit to specify the mode as illustrated in Fig 1(a).

A register – reference instruction, shown in Fig 1(b), specifies an operation on or a test of the AC or E register. An operand from memory is not needed; therefore, the last 12 bits are used to specify the operation or test to be executed. A register – reference instruction is recognized by the operation code 111 with 0 in the first bit of the instruction.

Similarly, an input – output instruction does not need a reference to memory and is recognized by the operation code 111 with a 1 in the first bit of the instruction. The remaining 12 bits are used to specify the type of input – output operation or test performed. Note that the first bit of the instruction code is not used as a mode bit when the last 12 bits are not used to designate an address. The input-output instruction format is shown in Fig 1(c).

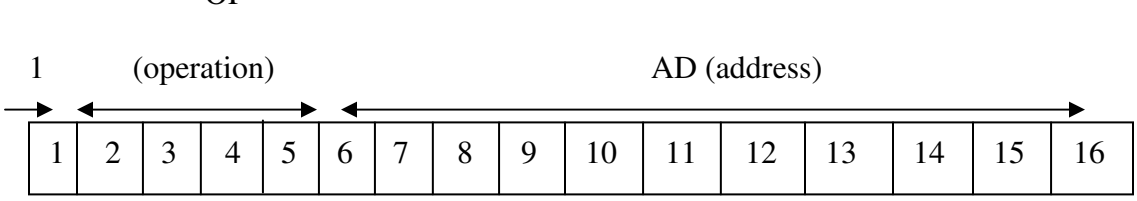


Fig 1(a) Memory – reference instruction

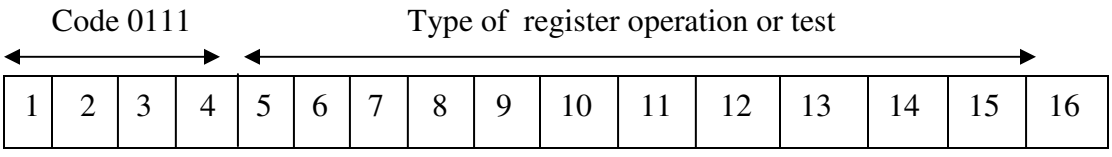


Fig 1(b) Register – reference instruction

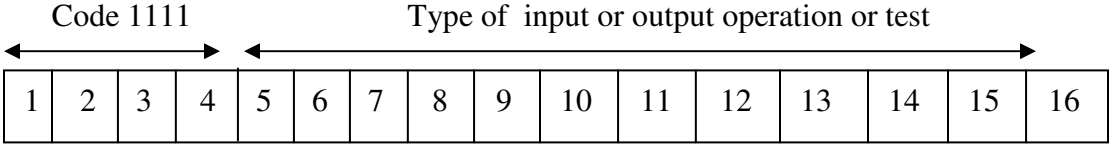


Fig 1(c) Input - output instruction

Q.110 List four alternatives for achieving a conditional branch operation in a control memory. **(7)**

Ans:

A computer that employs a microprogrammed control unit will have two separate memories: a main memory and a control memory. The main memory is available to the user for storing his programs. The contents of main memory may alter when the program is executed and every time the program is changed, the user's program in memory consists of machine instructions and data. The control memory holds a fixed microprogram that cannot be altered by the occasional user. The micoprogram consists of microinstructions that specify various internal control signals for execution of register micro – operations. Each machine instruction initiates a series of microinstructions in control memory. These microinstructions generate the micro – operations to

(1) fetch the instruction from main memory;
(2) evaluate the effective address of the operand;
(3) execute the operation specified by the instruction; and
(4) return control to the beginning of the fetch cycle to repeat the sequence again for the next instruction.

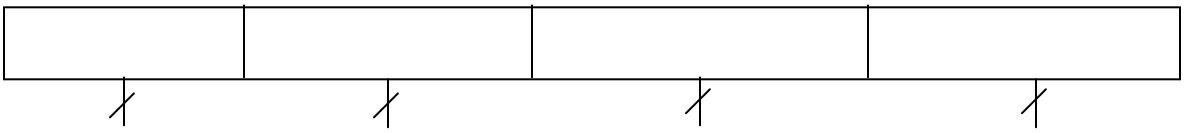
Mnemonic	**Branch condition**	**Tested condition**
BZ	Branch if zero	Z = 1
BNZ	Branch if not zero	Z = 0
BC	Branch if carry	C = 1
BNC	Branch if no carry	C = 0
BP	Branch if plus	S = 0
BM	Branch if minus	S = 1
BV	Branch if overflow	V = 1
BNV	Branch if no overflow	V = 0
Unsigned compare conditions (A – B)		
BHI	Branch if higher	A > B
BHE	Branch if higher or equal	A ≥ B
BLO	Branch if lower	A < B
BLOE	Branch if lower or equal	A ≤ B
BE	Branch if equal	A = B
BNE	Branch if not equal	A ≠ B

Signed compare conditions (A – B)		
BGT	Branch if greater than	A > B
BGE	Branch if greater or equal	A ≥ B
BLT	Branch if less than	A < B
BLE	Branch if less or equal	A ≤ B
BE	Branch if equal	A = B
BNE	Branch if not equal	A ≠ B

Q.111 Show how a microprogrammed control unit works. List three advantages over conventional control unit. **(7)**

Ans:
Microprogrammed Control unit uses a fast memory (called Control ROM) to store microinstructions. Each microinstruction generates control signals to cause one or more micro-operations. A set of microinstructions make a microprogram, which corresponds to an instruction, and is used to FETCH, DECODE and execute the instructions.

Data in a microinstruction of the Control ROM may be organized as shown below



To execute an instruction, following steps are involved

1. Requisite control signals are turned ON
2. If the jump condition is FALSE, next microinstruction in sequence is executed
3. If jump condition is TRUE, microprogram jumps to address specified in the last field.

Q.112 Describe either source-initiated or destination initiated data transfer using handshaking protocol. What advantage does handshaking provide over strobe based data transfer? **(7)**

Ans: Refer page 395-397 from Morris mano (3rd Edition)

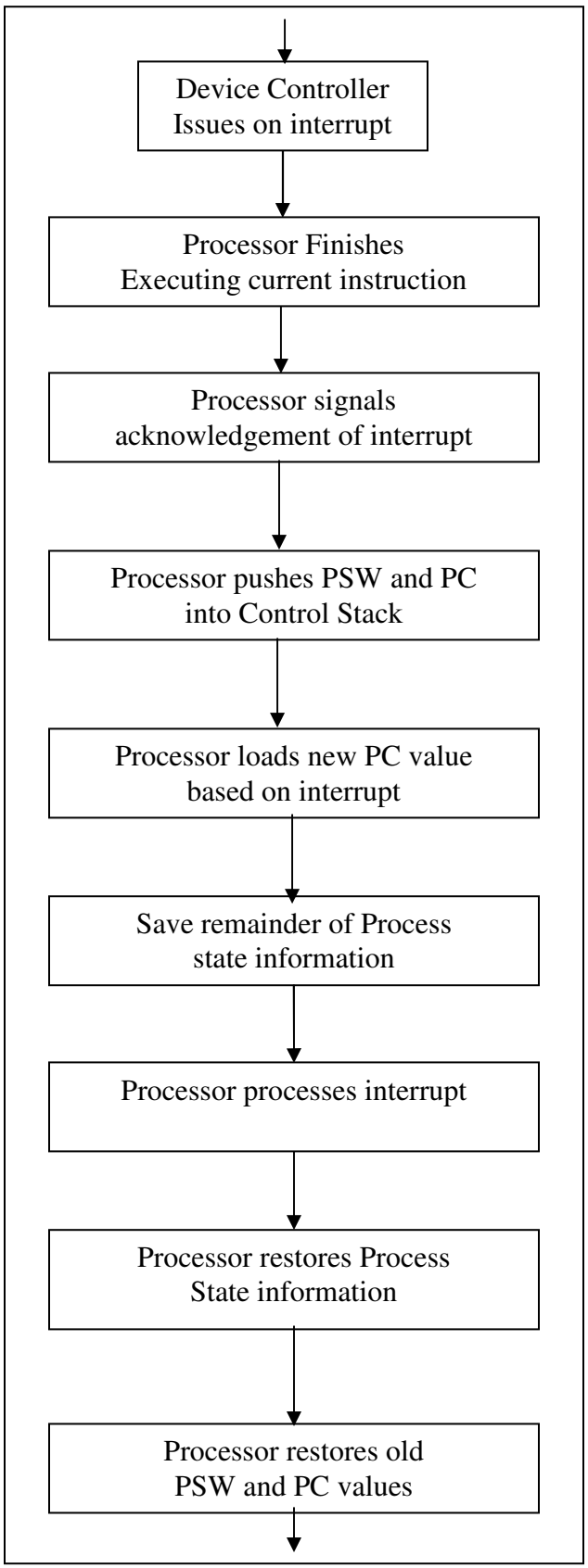
Q.113 Why does I/O interrupt make more efficient use of the CPU? **(7)**

Ans: I/O Interface:
Input – output (I/O) interface provides a method for transferring binary information between internal storage, such as memory and CPU registers, and external I/O devices. Peripherals connected on – line to a computer need special communication links for interfacing them with the central processor. The purpose of the interface and the communication link is to resolve the differences that exist between the central computer and each peripheral. The major differences are:

1. Peripherals are electromechanical devices and their manner of operation is different from the operation of the CPU and memory which are electronic devices.
2. The data transfer rate of peripherals is much slower then the transfer rate in the central computer.
3. The operation of the peripherals must be synchronized with the operation of the CPU and memory unit.
4. Data formats in peripherals differ from the word format in the central processor.
5. The operation of each peripherals must be controlled so as not to disturb the operation of the central computer and other peripherals connected to the system. The I/O interrupts the CPU only when it requires to transfer data, thus making an efficient use of CPU time.

Q.114 Explain the sequence of operations that take place in an interrupt driven I/O transfer. **(8)**

Ans:



Q.115 40 numbers are stored in memory starting from location START. Write a program in assembly language to count number of positive elements, negative elements and number of zeros. **(14)**

Ans:

```
lxi h, (START+44)h
mvi m,00h

lxi h, (START+49)h
mvi m,00h

lxi h, (START+53)h
mvi m,00h

lxi h,STARTh
mvi c,28h
mvi a,00h

loop:
dcr c
cmp c
jz ending
cmp m
jz zero
jn pos
jp neg

neg:
push h
lxi h, (START+44)h
inr m
pop h
inx h
jmp loop

pos:
push h
lxi h, (START+49)h
inr m
pop h
inx h
jmp loop

zero:
push h
lxi h, (START+53)h
inr m
```

```
pop h
inx h
jmp loop

ending:
hlt
```

Q.116 Give the differences between Demultiplexers and multiplexers. **(6)**

Ans:
Demultiplexers: A decoder with one or more enable inputs can function as demultiplexer. A demultiplexer is a digital function that receives information on a single line and transmits this information in one of $_2$n possible output lines.

Multiplexers: A multiplexer is a digital function that receives binary information 2n lines and transmits information on a single output line.

Q.117 Differentiate between serial and parallel transfer of information. **(3)**

Ans:
Parallel Transfer: Information transfer from one register to another can be performed either in parallel or in serial. Parallel transfer is simultaneous transfer of all bits from the source register to the destination register and is accomplished during on clock pulse.

Serial Transfer: For serial transfer, both the source and destination registers are shift-registers. The information is transferred one bit at a time by shifting the bits out of the source register into destination register.

Q.118 Define a bus. **(3)**

Ans:
Bus: A group of wires through which binary information is transferred among registers is called a bus.

Q.119 How is ring counter useful in timing sequence in digital computers? **(4)**

Ans:
A ring counter is a circular shift register with only one flip-flop being sent at any particular time, all others are cleared. The single bit is shifted from one flip-flop to the other to produce the sequence of timing signals.

Q.120 How many references to memory are required to bring an operand into a processor register? **(4)**

Ans:

No. of references to memory for obtaining the operand in to a processor register depends on the type of the instruction.

For immediate address instruction –One i.e for fetching operand part of the instruction.

For direct address instruction –Two

For indirect address instruction –Three

In all the cases, the first memory reference is that of fetching operand part of the instruction.

Q.121 Define the following:
(i) Logical shift (ii) Circular shift. **(6)**

Ans:
Refer Section 4-6 page 114 from Morris mano (3rd Edition)

Q.122 Define the following:

(i) Fetch Cycle. (ii) Indirect Cycle. **(8)**

Ans:
i) Fetch Cycle: An instruction is read from memory during the instruction fetch cycle. The fetch cycle is recognized by variable c_0. The four timing signals that occur during the cycle initiate the sequence of micro operations for the fetch cycle. The address, which is in PC, is transferred into MAR. The memory reads the instruction and places it in MBR. At the same time, the program counter is incremented by one to prepare it for the address of the next instruction.

ii) Indirect Cycle: The indirect cycle is recognized by the variable c_1. During this cycle, control reads the memory word where the address of the operand is to be found.

Q.123 Write down the three types of CPU organization with the help of examples. **(8)**

Ans:
Refer page 258,259 from Morris mano (3rd Edition)

Q.124 Explain the concept of program interrupt with suitable examples. **(8)**

Ans:
Refer page 281,282 from Morris mano (3rd Edition)

Q.125 What are pseudo-instructions and define some common symbols used in assembly language.

(8)

Ans:

Refer page 182 from Morris mano (3rd Edition)

Some common symbols used in assembly language are as follows:
AND
ADD
STA
CLA
INP
OUT etc.

Q.126 Explain Binary counters and Binary counter sequence in detail. **(8)**

Ans: Refer Section 2-6 page 55-56 from Morris mano (3rd Edition)

Q.127 Write down an algorithm for adding and subtracting numbers in signed – 2's complement representation. **(8)**

Ans:

Subtract	ADD
↓	↓
Minuend in AC	Augend in AC
Subtrahend in BR	Addend in BR
↓	↓
AC ← AC + BR + 1	AC ← AC + BR
.V ← overflow	V ← overflow
↓	↓
END	END

Q.128 Discuss how Booth's algorithm treats positive and negative multipliers uniformly. **(8)**

Ans:
Refer page 345-346 from Morris mano (3rd Edition)

Q.129 Give the difference between priority interrupt and daisy chain priority interrupt. **(8)**

Ans:
Refer page 410-411 from Morris mano (3rd Edition)

Q.130 Define the following:
(i) Control command. (ii) Test command.
(iii) Data-output command. (iv) Data-input command. **(8)**

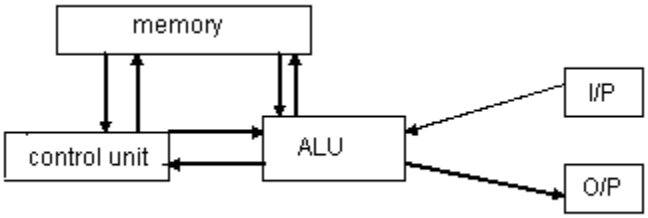
Ans:
Refer page 389 from Morris mano (3rd Edition)

Q.131 Explain Von Neumann Architecture. What are its drawbacks ? **(3+3)**

Ans:

The von Neumann architecture allows instructions and data to be mixed and stored in the same memory module and the contents of this memory are addressable by location only. The execution occurs in a sequential fashion.

The Von Neumann machine had five basic parts:- (i) Memory (ii) ALU (iii) Program control unit (iv) Input equipment & (v) output equipment.



Disadvantages:- The circuit speed can't be increased indefinitely. The most ironic aspect of the quest for even increasing speed is that most of the transistors are idle all the time. The traditional structure of a computer with a single CPU that issues sequential requests over a bus to a memory that responds to one request at a time has became known as the von-Neumann bottle neck.Modern computer predominantly follow the vonNeuman architecture, but use some elements of Harvard architecture .

Q.132 What is meant by Addressing Mode? Explain at least five different Addressing Modes with an example. **(2+6)**

Ans:

When a processor accesses memory, to either read or work data, it must specify the memory address it needs to access. An assembly language instruction may use one of several addressing modes to generate this address. These one –

(i) Direct mode
(ii) Indirect mode
(iii) Register direct & register indirect mode
(iv) Immediate mode
(v) Implicit mode
(vi) Relative mode
(vii) Index mode & Base address mode.

Direct Mode:-In this mode,the instruction includes a memory address
Ex. LDAC 5,
Data from memory location 5 is stored in accumulator.

Indirect Mode:-The address specified in the instruction is not the address of the operand. It is the address of a memory location that contains the address of the operand.

Ex. LDAC @ 5.

First it retrieve the content of memory location 5 say 10. Then the CPU goes to location 10, reads the contents of that location and loads the data into the CPU.

Register Direct & Register indirect modes:-

Register modes work the same as direct & indirect modes discussed above, except they do not specify a memory address. Instead they specify a register.

Example: LDAC R $\Rightarrow$ Content of Reg. 'R' is copied in accumulator

LDAC (R) $\Rightarrow$ The content of Reg. 'R' gives the memory address whose content is to be copied in to accumulator.

Immediate Mode:- The operand specified in the instruction is not an address, it is the actual data to be used.

Example:LDAC # 5, It moves the datavalue 5 to accumulator.

Implicit Mode:- It does not explicitely specify an operand.The instruction implicitely specify the operand because it always applies to a specific register.

Example: CLAC $\Rightarrow$ Clear accumulator

CMC $\Rightarrow$ Complement accumulator.

Q.133 Discuss the general characteristics of memory systems. What is the use of virtual memory and discuss its concept? **(2+2+6)**

Ans:

Memory is required in a computer system to store program and datas. The memory is of different types primary memory and secondary memory.

ROM, RAM, EPROM are belongs to primary memory. Where as the hard disk, magnetic tape, magnetic dram, floppy disk memories are known as secondary memory.

Primary memories are also learn as main memory. The memory unit that communicates directly with the CPU is called main memory. The devices that provides back up storage are called auxiliary memory / secondary memory.

The time required to access a memory unit for reading & writing is depends on the technology on which the memory unit is built. Primary memories are more factor than secondary memory. But they are costlier than secondary memory. To minimize the

cost and to improve the performance a large auxiliary memory is used along with a small quantity of main memory. When programs or data required by CPU is not available in main memory. Then the I/O processor brings those information's from auxiliary memory.

Virtual memory is a concept used in some large computer systems that permits the user to construct programs as though a large memory space were available equal to the totality of auxiliary memory. Each address that is referenced by the CPU goes through an address mapping from the so called virtual address to a physical address in main memory. Virtual memory is used to give programmer the illusion that they have a large memory at their disposal, even though the computer actually has a relatively small main memory. Virtual memory system provides a mechanism for translating program generated addresses in to correct main memory locations. This is done dynamically. The translation or mapping is handled automatically by the hardware by means of mapping table.

Q.134 How many 256×8 ROM chips are required to produce a memory capacity of 4000 bytes? How many address lines are required to access the 4000 bytes? How many of these addresses will be common to all these chips? **(1+1+2)**

Ans:

4000 bytes ≈ 4K bytes = $2^2\ 2^{10}$ bytes = 2^{12} bytes.

ROM size = 256 byte = 2^8 bytes.

No. of ROM chip req. = $\frac{2^{12}}{2^8} = 2^4 = 16$.

No. of add. Lines required to access 4 K bytes = 12.

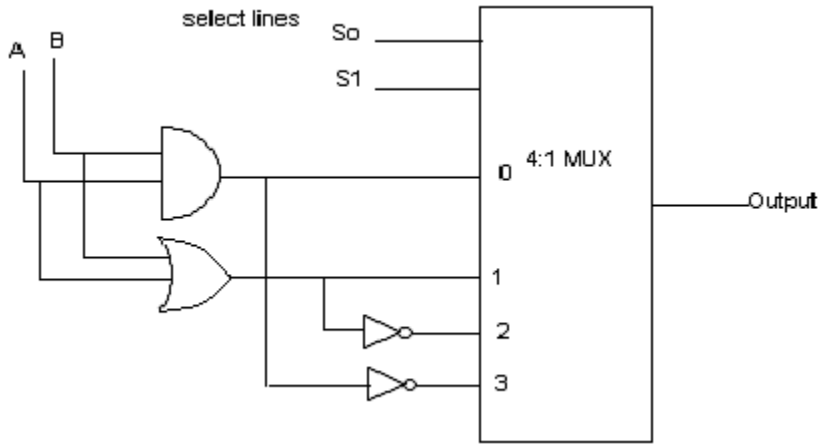
256 No. of addresses will be common to all these chips.

Q.135 Draw the block diagram of Arithmetic Logic Unit (ALU) to perform the following operations. (use MUX) (Assume the length of the data).

(i) AND (ii) OR

(iii) NOR (iv) NAND **(8)**

Ans:



Q.136 Show the function table for above design and explain. **(6)**

Ans:

Function Table

S_1	S_0	*O/p*	*Function realized*
0	0	$A \wedge B$	AND
0	1	$A \vee B$	OR
1	0	$\overline{A \vee B}$	NOR
1	1	$\overline{A \wedge B}$	NAND

Q. 137 What is meant by pipelining? Why do we require instruction pipelining? Explain its working procedure. Discuss the pipeline performance measures. **(8)**

Ans:

Pipe lining is a technique of decomposing a sequential process into sub-operations, with each sub process being executed in a special dedicated segment that operates competently with all other segments. It can be visualized as a collection of processing segments through which binary information flows. Each segment performs partial processing dictated by the way the task is partitioned. The result obtained from the computation in each segment is transferred to the next segment in the pipeline. The final result is obtained after the data have passed through all segments. The name

'Pipeline' implies a flow of information analogous to an industrial assembly line. Its characteristic is that several computations can be in progress in distinct segments at the same time.

An instruction pipeline operates on a stream of instruction by overlapping the fetch, decode & execute phases of the instruction cycle. The pipe line technique provides a factor operation over a pinely serial sequence even through the maximum theoretical speed is never fully achieved.

Working of Instructional Pipelining:

An instructional pipeline reads consecutive instructions from memory while previous instructions are being executed in other segments. This causes the instruction fetch & execute phases to overlap and perform simultaneous operations. When branch instruction is encountered, pipeline must be emptied and all the instructions that have been read from memory after the branch instruction must be discarded.

Instruction pipeline may be divided on to four segments such as-

1. Fetch the instruction from memory.
2. Decode instruction & calculate effective address.
3. Fetch operand from memory.
4. Execute instruction

Pipeline performance measure is in terms of time taken in executing a program. If a non-pipe line unit that performs a given task and takes a time equal to 't_n' to complete. The speed up of a pipe line processing over an equivalent non-pipe line processing is defined by the ratio: $S = \frac{nt_n}{(K + n - 1)t_p}$

Where K = No. of segments in pipe line.

T_p = Time taken by each segment to process a sub-operation.

n = No. of tasks.

Q.138 What are the different conflicts that will arise in pipeline (elaborate)? How do you remove the conflicts? **(3+3)**

Ans:

There are three major difficulties that causes the instruction pipe line to deviate from its normal operation.

1. Resource conflicts.
2. Data dependency
3. Branch difficulties.

Resource conflicts : It is caused by access to memory by two segments at the same time. Most of these conflicts can be resolved by using separate instruction & data memories.

Data Dependency: This conflicts arises when an instruction depends on the result of a previous instruction but this result is not yet available.

Branch difficulties : It arises from branch and other instructions that change the value of PC.

Address dependency : It may occur when an operand address can not be calculated because the information needed by the addressing mode is not available. For example, an instruction with register indirect mode cannot proceed to fetch the operand, if the previous instruction is loading the address into the register. So the operand access to memory must be delayed until the required address is available.

Handling data dependency :-

The method used for resolving data conflicts are

(i) Hardware Inter lock : It detects instructions whose source operands are destinations of instruction father up in the pipe line. Detention of this situation causes the instruction whose source is not available to be delayed by sufficient lock cycles to resolve the conflict.

(ii) Operand forwarding : It was special hardware to detect a conflict and then avoid it by routing the data through special paths between pipeline segments. For example, instead of transferring as ALU result into a destination register, the hardware checks the destination operand and it if is needed as a source in the next instruction, it passes the result directly into the ALU, by passing the register file.

(iii) Delayed Load : The responsibility for solving data conflicts problems is given to compiler that translates the high level programming language into a machine language program. The compiler for such computers is designed to detect a data conflict and re-order the instructions by inserting no-operation instructions.

Similarly for handling **branch address conflicts,** The following methods are used

(1) Pre-fetch target instruction.

(2) Branch target buffer

(3) Loop buffer

(4) Branch Prediction

(5) Delayed branch-Employed in most RISC processes.

Q.139 Explain how addition and subtraction of floating point operations are carried out with an example and show the algorithm / flow chart. **(4+4)**

Ans:

Floating point addition and subtraction are carried out in following steps.

(i) Check for zeros

(ii) Align the mantissas

(iii) Add or subtract the mantissas

(iv) Normalize the result.

Example:

Let one operand is in AC $\rightarrow 0.23 \times 10^2 = A \times 10^a$

The other operand is in BR $\rightarrow 0.15 \times 10^1 = B \times 10^b$

Ist Step:-

Check for zero.

AC is not zero

BR is also not zero

It AC = 0. Than the answer is BR in case of addition & is equal to BR with opposite sign in case of subtraction.

If BR = 0, Than the answer is AC for both addition & subtraction.

Step-2:- Allign the mantissas:-

This is done in order to make the exponent part of both operands same before addition or subtraction.

If a < b, than shift right 'A' and increment a till a = b.

If a > b, shift right 'B' and increment b till a = b.

In this case a = 2, b = 1, then a > b.

Shr B and b = b+1 gives B = 0.015, b = 2. so now a = b.

Step-3 Add or sub the mantissas.

Now A = 0.23 B = 0.015

For addition A = A+B = 0.245. For subtraction A = A-B = 0.215

Step-4:- Normalise the result.

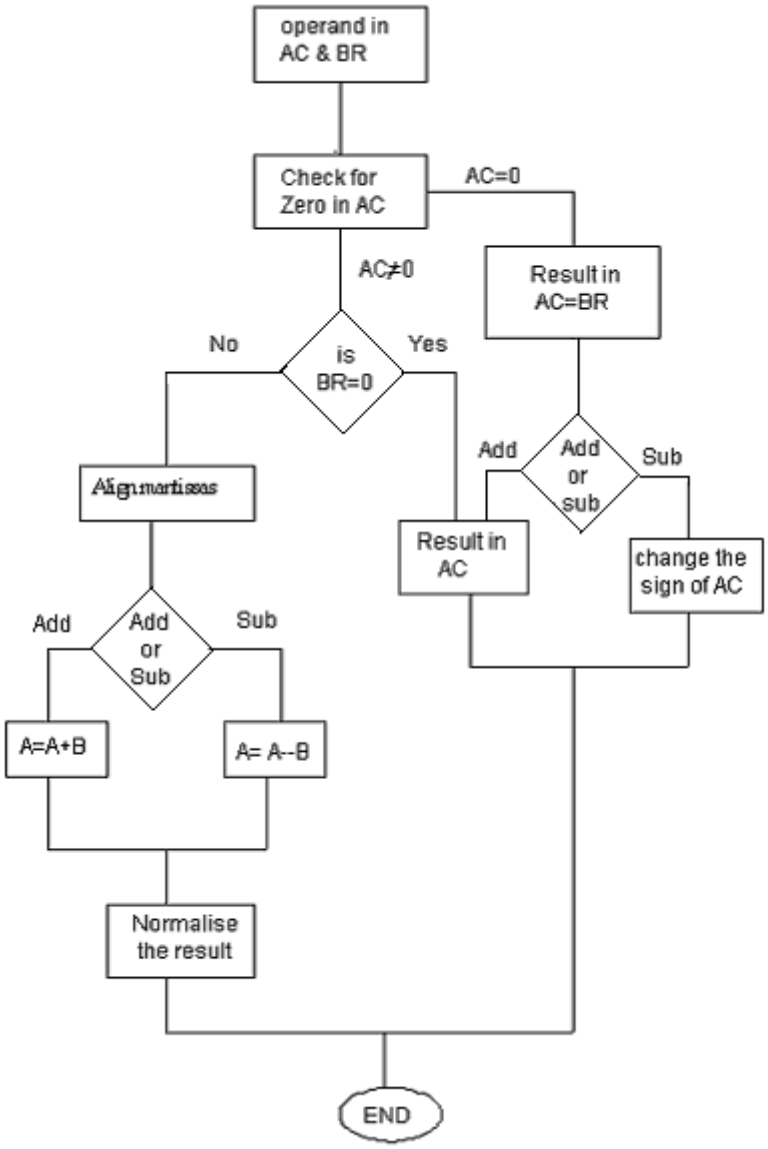
For addition the result is 0.245×10^2

For subtraction, the result is 0.215×10^2

Both are in normalized state. If not than mantissas needs to be shifted left & exponent is decremented until a non-zero digit appears in the first position of mantissas.

Example - If the result is 0.0035×10^5

Than after shifting left two times to mantissas & documenting exponent by e, we get the normalized result as 0.35×10^3.



Flow Chart:-

Q.140 What is meant by normalization? Why we do normalization of floating point numbers? **(6)**

Ans:

When two normalized mantissas are added, the sum may contain an over flow digit. An over flow can be corrected easily by shifting the sum once to the right and incrementing the exponent.

Example 0.35×10^5

$$\begin{array}{r} +\ 0.73 \times 10^5 \\ \hline 1.08 \times 10^5 \end{array}$$ over flow occur.

After normalization it becomes. 0.108×10^6.

Similarly when two normalized mantissas are subtracted, the result may contain most significant zero as shown below.

$$\begin{array}{r} 0.56 \times 10^5 \\ -\ 0.53 \times 10^5 \\ \hline 0.03 \times 10^5 \end{array}$$ under flow occurs.

To normalize the under flow condition, mantissas is shifted left and decrement the exponent until a non-zero digit appears in the first position.

$$0.03 \times 10^5 \xrightarrow{\text{normalised}} 0.3 \times 10^4$$

Normalization means to have the most significant digit of mantissa to be non-zero.

It is done in case of floating point addition and subtraction to over come overflow and underflow as discussed in above example.

Q.141 Explain the role of stacks for handling interrupts. **(4)**

Ans:

When an interrupt comes to the processor, the processor recognize it only if the interrupt enable FF is ON i.e. IEN = 1. The processor complete the execution of the instruction currently giving on and sends a acknowledgement signal the I/O device generating interrupt signal. On receiving the interrupt acknowledge signal from CPU.

The hardware for interrupt handling puts out the address of the memory where the interrupt service routine (ISR) is stored. If the program counter content of the processor w.r.t. its first program and other register contents associated with the first program along with flag register, accumulator are not stored properly, than the first program can't be continued after ISR being executed. So it is essential for CPU to save the contents of PC, AC & other processor register in some designated memory location. For this work, stack memory is used by most of the processor. As the stack works on LTFO principle, so it is convenient to save register contents before servicing the I/O device interrupting the CPU and also it is convenient to POP up all register information's sequentially after the processor returns back to main program at the end of ISR.

In this way stack memory is used in handling the interrupts by the processor in a interrupt drives circumstance.

Q. 142 What is the sequence of steps that will take place when an interrupt occurs? **(6)**

Ans:

When an interrupt is recognized the following steps are carried out by the processor-

1. Complete the execution of current instruction.
2. Document the stack pointer. SP ← SP-1.
3. Save the program counter content in to stack

   M[SP] ← PC.
4. Save the contents of processor registers.
5. Send interrupt acknowledgement signal to interrupt controller.
6. Transfer the vector address of the corresponding interrupt to program counter.
7. Disable the interrupt enable FF, so that other interrupt signal can't be recognized till the end of ISR.
8. Go to fetch the first instruction of ISR & execute the ISR.

When the processor came access the return instruction of ISR, it dies the following-

1. activate the interrupt enable FF to allow it to be interrupted.
2. restore contents of processor registers.
3. restore the return address saved in stack in to PC.
4. continue execution of main program.

Q.143 Define hit ratio and explain its significance. **(4)**

Ans:

When the CPU refers to memory, and finds the word in cache, it is said to produce a hit. If the word is not found in cache, it is in main memory and it counts as a Miss. The performance of cache memory is frequently measured in terms of a quality called hit ratio. The ratio of the number of hits divided by the total CPU reference to memory (hits + misses) is called hit ratio.

Say, the memory access time for main memory = A_m & cache memory access time = A_c, If the hit ratio = 0.9, than the average memory access time = (0.1 A_m + 0.9 A_C).

The performance of cache memory improves if the hit ratio is high.

Q.144 Show the hardware to be used for addition and subtraction of signed 2s compliment representation. Explain how it operates and how an over flow will be handled. **(14)**

Ans:

In signed 2's complement representation, the left most bit of a binary number represents the sign bit, '0' for + ve & 1 for –ve. If the sign bit is 1, the entire number is represented in 2's complement form.

Addition:- both the operands are added up along with the sign bit. A carry out of the sign bit position is discarded.

Subtraction

Take 2's complement form of the subtracted including the sign bit and add it to the minuend including the sign bit. A carry out of the sign bit position is discarded.

Example (–6) – (–13) = +7.

$-6 \rightarrow 1111\,1010$

$+13 \rightarrow 0000\,1101$

$\textcircled{1}\ \overline{0000\,0111} = +7$

discarded

Overflow:- During addition / subtraction of signed 2's complement representation, if the result occupies (η+1) digits, than an over flow occurred

$+70 \rightarrow 0100\,0110$

$+80 \rightarrow 0101\,0000$

$\overline{1001\,0110}$

↓

Sign bit has been changed

$-70 \rightarrow 1011\,1010$

$+80 \rightarrow 1011\,0000$

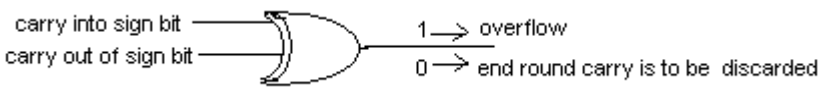
$\overline{-150}\quad \overline{1011\,0\,1010}$

↓

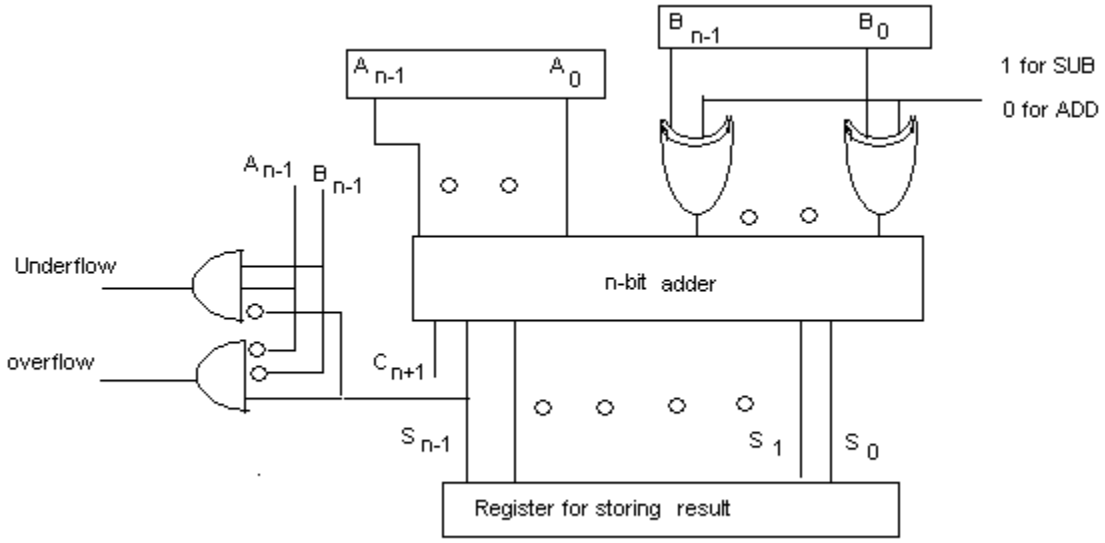
result is (η+1) bits.

In above two cases, overflow has occurred. It is not that the carry out of sign bit is to be discarded. Hence to distinguish the situation where end round carry is to be discarded & when it is an over flow, the following observations must be made.

If the carry in to the sign bit position is equal to the carry out of sign bit position than the end round carry is to be discarded & there is no overflow. Other wise if these two carry are not equal. Then it is considered to be overflow. Hence, the overflow condition can be checked as follows.



The hardware realization of signed 2's complement addition & subtraction is as given below.



Q.145 What is an instruction? What are the different parts of an instruction? Explain the significance of each part of an instruction with an example? **(8)**

Ans:

A computer instruction is a binary code that specifies a sequence of micro operations for the computer. Instruction codes together with data are stored in memory. The computer reads each instruction from memory and places it in a control register. The control then interprets the binary code of the instruction and proceeds to execute it by issuing a sequence of micro-operations.

op-code	Operand

→ Instruction code.

An instruction code consists of two ports.

(i) Opcode → operators part

(ii) Operands → Data / Address of data

The operation code of an instruction is a group of bits that define such operations as add, subtract shift, multiply etc.

The operand part of the instruction can be a data on which operation as specified by opcode is to be carried out or it can be a address of the memory where the data is available.

Example : **Add 450**, is an instruction where the opcode is **Add** i.e. the operation is to be carried out is addition of two operand. One operand is available in accumulator. The other one is available in the memory having address 450. The result of accumulator shall be stored in accumulator.

In case of indirect addressing mode, the instruction is **Add @ 450**. i.e. operand is in accumulatorhe other operand is to be obtained indirectly. The operand part of the instruction is 450, is the address of the memory location, where not the data but the address of the data is available. So after reading the memory location 450, it gets another address of memory location, where data is to be found.

If the Opcode part of the instructions is of n bits, than there can be 2^n no, of distinct operations in that computer. The bit length of operand part signifies the size of the memory. If the bit length of operand part is of K-bits, than the CPU can address 2^k memory words. So each part of the instruction has their own significance.

Q. 146 If a Computer has 128 operation codes and 512 k addresses, how many bits would be required for
(i) Single address instruction (ii) Two address instruction. **(6)**

Ans:

No. of Op codes = 128 = 2^7

Size of memory = 512 K = $2^8 \times 2^{10}$ bits.

(a) Size of single address instructions = 7 + 18 = 25 bits.

(b) Size of two address instruction = 7 + 2(18) = 7+36 = 43 bits.

Q. 147 What are the different modes of data transfer? Explain the DMA Controller with a block diagram. What is meant by block transfer? **(2+10+2)**

Ans:

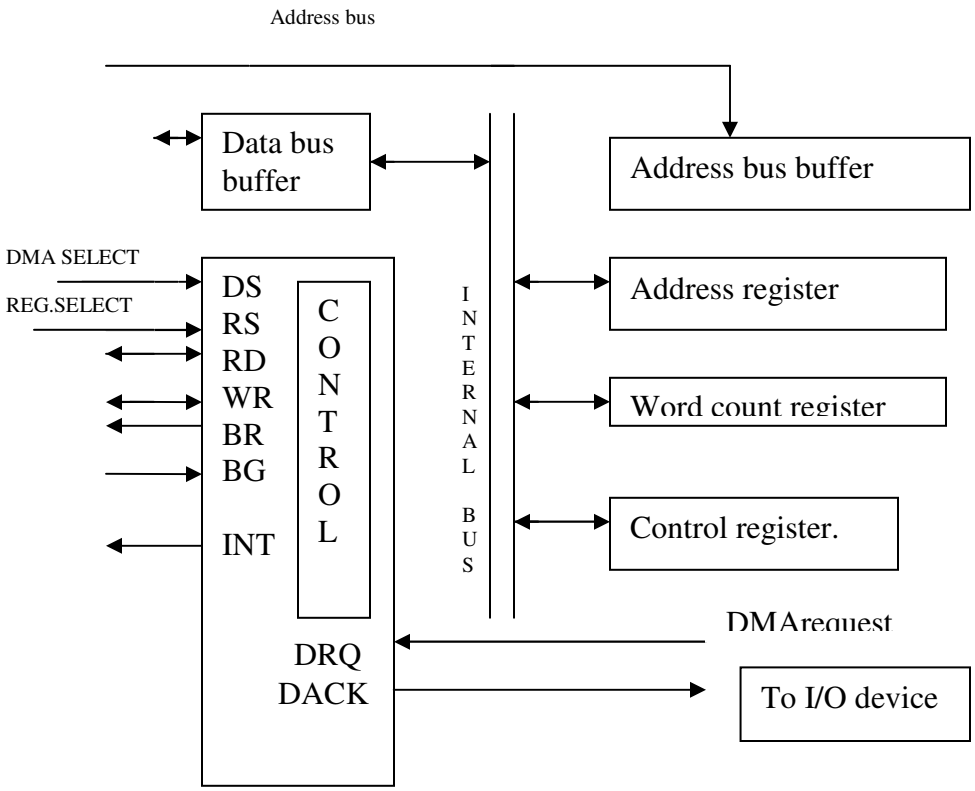
Different modes of data transfer are-

1. Programmed I/O
2. Interrupt initiated I/O
3. Direct memory access (DMA)

Direct memory access.

The transfer of data between a fast storage device such as magnetic disk and memory is often limited by the speed of the CPU. Removing the CPU from the path and letting the peripheral device manage the memory buses would improve the speed of transfer. This transfer technique is called direct memory access. Dining DMA controller takes over the buses to manage the transfer directly between I/O device & memory.

The block diagram of DMA controller is given below:-



DMA controller communicate with the CPU via the data bus and control lines. The registers in the DMA are selected by the CPU through the address bus by enabling the DS and RS inputs. BR input is used by the DMA controller to request the CPU to relinquish control of the buses. When this input is active, the CPU terminates the execution of the current instruction and places the address bus, data bus and the read and write lines in to a high impedance state. The CPU activates the bus grant (BG) output to inform the DMA controller that the buses are in the high impedance state. Now the DMA can conduct memory transfer without processor's intervention. The address register contains an address to specify the desired location of memory. The address register is incremented after each word that is transferred to memory.

The word count register holds the No. of words to be transferred. This register is decremented after each word transfer and is tested for zero internally.

The DMA controller is first initialized by CPU by sending the following information's through the data bas.

1. The starting address of memory block were data are available for DMA read operation or where data are to be stored for DMA write operation.

2. The word count, to word count register.

3. Control to specify the mode of transfer such as read or write.

4. A control to start DMA transfer.

The transfer of data between memory & I/O device can be in two ways-

1. Burst transfer

2. Cycle stealing

Burst Transfer :- A block sequence consisting of a number of memory words is transferred in a continuous burst while DMA controller is master of the memory bus. This mode of transfer is needed for fast I/O devices.

Q.148 Explain the different shift operations with examples. What is the status of flag bit in each case? **(7)**

Ans:

Shift micro operations are used for serial transfer of data. They are also used in conjunction with arithmetic, logic and other data processing operations. The content of the register can be shifted left or right. There are three types of shifts.

1. Logical shift
2. Circular shift
3. Arithmetic shift

Logical Shift:- In this shift a zero is transferred through the serial input.

Examples: $R_1 \rightarrow 11000011$

$ShrR_1 \rightarrow 01100001$, LSB is lost.

$Shl\ R_1 \rightarrow 10000110$, MSB is lost

Circular Shift:- Also known as rotate operation.

It circulate the bits of the register around the two ends without loss of intimation.

Let the register content of R_1 = 11000011

Cil → Circular shift left = 10000111

Cir → Circular shift right = 1110001

Arithmetic Shift:-This shift a signed binary number to left or right. An arithmetic shift lest multiplies a signed binary number b2 and an arithmetic shift right divides the number by 2. Arithmetic shift must leave the sign bit un changed. The left most bit in a register holds the sign bit.

Let the register contains 10111100 = (–68)

Then shift right operation gives 11011110 = (–34)

The sign bit is shifted right & also remain in its position, so as to get sign unchanged. The arithmetic shift right results the original number divided by 2. In case of arithmetic shift left operation, the original sign bit is lost. So in order to ensure that the sign bit remains same even after the shift left operation, the two MSB bits must be same. Other wise change of sign takes place is known as over flow. This overflow set a FF 'V_s' to be 1 to indicate that there is an overflow.

$V_S = R_{n-1} \oplus R_{n-2}$.

Example $R_1 = 11000011$, $R_{n-1} = R_{n-2}$.

ashl $R_1 = 10000110$, Sign bit remain unchanged.

Originally $R_1 = 11000011 = (-61)$

After ashl $R_1 = 10000110 = (-112)$ which is equal to two times of (-61).

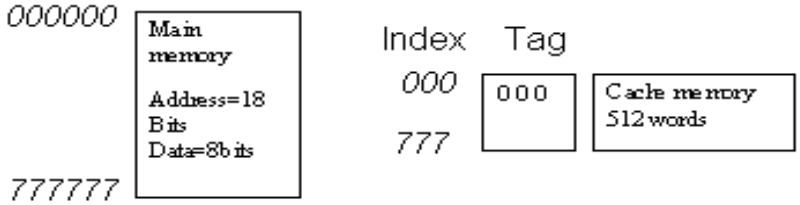
Q. 149 A personal computer has main memory of $32K \times 8$ bytes and cache memory of 512 words. The cache is directly mapped with block size of 4 words.
(i) How many bits are required in tag, index block and word fields of the address format?
(ii) Show the addressing format?
(iii) What are the advantages of direct addressing scheme? **(3+2+2)**

Ans:

Size of main memory = 32 K × 8 bytes = $2^5 \times 2^{10} \times 2^3$ bytes = 2^{18} bytes.

Size of cache memory = 512 words = 2^9 words.

Mapping method is direct mapping Block size = 4 words



(i) No. of index bits = 9, No. of tag bits = 18-9 = 9

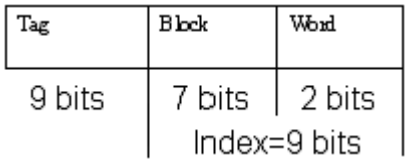
Total no. of bits in each word of cache = Data bits + Tag bits

= 8 + 9 =17 bits

No. of bits for representing block = 7

No. of bits for representing the word in the block = (9-7) = 2.

(ii) The addressing format for cache memory is by the index bits starting from (000) to (777) in octal representation.



(iii) Advantages of direct mapping scheme are-

(a) Less expensive than associative memory due to decrease in word size of cache memory.

(b) Mapping logic is simpler & less complex as only the tag field is to be matched instead of the whole address of the main memory.

Q.150 Simplify the Boolean function F together with the don't-care conditions d in (1) sum-of-products form and (2) product-of-sums form.

$$F(w, x, y, z) = \sum (0,1,2,3,7,8,10)$$

$$d(w, x, y, z) = \sum (5,6,11,15)$$

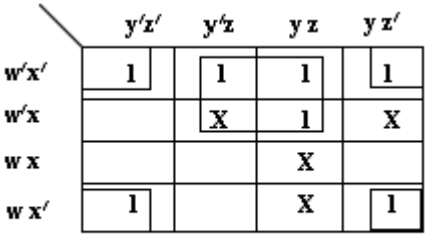
Draw the logic diagrams also. **(8)**

Ans:
Given function

$$F(w, x, y, z) = \Sigma(0,1,2,3,7,8,10)$$

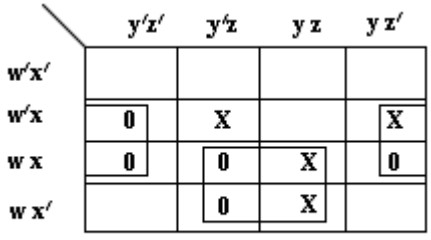
$$F(w, x, y, z) = \sum (0, 1, 2, 3, 7, 8, 10)$$
$$D(w, x, y, z) = \sum (5, 6, 11, 15)$$

Sum of products



$F = x'z' + w'z$

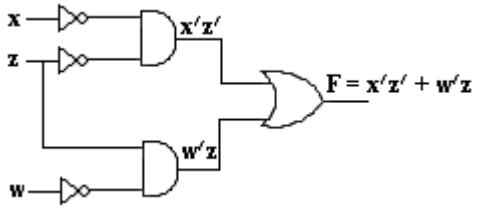
Product of sums



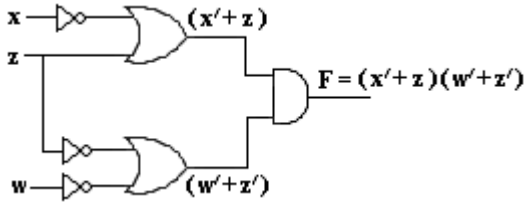
$F = (x' + z)(w' + z')$

Logic Diagram

SOP



POS



Q. 151 Represent the following decimal numbers in BCD, Ex - 3 and gray code.
(i) 56 (ii) 93

Ans:
(i) 56
Binary equivalent of 56 is 111000

BCD representation of 56 is 0101 0110

Er-3 representation of 56 is 10001001

Gray Code 56 = $(111000)_2$

Gray Code = (100100)

(ii) 93
Binary equivalent of 93 is 1011101

BCD representation of 93 is 1001 0011

Er-3 representation of 93 is 11000110

Gray Code 93 = $(1011101)_2$

Gray Code = (1110011)

Q. 152 Multiply $+.1001 \times 2^{-0011}$ and $-.1010 \times 2^{-0001}$ using binary multiplication algorithm. Show all steps. **(8)**

Ans: Let $X = +.1001 \times 2^{-0011}$

$Y = -.1010 \times 2^{-0001}$

Steps

1 : Add Exponents
(–0011) + (–0001)
= –0100

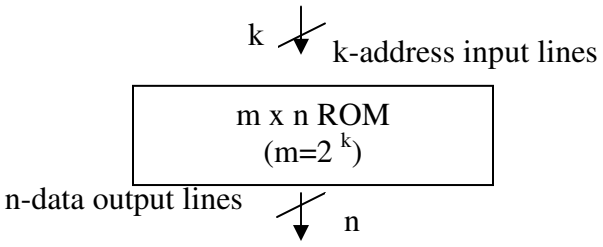
2 : Multiply mantissas
+ .1001 x –.1010
= .01011010

3 : Normalize and round the final product.
Hence, the final product is $1.011010 \times 2^{-0110}$

Q.153 Explain what do you understand by ROM ? **(4)**

Ans:

R.O.M.stands for (read-only-memory).it is a memory unit that performs the read operation only. It does not have a write capability. The binary information stored in it is made permanent during the hardware production. R.O.M. comes with special internal electronic fuses that can be programmed for a specific configuration by blowing out fuses of required position by ROM programmer.



A m x n ROM is an array of binary cells organized in to the m words of 'n' bits each as shown in the block diagram, it has K-address input lines to select one of the 2^k=m words of memory. As it has 'n' output lines, so the word length of each memory cell is of n-bits it is used for storing fixed programs that are not to be altered.

Example: design of control unit for stored-program computer.

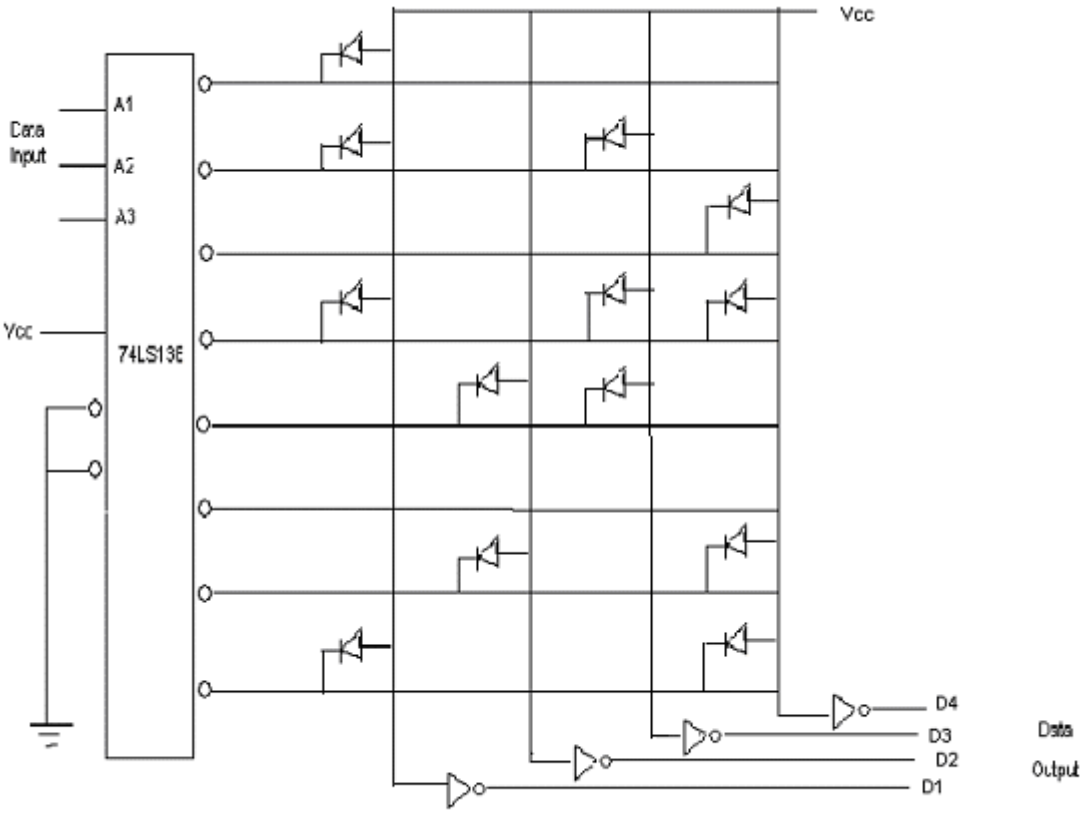
Q.154 With the help of a diagram using diodes in a matrix form, explain the concept underlying ROM ? **(6)**

Ans:

A ROM implementing the following truth table using diodes in a matrix form is shown below.

Truth table

Address inputs			output data			
A3	A2	A1	D4	D3	D2	D1
0	0	0	0	0	0	1
0	0	1	0	1	0	1
0	1	0	1	0	0	0
0	1	1	1	1	0	1
1	0	0	0	1	1	0
1	0	1	0	0	0	0
1	1	0	1	0	1	0
1	1	1	1	0	0	1



Diodes are used to connect the output lines of the decoder to the data output lines that corresponds to logic 1's in the truth table for example when the input address lines are 000, decoder output '0' is selected. According to the truth table, the required output is 0001, so data line D1 must be a '1'.

Q.155 Define PROM and EPROM. **(4)**

Ans:

PROM: is the programmable read only memory. The user programs it. In PROM, FET or BJT are used in place of diodes of a ROM. PROM have each address select line connected to a FET gate (or bipolar transistor base) with the source (emitter) grounded and the drain (collector) connected to the vertical data lines. A fuse link exists between grounds & the FET source (or bipolar transistor emitter). Because PROMS are programmed by the user, each address select & data lines intersection has its own fused FET or transistor. When the fuse is intact the memory cell is configured as logical '1', and when the fuse is blown (open circuit) the memory cell is logical'0'. Logical 0's are program by selecting the appropriate select line then driving the vertical data line with a pulse of high current.

EPROM: stands for erasable programmable read only memory. Instead of fused bi-polar or FET transistor switch as in PROM, it has a MOSFET that has two gates. The memory cell is configured like PROM cell, however no fuse is present. When an EPROM is programmed, a charge is placed on the floating gate, permitting channel current to flow & thus establishing the logic level of the memory cell .The floating gate to channel voltage is discharged when placing it under ultra violet light erases the device. It takes approximately 5 to 20 minute of exposure to the UV light to erase the device.

Q.156 Give the three methods by which the subtraction of two n-digit unsigned numbers M-N ($N \neq 0$) in base r can be carried out. **(3)**

Ans:

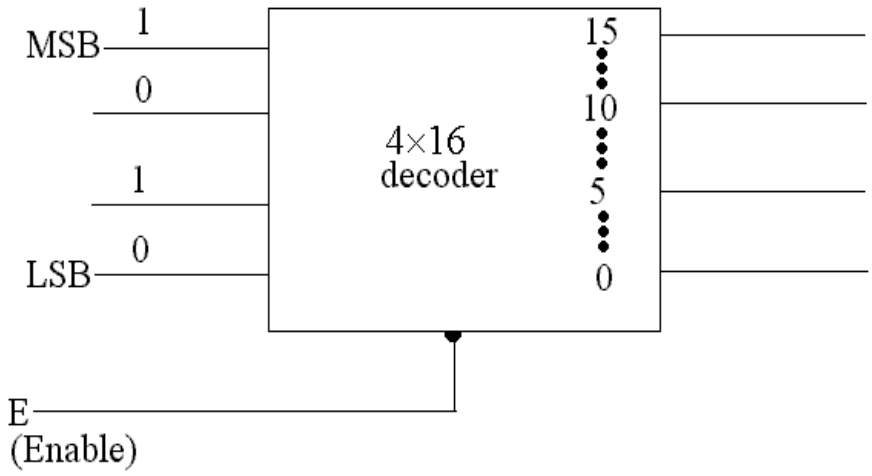
The three methods by which the subtraction of two n-digit unsigned numbers i.e.
M- N (where N is not equal to 0) in base r can be carried out are:

i. **Direct method of subtraction:** in which the subtraction in which we borrow a 1 from the higher significant position when the minuend digit is smaller then the corresponding subtrahend digit.

ii. **(r-1)'s complement method:** where the (r-1)s complement of subtrahend is added to minuend i.e. $M + (r^n - N)$

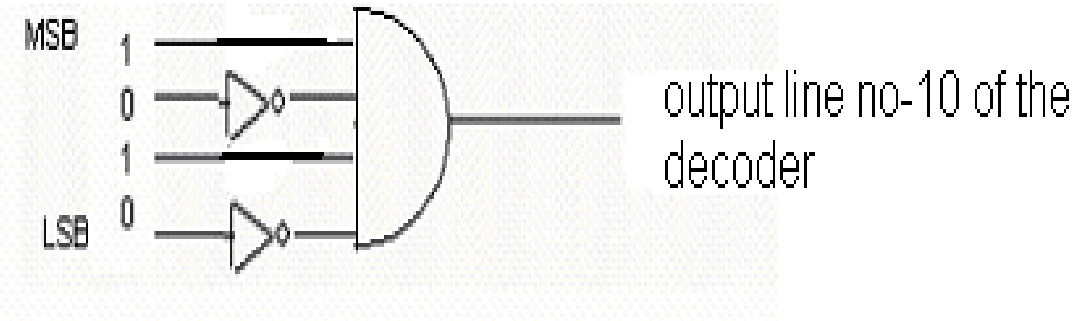
iii. **r's complement method:** where the r's complement of subtrahend is added to minuend i.e. $M+[(r^n - N)+1]$.

Q.157 In the 4 – to –16 decoder shown in the figure, mark the output line of the decoder which goes low, when the input to the decoder is as shown in this logic diagram. Give reasons for your answer. **(5)**

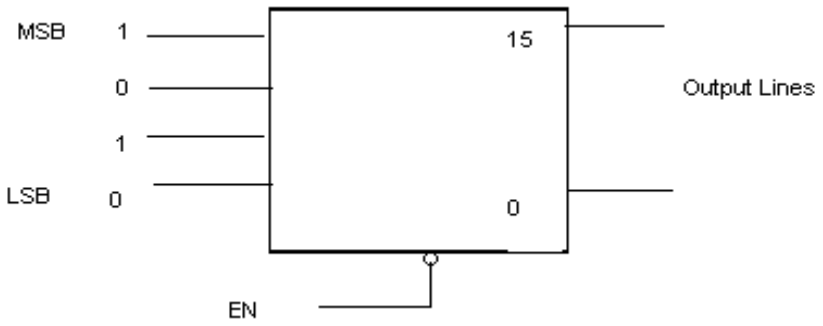


Ans:

When the input to (4x16) decoder is (1010), with enable active low, the output line marked with decimal '10' shall be active high. Because the AND gate corresponding to output line 10 shall produce a high output.



(If the output are buffered, than with the same configuration a zero shall be produced on the output line---10)

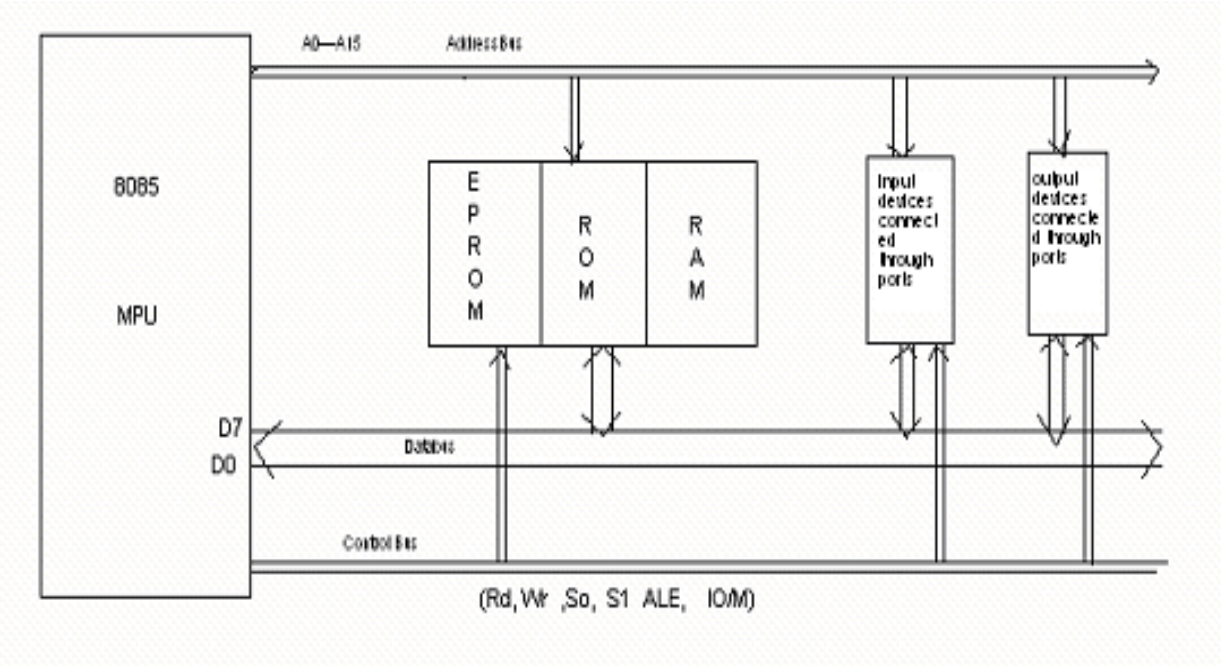


The (4x16) decoder consists of 16 AND gates which decodes all the 16 possible minterms, one of the AND gate out of 16 shall produce a high output depending on the input bits combination.

Q.158 With the help of a diagram, explain a microcomputer system having microprocessor 8085, EPROM and RAM memories, input and output, and buses linking all peripherals (memory and I/Os) to the microprocessor unit. **(6)**

Ans:

8085 microprocessor is a 8 bit microprocessor i.e. it can operate on the data of 8-bit at a time hence it has an 8-bit ALU to perform arithmetic operation like add, substraction and logical operation like ANDing, ORing, complementing etc….



It has got 16-bit address line (A0 toA15) where A_0 toA_7 are multiplexed with the data lines D0-D7. It can address up to 2^{16} =64k of memory locations. Further it has a control signal $IO/\overline{M}$ which is 1 for addressing to an I/O device & is '0' while addressing memory locations. In I/O mapped I/O, 8085 microprocessor can address up to 2^8 = 256 I/O devices.

The memory unit consists of EPROM, ROM & RAM

The monitor programs & other essential programs and data are stored in EPROM or ROM whereas the RAM memory is used by the user for storing the programs & data, which is not of permanent nature. Besides address lines & data lines, a microprocessor system has got control bus consisting of control signals. These control signals are generated by the control unit of 8085 microprocessor and are connected to all peripherals. Control signals controls the flow of operations sequentially. RD, WR, $IO/\overline{M}$, S0, S1, ALE are some of the control signals of 8085 microprocessor . MPU generates specific control signals for every operation such as memory read, memory write, I/O read, I/O write, DMA operation, interrupt cycles etc….

The data bus of the computer system is bi-directional, where as the address bus is unidirectional.

Input devices are connected to MPU through their respective interface circuits to feed data from external world to the computer system. The example of input devices are keyboard, mouse, joystick, etc….similarly, output devices are used to transfer data from the microprocessor to the outside world. They include devices such as light emitting diodes, CRT or video screen, a printer, a magnetic tape x-y plotter, D/A converter etc…

Q.159 Differentiate between :

(i) micro instructions and micro operations. **(2)**

(ii) mini computers and microcomputers. **(2)**

(iii) RAM and ROM chips. **(2)**

Ans:

(i) **Microinstructions:** A microinstruction specifies one or more micro-operation for the system and a sequence of microinstruction constitutes a micro program, which besides in control memory.

Micro operation: it is an elementary operation performed with the data stored in registers.

To perform a micro operation microinstruction generates necessary control signals in proper sequence.

ii) **Mini computer**: These are the computer of medium sized, which are slower, then the mainframes and also have smaller memory capacity. In earlier days these machines were used to meet the needs of manufacturing needs of small factories, data processing task of medium sized business & colleagues. Now a day, even the high end microcomputers are more powerful then earlier minicomputers.

Microcomputers: In earlier days 4bits & 8bit microprocessor oriented computer systems were called as microcomputers. But now a days due to invent of powerful microprocessors of 16, 32 and 64-bit the microcomputers are quite powerful with large memory capacity. The personal computers come under the category of microcomputers.

iii) **RAM:** it is a read- write semiconductor memory. In this memory, user can write data many a times and also can read it.
ROM: it is a read only memory. Although it also works on the principle of random access but it cannot be written more than once. As fusible links are burnt while writing this type of memory, so it can be written once and the data is of permanent nature.

Q.160 Write short notes on :
(i) Cache memory. **(4)**
(ii) Shift instructions. **(4)**

Ans:
(i) **Cache, memory:** it is a special high-speed memory called a cache, is used to increase the speed of processing by making current programs and data available to CPU at a rapid rate. The cache memory is employed in computer systems to compensate for the speed differential between main memory access time and processor logic. CPU logic is usually faster than main memory access time, with the result that processing speed is limited primarily by the speed of main memory.
To compensate for the mismatch in operating speed, extremely fast small cache between CPU & main memory, whose access time is close to processor logic clock cycle time, is used. Cache is primarily used for storing segments of programs currently being executed.
ii) **Shift instruction:** shift instructions are used for serial transfer of data in a register. These are also used in conjunction with arithmetic, logic & other data processing operations. The content of the register can be shifted left or right. There are three types of shift operations.
a) Logical shift: it transfers 0 through the serial input.
b) Circular shift: it does the rotate operation .it circulates the bits of the register around the two ends without loss of information.
c): arithmetic shift: it shifts a signed binary number to the left or right. An arithmetic shift left multiplies a signed binary number by 2 and an arithmetic shift right divides the number by 2. It leaves the sign bit unchanged.
All the above-mentioned shift instructions are of two types depending on the direction of shifting of bits in the register. i.e. left shift or right shift. The examples of shift instruction are SHR, SHL, SHRA, SHLA, ROR, ROL, RORC, & ROLC. Where RORC &ROLC are the instructions for shifting through carry.

Q.161 In connection with the assembly language, what do you understand by

(i) field
(ii) symbolic address
(iii) pseudo instructions **(9)**

Ans:

i) **Field:** The bits of the instruction are divided in to groups called fields. The most common fields found in instruction formats are: opcode field, address field & mode field. Other special fields are some times employed under certain circumstances. The group of bits of any particular field specifies various processing information to the processor. E.g.: ADD XXXX, where ADD represents the opcode field & specifies the operation and XXXX represents the address field where the operand is found and the other operand is always in accumulator.

(ii) **Symbolic address:** A symbolic address consists of one, two or three but not more than three alphanumeric characters. The first character must be a letter; the next two may be letters or numbers. The symbol can be chosen arbitrary by the programmer. A symbolic address in the label field is terminated by a comma (,), so that it will be recognized as a label by the assembler.

(iii) **Pseudo instruction:** These are not machine instructions, rather an instruction to the assembler giving information about some phase of the translation. e.g.: ORG(origin) informs the assembler that the instruction or operand in the following line is to be placed in a memory location specified by the number next to ORG. similarly END,DEC,HEX are also pseudo instructions which instruct the assembler about end of program, the number is decimal, the number is hexadecimal respectively.

Q.162 (i) Explain subroutine. **(2)**

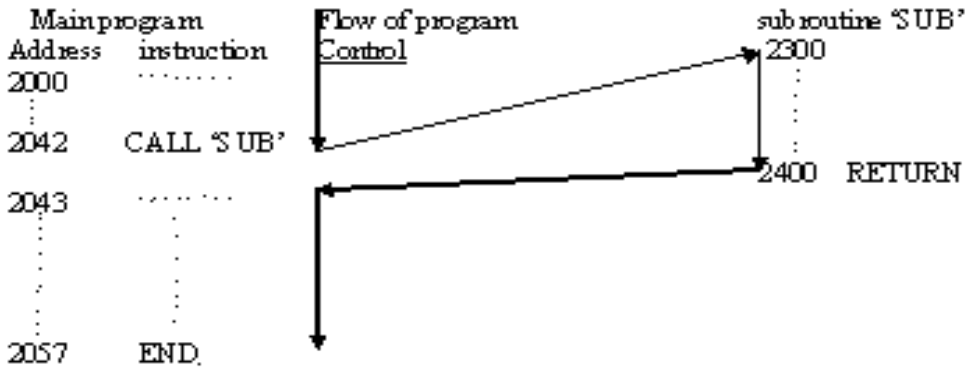
(ii) Demonstrate the use of subroutines with the help of suitable diagram. **(3)**

Ans:

(i) **Subroutine:** A set of common instructions that can be used in a program many times is called a subroutine. Each time that a subroutine is used in the main part of the program, a branch is executed to the beginning of the subroutine. After the subroutine is executed, a branch is made back to the main program. Subroutine consists of a self-contained sequence of instructions that caries out a given task.

(ii) A subroutine is called in a main program by using suitable CALL instruction. When this CALL instruction followed by the name of the subroutine is executed, the return address of the main program is stored in the stack memory or any specified register. The program counter holds the starting address of the subroutine program.

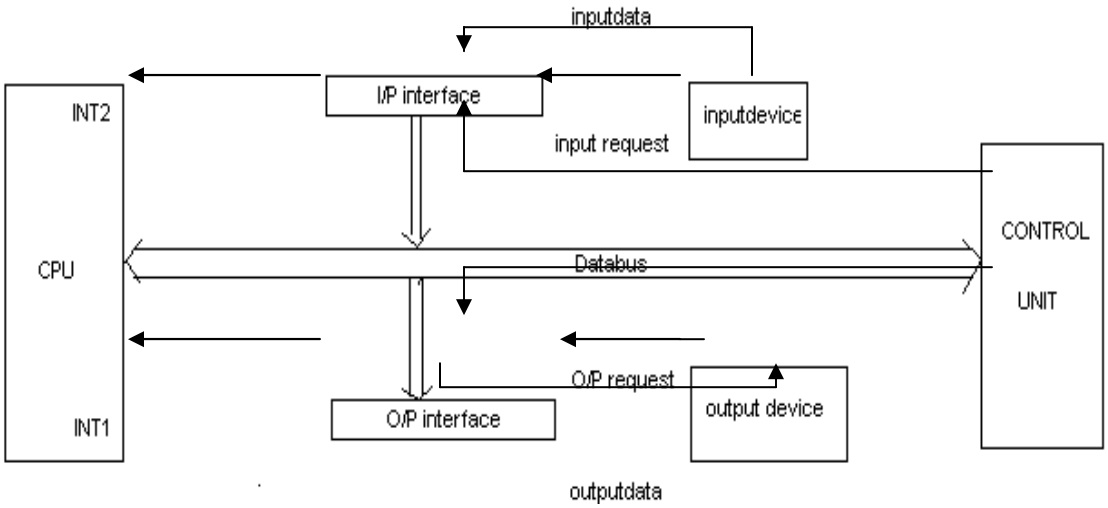
At the end of the subroutine there must be a RETURN instruction. This enables the processor to return the control of program flow again to main program. The return address that is stored in the stack is loaded in program counter and the execution of main program proceeds.



Q.163 What do you understand by the term 'Program Interrupt'. Explain with the help of suitable diagrams. **(8)**

Ans:

A computer system is used to communicate with the input and output devices. The data flow is from input device to computer and from computer to output device .Now the simplest way of transfer of data between I/O device and computer is called program control data transfer. In which the CPU keep on searching for the data buffer of input and output devices in order to receive a data or to send a data. Hence CPU does no useful work & CPU time is wasted in waiting for data transfer between I/O devices. To improve the efficiency of CPU, program interrupt technique is used. In which CPU does its own work without checking the input and output flag. Whenever an I/O device needs to communicate, it sends an interrupt signal. On receiving the interrupt, the normal processing of CPU gets stopped temporarily and it saves the return address of the program in stack and the program for data transfer between the CPU & I/O device is executed. Once this communication is over, again CPU comes back to the 1st program by obtaining the return address from stack. In this way CPU efficiency is improved. Hence, this program interrupt is applicable in multiprogramming environment that is when two or more program resides in memory at the same time. The block diagram arrangement of program interrupt technique is shown below.



When the input device needs to send a data to CPU, its interface circuit initiate a interrupt to CPU through INT2 line and similarly when O/P device need a data from CPU, its control circuit generates INT1 signal to CPU. Both input & output of data is carried out by generating interrupt signal to CPU Hence, in this scheme, interrupt is generated by external device only.

Q.164 Differentiate between the terms external, internal and software interrupts. **(6)**

Ans:

There are three major types of interrupt that causes a break in the normal execution of a program. They are:

(i)External interrupt: it comes from I/O devices on from any external sources. Any microprocessor has got same interrupt lines. For example 8085 has got RST 7.5,RST 6.5,RST 5.5,INTR & TRAP lines Any I/O device connected to these lines sends interrupt signal when need the service of microprocessor. As these interrupts come from an external I/O device, so it is called external interrupt. These are asynchronous with the program and independent of it.

(ii)Internal interrupt: it arises from illegal or erroneous use of an instruction or data. Examples of interrupts caused by internal error conditions are register overflow, attempt to divide by zero, an invalid operation code, stack overflow and protection violation. These error conditions usually occur as a result of premature termination of the instruction execution. The service program that process the internal interrupt determines the corrective measures to be taken hence the internal interrupt is initiated by some exceptional condition caused by the program itself rather than by an external event. These are synchronous with the program. If the program reruns, internal interrupts will occur in the same place each time.

(iii)Software interrupts: initiated by executing an instruction. Software interrupts are special call instructions that behave like an interrupt rather than a subroutine call. It can be used by the programmer to initiate an interrupt procedure at any desired point in the program.

Q.165 Write short notes on
(i) Conditional branching **(3)**
(ii) Flags **(3)**

Ans:

(i). **Conditional branching**: branching from main program to a subroutine or to another part of the main program when a particular condition is satisfied is called conditional branching. Hence before branching from normal flow of program execution, the condition is tested. If satisfied, then the program flow jumps to a new location of the program or to a subroutine. The conditions are checked with the status of flag bits. For example, jump on +ve is a conditional branching. It checks the sign flag associated with accumulator content and it branches to the new location or address as specified by the condition at jump instruction is not satisfied, then the normal flow of execution of the program continue. The other examples of conditional branching are jump on minus, jump on even parity, jump on carry, jump on no carry etc….

In case of conditional jump instructions, no return address is saved but in care of conditional call instruction, the return address is to be saved.

(ii). **Flags**: These are nothing but a single bit Flip-Flops, the value of the flip-flop can be zero or one depending on same information associated with accumulator content. For example sign flag is a single bit FF, when holds 1, if the sign of accumulator content is negative in case of 2's complement arithmetic otherwise it is zero. Similarly, parity flag, direction flag, zero flag, auxiliary carry flag, etc… the content of flag bits keeps on changing with the program flow depending on the accumulator content. Flags are tested during conditional branching instructions. The content of flag register & accumulator is called program status word.

Q.166 Explain 'Assemblers' and 'Cross Assemblers'. Give advantages of using them. **(8)**

Ans:

Assembler: the assembler is a program that translates source code or mnemonics in to binary code of the microprocessor and generates a file called object file. It performs various functions such as error checking & memory allocation.

Cross assembler: the cross assembler is a program that develop the assembly language program of a microprocessor running on a computer system which have some other microprocessor. For example we can use PCs based on Pentium microprocessor to develop the program for 8085 microprocessor. Hence, cross assembler is used in PC's to develop the program for other microprocessor.

If we want to develop assembly language program for the same processor, i.e. the processor of the system, then only assembler is required.

Advantage of using assembler & cross assembler: -

(i) Assembler translates mnemonics in to binary code with speed and accuracy, thus elementary human error.

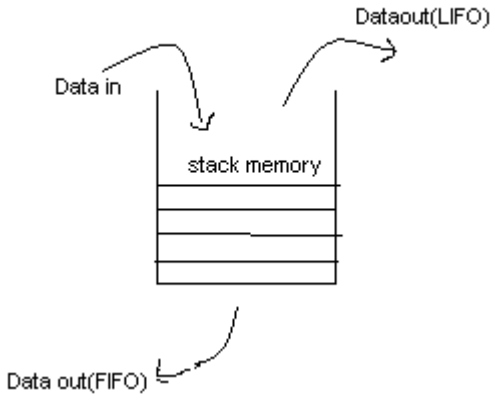
(ii) The assembler assigns appropriate values to the symbol used in the program, this facilitates specifying jump locations.

(iii) Insert/delete instruction in a program is made easy by using assembler. New memory location is also calculated by assembler.

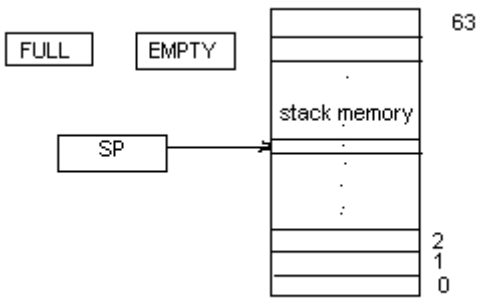
(iv) It can check syntax error.
(v) It can reserve memory for data or result.
(vi) It can provide files for documentation.

Q.167 Explain the principle of stack ; what are LIFO and FIFO operations.**(9)**

Ans:
A stack is a storage device that stores information in such a manner that the item stored can be retrieved in a sequential manner. The stack in digital computers is essentially a memory unit with an address register that can count. This register is called stack pointer. This needs initialization before using a portion of r/w memory as stack memory. The value of the stack pointer always points to the top item in the stack.
The insertion of data in to stack is called push operation and deletion of data from stack is called pop operation. When a push operation is carried out, SP is decremented & incremented with pop operation.
Stack memory deletion process can be of two types.
(i) LIFO (ii) FIFO



LIFO stands for last in first out i.e. the most recent data, which is put in to the stack, can be taken out first. Where as in case of FIFO principle, the data which was inserted first can be taken out first.
Let us see the principle of operation of a stack memory, where the stack pointer is of 6 bits. Hence the total number of memory words that this stack can hold is 2^6 = 64 words as shown in figure below,



To indicate the situation of stack, two flags named as FULL & EMPTY are used. When the stack is full, FULL flag is set to 1 and when the stack is empty of items, EMPTY flag is set to 1.

The micro-operations associated with insertion of data into stack is

SP ← SP+1
M[SP] ← DR where DR stands for data register.
If(SP=0), then (FULL ← 1)
EMPTY ← 0

Similarly, the micro operations associated with POP is

DR ← M [SP]
SP ← SP-1
If (SP=0) then (EMPTY ← 1)
FULL ← 0

Stack memory is very useful for storing return address and other registers content by the processor when it encounters an interrupt or a call instruction. This stack memory can also be used for efficient evolution of arithmetic operations with reverse polish notation (RPN)

Q.168 List the instructions necessary for using stack. **(5)**

Ans:

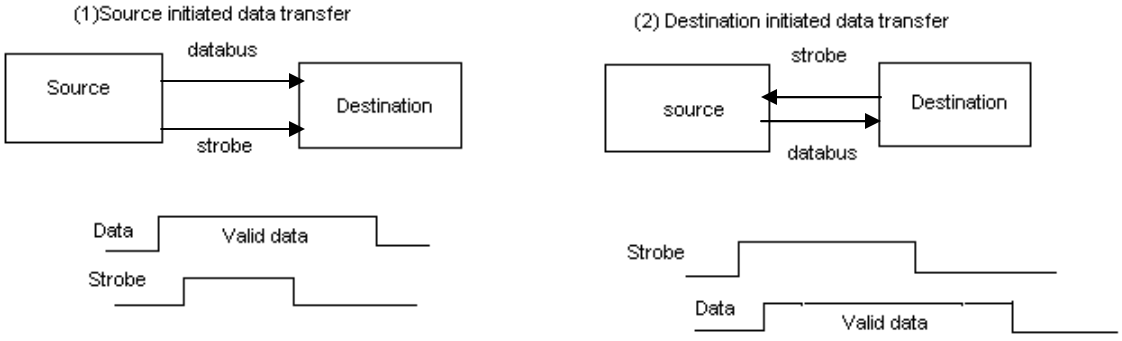
The instructions that are necessary for using stack are:
(I) PUSH X: it will PUSH the word at address X to the top of the stack. The stack pointer is updated automatically.
(ii) POP X: it will take out the word which is available at the top of the stack and the stack pointer is updated automatically.
In case of 8085 microprocessor, the other instructions used for stack is PUSH rp, POP rp, SPHL, XTHL
Where the instruction XTHL exchanges the content of H & L register with the top two data of the stack memory. The content of SP remains same.
The instruction SPHL, copies the content of H register in $(SP)_H$& the content of register L is copied in $(SP)_L$.

Q.169 Explain the Strobe Control method of Asynchronous data transfer. What are the disadvantages of this method? **(14)**

Ans:

When the two units under communication uses their own private clock for internal registers, then it is called asynchronous communication. In this mode of data transfer, it requires some control signals between the communicating devices to indicate the time at which data is being transmitted. One way of achieving this is by mean of Strobe pulse, supplied by one of the unit to indicate to the other unit when data transfer has to occur.
In the strobe control of data transfer, there is only one control line between the destination device & source device. There is no confirmation of receipt of data by destination unit. This is the disadvantage of strobe control method.
The timing diagram that shows the time relationship that must exist between the control signal & the data in the buses are different depending on whether the transfer is initiated by

the source or by the destination device. The strobe signal may be activated by the source or by destination device. If it is activated by source, then it is called source initiated data transfer, in other case it is called as destination initiated data transfer. In first case, data is placed in data bus by source, & than strobe signal is sent to destination to inform the availability of data on data on data bus. The destination unit uses the falling edge of the strobe pulse to transfer the contents of the data bus in to one of its internal registers. In the second case the destination units activates the strobe pulse, informing the source to provide the data. Source places the data on the data bus for acceptance of destination unit. After receiving data, destination unit then disable the strobe. The source removes the data from the bus after a predetermined time interval. The timing diagram is shown below



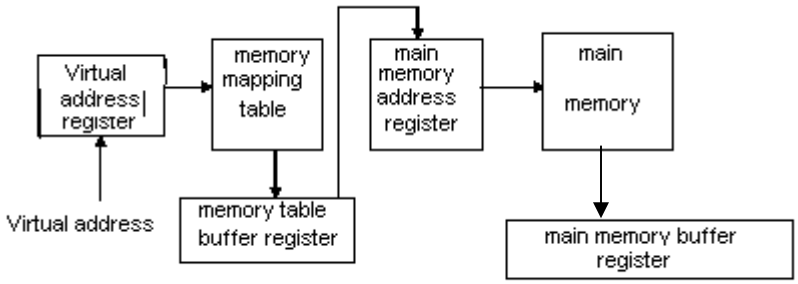
The disadvantage of the strobe method is that the source unit that initiates the transfer has no way of knowing whether the destination unit has actually received the data item that was placed in the bus. Similarly, a destination unit that initiates the transfer has no way of knowing whether the source unit has actually placed the data on the bus.

Q.170 Explain the term Virtual Memory. **(7)**

Ans:

Virtual memory is a concept used in large computer system that permits the user to construct programs as though a large memory space is available. The user shall feel that the system has a main memory equal to the totally of auxiliary memory. Each address that is referenced by the CPU goes through an address mapping from the so called virtual address to a physical address in main memory. It gives the programmer the illusion that they have a very large memory at this disposal even though the computer actually has a relatively small main memory.

Say, for a PC with 256 MB RAM & 40 GB hard disk, the auxiliary memory is 40 GB, but the main memory is 256 Megabyte, but in virtual memory concept, the user shall have a feeling of 40Gbyte main memory. A virtual memory system provides a mechanism for translating program generated addresses into correct main memory locations. This done dynamically, while programs are being executed in the CPU. The translation or mapping of virtual address to physical address is handled by the hardware by the means of a mapping table. The address used by the programmer is called virtual address. The address in the main memory is called physical address. For the specification cited above, the address space is of 40×10^{12} and the memory space is of 256×10^{6} words. The memory table for mapping a virtual address is given below



Q.171 Construct an associative memory page table with the number of words equal to the number of blocks in the main memory. **(7)**

Ans:

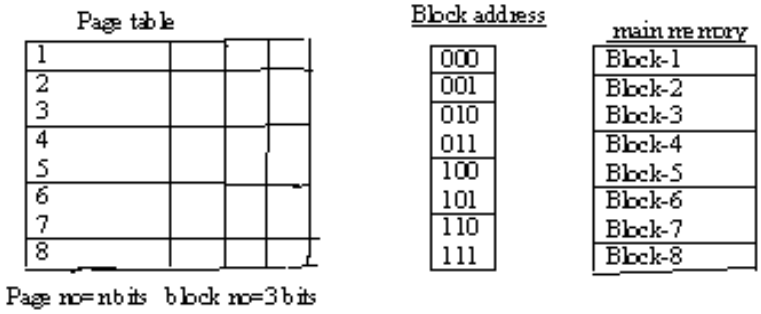
Given ; the no. of words of page table= no. of blocks in main memory.

Let the number of words of a page table=8

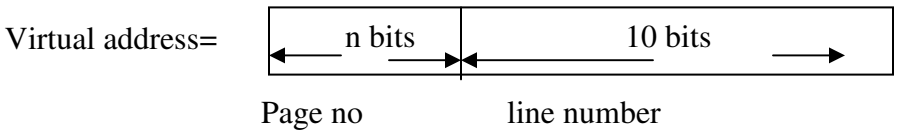
No. of blocks in main memory is 8.

Let each block is of 1k, i.e. 1024 words. So the size of each page shall be also 1 k. so main memory is of 8k. No. of pages= 2^n

Size of address space= $2^n \times 1024$ words.



The virtual address is of n bits plus 10 bits= (10+n) bits 10 bits are required to represent line number & n-bits are required to represent the page number.



The page no of virtual address is compared with the page no. stored in the page table of 8 words. If it matches, than the corresponding word of page table, consisting of (n+3) bits are read out and the block number contained in last three bits of the word gives the

location where this page is stored in main memory. The block numbers along with the line number (10 bits) constitute the physical address of the data, which is required by the program. In this method, the memory space used for storing the page table is utilized fully.

Q.172 Explain the term 'Booth's multiplication algorithm'.**(4)**

Ans:

Booths multiplication algorithm gives a procedure for multiplying binary integers in signed 2's complement representation. It operates on the fact that strings of 0's in a multiplier needs no addition but just shifting and a string of 1's in the multiplier from bit weight 2^k *to* 2^m can be treated as $2^{k+1} - 2^m$. For example, the binary number 001110(+14) has a string of 1's from 2^3 to 2^1 .i.e. m=1, k=3. Therefore the number can be represented as $2^{k+1} - 2^m = 2^4 - 2^1 = 16 - 2 = 14$.

Therefore the multiplicand 'M' is to be multiplied by 14.i.e. the multiplier can be taken as $M\ 2^4 - M\ 2^1$.This is equivalent to shifting the multiplicand 'M' to left for 4 bits and subtracting the multiplicand shifted to left by 1 bit.

Hence the booth's multiplication algorithms requires examination of the multiplier bits and shifting if the partial product. Prior to the shifting, the multiplicand may be added to the partial product, subtracted from the partial product or left unchanged according to the following rules.

(1) The multiplicand is subtracted from the partial product upon encountering the first least significant 1 in a string of 1's in the multiplier.

(2) The multiplicand is added to the partial product upon countering the first 0 (provided that there was a previous 1) in a string of 0's in the multiplier.

(3) The partial product does not change when the multiplier bits is identical to previous multiplier bit.

Q.173 Explain the working of a Half substractor with the help of a diagram. **(4)**

Ans:

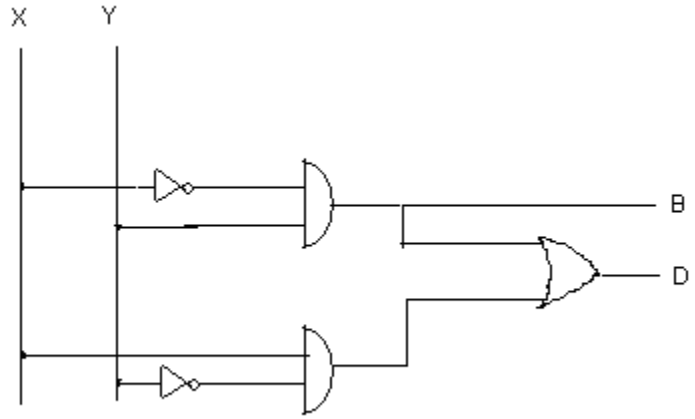
Half substractor is a combinational circuit that is used for subtracting two single bits. The result is of two bits, known as borrow & difference. The truth table is given below.

X	Y	B (borrow)	D (difference)
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

$B = \overline{x}\, y$

$D = x \oplus y$

The realization of a half substractor is

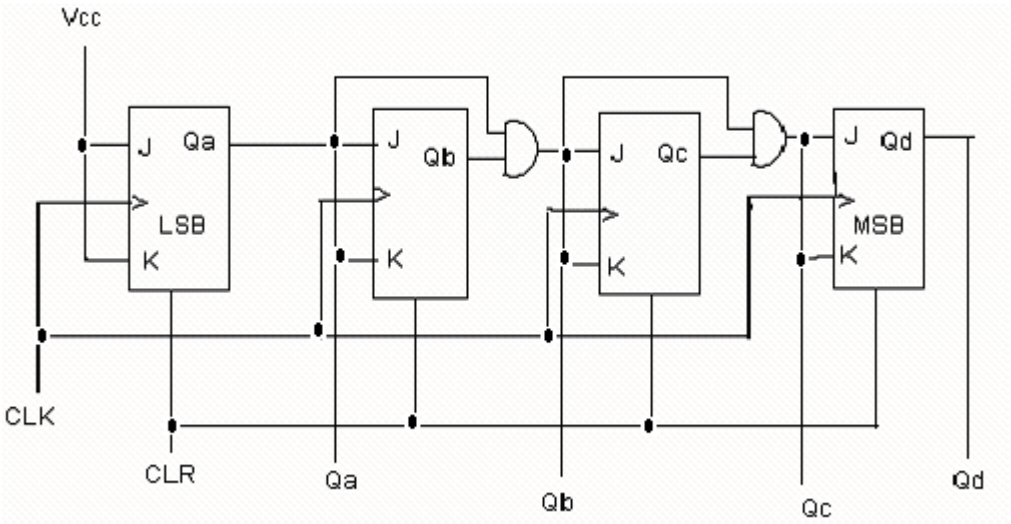


Half Substractor

Q.174 Explain with a neat diagram the working of a 4-bit synchronous binary counter. **(6)**

Ans:

4 bit synchronous counter is shown below.



To start the operation of the counter it is cleared to '0000, state by asynchronous clear input .With the clock pulse, only the LSB FF shall toggle to output 1, as its input J & k are tied to Vcc but the other FFs shall toggle only when the AND of the outputs of FFs preceding it are 1.

Hence the translation table shall be

Clk pulse	Qd	QC	Qb	Qa
CLR	0	0	0	0
↓	0	0	0	1
↓	0	0	1	0
↓	0	0	1	1
↓	0	1	0	0
.	0	1	0	1
.	0	1	1	0
.	0	1	1	1
.	1	0	0	0
.	1	0	0	1
.	.	.	.	.
	.	.	.	.
.	.	.	.	.
	.	.	.	.
↓	1	1	1	1
↓	0	0	0	0 (initial state)

So, the counter will proceed from 0000 to 1111 with the falling edge of the clock pulses. This counter is called synchronous counter as the clock input to all the FFs is connected to same signal source, which produces the clock signals for the counter.

Q.175 Implement the following Boolean function after simplifying using Karnaugh map technique. $f(A,B,C,D)=\sum m(1,2,4,15)+\sum d(0,3,14)$ **(8)**

Ans: $f(A,B,C,D)=\sum m(1,2,4,15)+\sum d(0,3,14)$

	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	$C\bar{D}$
$\bar{A}\bar{B}$	X	1	X	1
$\bar{A}B$	1	0	0	0
AB	0	0	1	X
$A\bar{B}$	0	0	0	0

$$f=\bar{A}\,\bar{B}+ABC+\bar{A}\,\bar{C}\,\bar{D}$$

Q.176 Give the logic diagram and truth table of a full subtractor. Realize the full subtractor using NAND gates only. **(6)**

Ans:

Full subtractor is a combinational logic circuit capable of doing substraction of a substrahend bit from minuend bit with borrow. The table is shown below:

← *input* →			← *outputs* →	
X	Y	Z	B	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$B = \bar{x}\bar{y}z + \bar{x}y\bar{z} + \bar{x}yz + xyz$

$= \bar{x}(\bar{y}z + y\bar{z}) + yz(\bar{x} + x)$

$= \bar{x}(y \oplus z) + yz$

$D = \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz$

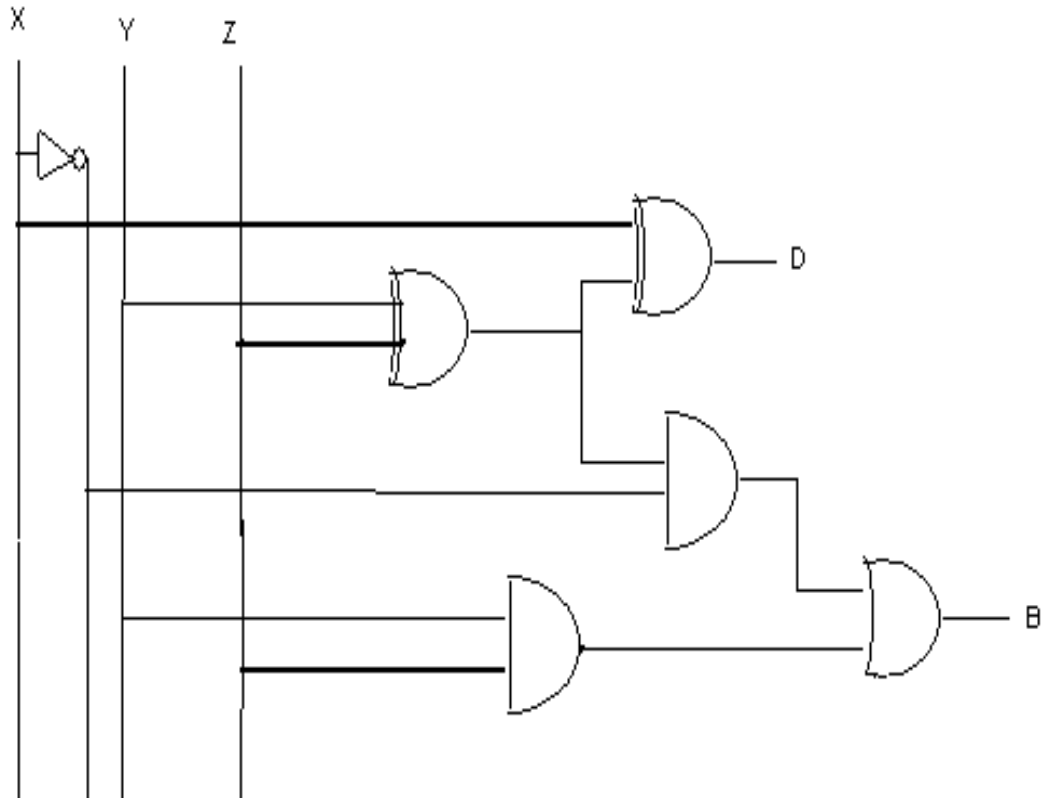
$= \bar{x}\bar{y}z + xyz + \bar{x}y\bar{z} + x\bar{y}\bar{z}$

$= \bar{z}(\bar{x}\bar{y} + xy) + \bar{z}(\bar{x}y + x\bar{y})$

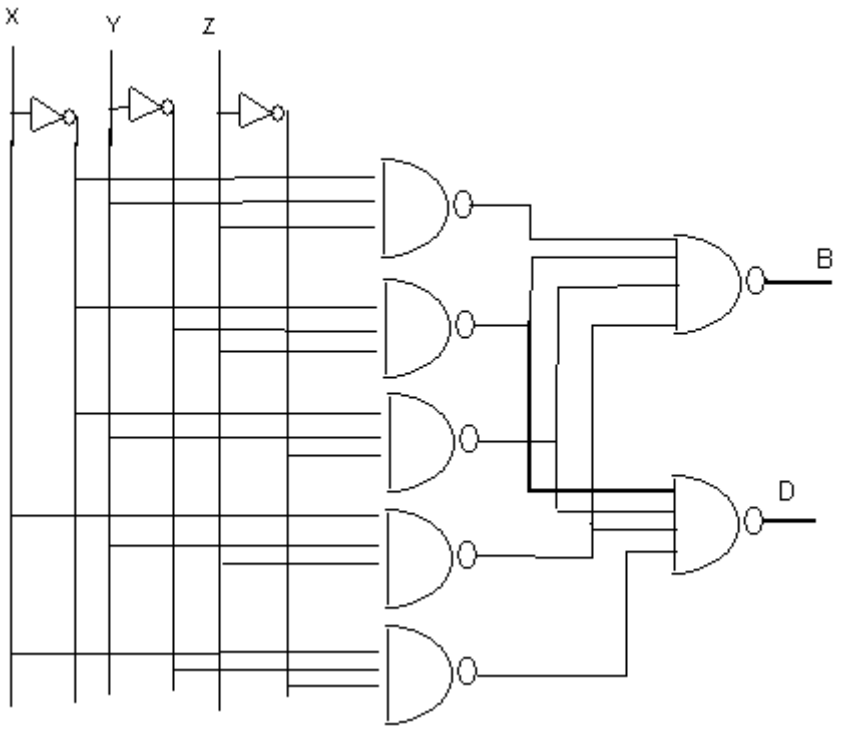
$= z\overline{(x \oplus y)} + \bar{z}(x \oplus y)$

$= x \oplus y \oplus z.$

The logic diagram of the above eqation is given below:



The NAND gate realization of the above circuit is given below:



Q.177 What are sequential circuits? Compare them with combinational circuits. **(4)**

Ans:

Sequential circuits are a part of digital circuit using feed back paths. Due to presence of feed back path, the output of the circuit at any time depends on the present inputs well as previous states available in the circuit. Sequential logic expands combinational logic functions to include the ability to store & latter retrieve binary information memories, Registers, counters & flip flops belongs to sequential circuits.

comparison between sequential ckt & combinational ckt.

Sequential	***Combinational***
(i) Feedback path in circuit	No feed back.
(ii) O/p depends on previous state and present input.	O/p depends on only present inputs
(iii) clock input is required to sequence the operation in time.	Not required
(iv) Circuit behaviour is represented by transition table	Circuit behaviour is represented by truth table.
(v) Circuit design and analysis is complicated.	Circuit design and analysis is easy.

Q.178 Explain the various phases of instruction cycle in a basic computer. **(6)**

Ans:

The program is stored in the memory get executed one instruction after other in a sequential manner. The process carried out by the CPU for execution of each instruction is called instruction cycle. This instruction cycle is subdivided into a sequence of sub cycles or phases. These are

(i) Fetch an instruction from memory.

(ii) Decode the instruction

(iii) Read the effective address from memory if the instruction has an indirect address.

(iv) Execute the instruction.

Upon the completion of step (iv), the control goes back to steo-1 to fetch, decode and execute the next instruction. This process continuous indefinitely unless HALT instruction is encountered. The micro operation that takes place for fetch and decode cy le is as under :-

T_0 : AR ← PC

T_1 : IR ← M[AR}, PC ← PC+1

T_2 : D_0, D_1, ……D_7 ← Decode IR

The necessary control signals are generated by the control unit during T_3, T_4… , for execution of the instruction. The program counter gets incemented during T_1, so as to enable to fetch the next instruction from memory after completion of execution of present instruction.

Q.179 Perform (–35) – (+40) with negative numbers in signed 2's complement representation. **(6)**

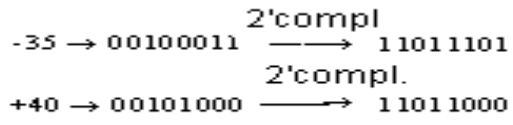
Ans:

-35 – (+40) in 2's complement representation is carried out as under :-

+ 35 = 010001

+ 40 = 0101000

In signed 2's complement representation, both the operands are represented as



As we have to substract (+40) from -35

So the result of substracton = Minuend in signed 2's complement from + 2's complement of substracted.

-35 – (+40) = 11011101

11011000

10110101

The overflow carry is neglected. So the answer is (10110101) which is in signed 2's complement from, which is equal to (-75).

Q.180 Use restoring method to divide 10100011 by 1011. **(8)**

Ans:

The restoring method of division to divide 10100011 by 1011 is given below. Assume both numbers are positive

	E	A	Q	SC	B	Remark
	0	1010	0011	4	1011	
EA←A + $\overline{B}$+1	0	1111				E=0,so A < B
EA←AA+B	1	1010				
Shl EAQ	1	0100	0110			E=1, A←A+B+1
A←A+ $\overline{B}$+1		1001				
Q_N←1			0111			
SC ←SC-1				3		
	1	1001	0111	3	1011	
Shl EAQ	1	0010	1110			E=1
A←A + $\overline{B}$+1		0111				
Q_N←1			1111			
SC ←SC-1				2		
Shl EAQ	0	1111	1110			E=0
A←A + $\overline{B}$+1	1	0100				E=1
Q_N←1			1111			
SC ←SC-1				1		
Shl EAQ	0	1001	1110			E=0
A←A + $\overline{B}$+1	0	1110				E=0
Q_N←1	1	1001				
SC ←SC-1				0		END

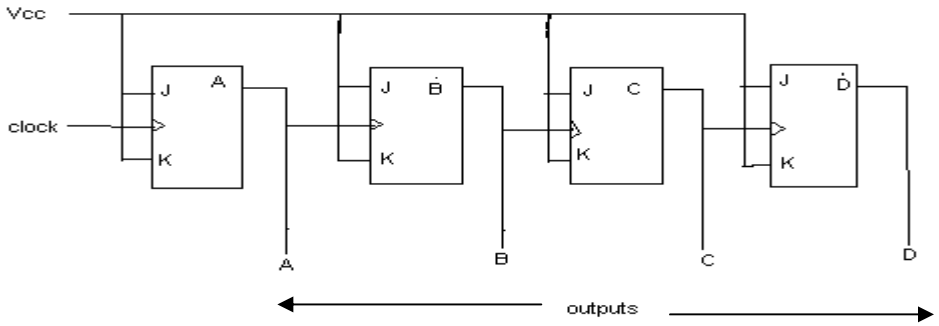
The quotient is in Q i.e $(1110)_2 = (14)_{10}$, Reminder is in a i.e $(1001)_2 = (9)_{10}$

and as we know $163 \div 11 = 14\frac{9}{11}$. Hence the restoring method is correct.

Q.181 Differentiate between synchronous and asynchronous counters. Explain the working of an asynchronous counter. **(6)**

Synchronous counter	***Asynchronous counter***
(i) The change of state of all FF_3 are synchronous with applied clock signal. (ii) As the state change takes place at one instant, so the propagation delay of FFs do not affect the performance. (iii) Design process is structured combinational circuit is designed to obtain required input for FFs. (iv) More reliable	(i) The change of state of all FFs are not synchronous with applied clock signal. The output of some FFs may act as an clock signal for other. (ii) Propagation delay of FFs may introduce some error in states (iii) Design process is not st4rucxtured one mainly combination circuit is designed to obtain the clock signal of FFs. (iv) less reliable.

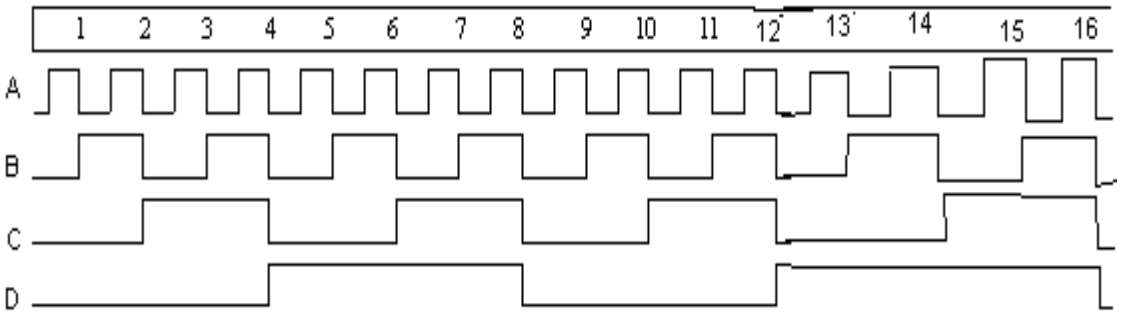
A four bit mod 16 asynchronous counter is shown below :-



clk	***Present state***				***Next state***			
	Q_D	Q_C	Q_B	Q_A	Q_D	Q_C	Q_B	Q_A
1	0	0	0	0	0	0	0	1
2	0	0	0	1	0	0	1	0
3	0	0	1	0	0	0	1	1
4	0	0	1	1	0	1	0	0
……	….	….	….	….	….	….	….	….
….	…	….	…..	….	….	….	….	….

15	1	1	1	0	1	1	1	1
16	1	1	1	1	0	0	0	0 (initial state)

Working Clock signal is connected to a clock input of LSB FF only. The J and K input terminals of each FFs are shorted.J & K are connected to V_{cc}. So the output Q_A toggles with negative transition of clock signal of the LSB FF.The output of A drives B, output of B drives C, and output of C drives D. When the output of a FF is used as clock input for the next flip-flop,we call the counter as asynchrous or ripple counter.In this type of counters all the flip-flops do not change their states at the same time.Because of this the overall propagation time is the sum of the induvisual delays.This counter gives the sequence of counting from 0000 to 1111.The timing diagram is given below:



Q.182 With the help of a neat diagram explain the working of a 4 – bit bidirectional shift register with parallel load. **(8)**

Ans:

A four bit bi-directional shift register with parallel load is shown above. Each state consist of a D-FF Q a 4x1 mux. Two selection inputs S_1l so select one of the multiplexer data input to the D FFs. The selection lines control the mode of operation of the register according to the function table given below :-

Mode control		Register Operation
S_1	S_0	
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	Parallel load

Q.183 Give the list of microoperations for implementing a subroutine call and return from a subroutine call, using stack. **(6)**

Ans:

When a subroutine is called in the main programme, the processor stores the return address in some specified location. It may be in the stock or in some specified memory location or in some specified register. Then the address of the first instruction of the subroutine is loaded into the program counter and the execution continues till the instruction for return to main program is encountered. On execution of return instruction, the processor loads the return address stored in specified location back to OPC, so that the main program continues its execution.

To implement this subroutine call return from subroutine call using stack the following micro operations are executed

Let CALL be the subroutine call instruction and RET be return to main program instruction.

For CALL SR

T_0 : AR ←PC

T_1 : Fetch: IR ← M[AR], PC ← PC + 1

T_2 : Decode D_0 …. D_7 ← IR (12-15), AR ← IR(0-11)

T_3 : TR← AR , Add. Of subroutine is stored in temporary register.

T_4 : AR ← SP
T_5 : M[AR] ← P, C } Return address is stored in stack where stack is build in R / W memory

T_6 : PC←TR, SC←0, address of first instruction of subroutine is loaded in program counter sequence counter is made zero.

For RET

T_0 : AR← PC ; Program counter is loaded in Add. Register.

T_1 : IR←M[AR], PC← PC+1 ; opcode fetch

T_2 : D_0 …D_7 ← IR[12-15], Decode.

T_3 : AR← SP ; Stack pointer address is transferred to Add. Register

T_4 : DR←M[AR]; M[AR] is transferred to Data Register.

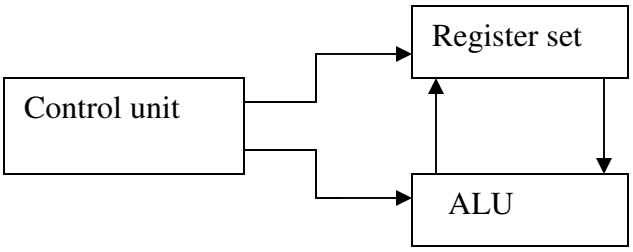
T_5 : PC ← DR, SC←0, Program counter is loaded with return address & sequence counter is made zero.

Q.184 Name the three types of CPU organizations. Explain about Accumulator organization with the help of examples. **(8)**

Ans:

The CPU controls the computer. It fetches instruction from memory, supplying the address and control signals needs by memory to access its data. The CPU decodes the instructions and controls the execution procedure. Internally CPU has three sections

(i) Register set **(ii)ALU** **(iii)Control unit**



Register Set : It includes a set of registers and a bus. The system address and data buses interact with this section of CPU. It also contains some special purpose registers not directly accessible by the programmer. The special purpose registers are – Instruction register, program counter, Address register, Data register, Stock pointer, Index register etc.

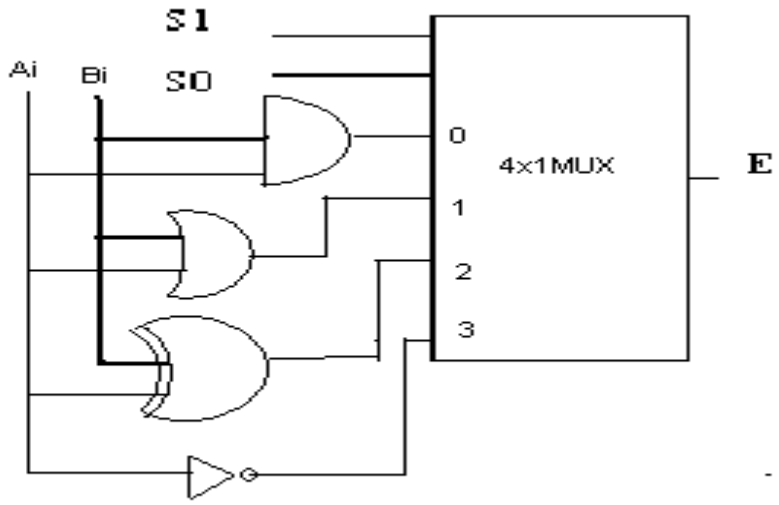
ALU : It performs moist arithmetic and logical operations such as add, subtract, complement, AND, OR, Ex-OR et. It receives its operand from register section of the CPU and stores its result back in register section.

Control Unit ; It controls all the operations of CPU. This unit generates the internal control signals that causes registers to load data, increment or clear their contents and output their contents, as well as cause the ALU to perform the correct function.

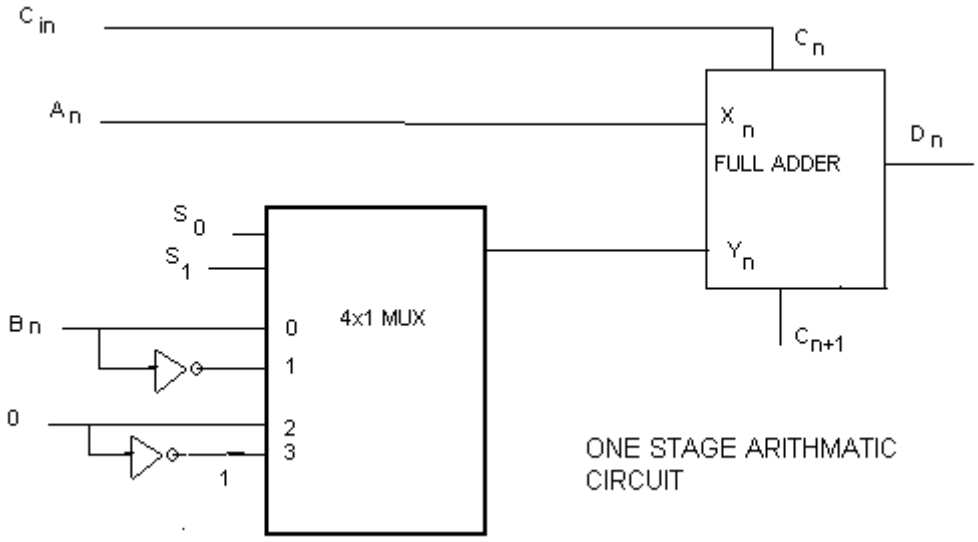
The control unit can be designed in two ways
(i) Hardwired control unit
(ii) Micro-programmed control unit

The hardware circuit for implementing logical operation is :-

S_1	S_0	*Output*	*Operation*
0	0	$E = A \wedge B$	AND
0	1	$E = A \vee B$	OR
1	0	$E = A \oplus B$	X OR
1	1	$E = \overline{A}$	Complement



The hard wave realization for implementing addition, substraction, increment, & decrement is given below (for one stage):



Function Table

S_1	S_0	C_{in}	*Output*	*Operation*
0	0	0	D = A + B	Add
0	0	1	D = A + B + 1	Add with carry
0	1	0	D = A + $\bar{B}$	Substract with borrow
0	1	1	D = A + $\bar{B}$ +1	Substract
1	0	0	D = A	Transfer A
1	0	1	D = A + 1	Increment A
1	1	0	D = A – 1	Decrement A
1	1	1	D = A	Transfer A

Q.185 What are pseudo operations? Name any three and explain about them. **(6)**

Ans:

Pseudo instructions are not assembly language instructions. These are instructions to the assembler. So the assembler does not convert these instructions to machine code. These are symbolic codes which specify the origin of the program, End of the program, type of the data like Decimal/Hex decimal etc.

Example: ORG, END, DEC, HEX

ORG: Its purpose is to specify on origin, that is, the memory location of the next instruction below it.

DEC: Decimal operands are specified following the symbol DCE, E, B, DEC-32, it means. The variable 'B' is a decimal number equal to -32,

END: It specifies the END of the assembly language program.

Q.186 A two-way set associative cache memory uses blocks of four words. The cache can accommodate a total of 2048 words from the main memory. The main memory size is $128\,K \times 32$.

(i) How many bits are there in the tag, index block and word fields of the address format?

(ii) What is the size of the cache memory? **(8)**

Ans:

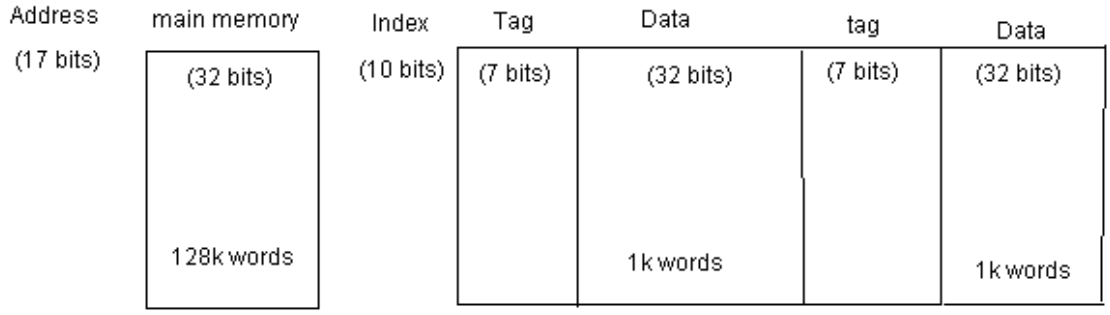
Size of cache memory = 2048 words = 2K words.

Size of main memory = $128\,K \times 32 = 2^7\,K \times 32$

Each word size of main memory = 32 bits.

Address length of main memory = 17 bits

Address length of cache memory = 10 bits (per set)



(i) Tag = 7 bits

Block index = (10-2) = 8 bit.

Word field = 2 bit as each block is of 4 words.

(ii) Word length of cache memory = 2(7+32) = 78 bits, size = (1024×78)

Q.187 What is pipelining? Name the two pipeline organizations. Explain about the arithmetic pipeline with the help of an example. **(6)**

Ans:

Pipe lining professing is an implementation technique where arithmetic sub-operations or the phases of a computer instruction cycle overlap are execution.

Pipe lining is a technique of decomposing a sequential process in to sub operations, with each sub process being executed is a special dedicated segment that operates concurrently with all other segments. A pipe line can be visualized as a collection of processing segments through which binary information flows. Each segment performs partial processing dictated by the way the task is partitioned the result obtained from the computation in each segment is transformed to the next segment in the pipe line. The final result obtained after the data have passed through all segments.

There are different types of pipe lining organizations

(i) Arithmetic pipe line

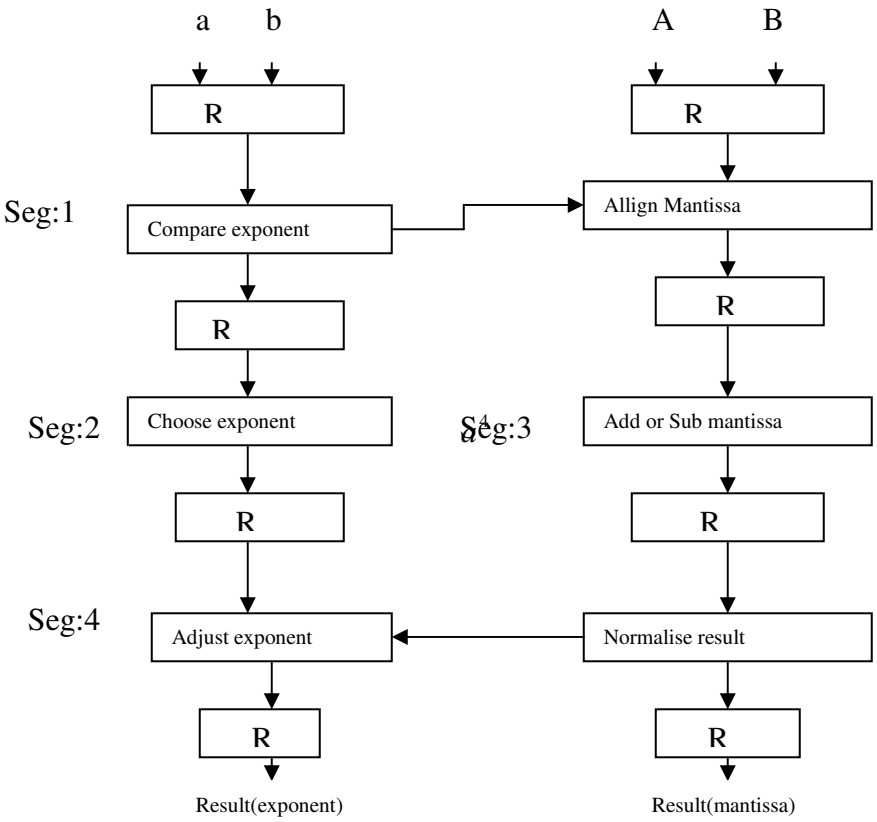
(ii) Instruction pipe line

Arithmetic Pipe Line

Let us consider arithmetic pipe line for floating point adder. Let two input are-

$$X = A \times 2^a$$
$$Y = B \times 2^b$$

Where A & B are two tractions that represents mantissas and a and b are the exponents. Arithmetic addition and subs traction can be performed in four segments.



. As per the flow chart

Segment-1: Two exponent are subtracted, to obtain

3-2 = 1

Segment-2: Longer exponent is selected.

Shift mantissa of Y to the night to obtain

$$X = 0.9504 \times 10^3$$
$$Y = 0.0820 \times 10^3$$

Segment-3: Addition of mantissa order same exponent results.

$$Z = 1.0324 \times 10^3$$

Segment -4: The sum is adjusted by normalizing the result.

$$Z = 0.10324 \times 10^4.$$

Q.188 What are the hazards of instruction pipelining? How are these taken care of. **(8)**

Ans:

There are three major difficulties that causes the instruction pipe line to deviate from its normal operation.

(i) Resource conflict: Caused by access to memory by two segments at the same time. Most of the conflict can be resolved by using separate instruction and data memories.

(ii) Data dependency: This conflict arises when an instruction depends on the result of a pervious instruction, but this result is not yet variable.

(iii) Branch difficulties: This arises from branch & other instructions that changes the value of PC.

The problem of data dependency can be solved through the followings.

Operand forwarding: The hardware avoid the conflict by routing the data through special paths between pipe line segments.

Through Compiler Programs: Insert the No. operation instruction in the program.

Handling branch difficulties: The methods used are –

(i) Prefetch target instructions

(ii) Use of branch target buffer

(iii) Use of loop buffer.

(iv) branch prediction

(v) Delayed branch.

Q.189 Explain the concept of parallel priority logic with four interrupt sources and give the logic diagram. **(8)**

Ans:

The parallel priority interrupt method uses a register whose bits are set separately by the interrupt signal from each device. Priority is established according to the position of the bits in the register. In addition to the interrupt register, the circuit may include a mark register whose purpose is to control the status of each interrupt request. The mark register can be programmed to disable lower priority interrupts while a higher priority device is being serviced. It can also provide a facility that allows a high priority device to interrupt the CPU, while a lower priority device is being serviced.

The interrupt registers individual bits are set by internal I_{10} devices requesting the service of CPU, and is cleared by program instructions. The I/O devices are origin

with some priority value depending on their nature of devices and the services rendered by them. For example magnetic disk may get higher priority than a printer.

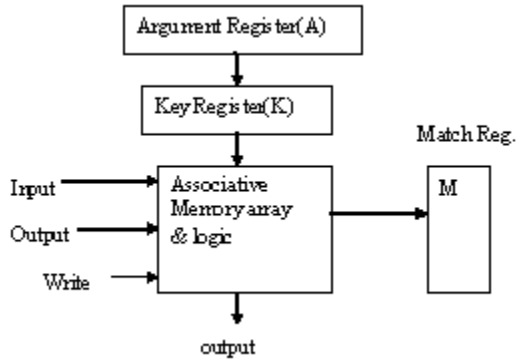
The mark register has same no. of bits as that of interrupt register. By means of program, it is possible to set or next any bit of the mark register. If its is 1, then the associated interrupt is recognized, other wise it is treated to be marked. Each interrupt bit along with its mark bit are applied to a AND gate to produce four inputs to a priority encoder. In this way the interrupts are recognized by CPU. The priority encoder output decides the vector address of the interrupt service subroutine, (ISR) which is to be loaded in to PC for executor of ISR darning interrupt cycle. Another output of priority encoder sets an interrupt status flip-flop (IST FF), when an interrupt is recognized. The interrupt enable FF(IEN) can be set or cleared by the program to provide an overall control over the interrupt system. It $I_{EN} = 1$, than interrupt is recognized by CPU other wise not.

If IST = 1 & IEN = 1. Than the interrupt signal goes to CPU, in return, CPU bends interrupt acknowledgement signal, which enables the vector address register to place the vector address of ISR in to program computer.

Q.190 Draw the logic diagram of the cell of one word in associative memory including the read and write logic. **(6)**

Ans:

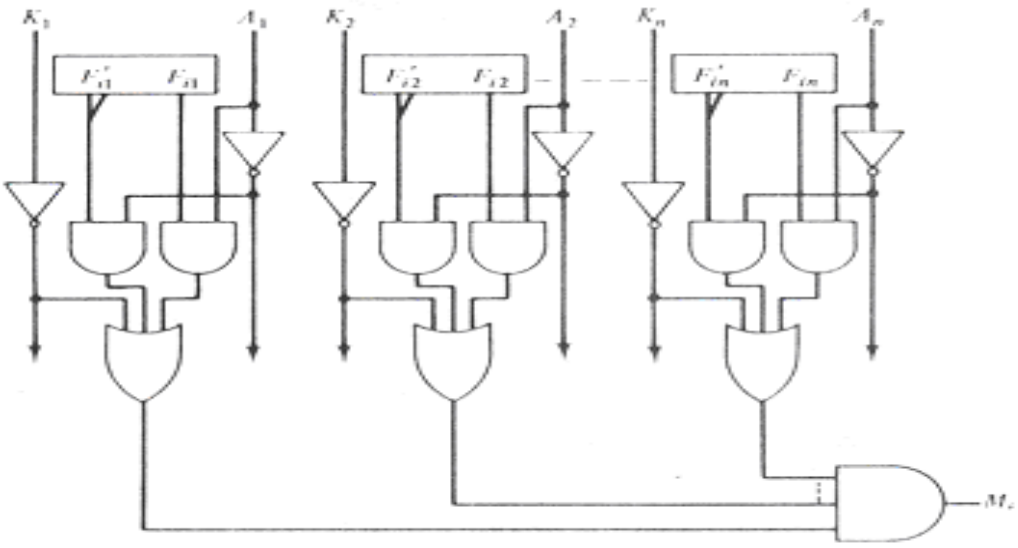
he logic diagram of the cell of one word in associative memory including the read and



write logic is sown below:-

This consists of a memory array of logic for 'M' words with n-bits per word. The argument register A. The key register K each has n-bits, one for each bit of a ward. The mach register 'M' has m bits, one for each memory word. Each word in the memory is compared in parallel with the content of the argument register. Words that match the bits of the argument register set a corresponding bit in the match register. After the matching process, those bits in the match register that have been set indicate the fact that their corresponding words have been matched.

The circuit. for match logic for one word of associative memory is given below:-



Match logic for one word of associative memory.

Q.191 Explain the working of a Microprogram Sequencer with the help of a neat diagram. **(8)**

Ans:

the function of control unit in a digital computer is to initiate sequences of micro-operations. When the control signals are generated by hardware using conventional logic design techniques, the control unit is said to be hard wined . Micro programming is the second alterative.

The control function that specifies micro operations is a binary variable. These binary control variables are stored in memory is called a micro programmed control unit. A sequence of micro instructions constitutes a micro program. Each machine instruction initiates a service of micro instructions in control memory. These micro instructions generates the micro operations to fetch the instruction from main memory to evaluate the effective address, to execute the operation specified by the instruction and to return control to fetch phase in order to repeat the cycle for new instruction. Control memory address register specifies the address of the micro interaction read from memory. The micro instruction contains a control word that specifies one or more micro operations for the data processes. Once these operations are excreted, the control must determine the next address. The location of the next micro instruction may be the one next in sequence or it may be located some where else in the control memory. For this reason it is necessary to use some bits of the present micro instruction to control the generation of the address of the next micro instruction. The next address may also be a function of external input conditions. The next address is computed by the circuit is called micro program sequences. The typical functions of a micro program sequences are

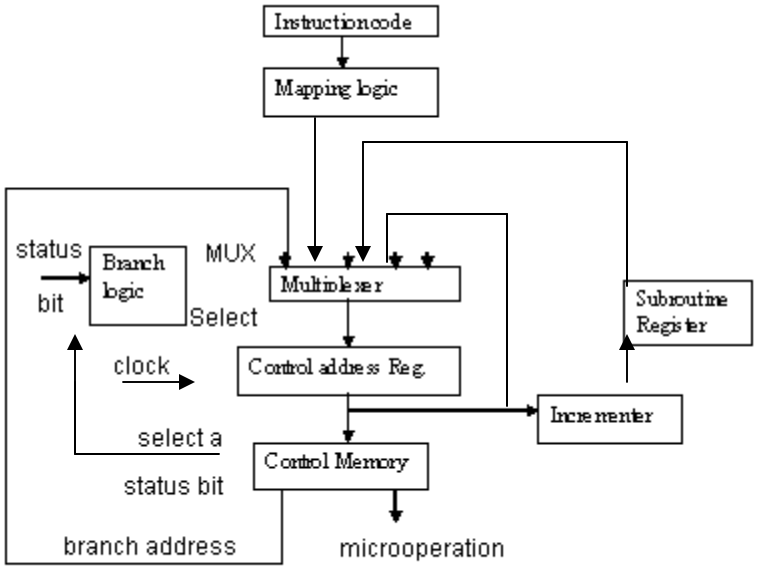
(i) Incrementing the control address registers by one.

(ii) Loading into the control address register an address from control memory

(iii) Transferring an external address loading an initial address to start control operations.

The sequencer should also have a facility for subroutine call and return.

This is shown in the following diagram.



Q.192 Name and explain the fields of a symbolic microinstruction. **(6)**

Ans:
The format of a micro operation is shown below:-

F1	F2	F_3	Condition for branching	Branch field	Address

Where, F_1, F_2, F_3 are micro operation field. The bit pattern of each filed are encoded to specify distinct micro operations. If there are three fields, them it means that no more than three micro operations can be chosen for a micro instruction.If fewer than three micro operations are used, then one or more of the fields will use the binary code for no operation.

Condition field

It consists of bit pattern, which is encoded to specify the status bit condition.

Branch field:

Its bit pattern provides an unconditional branch operation. The branch field bits are used in conjunction with the address field to choose the address of next micro instruction.

Q.193 Write an assembly language program to sort the given list in ascending order. The length of the series is given in the memory location 203FH and the series itself starts from 2040H onwards. **(10)**

Ans:

Length of the series is in memory location -203F H. The series starts from 2040_H on words.

Aim: To arrange the series in ascending order.

Program:-

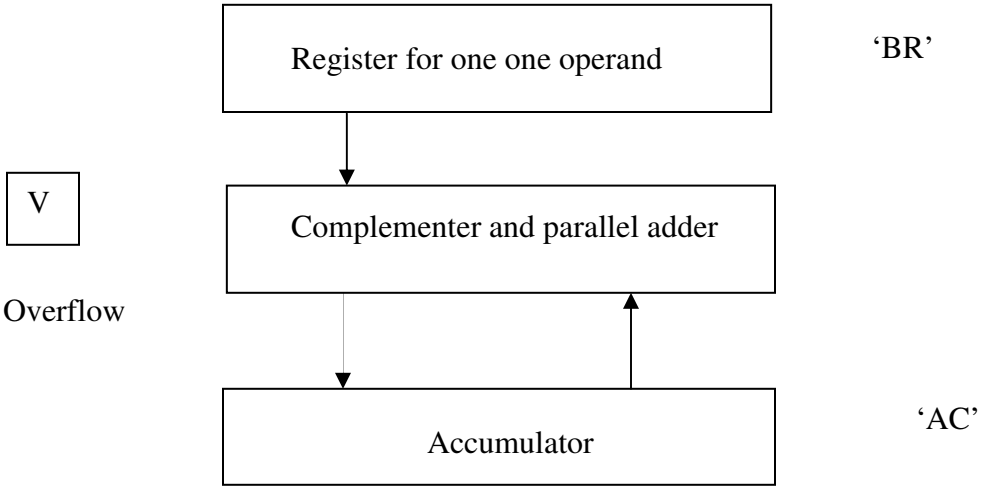
```
LXI  H,203F
MOV B,M        :Length of the series is in reg B
DCR B
INX H
LOOP1: MOV C,B      : (Length of the series -1) is in reg C
LOOP2: MOV A,M   :First element of the series goes to Accumulator
    INX H
    CMP M        :comparison between two elements
    JNC  LOOP3
    MOV D,M
     MOV M,A
     DCX H
     MOV M,D
     INX H
LOOP3:    DCR C
     JNZ  LOOP2
     LXI H,2040
     DCR B
     JNZ LOOP1
     HLT
```

This programme gives the series in ascending order in the same location.

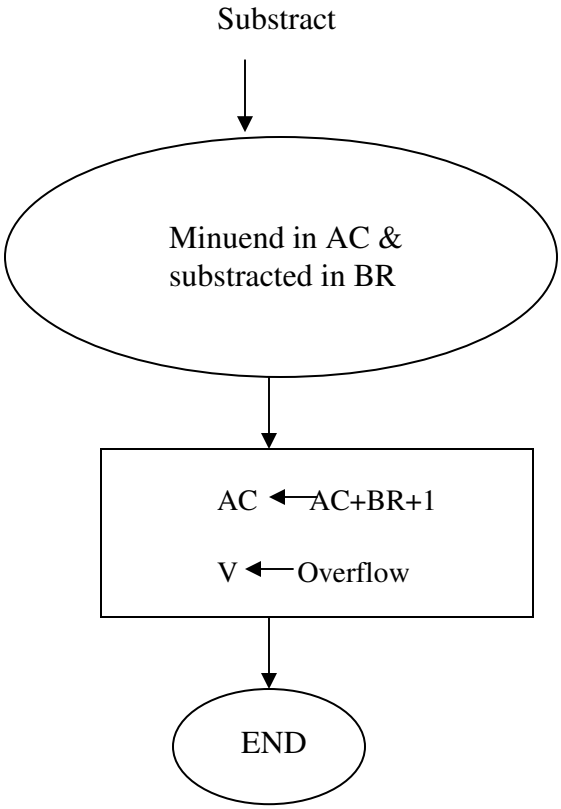
Q.194 Give an algorithm for subtracting numbers in signed 2's complement form. **(4)**

Ans:

The hardware arrangement for signed 2's complement addition & subs-traction is given below.



The algorithm is shown in the chart



TYPICAL QUESTIONS & ANSWERS

PART - I

OBJECTIVE TYPE QUESTIONS

Each Question carries 2 marks.

Choose correct or the best alternative in the following:

Q.1 What is the output of the following program?

```
main ( )
{     int x = 2, y = 5;
       if (x < y) return (x = x+y); else printf ("z1");
       printf("z2");
}
```

(A) z2 **(B)** z1z2
(C) Compilation error **(D)** None of these

Ans: D
There is no compilation error but there will no output because function is returning a value and if statement is true in this case.

Q.2 Choose the correct one

(A) Address operator can not be applied to register variables
(B) Address operator can be applied to register variables
(C) Use of register declaration will increase the execution time
(D) None of the above

Ans: D
A register access is much faster than a memory access, keeping the frequently accessed variables in the register will lead to faster execution of programs.

Q.3 What is the following program doing?

```
main ()
{ int d = 1;
do
            printf("%d\n", d++);
while (d < = 9);}
```

(A) Adding 9 integers **(B)** Adding integers from 1 to 9
(C) Displaying integers from 1 to 9 **(D)** None of these

Ans: C
d starting from 1 is incrementing one by one till d=9 so the printf statement is printing numbers from 1 to 9.

Q.4 What is the output of the following program?

```
main ( )
{    extern int x;
      x = 20;
       printf("\n%d", x);
}
```

(A) 0 **(B)** 20
(C) error **(D)** garbage value

Ans: C
Output of the given program will be "Linker error-undefined symbol x". External variables are declared outside a function.

Q.5 If x is one dimensional array, then pick up the correct answer
(A) *(x + i) is same as &x[i] **(B)** *&x[i] is same as x + i
(C) *(x + i) is same as x[i] +1 **(D)** *(x + i) is same as *x[i]

Ans: A
num[i] is same as *(num+i)

Q.6 Consider the following declaration

```
int a, *b = &a, **c = &b;
```

The following program fragment

```
a = 4;
**c = 5;
```

(A) does not change the value of a **(B)** assigns address of c to a
(C) assigns the value of b to a **(D)** assigns 5 to a

Ans: D
The given statements assigns 5 to a

Q.7 Choose the correct answer
(A) enum variable can not be assigned new values
(B) enum variable can be compared
(C) enumeration feature increase the power of C
(D) None of the above

Ans: C
The enumerated data types give an opportunity to invent our own data typeand define what value the variable of this data type can take.

Q.8 The content of file will be lost if it is opened in
(A) w mode **(B)** w+ mode
(C) a mode **(D)** a+ mode

Ans: A
When the mode is writing, the contents are deleted and the file is opened as a new file.

Q.9 Consider the following code segment:

```
int a[10], *p1, *p2;
p1 = &a[4];
p2 = &a[6];
```

Which of the following statements is incorrect w.r.t. pointers?
(A) p1 + 2 **(B)** p2 – 2
(C) p2 + p1 **(D)** p2 – p1

Ans: C

Addition of two pointers is not allowed.

Q.10 The second expression (j – k) in the following expression will be evaluated

```
(i + 5) && (j - k)
```

(A) if expression (i + 5) is true.
(B) if expression (i + 5) is false.
(C) irrespective of whether (i + 5) is true or false.
(D) will not be evaluated in any case.

Ans: A
In a compound logical expression combined with &&, the second expression is evaluated only if first is evaluated in true.

Q.11 In the **for** statement: `for (exp1; exp2; exp3) { … }`
where exp1, exp2 and exp3 are expressions. What is optional?
(A) None of the expressions is optional.
(B) Only exp1 is optional.
(C) Only exp1 and exp3 are optional.
(D) All the expressions are optional.

Ans: D
All the expressions are optional. For (;;) is a valid statement in C.

Q.12 The output of the following code segment will be

```
char x = 'B';
switch (x) {
case 'A': printf("a");
case 'B': printf("b");
case 'C': printf("c");
}
```

(A) B **(B)** b
(C) BC **(D)** bc

Ans: D
Since there is no break statement, all the statement after case'B' are executed.

Q.13 What will be the output of the following code segment?

```
main( ) {
char  s[10];
strcpy(s, "abc");
printf("%d %d", strlen(s), sizeof(s));
}
```

(A) 3 10 **(B)** 3 3
(C) 10 3 **(D)** 10 10

Ans: A
strlen(s) give the length of the string, that is 3 and sizeof(s) give the size of array s that is 10.

Q.14 Which of the following is the odd one out?
(A) j = j + 1; **(B)** j =+ 1;
(C) j++; **(D)** j += 1;

Ans: B
j=+1 is odd one out as rest all means incrementing the value of variable by 1.

Q.15 Which of the following is true for the statement:

```
NurseryLand.Nursery.Students = 10;
```

(A) The structure Students is nested within the structure Nursery.
(B) The structure NurseryLand is nested within the structure Nursery.
(C) The structure Nursery is nested within the structure NurseryLand.
(D) The structure Nursery is nested within the structure Students.

Ans: C
The structure Nursery is nested within the structure NurseryLand.

Q.16 What will be the output of the following code segment, if any?

```
    myfunc ( struct test t) {
    strcpy(t.s, "world");
    }
 main( ) {
struct test { char s[10]; } t;
    strcpy(t.s, "Hello");
    printf("%s", t.s);
    myfunc(t);
    printf("%s", t.s);
}
```

(A) Hello Hello
(B) world world
(C) Hello world
(D) the program will not compile

Ans: D
The program will not compile because undefined symbol s for myfunc( ) function. Structure should be defined before the main and the function where it is called.

Q.17 If a function is declared as **void fn(int *p)**, then which of the following statements is valid to call function **fn**?

(A) **fn(x)** where x is defined as **int x;**
(B) **fn(x)** where x is defined as **int *x;**
(C) **fn(&x)** where x is defined as **int *x;**
(D) **fn(*x)** where x is defined as **int *x;**

Ans: B
Function **void fn(int *p)** needs pointer to int as argument. When x is defined as int *x, then x is pointer to integer and not *x.

Q.18 What is the following function computing? Assume a and b are positive integers.

```
 int fn( int a, int b) {
   if (b == 0)
         return b;
   else
         return (a * fn(a, b - 1));
}
```

(A) Output will be 0 always
(B) Output will always be b
(C) Computing a^b
(D) Computing a + b

Ans: A
The output is always be 0 because b is decremented in recursive function fn each time by 1 till the terminating condition b==0 where it will return 0.

Q.19 What is the output of the following C program?

```
# include <stdio.h>
main ( )
{
 int a, b=0;
 static int c [10]={1,2,3,4,5,6,7,8,9,0};
 for (a=0; a<10;+ + a)
 if ((c[a]%2)= = 0) b+ = c [a];
 printf ("%d", b);
  }
```

(A) 20 **(B)** 25
(C) 45 **(D)** 90

Ans: A
printf statement will print b which is sum of the those values from array c which get divided by 2, that is 2+4+6+8=20.

Q.20 If a, b and c are integer variables with the values a=8, b=3 and c=-5. Then what is the value of the arithmetic expression:

```
2 *  b + 3 * (a-c)
```

(A) 45 **(B)** 6
(C) -16 **(D)** -1

Ans: A
the value of the arithmetic expression is 45 as 2*3+3*(8—5)=6+3*13=6+39=45

Q.21 A global variable is a variable
(A) declared in the main ( ) function.
(B) declared in any function other than the main ( ) function.
(C) declared outside the body of every function.
(D) declared any where in the C program.

Ans: C
A global variable is declared outside the body of every function.

Q.22 main ( ) is an example of
(A) library function **(B)** user defined function
(C) header **(D)** statement

Ans: A
main() is a special function used by C system to tell the computer where the program starts.

Q.23 While incrementing a pointer, its value gets increased by the length of the data type to which it points. This length is called
(A) scale factor **(B)** length factor
(C) pointer factor **(D)** increment factor
Ans: D

While incrementing a pointer, its value gets increased by the length of the data type to which it points.

Q.24 The first digit of a decimal constant must be

(A) a zero **(B)** a non zero number
(C) a negative number **(D)** an integer

Ans: D
Decimal constants consist of a set of digit, 0 to 9, preceded by an optional – or + sign.

Q.25 What is the output of the following statement:

```
printf ("%-3d",12345);
```

(A) 1 2 3 **(B)** -1 2 3
(C) 1 2 3 4 5 **(D)** 12

Ans: C
printf statement would print 12345.

Q.26 A single character input from the keyboard can be obtained by using the function.

(A) printf ( ) **(B)** scanf ( )
(C) putchar ( ) **(D)** getchar ( )

Ans: D
Reading a single character can be done by using the function getchar( ).

Q.27 The function ftell ( )

(A) reads a character from a file
(B) reads an integer from a file
(C) gives the current position in the file
(D) sets the position to the beginning of the file.

Ans: C
ftell( ) takes a file pointer and returns a number of type long, that corresponds to the current position.

Q.28 If the variables i, j and k are assigned the values 5,3 and 2 respectively, then the expression `i = j + ( k + + = 6 ) + 7`

(A) gives an error message **(B)** assigns a value 16 to i
(C) assigns a value 18 to i **(D)** assigns a value 19 to i

Ans: A
It gives an error message-Lvalue required.

Q.29 If an integer needs two bytes of storage, then the maximum value of a signed integer is

(A) 2^{16}-1 **(B)** 2^{15}-1
(C) 2^{16} **(D)** 2^{15}

Ans: B
If we use a 16 bit word length, the size of the integer value is limited to the range -2^{15} to 2^{15}-1

Q.30 Literal means

(A) a string **(B)** a string constant
(C) a character **(D)** an alphabet

Ans: B
Literal means a string constant.

Q.31 If 'y' is of integer type then the expressions
$3*(y-8)/9$ and $(y-8)/9*3$
(A) must yield the same value.
(B) must yield different values.
(C) may or may not yield the same value.
(D) none of the above.

Ans: C
The expression may or may not yield the same value.

Q.32 In the following code fragment

```
int x, y = 2, z, a;
x=(y*=2) + (z=a=y);
printf ('%d',x);
```

(A) prints 8
(B) prints 6
(C) prints 6 or 8 depending on the compiler
(D) is syntactically wrong

Ans: A
It will print 8 because x=(y*=2)+(z=a=y)=4+4=8.

Q.33 A possible output of the following program fragment is

```
for (i=getchar();; i=get.char())
          if (i=='x') break;
          else putchar(i);
```

(A) mi **(B)** mix
(C) mixx **(D)** none of the above

Ans: D
None of the above as it is wrong syntax.

Q.34 In a for loop, if the condition is missing, then,
(A) It is assumed to be present and taken to be false.
(B) It is assumed to be present and taken to be true.
(C) It results in a syntax error.
(D) Execution will be terminated abruptly.

Ans: B

Q.35 If storage class is missing in the array definition, by default it will be taken to be
(A) automatic
(B) external
(C) static

(D) either automatic or external depending on the place of occurrence.

Ans: A
A variable declared inside inside a function without storage class specification is, by default, an automatic variable.

Q.36 The maximum number of dimensions an array can have in C is
(A) 3 **(B)** 4
(C) 5 **(D)** compiler dependent

Ans: D
C allows arrays of three or more dimensions. The exact limit is determined by the compiler.

Q.37 puts(argv[0]);
(A) prints the name of the source code file.
(B) prints argv.
(C) prints the number of command line arguments.
(D) prints the name of the executable code file.

Ans: D
argv[0] represent the filename where the executable code of the program is stored.

Q.38 `printf("%-10s", "ABDUL");` displays
(A) ABDUL␣␣␣␣␣ **(B)** ␣␣␣␣␣ABDUL
(C) ABDUL␣␣␣␣␣␣␣␣␣␣ **(D)** ␣␣␣␣␣␣␣␣␣␣ABDUL

Ans: A
-10s will print ABDUL in 10 space ABDUL followed by 5 blank space.

Q.39 Which amongst the following expression uses bitwise operator?
(A) a++ **(B)** !a>5
(C) a|b **(D)** a!=b

Ans: C
| is bitwise OR.

Q.40 The output of the following program is

```
main( )
{    float y;
     y=198.7361;
     printf("%7.2f", y);
}
```

(A)

1	9	8	.	7	3	6

(B)

1	9	8	.	7	3	

(C)

	1	9	8	.	7	4

(D)

1	9	8	.	7	4	

Ans: C
The printf statement is giving formatted output till two places of decimal.

Q.41 Which is not dynamic memory allocation function?

(A) malloc **(B)** free
(C) alloc **(D)** calloc

Ans: C
Three dynamic memory allocation functions are: malloc, calloc and free

Q.42 Which header file is used for screen handling function:-

(A) IO.H **(B)** STDLIB.H
(C) CONIO.H **(D)** STDIO.H

Ans: D
The header file stdio.h contains definitions of constants,macros and types, along with function declarations for standard I/O functions.

Q.43 Choose the directive that is used to remove previously defined definition of the macro name that follows it -

(A) # remdef **(B)** # pragma
(C) # undef **(D)** # define

Ans: C
The preprocessor directive #undef OKAY would cause the definition of OKAY to be removed from the system.

Q.44 The output of the following is

```
x = 'a';
printf("%d", x);
```

(A) 'a' **(B)** a
(C) 97 **(D)** None of the above

Ans: C
The printf statement is printing ascii value of a, that is 97.

Q.45 Consider the following statement

```
int j, k, p;
float q, r, a;
a = j/k;
p=q/r;
If q=7.2, r=20, j=3, k=2
```

The value of a and p is

(A) a=1.5, p=3.6 **(B)** a=2, p=3
(C) a=1.5, p=4 **(D)** a=1, p=3

Ans: C
a=3/2=1.5 and p=q/r=7.2/2=3.6 is rounded off to 4.

Q.46 Choose the function that returns remainder of x/y -

(A) remainder( ) **(B)** mod( )
(C) modulus( ) **(D)** rem( )

Ans: C

modulus( ) function produces the reminder of an integer division.

Q.47 What is the output of following program:-

```
int q, *p, n;
q = 176;                If the address of q is 2801
p = &q;                 and p is 2600
n = *p;
printf("%d", n);
```

(A) 2801 **(B)** 176
(C) 2600 **(D)** None of the above

Ans: B
n is assigned a the value which is present at the address of q and that value is 176.

Q.48 Consider the following statements-

```
x = 5;
y = x >3 ? 10 : 20;
```

The value of y is
(A) 10 **(B)** 20
(C) 5 **(D)** 3

Ans: A
Since x=5 is greater than 3 so y is assigned value 10. It is equivalent to if-else statements.

Q.49 Determine which of the following is an invalid character constant.
(A) '\a' **(B)** 'T'
(C) '\0' **(D)** '/n'

Ans: D
newline character constant is "\n" not "/n".

Q.50 What is the name of built-in function for finding square roots?
(A) square(x) **(B)** sqr(x)
(C) sqrt(x) **(D)** No built-in function

Ans: C
sqrt(x) is a built-in function for finding square roots.

Q.51 What is the output of following statement?

```
for(i=1; i<4; i++)
printf("%d",(i%2) ? i : 2*i);
```

(A) 1 4 3 **(B)** 1 2 3
(C) 2 4 6 **(D)** 2 2 6

Ans: A
for i=1, (i%2) is true so the statement will print 1; for for i=2, (i%2) is false so the statement will print 2*i=2*2=4; for for i=3, (i%2) is again true so the statement will print 3; for i=4, the statement is out from the loop.

Q.52 Which of the following statement is true about a function?
(A) An invoking function must pass arguments to the invoked function.
(B) Every function returns a value to the invoker.

(C) A function may contain more than one return statement.
(D) Every function must be defined in its own separate file.

Ans: A
An invoking function must pass arguments to the invoked function.

Q.53 What is the output of the following program?

```
main( )
{
    int i=4, z=12;
    if(i=5 || z>50)
       printf("hello");
    else
       printf("hye");
}
```

(A) hello **(B)** hye
(C) syntax error **(D)** hellohye

Ans: A
i=5 will assign value 5 to i so the if statement (i=5||z>50) is true so "printf" statement will print hello.

Q.54 For implementing recursive function the data structure used is:
(A) Queue **(B)** Stack
(C) Linked List **(D)** Tree

Ans: B
For implementing recursive function, stack is used as a data structure.

Q.55 The size of array int a[5]={1,2} is
(A) 4 **(B)** 12
(C) 10 **(D)** 6

Ans: C
The size of int array is 2*5=10 bytes as int takes 2 bytes of storage.

Q.56 Which of the following is not an escape sequence?
(A) \n **(B)** \r
(C) \' **(D)** \p

Ans: D
\p is not an escape sequence.

Q.57 The output of the following statements is

```
char ch[6]={'e', 'n', 'd', '\0', 'p'};
printf("%s", ch);
```

(A) endp **(B)** end0p
(C) end **(D)** error

Ans: C
printf statement will print end because string is terminated at "\0" and in array after d, we have null character.

Q.58 How many times the following code prints the string "hello".

```
for(i=1; i<=1000; i++);
printf("hello");
```

(A) 1 **(B)** 1000
(C) Zero **(D)** Syntax error

Ans: A
The "for" loop is terminated by a semicolon so the next statement is execute that is printing hello.

Q.59 Find the invalid identifiers from the following:-
(i) nA (ii) 2nd (iii) ROLL NO (iv) case
(A) (i), (ii) and (iv) **(B)** (i) and (iii)
(C) (ii), (iii) and (iv) **(D)** (ii), (i) and (iii)

Ans: C
Identifier cannot start with a digit; it cannot have a space and case is a keyword.

Q.60 The void type is used for

(A) Returning the value **(B)** creating generic pointers
(C) Creating functions **(D)** Avoid error

Ans: B
The void type is used to create generic pointers.

Q.61 The valid octal constants from the following
(i) 0245 (ii) 0387 (iii) 04.32 (iv) –0467
(A) (i) and (ii) **(B)** (iii) and (iv)
(C) (ii) and (iii) **(D)** (i) and (iv)

Ans: D
(i) and (iv) are valid octal constants.

Q.62 The variable that are declared outside all the functions are called _____.
(A) Local variable **(B)** Global variable
(C) Auto variable **(D)** None of the above
Ans: B
The variables that are declared outside all functions are called global variable.

Q.63 Consider the following statements:-

```
int x = 6, y=8, z, w;
y = x++;
z = ++x;
```

The value of x,y,z by calculating the above expressions are:-
(A) y=8, z=8, x=6 **(B)** y=6, x=8, z=8
(C) y=9, z=7, x=8 **(D)** y=7, x=8, z=7

Ans: B
y is assigned value of x that is 6, then x in incremented that is value of x=7, z is assigned value of x after incrementing that is z =8 so value of x =8.

Q.64 To declare an array S that holds a 5-character string, you would write

(A) char S[5] **(B)** String S[5]
(C) char S[6] **(D)** String S[6]

Ans: A
A string is nothing but a char array.

Q.65 The function used to read a character from a file that has been opened in read mode is

(A) putc **(B)** getc
(C) getchar **(D)** putchar

Ans: B
getc is used to read a character from a file that has been opened in read mode.

Q.66 The function that allocates requested size of bytes and returns a pointer to the first byte of the allocated space is -

(A) realloc **(B)** malloc
(C) calloc **(D)** none of the above

Ans: B
malloc allocates requested size of bytes and returns a pointer to the first byte of the allocated space.

Q.67 The constructed datatype of C is known as

(A) Pointers **(B)** String
(C) Structure **(D)** Array

Ans: C
Structure is a constructed datatype of C

Q.68 The postfix form of the following infix notation is : $(A+B)*(C*D-E)*F$

(A) $AB+CD*E-*F*$ **(B)** $AB+CDE+-*F*$
(C) $AB+CD-EF+-**$ **(D)** $ABCDEF*-+*+$

Ans: (A)

Q.69 The number of nodes in a complete binary tree of depth d (with root at depth 0) is

(A) $2^{d-1}+1$ **(B)** $2^{d+1}-1$
(C) $2^{d-1}-1$ **(D)** $2^{d+1}+1$

Ans: (B)

Q.70 The average case of quick sort has order

(A) $O(n^2)$ **(B)** $O(n)$
(C) $O(n \log n)$ **(D)** $O(\log n)$

Ans: (C)

Q.71 Inorder to get the information stored in a BST in the descending order, one should traverse it in which of the following order?

(A) left, root, right **(B)** root, left, right
(C) right, root, left **(D)** right, left, root

Ans: (C)

Q.72 Every internal node in a B-tree of minimum degree 2 can have

(A) 2, 3 or 4 children **(B)** 1, 2 or 3 children
(C) 2, 4 or 6 children **(D)** 0, 2 or 4 children

Ans: (B)

Q.73 Which sorting algorithm is the best if the list is already in order?

(A) Quick sort **(B)** Merge sort
(C) Insertion sort **(D)** Heap sort

Ans: (C)

Q.74 In _____ the difference between the height of the left sub tree and height of right sub tree, for each node, is not more than one

(A) BST **(B)** Complete Binary Tree
(C) AVL-tree **(D)** B-tree

Ans: (C)

Q.75 The number of comparisons required to sort 5 numbers in ascending order using bubble sort is

(A) 7 **(B)** 6
(C) 10 **(D)** 5

Ans: (C)

Q.76 The complexity of adding two matrices of order m*n is

(A) m + n **(B)** mn
(C) max(m, n) **(D)** min(m, n)

Ans: (B)

Q.77 The second largest number from a set of n distinct numbers can be found in

(A) O(n) **(B)** O(2n)
(C) $O(n^2)$ **(D)** O(log n)

Ans: (A)

Q.78 If the inorder and preorder traversal of a binary tree are D,B,F,E,G,H,A,C and A,B,D,E,F,G,H,C respectively then the postorder traversal of that tree is

(A) D,F,G,A,B,C,H,E **(B)** F,H,D,G,E,B,C,A
(C) C,G,H ,F,E,D,B,A **(D)** D,F,H,G,E,B,C,A

Ans: (D)

Q.79 In a binary tree, the number of terminal or leaf nodes is 10. The number of nodes with two children is

(A) 9 **(B)** 11
(C) 15 **(D)** 20

Ans: (A)

Q.80 Which amongst the following cannot be a balance factor of any node of an AVL tree?

(A) 1 **(B)** 0
(C) 2 **(D)** –1

Ans: (C)

Q.81 How many distinct binary search trees can be formed which contains the integers 1, 2, 3?

(A) 6 **(B)** 5
(C) 4 **(D)** 3

Ans: (B)

Q.82 The sort which inserts each elements A(K) into proper position in the previously sorted sub array A(1), ..., A(K–1)

(A) Insertion sort **(B)** Radix sort
(C) Merge sort **(D)** Bubble sort

Ans: (A)

Q.83 Direct or random access of elements is not possible in

(A) Linked list **(B)** Array
(C) String **(D)** None of these

Ans: (A)

Q.84 Level of any node of a tree is

(A) Height of its left subtree minus height of its right subtree
(B) Height of its right subtree minus height of its left subtree
(C) Its distance from the root
(D) None of these

Ans: (C)

Q.85 A desirable choice for the partitioning element in quick sort is

(A) First element of the list
(B) Last element of the list
(C) Randomly chosen element of the list
(D) Median of the list

Ans: (A)

Q.86 lg (n!) =_____

(A) O (n) **(B)** O (lg n)
(C) O (n^2) **(D)** O (n lg n)

Ans: (D)

$n! = n(n-1)(n-2)\text{-----}3X2X1$
$\geq (n/2)^{n/2}$
$\log n! \geq n/2 \log n/2$
$\geq n/2(\log n - \log 2)$
$\geq n/2 (\log n - 1)$
$\leq n \log n$
$= O(n \log n)$

Q.87 The result of evaluating the following postfix expression is
5, 7, 9, *, +, 4, 9, 3, /, +, -

(A) 50 **(B)** 65
(C) 61 **(D)** 69

Ans: (C)

Q.88 A graph with **n** vertices will definitely have a parallel edge or self loop if the total number of edges are

(A) more than **n** **(B)** more than **n+1**
(C) more than **(n+1)/2** **(D)** more than **n(n-1)/2**

Ans: (D)

Q.89 Out of the following, the slowest sorting procedure is

(A) Quick Sort **(B)** Heap Sort
(C) Shell Sort **(D)** Bubble Sort

Ans: (D)

Q.90 In _____, it is possible to traverse a tree without using stacks either implicitly or explicitly.

(A) Threaded binary trees. **(B)** AVL Tree
(C) B^+ tree **(D)** Heap

Ans: (C)

Q.91 The order of a B-Tree with 2, 3, 4 or 5 children in every internal node is

(A) 2 **(B)** 3
(C) 4 **(D)** 5

Ans: (C)

Q.92 The number of nodes that have no successors in a complete binary tree of depth 4 is

(A) 0 **(B)** 8
(C) 16 **(D)** 4

Ans: (B)

Q.93 One can make an exact replica of a Binary Search Tree by traversing it in
(A) Inorder **(B)** Preorder
(C) Postorder **(D)** Any order

Ans: (B)

Q.94 A complete Binary Tree with 15 nodes contains_____edges
(A) 15 **(B)** 30
(C) 14 **(D)** 16

Ans: (C)

Q.95 The minimum number of comparisons required to find the largest number from 4 different numbers are
(A) 4 **(B)** 3
(C) 5 **(D)** 6

Ans: (B)

Q.96 An infix expression can be converted to a postfix expression using a
(A) Stack **(B)** Queue
(C) Dequeue **(D)** None of these

Ans: (A)

Q.97 A data structure in which an element is added and removed only from one end, is known as
(A) Queue **(B)** Stack
(C) In-built structure **(D)** None of the above

Ans: (B)

Q.98 A complete binary tree with the property that the value of each node is at least as large as the values of its children is known as
(A) Binary Search Tree. **(B)** AVL Tree.
(C) Heap. **(D)** Threaded Binary Tree.

Ans: (C)

Q.99 A sorting algorithm is stable if
(A) its time complexity is constant irrespective of the nature of input.
(B) preserves the original order of records with equal keys.
(C) its space complexity is constant irrespective of the nature of input.
(D) it sorts any volume of data in a constant time.

Ans: (B)

Q.100 A tree in which, for every node, the difference between the height of its left subtree and right subtree is not more than one is

(A) AVL Tree. **(B)** Complete Binary Tree.

(C) B – Tree. **(D)** B^+ Tree.

Ans: (A)

Q.101 The data structure needed to convert a recursion to an iterative procedure is

(A) Queue. **(B)** Graph.

(C) Stack. **(D)** Tree.

Ans: (C)

Q.102 A binary tree stored using linked representation can be converted to its mirror image by traversing it in

(A) Inorder. **(B)** Preorder.

(C) Postorder. **(D)** Any order.

Ans: (B)

Q.103 The prefix form of an infix expression A+B-C*D is

(A) +AB-*CD. **(B)** -+A B C * D.

(C) -+A B * C D. **(D)** - + *ABCD.

Ans: (C)

Q.104 The number of edges in a simple, n-vertex, complete graph is

(A) n*(n-2). **(B)** n*(n-1).

(C) n*(n-1)/2. **(D)** n*(n-1)*(n-2)

Ans: (C)

Q.105 The largest and the second largest number from a set of n distinct numbers can be found in

(A) O (n). **(B)** O (2n).

(C) O $\left(n^2\right)$. **(D)** O (log n).

Ans: (A)

Q.106 To implement Sparse matrix dynamically, the following data structure is used

(A) Trees **(B)** Graphs

(C) Priority Queues **(D)** Linked List

Ans: (D)

Q.107 The depth $d_{n,}$ of complete binary tree of n nodes, where nodes are labeled from 1 to n with root as node 1 and last leaf node as node n is

(A) $\lfloor \log_2 n - 1 \rfloor$ **(B)** $\lfloor \log_2 n + 1 \rfloor$

(C) $\lceil \log_2 n + 1 \rceil$ **(D)** $\lceil \log_2 n - 1 \rceil$

Ans: (C)

Q.108 The balance factor for an AVL tree is either
(A) 0,1 or –1 **(B)** –2,–1 or 0
(C) 0,1 or 2 **(D)** All the above

Ans: (A)

Q.109 Applications of Linked List are
(A) Simulation , event driven systems
(B) Postfix and prefix manipulations
(C) Dictionary systems, polynomial manipulations
(D) Fixed block storage allocation, garbage collection

Ans: (D)

Q.110 AVL trees have LL, LR, RR, RL rotations to balance the tree to maintain the balance factor (LR : Insert node in Right sub tree of Left sub tree of node A, etc). Among rotations the following are single and double rotations
(A) LL, RL and LR, RR **(B)** LL, RR and LR, RL
(C) LR, RR and LL, RL **(D)** LR, RL and LR, RL

Ans: (B)

Q.111 Hashing collision resolution techniques are
(A) Huffman coding, linear hashing **(B)** Bucket addressing, Huffman coding
(C) Chaining, Huffman coding **(D)** Chaining, Bucket addressing

Ans: (D)

Q.112 The running time of the following sorting algorithm depends on whether the partitioning is balanced or unbalanced
(A) Insertion sort **(B)** Selection sort
(C) Quick sort **(D)** Merge sort

Ans: (C)

Q.113 Graphs are represented using
(A) Adjacency tree **(B)** Adjacency linked list
(C) Adjacency graph **(D)** Adjacency queue

Ans: (B)

Q.114 The average case complexity of Insertion Sort is
(A) $O(2^n)$ **(B)** $O(n^3)$
(C) $O(n^2)$ **(D)** $O(2n)$

Ans: (C)

Q.115 Infinite recursion leads to
(A) Overflow of run-time stack **(B)** Underflow of registers usage
(C) Overflow of I/O cycles **(D)** Underflow of run-time stack

Ans: (A)

Q.116 The number of unused pointers in a complete binary tree of depth 5 is
(A) 4 **(B)** 8
(C) 16 **(D)** 32

Ans: (C)

Q.117 The running time for creating a heap of size n is
(A) O (n) **(B)** O (log n)
(C) O (n log n) **(D)** O (n^2)

Ans: (C)

Q.118 What would be returned by the following recursive function after we call test (0, 3)

```
int test (int a, int b)
{
     if (a==b) return (1);
               else if (a>b) return(0);
               else return (a+test(a+1, b));
 }
```

(A) 1 **(B)** 2
(C) 3 **(D)** 4

Ans: (D)

Q.119 The extra key inserted at the end of the array is called a
(A) End Key **(B)** Stop Key
(C) Sentinel **(D)** Transposition

Ans: (C)

Q.120 Which of the following operations is performed more efficiently by doubly linked list than by singly linked list
(A) Deleting a node whose location is given.
(B) Searching of an unsorted list for a given item.
(C) Inserting a new node after node whose location is given.
(D) Traversing the list to process each node.

Ans: (A)

Q.121 One can determine whether a Binary tree is a Binary Search Tree by traversing it in
(A) Preorder **(B)** Inorder
(C) Postorder **(D)** Any of the three orders

Ans: (B)

Q.122 The spanning tree of connected graph with 10 vertices contains
(A) 9 edges **(B)** 11 edges
(C) 10 edges **(D)** 9 vertices

Ans: (A)

Q.123 A sorted file contains 16 items. Using binary search, the maximum number of comparisons to search for an item in this file is

(A) 15 **(B)** 8
(C) 1 **(D)** 4

Ans: (D)

Q.124 One can determine whether an infix expression has balanced parenthesis or not by using

(A) Array **(B)** Queue
(C) Stack **(D)** Tree

Ans: (C)

Q.125 The average number of key comparisons done in successful sequential search in a list of length n is

(A) log n **(B)** (n-1)/2
(C) n/2 **(D)** (n+1)/2

Ans: (D)

Q.126 The maximum number of nodes in a binary tree of depth 5 is

(A) 31 **(B)** 16
(C) 32 **(D)** 15

Ans: (A)

Q.127 n elements of a Queue are to be reversed using another queue. The number of "ADD" and "REMOVE" operations required to do so is

(A) 2*n **(B)** 4*n
(C) n **(D)** The task cannot be accomplished

Ans: (D)

Q.128 A complete binary tree with n leaf nodes has

(A) n+1 nodes **(B)** 2n-1 nodes
(C) 2n+1 nodes **(D)** n(n-1)/2 nodes

Ans: (B)

Q.129 A binary tree can be converted in to its mirror image by traversing it in

(A) Inorder **(B)** Preorder
(C) Postorder **(D)** Anyorder

Ans: (B)

Q.130 One can convert an infix expression to a postfix expression using a

(A) stack **(B)** Queue
(C) Deque **(D)** none of these

Ans: (A)

Q.131 Which of the following types of expressions do not require precedence rules for evaluation?

(A) fully parenthesised infix expression
(B) postfix expression
(C) partially parenthesised infix expression
(D) more than one of the above

Ans: (A)

Q.132 Overflow condition in linked list may occur when attempting to_____

(A) Create a node when free space pool is empty.
(B) Traverse the nodes when free space pool is empty.
(C) Create a node when linked list is empty.
(D) None of these.

Ans: (A)

Q.133 Linked lists are not suitable data structures for which one of the following problems

(A) insertion sort **(B)** binary search
(C) radix sort **(D)** polynomial manipulation

Ans: (B)

Q.134 The sorting technique where array to be sorted is partitioned again and again in such a way that all elements less than or equal to partitioning element appear before it and those which are greater appear after it, is called

(A) merge sort **(B)** quick sort
(C) selection sort **(D)** none of these

Ans: (B)

Q.135 The search technique for searching a sorted file that requires increased amount of space is

(A) indexed sequential search (B) interpolation search
(C) sequential search (D) tree search

Ans: (A)

The search technique for searching a sorted file that requires increased amount of space is indexed sequential search. Because in this search technique we need to maintain a separate index file which requires additional storage space.

Q.136 The postfix form of A ^ B * C - D + E/ F/ (G + H),

(A) AB^C*D-EF/GH+/+ (B) AB^CD-EP/GH+/+*

(C) ABCDEFGH+//+-*^ (D) AB^D +EFGH +//*+

Ans: (A)

Q.137 The prefix of (A+B)*(C-D)/E*F

(A) /+-AB*CD. (B) /*+-ABCD*EF.

(C) */*+AB-CDEF. (D) **AB+CD/EF.

Ans: (C)
Prefix of (A+B) * (C - D) / E*F
(+AB) * (-CD) / E*F
*+AB-CD E*F
/+AB-CDEF

Q.138 A sorting technique which uses the binary tree concept such that label of any node is larger than all the labels in the subtrees, is called
(A) selection sort. (B) insertion sort.

(C) Heap sort. (D) quick sort.

Ans: (C)
A Sorting technique which uses the binary tree concept such that label of any node is larger than all the, labels in the sub trees, is called Heap sort because heap sort works on a complete binary tree with the property that the value at any node 'N' of the tree should be greater than or equal to the value at all its children nodes.

Q.139 A full binary tree with 'n' non-leaf nodes contains

(A) $\log_2$ n nodes . (B) n+1 nodes.

(C) 2n nodes. (D) 2n+l nodes.

Ans: (D)

Q.140 A graph 'G' with 'n' nodes is bipartite if it contains
(A) n edges. (B) a cycle of odd length.
(C) no cycle of odd length. (D) n^2 edges.

Ans: (C)

Q.141 Recursive procedures are implemented by using ____ data tructure.
(A) queues. (B) stacks.

(C) linked lists. (D) strings.

Ans: (B)
Recursive procedures are implemented by using stacks because stacks are LIFO data structure and we need this feature to store return addresses of various recursive calls in recursive procedures.

Q.142 In _____, the difference between the height of the left sub tree and height of the right tree, for each node, is almost one.

(A) Binary search tree (B) AVL - tree

(C) Complete tree (D) Threaded binary tree

Ans: (B)

PART – II

DESCRIPTIVES

Q.1 Write a C program to compute the sum of first n terms (n ≥ 1) of the following series using 'for' loop.

1 – 3 + 5 – 7 + 9 - …… **(6)**

Ans:

A C program to compute the sum of first n terms of the series 1-3+5-7+9-… is listed below:

```
main()
{
 int start=1,i=1,no=0,sum=0;
 clrscr();
 printf ("\nEnter number of terms to be added:-> ");
 scanf("%d",&no);
 for (i=1;i<=no;i++)
 {
  if (i%2!=0)
  {
         sum+=start;
         if (i==1)
                printf ("%d",start);
         else
                printf ("+%d",start);
  }
  else
  {
         sum-=start;
         printf ("-%d",start);
  }
  start+=2;
 }
 printf ("=%d",sum);
 getch();
}
```

Q.2 Write a C program to convert a binary number to its corresponding octal number. **(8)**

Ans:

A C program to convert a binary number to its binary octal number is as follows:

```
main()
{
 long int bin_no,no;
 int oct_no,oct_p,i,rem,pow2,pow8,inter_oct;
 clrscr();
 printf ("\nEnter the Binary number: -> ");
 scanf("%ld",&bin_no);
 no=bin_no;
 pow8=1;
 oct_no=0;
 while (no>0)
 {
        i=0;
        inter_oct=0;
        pow2=1;
        while (i<=2)
        {
               if (no==0)break;
```

```
                rem=no%10;
                inter_oct+=rem*pow2;
                i++;
                pow2*=2;
                no/=10;
            }
            oct_no+=inter_oct*pow8;
            pow8*=10;
        }
        printf ("\nOctal Equivalent for %ld = %d",bin_no,oct_no);
        getch();
    }
```

Q.3 Write a C program to print out n values of the following sequence.
1 –1 1 –1 1 … **(6)**

Ans:
A C program to print n values of following sequence 1 –1 1 –1 1 … is listed below:

```
#include<stdio.h>
#include<conio.h>
main()
{
int i,j,n,k;
clrscr();
printf("\n enter number \n");
scanf("%d",&n);
k=1;
for(i=1;i<=n;i++)
{
for(j=1;j<=i;j++)
printf("%d",k);
printf("\t");
}
getch();
}
```

Q.4 Write a C program to test whether a given pair of numbers are amicable numbers. (Amicable number are pairs of numbers each of whose divisors add to the other number) **(8)**

Ans:
A C program to test whether a given pair of numbers is amicable numbers is as follows:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
int i,j,n,sum_i=0,sum_j=0;
clrscr();
printf("enter any two numbers  >");
scanf("%d%d",&i,&j);
 for(n=1;n<i;n++)
 {
  if (i%n==0)
  sum_i+=n;
 }
 for (n=1;n<j;n++)
```

```
 {
  if (j%n==0)
  sum_j+=n;
 }
if ((sum_j==i) && (sum_i==j))
 printf ("\nAmicable");
else
 printf ("\nNot Amicable");
getch();
}
```

Q.5 Write a C program to rearrange the elements of an array so that those originally stored at odd suffixes are placed before those at even suffixes. **(6)**

Ans:

A C program to rearrange the elements of an array so that those originally stored at odd suffixes are placed before those at even suffixs:

```
main()
{
 char a[25],ch,temp;
 int i;
 clrscr();
 printf ("\nEnter the string:-> ");
 gets(a);
 for (i=0;a[i]!='\0';i++)
 {
      if (a[i+1]=='\0')break;
      temp=a[i];
      a[i]=a[i+1];
      a[i+1]=temp;
      i++;
 }
 puts(a);
 getch();
}
```

Q.6 Admission to a college in science branch is given if the following conditions are satisfied

(i) Maths marks >= 80
(ii) Physics marks >= 75
(iii) Chemistry marks >= 70
(iv) Total percentage in all three subjects >= 80

Given the marks in three subjects, write a program to process the applications to list the eligible candidates. **(8)**

Ans:

A C program to process the applications to list the eligible candidates is listed below:

```
 #include<stdio.h>
 #include<conio.h>
void main()
{
 float math,phy,che;
 float per;
 char ch;
 clrscr();
 printf("\nDo u want to enter any record:-> ");
 ch=getch();
```

```
while(ch=='y' || ch=='Y')
{
 clrscr();
 printf("\nEnter Marks in Math:-> ");
 scanf("%f",&math);
 printf("\nEnter Marks in Physics:-> ");
 scanf("%f",&phy);
 printf("\nEnter Marks in Chemistry:-> ");
 scanf("%f",&che);
 per=(math+phy+che)/3.00;
 if(math>=80.00 && phy>=75.00 && che>=70.00 && per>=80.00)
       printf("\nYou are eligible for selection.");
 else
       printf("\nSorry you are not eligible!!");
 printf("\nDo u want to enter new record:-> ");
 ch=getch();
}
printf ("\nThanks for using it!");
getch();
}
```

Q.7 Write a C program using while loop to reverse the digits of a given number. (for example, If number is=12345 then output number is= 54321) **(5)**

Ans:

A C program to reverse the digits of a given number:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n,r;
clrscr();
printf("enter an integer");
scanf("%d",&n);
printf("\nreverse of %d : ",n);
while(n>0)
{
r=n%10;
printf("%d",r);
n=n/10;
}
getch();
}
```

Q.8 Write C program to produce 10 rows of the following form of Floyd's triangle **(5)**

```
1
0   1
1   0   1
0   1   0   1
```

Ans:

A C program to produce 10 rows of Floyd's triangle is:

```
#include<conio.h>
void main()
{
  int i,j;
clrscr();
for(i=1;i<=10;i++)
{
```

```
   for(j=1;j<=i;j++)
   {    printf(" ");
 if(i%2==0)
     if(j%2==0)
        printf("1");
     else
        printf("0");
      else
        if(j%2==0)
          printf("0");
        else
          printf("1");
        }
 printf("\n\n");
 }
   getch();
}
```

Q.9 Consider the following macro definition

```
#define root (a, b) sqrt((a) * (a) + (b) * (b))
```

What will be the result of the following macro call statement

```
root(a++, b++) if a = 3 and b = 4
```

(3)

Ans:

Result of this macro is:5

sqrt(3*3+4*4)

sqrt(9+16)=sqrt(25)=5

Q.10 Write a nested macro that gives the minimum of three values. **(3)**

Ans:

A nested macro that gives the minimum of three variables is listed below:

```
#include<stdio.h>
#include<conio.h>
#define min(a,b) ((a>b)?b:a)
#define minthree(a,b,c) (min(min(a,b),c))
void main()
{
int x,y,z,w;
clrscr();
printf("enter three numbers :\n");
scanf("%d%d%d",&x,&y,&w);
z=minthree(x,y,w);
printf("Minimum of three value is %d",z);
getch();
}
```

Q.11 Given are two one dimensional arrays A and B which are stored in ascending order. Write a program to merge them into a single sorted array C that contains every element of A and B in ascending order.

(8)

Ans:

A program to merge two arrays into single sorted array that contains every element of arrays into a ascending order:

```
#include<stdio.h>
#include<conio.h>
void sort(int*,int);
void merge(int*,int*,int,int);
void main()
```

```
{
 int a[10],b[10];
 int i,j,m,n;
 clrscr();
 printf("how many numbers u want to enter in 1st array : ");
   scanf("%d",&n);
 printf("enter numbers in ascending order :\n");
 for(i=0;i<n;i++)
   scanf("%d",&a[i]);
 printf("how many numbers u want to enter in 2nd array : ");
   scanf("%d",&m);
 printf("enter numbers in ascending order :\n");
 for(i=0;i<m;i++)
   scanf("%d",&b[i]);
 merge(a,b,n,m);
  getch();
}
void merge(int *a,int *b,int n,int m)
{
 int i=0,c[20],j=0,k=0,count=0;
 while(i<=n&&j<=m)
 {
     if(a[i]<b[j])
    {
       c[k]=a[i];
       i++;
       k++;
    }
    if(a[i]>b[j])
   {
      c[k]=b[j];
      j++;
      k++;
    }
    if(a[i]==b[j])
   {
      c[k]=a[i];
      k++;
      i++;
      j++;
      count++;
    }
 }
 if(i<=n&&j==m)
 {
    while(i<=n)
    {
        c[k]=a[i];
        i++;
        k++;
     }
  }
 if(i==n&&j<=m)
 {
    while(j<=m)
   {
       c[k]=b[j];
       i++;
       j++;
    }
  }
    for(i=0;i<m+n-count;i++)
       printf("%d\t",c[i]);
```

```
}
```

Q.12 Write a C program that reads the text and counts all occurrences of a particular word. **(6)**

Ans:

A C program that reads the text and count all occurrences of a particular word:

```
#include <string.h>
       void main()
       {
char a[100],b[20],c[20];
int i,count=0,k=0,len;
clrscr();
printf ("\nEnter a string:-> ");
gets(a);
printf("\nEnter the word to be searched:-> ");
gets(b);
len=strlen(a);
for(i=0;i<=len;i++)
{
 //Assuming Space , Tab and NULL as word separator
 if(a[i]==32 || a[i]=='\t' || a[i]=='\0')
 {
       c[k]='\0';
       if(strcmp(b,c)==0)count++;
       k=0;
 }
 else
 {
       c[k]=a[i];
       k++;
 }
}
printf("Occurance of %s is = %d",b,count);
getch();
         }
```

Q.13 Write a C program that reads a string from keyboard and determines whether the string is palindrome or not. (A string is palindrome if it is read from left or right gives you the same string) **(8)**

Ans:

A C program to check if input string is palindrome or not:

```
#include<stdio.h>
      #include<conio.h>
      void main()
      {
 int i,j,k,flag=1;
 char a[20];
 clrscr();
 printf("enter any word:\n");
 scanf("%s",a);
 i=0;
 while(a[i]!='\0')
 i++;
 k=i;
 for(j=0;j<k/2;)
 {
       if(a[j]!=a[i-1])
       {
           flag=0;
```

```
            break;
          }
          if(a[j]==a[i-1])
         {
              j++;
              i--;
          }
}
if(flag==0)
printf("it is a not palindrome");
else
printf("it is a palindrome");
getch();
         }
```

Q.14 Write a C function strend(s, t), which returns 1 if the string t occurs at the end of the string s, and zero otherwise. **(7)**

Ans:

C function strend(s, t), which returns 1 if the string t occurs at the end of the string s, and zero otherwise.

```
#include<conio.h>
#include<string.h>
strend ()
{
 char s[100],t[20],c[20];
 int i,j,k=0,len1,len2;
 clrscr();
 printf("\n Enter the string : ");
 gets(s);
 printf("\n Enter the string to be searched for : ");
 gets(t);
 len1=strlen(s);
 len2=strlen(t);
 for(i=len1-(len2);i<=len1;i++)
 {
        c[k++]=s[i];
 }
 c[k]='\0';
 if(strcmp(t,c)==0)
        return 1;
 else
        return 0;
 getch();
}
```

Q.15 Write a function rightrot(x, n) that returns the value of the integer x rotated to the right by n positions. **(7)**

Ans:

A function rightrot(x, n) that returns the value of the integer x rotated to the right by n positions.

```
#include<conio.h>
rightrot(x,n)
{
 int x,n,k=0,num,rem,i,j,dig;
 int arr1[10],arr2[10];
 clrscr();
 printf("\n Enter the integer to be rotated : ");
 scanf("%d",&x);
```

```
 printf("\n Enter the positions by which %d is to be rotated
: ",x);
 scanf("%d",&n);
 for(num=x;num!=0;num/=10)
 {
        rem=num%10;
        arr1[k++]=rem;
 }
 i=k-1;
 k=0;
 for(;i>=0;i--)
        arr2[k++]=arr1[i];
 k-=1;
 for(i=1;i<=n;i++)
 {
        dig=arr2[k];
        for(j=k;j>0;j--)
               arr2[j]=arr2[j-1];
        arr2[0]=dig;
 }
 printf("\n\n The original number is : “);
  return x;
 printf("\n The rotated number is : ");
 for(i=0;i<=k;i++)
        printf("%d",arr2[i]);
 getch();
}
```

Q.16 Write recursive function in C to obtain n^{th} term of the Fibonacci series. **(7)**

Ans:

recursive function in C to obtain n^{th} term of the Fibonacci series

```
    #include<stdio.h>
    #include<conio.h>
     void main()
     {
int n;
void fibonac(int );
clrscr();
printf("enter the length for the series");
scanf("%d",&n);
printf("Fibonacci sequence upto %d terms is:\n\n",n);
fibonac(n);
getch();
      }
void fibonac(int n)
{ static int f1=0,f2=1;
   int temp;
   if(n<2)
   { f1=0;
     f2=1;
   }
  else
  {
     fibonac(n-1);
     temp=f2;
     f2=f1+f2;
     f1=temp;
  }
  printf("%5d",f1);
}
```

Q.17 Write C function named 'fiddle' that takes two arguments, x and y and changes both values. x is an int while y is a pointer to int **(7)**

Ans:

C function named 'fiddle' that takes two arguments, x and y and changes both values. x is an int while y is a pointer to int

```
#include<conio.h>
void main()
{
  int x,z,*y,temp;
  clrscr();
  printf("\n Enter two numbers : \n");
  scanf("%d %d",&x,&z);
  y=&z;
  printf("\n\n x = %d and y = %d",x,*y);
  fiddle(&x,y);
  printf("\n After changing their values ");
  printf("\n x = %d and y = %d",x,*y);
  getch();
}
fiddle(int *a,int *b)
{
  int temp;
  temp=*a;
  *a=*b;
  *b=temp;
  getch();
}
```

Q.18 What is an unsigned integer constant? What is the significance of declaring a constant as unsigned? **(4)**

Ans:

An integer constant is any number in the range -32768 to +32767, because an integer constant always occupies two bytes in memory and in two bytes we cannot store a number bigger than +32767 or smaller than -32768. Out of the two bytes to store an integer, the highest bit is used to store the sign of the integer. This bit is 1 if the number is negative and 0 if number is positive. Unsigned integer constant is an integer constant which has the permissible range from 0 to 65536. Thus significance of declaring a constant as unsigned almost doubles the size of the largest possible value. This happens because on declaring a constant as unsigned, the sixteenth bit is free and is not used to store the sign of the constant.

Q.19 Explain pointers and structures by giving an example of pointer to structure variable? **(5)**

Ans:

We can have a pointer pointing to a structure just the same way a pointer pointing to an int, such pointers are known as structure pointers. For example consider the following example:

```
#include<stdio.h>
#include<conio.h>
struct student
{
     char name[20];
     int roll_no;
};
```

```
void main()
{
     struct student stu[3],*ptr;
     clrscr();
     printf("\n Enter data\n");
     for(ptr=stu;ptr<stu+3;ptr++)
    {  printf("Name");
        scanf("%s",ptr->name);
        printf("roll_no");
        scanf("%d",&ptr->roll_no);
    }
    printf("\nStudent Data\n\n");
    ptr=stu;
    while(ptr<stu+3)
   {
        printf("%s   %5d\n",ptr->name,ptr->roll_no); ptr++;
    }
getch();
}
```

Here ptr is a structure pointer not a structure variable and dot operator requires a structure variable on its left. C provides arrow operator "->" to refer to structure elements. "ptr=stu" would assign the address of the zeroth element of stu to ptr. Its members can be accessed by statement like "ptr->name". When the pointer ptr is incremented by one, it is made to point to the next record, that is stu[1] and so on.

Q.20 Compare the following pairs of statements with respect to their syntax and function:

a. break and continue.
b. goto and break. **(4)**

Ans:

a. break and continue

Two keywords that are very important to looping are break and continue. The **break** command will exit the most immediately surrounding loop regardless of what the conditions of the loop are. Break is useful if we want to exit a loop under special circumstances.

```
#include <stdio.h>
void  main() {
 int a;
  printf("Pick a number from 1 to 4:\n");
 scanf("%d", &a);
  switch (a) {
   case 1:
     printf("You chose number 1\n");
     break;
   case 2:
     printf("You chose number 2\n");
     break;
   case 3:
     printf("You chose number 3\n");
     break;
   case 4:
     printf("You chose number 4\n");
     break;
   default:
     printf("That's not 1,2,3 or 4!\n");
 }
 getch();
```

```
}
```

Continue is another keyword that controls the flow of loops. If we are executing a loop and hit a continue statement, the loop will stop its current iteration, update itself (in the case of for loops) and begin to execute again from the top. Essentially, the continue statement is saying "this iteration of the loop is done; let's continue with the loop without executing whatever code comes after me."
The syntax of continue statement is simple continue;

b. goto and break

C support **goto statement** to branch unconditionally from one point to another in the program.The goto keyword is followed by a label, which is basically some identifier placed elsewhere in the program where the control is to be transferred.
During running of a program, the statement like "goto label1;" cause the flow of control to the statement immediately following the label "label1". We can have a forward jump or a backward jump.

```
#include <stdio.h>
        void main()
         {
 int attempt, number = 46;
 looping: /* a label */
  printf("Guess a number from 0-100\n");
  scanf("%d", &attempt);
  if(number==attempt) {
   printf("You guessed correctly!\n\n");
  }
  else {
   printf("Let me ask again...\n\n");
   goto looping; /* Jump to the label*/
  }
getch();
    }
```

Q.21 Explain the difference between the following:
(i) Program testing and debugging.
(ii) Top down and bottom up approaches.
(iii) Interpreted and compiled languages. **(6)**

Ans:

(i) Program testing and debugging:

Program testing is the process of checking program, to verify that it satisfies its requirements and to detect errors. These errors can be of any type-Syntax errors, Run-time errors, Logical errors and Latent errors. Testing include necessary steps to detect all possible errors in the program. This can be done either at a module level known as unit testing or at program level known as integration testing.
Debugging is a methodical process of finding and reducing the number of bugs in a computer program making it behave as expected. One simple way to find the location of the error is to use print statement to display the values of the variables. Once the location of the error is found, the error is corrected and debugging statement may be removed.

(ii) Top Down and Bottom up approaches

A **top-down** approach is essentially breaking down a system to gain insight into its compositional sub-systems. In a top-down approach an overview of the system is first formulated, specifying but not detailing any first-level subsystems. Each subsystem is then refined in greater detail, sometimes in many additional

subsystem levels, until the entire specification is reduced to base elements. In short the top down approach means decomposing of the solution procedure into subtasks. This approach produces a readable and modular code that can be easily understood and maintained.

A **bottom-up** approach is essentially piecing together systems to give rise to grander systems, thus making the original systems sub-systems of the emergent system. In a bottom-up approach the individual base elements of the system are first specified in great detail. These elements are then linked together to form larger subsystems, which then in turn are linked, sometimes in many levels, until a complete top-level system is formed.

(iii)Interpreted and Compiled Languages

In **interpreted languages**, the instructions are executed immediately after parsing. Both tasks are performed by the interpreter. The code is saved in the same format that we entered. Interpreted languages include the MS-DOS Batch language (the OS itself is the interpreter), shell scripts in Unix/Linux systems, Java, Perl etc. Advantages of interpreted languages include relative ease of programming and no linker requirement. Disadvantages include poor speed performance and that we do not generate an executable (and therefore distributable) program. The interpreter must be present on a system to run the program.

In **compiled languages** the instructions into machine code and store them in a separate file for later execution. Many modern compilers can compile (parse) and execute in memory, giving the 'appearance' of an interpreted language. However, the key difference is that parsing and execution occurs in two distinct steps. Examples of compiled languages include Visual Basic, C/C++, Delphi and many others. In a compiled environment, we have several separate files like: source code (the text instructions), object code (the parsed source code) and the executable (the linked object code). There is definitely an increase in complexity in using compilers, but the key advantages are speed performance and that one can distribute stand-alone executables.

Q.22 Write a complete C program for reading an employees file containing {emp_number, name, salary, address}. Create an output file containing the names of those employees along with their salary and address whose salary is > 20,000. **(8)**

Ans:

```
#include<stdio.h>
struct employee
{
 int code;
 char name[30];
 char address[50];
 char phone[10];
 float sal;
}emp,tmp;
FILE *fp1,*fp2;
void main()
{
 char ch='y';
 int i=1;
 //Assuming that the file already exists
 fp1=fopen("employee.txt","r");
 if (fp1==NULL)
 {
fp1=fopen("employee.txt","w");
if (fp1==NULL)
```

```
        {
              printf ("\nUnable to create Employee.txt file");
              getch();
              exit();
        }
        printf ("\nNo Data found in Employee.txt. Add new
      records");
        while (1)
        {
              printf ("\nEnter Employee Code:-> ");
              flushall();
              scanf ("%d",&emp.code);
              printf ("\nEnter Employee Name:-> ");
              flushall();
              gets(emp.name);
              printf ("\nEnter Employee Address:-> ");
              flushall();
              gets(emp.address);
              printf ("\nEnter Employee Phone Number:-> ");
              flushall();
              gets(emp.phone);
              printf ("\nEnter Employee Salary:-> ");
              flushall();
              scanf("%f",&emp.sal);
              //Writing records to Employee.txt file
              fwrite(&emp, sizeof(emp),1,fp1);
              printf ("\nDo u want to add more records (y/n):->
             ");
              flushall();
              scanf("%c",&ch);
              if (ch=='n' || ch=='N')break;
        }
        fclose(fp1);
        fp1=fopen("employee.txt","r");
}
printf ("\nList of Employees having Salary greater than
 20000\n");
printf ("\nCode\tName\t\tAddress\tPhone No.\tSalary\n");
i=1;
fp2=fopen("Output.txt","w");
getch();
if (fp2==NULL)
{
       printf ("\nUnable to create Output file\n");
       getch();
       exit();
}
while (1)
{
       i=fread(&emp,sizeof(emp),1,fp1);
       if(i==0)break;
       if (emp.sal>20000)
              fwrite (&emp,sizeof(emp),1,fp2);
}
fclose (fp2);
fp2=fopen("output.txt","r");
i=1;
while (1)
{
       i=fread(&emp,sizeof(emp),1,fp2);
       if(i==0)break;
```

```
 printf
("\n%d\t%s\t\t%s\t%s\t%f",emp.code,emp.name,emp.address,emp.phone,e
mp.sal);
 }
 getch();
 }
```

Q.23 Write a function to display the binary number corresponding to the integer passed to it as an argument. **(6)**

Ans:

```
#include<stdio.h>
#include<conio.h>
void dis_bin(int n)
{
  num=n;
  for(i=0;num!=0;i++)
  {
        q=num/2;
        r[i]=num%2;
        num=q;
  }
  clrscr();
  printf("\n\n The integer the number is : %d\n",n);
  printf("\n The binary conversion is : ");
  for(i-=1;i>=0;i--)
        printf("%d",r[i]);
  getch();
}
```

Q.24 Write a function, my_atoi, similar to the library function **atoi** that returns the numeric value corresponding to the string passed to it as an argument. **(6)**

Ans:

```
#include <stdio.h>
main()
{
    long int n=0;
    char str[25];
    clrscr();
    printf ("\nEnter the string:-> ");
    gets(str);
    my_atoi(str,&n);
    printf ("\nInteger Equvalent of string %s=%ld",str,n);
    getch();
}
my_atoi(char str[25],long int *num)
{
   int i=1;
   long int s=1,l=0,j;
   float po10=1;
   //Trimming the string for integers
   for (i=0;str[i]!='\0';i++)
   {
         if (str[i]==' ')
         {
               for (j=i;str[j]!='\0';j++)
                     str[j]=str[j+1];
                     i--;
         }
```

```
            else if (str[i]>=48 && str[j]<=57)
                  continue;
            else
            {
                  str[i]='\0';
                  break;
            }
      }
      l=strlen(str);
      for (i=l-1;i>=0;i--)
      {
       switch (str[i])
       {
       case 48: s*=0;break;
       case 49: s*=1;break;
       case 50: s*=2;break;
       case 51: s*=3;break;
       case 52: s*=4;break;
       case 53: s*=5;break;
       case 54: s*=6;break;
       case 55: s*=7;break;
       case 56: s*=8;break;
       case 57: s*=9;break;
       }
       *num+=s;
       po10*=10;
       s=po10;
      }
}
```

Q.25 Briefly explain what do you understand by stepwise refinement of the program? **(4)**

Ans:

Stepwise refinement of the Program:

The concept of "Stepwise refinement" means to take an object and move it from a general perspective to a precise level of detail and this cannot be done in one jump but in steps. The number of steps needed to decompose an object into sufficient detail is ultimately based on the inherent nature of the object. "Stepwise refinement" of program represents a "divide and conquer" approach to design. In simple words, break a complex program into smaller, more manageable modules that can be reviewed and inspected before moving to the next level of detail. Step-wise refinement is a powerful paradigm for developing a complex program from a simple program by adding features incrementally.

Some of the steps taken for refinement of a program are:

1. Analysis of the algorithm used to develop the program, and analysis of various parts of the program and then making appropriate changes to improve the efficiency of the program.
2. Keep the program simple to improve memory efficiency.
3. Evaluation and incorporation of memory compression features.
4. Debugging and testing performed first to module level and then to interface level.
5. Take care of correct working and clarity while making it faster.

Q.26 Write a function to remove duplicates from an ordered array. For example, if input is: a,a,c,d,q,q,r,s,u,w,w,w,w; then the output should be a,c,d,q,r,s,u,w. **(8)**

Ans:

A C program to remove duplicates from an odered array:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
   int i,j,k,l,flag=0;
   char a[50];
   clrscr();
   printf("enter the characters in the array");
   gets(a);
   l=strlen(a);
   for(i=0;i<l-1;i++)
    for(j=i+1;j<l;j++)
   {
      if(a[i]==a[j])
      { l=l-1;
        for(k=j;k<l;k++)
        a[k]=a[k+1];
        flag=1;
        j=j-1;
      }
   }
   if(flag==0)
      printf("No duplicates found");
   else
      for(i=0;i<l;i++)
   printf("%c",a[i]);
   getch();
}
```

Q.27 Write a function to sort the characters of the string passed to it as argument. **(8)**

Ans:

A C function to sort the characters of the string:

```
main()
{
 char a[100],ch;
 int i,j;
 printf ("\nEnter the string:-> ");
 gets(a);
 for (i=0;a[i]!='\0';i++)
 {
  for (j=i+1;a[j]!='\0';j++)
  {
         if (a[i]>a[j])
         {
                ch=a[i];
                a[i]=a[j];
                a[j]=ch;
         }
  }
 }
 printf ("\nString after sorting\n");
 puts(a);
 getch();
}
```

Q.28 Give the outputs of the following code segments, if any and justify your answers.

```
(i)   #define CUBE(x)                (x * x * x)
main( )
{
    printf("%d", CUBE(4+5));
}
(ii)  int j = 5;
        printf("%d", j = j == 6);
        printf("%d", j = ++j == 6);
(iii)for (j = 0; j = 3; j++)
        printf("%d", j);
(iv)  main( )
    {
      static char a[ ] = "Test String";
      static char *b = "Test String";
      printf("%d %d", sizeof(a), sizeof(b));
    }
(v) main( ) {
    enum test {RED, BLUE, GREEN};
    enum test t = BLUE;
    printf("%d", t);
    }
(vi)  main( ) {
      union U { int j; char c; float f; } u;
      u.j = 10; u.c = 'A'; u.f = 99.99;
      printf("u.j = %d u.c = %c u.f = %f", u.j, u.c, u.f);
       }
```

(2 x 6)

Ans:

(i) Output is: 49
macro cube(x *x*x) is expanded into (4+5*4+5*4+5) which is evaluated as 4+20+20+5=49 due to priority of operators. So output is 49.

(ii) Output is: 0 0
Both the expressions are evaluated as 0 so the output is 0.

(iii) Output is: infinite loop
There is an incorrect assignment, j=3 in test expression.

(iv) Output is:12 2,
The printf is printing the size of char array a and char b, null char is included.

(v) Output is:1,
The compiler automatically assigns integer digits beginning with 0 to all the enumeration constants so BLUE has value 1.

(vi) Output is: u.j=1311,u.c='β',u.f=99.989998
The first two outputs are erroneous output which is machine dependent. During accessing a union member, we should make sure that we are accessing the member whose value is currently stored otherwise we will get errors.

Q.29 What are preprocessor directives? List three types of them. What is the difference between the following directives: #include <filename> and #include "filename"?

(4)

Ans:

Preprocessor directives: These are the commands given to a program known as preprocessor that processes the source code before it passes through the compiler. Each of these preprocessor directives begins with a # symbol. These can be placed anywhere in the program but usually placed before main ( ) or at the beginning of the program. Before the source code passes through the compiler, it is examined by the preprocessor for any directives. If there are any, appropriate action is taken and the source program is handed over to compiler. These directives can be divided into following three categories:

1. Macro substitution directives
2. File inclusion directives.
3. Compiler control directives.

#include "filename": The search for the file is made first in the current directory and then in the standard directories as mentioned in the include search path.

#include <filename>: This command would look for the file in the standard list of directories.

Both of these directives cause the entire contents of filename to be inserted into the source code at that point in the program.

Q.30 Write a function to compute the frequency of 'the' in a file. **(8)**

Ans:

A C function to compute the frequency of 'the':

```
void main()
{
  int i=0,j=0,k,n=0,m=0,l;
  char a[100],b[10],c[10]="the";
  clrscr();
  printf("enter the sentence");
  gets(a);
  while(a[i]!='\0')
  {
       if(a[i]==' '|| a[i]=='.')
       m++;
       i++;
  }
  i=0;
  l=0;
  while(l!=m+1)
  {
     while(a[i]!=' ' && a[i]!='.' && a[i]!='\0')
     {
         b[j]=a[i];
         j++;
         i++;
     }
     b[j]='\0';
     k=strcmp(b,c);
     if(k==0)
     {
         n++;
         j=0;
         if(a[i]!='\0')
          i++;
     }
     else
```

```
        {
            j=0;
            i++;
        }
        l++;
    }
    if(n!=0)
    printf("it is present %d times",n);
    else
    printf("it is not present");
    getch();
}
```

Q.31 Write a recursive function to print the reverse of a string passed to it as an argument. **(8)**

Ans:

A recursive function to print the reverse of a string is given below:

```
void main()
{
   char str[100];
   clrscr();
   printf("enter a string\n");
   gets(str);
   printf("reverse of the entered string is\n");
   rev(str);
   getch();
 }
    rev (char *string)
    {
    if (*string)
    {
       rev(string+1);
     putchar(*string);
    }
}
```

Q.32 Write a program which accepts two file names from the command line and prints whether the contents of the two files are same or not. If not, then print the first line number and then the contents of the lines from which they differ. Assume that the lines in both the files have at most 80 characters. **(12)**

Ans:

A program that accepts two file names from command prompt and check if two files are same or not:

```
#include<stdio.h>
main(int argc, char *argv[])
{
 FILE *fp1, *fp2;
 char ch1[80],ch2[80];
 int i=1,j=1,l1=0,l2=0;
 if (argc!=3)
 {
        printf ("\nWrong number of arguments.");
        getch();
        exit();
 }
 fp1=fopen(argv[1],"r");
 if (fp1==NULL)
```

```
{
        printf ("\nunable to open the file %s",argv[1]);
        getch();
        exit();
}
fp2=fopen(argv[2],"r");
if (fp2==NULL)
{
        printf ("\nunable to open the file %s",argv[2]);
        getch();
        exit();
}
l1=0;
l2=0;
while (i!=0 && j!=0)
{
        i=fgets(ch1,80,fp1);l1++;
        j=fgets(ch2,80,fp2);l2++;
        if (strcmp(ch1,ch2)!=0)
        {
                printf ("\nContents of both Files are not
Equal");
                printf     ("\nLine     Number\tContents\tFile
name\n");
                printf ("%d\t\t%s\t\tFrom %s",l1,ch1,argv[1]);
                printf ("%d\t\t%s\t\tFrom %s",l2,ch2,argv[2]);
                exit();
        }
}
if (i==0 && j==0)
        printf ("\nBoth Files are equal");
else
        printf("\nBoth files are not equal");
getch();
}
```

Q.33 Differentiate between the following:

(i) call by value and call by reference

(ii) do..while and while loops **(4)**

Ans:

(i) **Call by value and Call by reference**

Call by value means sending the values of the arguments- The value of each of the actual arguments in the calling function is copied into corresponding formal arguments of the called function. The changes made to the formal arguments have no effect on the values of actual arguments in the calling function. This technique of passing arguments is called call by value illustrated by swapv(int x, int y) function in the following example.

Call by reference means sending the addresses of the arguments- the addresses of actual arguments in the calling function are copied into formal arguments of the called function. Using these addresses we are actually working on actual argument so changes will be reflected in the calling function. This technique of passing arguments is called call by reference, illustrated by swapr(int *x,int *y) in following example.

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main(){
```

```
int i=10,j=20;
clrscr();
printf("The values before swap is i: %d, j:%d\n",i,j);
swapv(i,j);
printf("The values after swap is i: %d, j:%d\n",i,j);
printf("\n");
swapr(&i,&j);
printf("The values after swap is i: %d, j:%d\n",i,j);
printf("\n");
getch();
}
swapv(int x,int y)
{ int temp;
   temp=x;
   x=y;
   y=temp;
  }
swapr(int *x,int *y)
{
  int temp;
   temp=*x;
   *x=*y;
   *y=temp;
  }
```

The value of i and j is 10 and 20 only after calling function swapv, that is call by value. However the result of calling swapr(), call by reference is i=20 and j=10

(ii) do-while and while loops

while statement: The basic format of while statement is

```
while (conditional expression)
{
        ...block of statements to execute...
}
```

The while loop continues to loop until the conditional expression becomes false. Once this expression become false, the control is transferred out of the loop. On exit, the program continues with the statement immediately after the body of the loop. For example:

```
i=0;
        while(i<10)
       {
              printf("Hello world\n");
       i++;
}
```

This statement will print "Hello world" 10 times in a new line and come out of the loop when 'i' become 10.

Do statement: This loop construct is of the form:

```
      do
        {    ...block of statements to execute...
      }while(conditional expression);
```

While construct checks the conditional expression before the loop is executed. Sometimes it is necessary to execute the body of the loop before the conditional expression is evaluated. Such situations are handled by do-while loop construct. On reaching the do statement, the body of the loop is evaluated and at the end of the loop, the conditional expression is checked for true or false. If true,it continues to evaluate the body again and when condition become false, the control is transferred to the statement immediately after the while statement.

For example:

```
do
```

```
    {
        printf( "Input a character\n");
ch=getch( );
  }while(ch='n');
```

This segment of program reads a character from the keyboard until 'n' is keyed in.

Q.34 Write a program to generate a triangle as shown below for n = 4. The program should take the input from the user.

```
          A
        A B A
      A B C B A
    A B C D C B A
```

(8)

Ans:

A C program to generate a triangle for n=4 is listed below:

```
#include<conio.h>
void main()
{
  int i,j,k,ch;
  clrscr();
  for(i=3;i>=0;i--)
  {
        ch=65;
        printf("\n\n");

        for(j=i;j>0;j--)
              printf("  ");
        for(k=i;k<4;k++)
              printf("%c ",ch++);
        ch-=2;
        for(j=i;j<3;j++)
              printf("%c ",ch--);
        for(k=i;k>0;k--)
              printf("  ");
  }
  getch();
}
```

Q.35 Write a function that accepts two strings str1 and str2 as arguments and finds which of the two is alphabetically greater (without using the library functions). The function should return 1 if str1 is greater than str2, 0 if str1 is equal to str2, and -1 is str1 is smaller than str2. **(8)**

Ans:

A function that accept two strings to check which one is alphabetically greater:

```
void main()
{
   char a[10],b[10];
   int k;
   clrscr();
   printf("enter 1st string");
   gets(a);
   printf("enter 2nd string");
   gets(b);
   k=comp(a,b);
   if(k==1)
     printf("%s is greater than %s",a,b);
   if(k==-1)
```

```
            printf("%s is less than %s",a,b);
         if(k==0)
            printf("%s is equal to %s",a,b);
         getch();
      }
      int comp(char *a,char *b)
      {
         int i=0,j=0,k;
         while(a[i]!='\0'&& b[j]!='\0)
         {  if (a[i]<b[j])
              return (-1);
            else if(a[i]>b[j])
               return (1);
            else
               i++;
               j++;
         }
         if(i==j)
          return (0);
         if (i<j)
           return (-1);
          if(i>j)
            return (1);
       }
```

Q.36 Consider a linked list to store a polynomial, that is, every node of the linked list has coefficient, exponent and pointer to the next node in the list.

(i) Define a structure for node of such a list.

(ii)Write a function to subtract two such polynomials. The function should accept pointers to the two polynomials as arguments and return the pointer to the resultant polynomial. Assume that the polynomials passed to the function are in decreasing order on the exponents. **(2 + 8)**

Ans:

(i) Structure for polynomial node

```
struct polynode
{
  float coeff ;
  int exp ;
  struct polynode *link ;
};
```

(ii) A function that subtract two polynomials:

```
struct polynode * poly_sub ( struct polynode *x, struct polynode
*y )
{
struct polynode *z,*temp,*s=NULL;
/* if both linked lists are empty */
if ( x == NULL && y == NULL )
       return ;
/* traverse till one of the list ends */
while ( x != NULL || y != NULL )
{
       /* create a new node if the list is empty */
       if ( s == NULL )
       {
              s = malloc ( sizeof ( struct polynode ) ) ;
              z = s ;
       }
       /* create new nodes at intermediate stages */
```

```
        else
        {
        z -> link = malloc ( sizeof ( struct polynode ) )  ;
               z = z -> link ;
        }
        if (y==NULL && x!=NULL)
        {
               z -> coeff = x -> coeff;
               z -> exp = x -> exp;
               z -> link = NULL;
               x=x -> link;
               continue;
        }
        if (x==NULL && y!=NULL)
        {
               z -> coeff = y -> coeff;
               z -> exp = y -> exp;
               z -> link = NULL;
               y = y -> link;
               continue;
        }
        /* store a term of the larger degree polynomial */
        if ( x -> exp < y -> exp )
        {
               z -> coeff = y -> coeff ;
               z -> exp = y -> exp ;
               z -> link = NULL;
               y = y -> link ;  /* go to the next node */
        }
        else
        {
               if ( x -> exp > y -> exp )
               {
                      z -> coeff = x -> coeff ;
                      z -> exp = x -> exp ;
                      z -> link = NULL;
                      x = x -> link ;  /* go to the next node */
               }
               else
               {
        /* add the coefficients, when exponents are equal */
                      if ( x -> exp == y -> exp )
                      {
                             /* assigning the added coefficient */
                      z -> coeff = x -> coeff - y -> coeff ;
                             z -> exp = x -> exp ;
                             z -> link = NULL;
                             /* go to the next node */
                             x = x -> link ;
                             y = y -> link ;
                      }
               }
        }
}
return (s);
}
```

Q.37 Differentiate between the following:

(i) compiler and interpreter
(ii) unit testing and integration testing
(iii) syntax errors and logical errors **(6)**

Ans:

(i) Compiler and Interpreter: These are two types of language translators.

A **compiler** converts the source program (user-written program) into an object code (machine language by checking the entire program before execution. If the program is error free, object program is created and loaded into memory for execution. A compiler produces an error list of the program in one go and all have to be taken care even before the execution of first statement begin. It takes less time for execution.

An **interpreter** is also a language translator that translates and executes statements in the program one by one. It work on one statement at a time and if error free, executes the instruction before going to second instruction. Debugging is simpler in interpreter as it is done in stages. An interpreter takes more time for execution of a program as compared to a compiler.

(ii) Unit testing and integration testing:

Unit testing focuses on the smallest element of software design viz. the module. Unit test is conducted on each of the modules to uncover errors within the boundary of the module. It becomes simple when a module is designed to perform only one function. Unit testing makes heavy use of white-box testing.

Integration testing is a systematic approach for constructing program structure while conducting tests to uncover errors associated with interfacing. There are likely to be interfacing problems, such as data mismatch between the modules. In *Incremental* integration the program is constructed and tested in small segments. We have two types of integration testing:

-Top-Down Integration testing

-Bottom-Up Integration testing

(iii) Syntax errors and logical errors:

Syntax errors also known as compilation errors are caused by violation of the grammar rules of the language. The compiler detects, isolate these errors and give terminate the source program after listing the errors.

Logical errors: These are the errors related with the logic of the program execution. These errors are not detected by the compiler and are primarily due to a poor understanding of the problem or a lack of clarity of hierarchy of operators. Such errors cause incorrect result.

Q.38 Write a function to reverse the links in a linked list such that the last node becomes the first and the first becomes the last by traversing the linked list only once. **(8)**

Ans:

A C function to reverse the linked list:

```
struct node
{
  int data ;
  struct node *link ;
} ;
void reverse ( struct node **x )
{
  struct node *q, *r, *s ;
  q = *x ;
  r = NULL ;
  /* traverse the entire linked list */
  while ( q != NULL )
  {
        s = r ;
        r = q ;
```

```
            q = q -> link ;
            r -> link = s ;
        }
        *x = r ;
    }
```

Q.39 Define a structure for an employee of an organization having employee code, name, address, phone number and number of dependents. Assume that "allEmployees" is an array of employees in ascending order on the employee code. Write a function to display the details of an employee given its employee code. **(8)**

Ans:

A structure for an employee and a function to display the details of an employee:

```
    struct employee
    {
      int emp_code;
      char emp_name[30];
      char emp_address[50];
      char emp_ph_num[10];
      int no_of_dep;
    }allEmployees[100],b;
void display()
{
int ctr=0;
fp=fopen("employee.c","r");
rewind(fp);
while(fread(&allEmployees,sizeof(allEmployees[i]),1,fp)==1)
{
       ctr++;
       clrscr();
       heading();
       printf("\n\n\n\tFollowing are the details :-");
       printf("\n\n\tRecord #%d",ctr);
       printf("\n\n\t\tCode : %d",allEmployees[i].emp_code);
       printf("\n\n\t\tName : %s",allEmployees[i].emp_name);
printf("\n\n\t\tAddress : %s",allEmployees[i].emp_address);
printf("\n\n\t\tPhoneNumber:%s",allEmployees[i].emp_ph_num);
 printf("\n\n\t\tNumber of Dependents
                    :%s",allEmployees[i].no_of_dep);
       printf("\n\n\n\n\t\tPlease Press Enter...");
       getch();
}
}
```

Q.40 Explain the concept of top-down design for a program. **(5)**

Ans:

Top down Design:

A **top-down** approach is essentially breaking down a system to gain insight into its compositional sub-systems. In a top-down approach an overview of the system is first formulated, specifying but not detailing any first-level subsystems. Each subsystem is then refined in greater detail, sometimes in many additional subsystem levels, until the entire specification is reduced to base elements. In short the top down approach means decomposing of the solution procedure into subtasks. This approach produces a readable and modular code that can be easily understood and maintained. Top-down programming is a programming style in which design begins by specifying complex pieces and then dividing them into successively

smaller pieces. Eventually, the components are specific enough to be coded and the program is written. The technique for writing a program using top-down methods is to write a main procedure that names all the major functions it will need. Later, the programming team looks at the requirements of each of those functions and the process is repeated. These compartmentalized sub-routines eventually will perform actions so simple they can be easily and concisely coded. When all the various sub-routines have been coded the program is done.

Q.41 What are compilers and interpreters? List the advantages of an interpreter over a compiler. Under which situations you will prefer to use interpreter over compiler. **(8)**

Ans:

Compiler and Interpreter: These are two types of language translators.

A compiler converts the source program (user-written program) into an object code (machine language by checking the entire program before execution. If the program is error free, object program is created and loaded into memory for execution. A compiler produces an error list of the program in one go and all have to be taken care even before the execution of first statement begin. It takes less time for execution. An interpreter is also a language translator that translates and executes statements in the program one by one. It work on one statement at a time and if error free, executes the instruction before going to second instruction. Debugging is simpler in interpreter as it is done in stages. An interpreter takes more time for execution of a program as compared to a compiler.

Q.42 Give any two characteristics of a good programming language. **(3)**

Ans:

Two characteristics of a good programming language are:

1. It should support a large number of data types, built-in functions and powerful operators. In simple terms, a programming language should be robust, only then it can efficient and fast.
2. It should be portable means that programs written for one computer can be run on another with little or no modification.

Q.43 Write a program in C to demonstrate the use of scope resolution operator. **(8)**

Ans:

The scope resolutions operator is used to find the value of a variable out of the scope of the variable.

Example:

```
int i=10;
 main(){
 int i =5;
 cout<<::i;
cout<<i;
}
```

::i refers to the value just before the scope (i.e.10).

Q.44 Why '&' operator is not used with array names in a scanf statement? **(2)**

Ans:

In scanf() statement, the "address of" operator (&) either on the element of the array (e.g marks[i]) or on the variables (e.g &rate) are used. In doing so, the address of this particular array element is passed to the scanf() function, rather then its value; which is what scanf() requires. BUT many times the address of zeroth element (also called as base address) can be passed by just passing the name of the array.

Q.45 Write a C program to read a set of numbers from the keyboard and to sort to given array of elements in ascending order using a function. **(7)**

Ans:

A C program to sort given array of elements in ascending order:

```
#include<stdio.h>
#include<conio.h>
void sort(int*,int);
void main()
{
    int a[10];
   int i,n;
   clrscr();
   printf("how many numbers u want to enter in 1st array :
");
   scanf("%d",&n);
   printf("enter numbers :\n");
   for(i=0;i<n;i++)
   scanf("%d",&a[i]);
   sort(a,n);
   for(i=0;i<n;i++)
   printf("%d",a[i]);
   getch();
}
void sort(int *a,int m)
{
    int i,j,temp;
    for(i=0;i<m-1;i++)
   {
       for(j=i+1;j<m;j++)
      {
           if(a[i]>a[j])
          {
              temp=a[j];
              a[j]=a[i];
              a[i]=temp;
           }
       }
    }
}
```

Q.46 Write a recursive function to calculate the factorial of a positive integer. **(6)**

Ans:

A recursive function to calculate the factorial of a given positive number:

```
void main()
{
  int n,c;
```

```
clrscr();
printf("enter the number   :");
scanf("%d",&n);
c=fact(n);
printf("factorial of %d = %d",n,c);
getch();
    }
    fact(int n)
   {
int factorial;
   if(n==1||n==0)
       return(1);
   else
       factorial=n*fact(n-1);
 return (factorial);
}
```

Q.47 How does an enumstatement differ from a typedef statement? **(3)**

Ans:

Typedef statement allows user to define an identifier that would represent an exiting data type. The user-defined data type identifier can be used further to declare variables. It has the following syntax

```
typedef   datatype identifier;
```

where datatype refers to exiting data type and identifier is the new name given to this datatype. For example

```
typedef int nos;
```

nos here symbolizes int type and now it can be used later to declare variables like

```
nos num1,num2,num3;
```

enum statement is used to declare variables that can have one of the values enclosed within braces known as enumeration constant. After declaring this, we can declare variables to be of this 'new' type. It has the following syntax

```
enum identifier {value1, value2, value3……,valuen);
```

where value1, value2 etc are values of the identifier. For example

```
enum day { Monday, Tuesday, Wednesday, Thursday, Friday,
Saturday}
```

We can define later

```
enum day working_day;
working_day=Monday;
```

The compiler automatically assigns integer digits beginning with 0 to all enumeration constants.

Q.48 Define a structure. How it is different from union? Write a C program to illustrate a structure. **(8)**

Ans:

Structures and union

A structure contains an ordered group of data objects. Unlike the elements of an array, the data objects within a structure can have varied data types. Each data object in a structure is a member or field. A union is an object similar to a structure except that

1. In union, one block is used by all the member of the union but in case of structure, each member has their own memory space. All of union members start at the same location in memory. A union variable can represent the value of only one of its members at a time.

2. The size of the union is equal to the size of the largest member of the union where as size of the structure is the sum of the size of all members of the structure.

```
For example
        struct book
       {
         char    name;
          int  pages;
          float price;
        };
```

Now if we define struct book book1, then it will assign 1+2+4=7 bytes of memory for book1.

If we define it as union like

```
union book
      {
         char name;
          int  pages;
          float price;
        };
```

The compiler allocates a piece of storage that is large enough to store the largest variable types in union. All three variables will share the same address and 4 bytes of memory is allocated to it. This program of structure will read name, roll no and marks in 6 subject of 3 students & then display name and roll of student who scored more than 70% marks in total.

```
#include<stdio.h>
#include<conio.h>
struct student
{
  char name[10];
  int roll_no;
  int marks[6];
  int total;
  int per;
};
void main()
{
   struct student stu[3];
   int i,j,req;
   clrscr();
   for(i=0;i<3;i++)
   {
     stu[i].total=0;
     printf("enter data for %d students :",i+1);
     printf("\nenter name");
     scanf("%s",stu[i].name);
     printf("\nenter roll no   ");
     scanf("%d",&stu[i].roll_no);
     printf("\nenter marks in subjects\t");
     for(j=0;j<6;j++)
     {
         printf("\nenter marks in %d subject\t",j+1);
         scanf("%d",&stu[i].marks[j]);
         stu[i].total=stu[i].total+stu[i].marks[j];
     }
     stu[i].per=stu[i].total/6;
     printf("\n");
   }
   for(i=0;i<3;i++)
   {
      if(stu[i].per>70)
      {
          printf("\nSTUDENT    %d",i+1);
```

```
                printf("\nname  :");
                printf("%s",stu[i].name);
                printf("\nroll no");
                printf("%d",stu[i].roll_no);
                for(j=0;j<6;j++)
               {
                   printf("\nmarks in %d subject\t",j+1);
                   printf("%d",stu[i].marks[j]);
                }
           printf("\nTOTAL    :%d",stu[i].total);
       }
       }
       getch();
   }
```

Q.49 Write a C program to concatenate two strings. **(8)**

Ans:

A program to concatenate two strings is given below:

```
#include<stdio.h>
#include<conio.h>
void main()
{
char a[100],b[100];
int i=0,j=0;
clrscr();
printf("enter the set of lines");
gets(a);
while(a[i]!='\0')
{
while(a[i]!=' '&&a[i]!='\t'&&a[i]!='\0')
{
b[j]=a[i];
j++;
i++;
}
while(a[i]==' '||a[i]=='\t')
i++;
}
b[j]='\0';
printf("%s",b);
getch();
}
```

Q.50 What is a pointer? How it is declared? Write a C program to reverse a string using pointers. **(2+3+4)**

Ans:

A pointer is a variable which contains the address in memory of another variable. We can have a pointer to any variable type. The unary or monadic operator **&** gives the "address of a variable". The indirection or dereference operator ***** gives the "contents of an object pointed to by a pointer".

A pointer is declared as follows:

```
int *ptr;
```

where *ptr is a pointer of int type.

A program to reverse a string using pointer is listed below:

```
#include<stdio.h>
#include<conio.h>
void main()
{
```

```
 int i,j;
 char *a;
 clrscr();
 printf("enter a string");
 scanf("%s",a);
 i=0;
 while(*(a+i)!='\0')
 i++;
 for(j=i-1;j>=0;j--)
 printf("%c",*(a+j));
 getch();
}
```

Q.51 Differentiate between pointers and arrays? Write a C program to display the contents of an array using a pointer arithmetic. **(2+5)**

Ans:

Pointer and arrays:

Pointers and arrays are very closely linked in C. Consider the following statements:

```
int a[10], x;
int *ptr; /* ptr is a pointer variable*/
ptr = &a[0];  /* ptr points to address of a[0] */
x = *ptr;
/* x = contents of ptr (a[0] in this case) */
```

A pointer is a variable so we can do

`ptr = a` and `ptr++;`

while an array is not a variable so statements

`a = pa` and `a++` are illegal.

A C program to display the contents of an array using a pointer arithmetic is listed below:

```
//display the contents of an array using pointer
#include<conio.h>
void main()
{
 int *p,sum,i;
 static int x[5] = {5,9,6,3,7};
 i=0;
 p=x;
 sum=0;
 clrscr();
 printf("\nElement   Value   Address\n\n");
 while(i<5)
 {
       printf("  x[%d]     %d       %u\n",i,*p,p);
       sum+=*p;
       i++;
       *p++;
 }
 printf("\n Sum   = %d\n",sum);
 printf("\n &x[0] = %u\n",&x[0]);
 printf("\n p     = %u\n",p);
 getch();
}
```

Q.52 What is a linked list? List different types of linked list. Write a C program to demonstrate a simple linear linked list. **(2+2+6)**

Ans:

Linked list: A linked list is a self referential structure which contain a member field that point to the same structure type. In simple term, a linked list is collections of nodes that consists of two fields, one containing the information about that node, item and second contain the address of next node. Such a structure is represented as follows:

```
struct node
{       int item;
         struct node *next;
};
```

Info part can be any of the data type; address part should always be the structure type.

Linked lists are of following types:

1. **Linear Singly linked list**: A list type in which each node points to the next node and the last node points to NULL.
2. **Circular linked list**: Lists which have no beginning and no end. The last node points back to the first item.
3. **Doubly linked list or Two-way linked list**: These lists contain double set of pointers, one pointing to the next item and other pointing to the preceding item. So one can traverse the list in either direction.
4. **Circularly doubly linked list**: It employs both the forward and backward pointer in circular form.

A program to demonstrate simple linear linked list is given below:

```
#include<stdio.h>
#include<stdlib.h>
#define NULL 0
struct linked_list
{
  int number;
  struct linked_list *next;
};
typedef struct linked_list node;
void main()
{
  node *head;
  void create(node *p);
  int count(node *p);
  void print(node *p);
  clrscr();
  head=(node *)malloc(sizeof(node));
  create(head);
  printf("\n");
  print(head);
  printf("\n");
  printf("\n Number of items  = %d \n",count(head));
  getch();
}
void create(node *list)
{
  printf("Input a number\n");
  printf("(type -999 to end) : ");
  scanf("%d",&list->number);
  if(list->number == -999)
         list->next=NULL;
  else
  {
         list->next=(node *)malloc(sizeof(node));
         create(list->next);
```

```
  }
  return;
}
void print(node *list)
{
  if(list->next!=NULL)
  {
        printf("%d - ->",list->number);
  if(list->next->next == NULL)
        printf("%d",list->next->number);
  print(list->next);
  }
  return;
}
int count(node *list)
{
  if(list->next == NULL)
        return(0);
  else
        return(1+count(list->next));
}
```

Q.53 Explain the different types of memory allocations in C. **(6)**

Ans:

Different types of memory allocation function are:

(i) malloc( ): It is a memory allocation function that allocates requested size of bytes and returns a pointer to the first byte of the allocated space. The malloc function returns a pointer of type void so we can assign it to any type of pointer. It takes the the following form:

```
ptr= (cast type *) malloc(byte-size);
```

where ptr is a pointer of type cast-type. For example, the statement

`x=(int *) malloc(10 *sizeof(int))` means that a memory space equivalent to 10 times the size of an int byte is reserved and the address of the first byte of memory allocated is assigned to the pointer x of int type.

The malloc function can also allocate space for complex data types such as structures. For example:

`ptr= (struct student*) malloc(sizeof (struct student));`

where ptr is a pointer of type struct student.

(ii) calloc( ): It is another memory allocation function that allocates space for an array of elements, initializes them to zero and then returns a pointer to the memory. This function is normally used for requesting memory space at run time. It takes the following form:

```
ptr= (cast type *) calloc(n,element-size);
```

This statement allocates contiguous space for n blocks, each of size element-size bytes.

(iii) realloc( ): realloc is a memory allocation function that modifies the size of previously allocated space. Sometime it may happen that the allocated memory space is larger than what is required or it is less than what is required. In both cases, we can change the memory size already allocated with the help of the realloc function known as reallocation of memory. For example, if the original allocation is done by statement

```
ptr= malloc(size);
```

then reallocation is done by the statement

`ptr=realloc(ptr,newsize);` which will allocate a new memory space of size newsize to the pointer variable ptr and returns a pointer to the first byte of the new memory block.

Q.54 Explain the following file functions.

(i) fgetc( ) (ii) ftell
(iii) fgets( ) (iv) rewind( )
(v) fseek **(5)**

Ans:

(i)getc( ): reads a character from a file. This function is used to read a character from a file that has been opened in the read mode. For example, the statement c=getc(fp2); would read a character from the file whose file pointer is fp2.

(ii)ftell( ): gives the current position in the file from the start. ftell takes a file pointer and returns a number of type long, that corresponds to the current position. For example:n=ftell(p); would give the relative offset n in bytes of the current position. This means that n bytes have already been read (or written).

(iii)fgets( ): To read a line of characters from a file,we use the fgets() library function. The prototype is `char *fgets(char *str, int n, FILE *fp);` The argument str is a pointer to a buffer in which the input is to be stored, n is the maximum number of characters to be input, and fp is the pointer to type FILE that was returned by fopen() when the file was opened.

(iv) rewind( ):sets the position to the beginning of the file. It also takes a file pointer and reset the position to the start of the file. For example:

```
rewind(fp);
```

`n=ftell(fp);` would assign 0 to n because file pointer has been set to the start of the file by rewind.

(v)fseek( ):sets the position to a desired point in the file. fseek function is used to move the file position to a desired location within the file. For example: `fseek(fp,m,0);` would move the file pointer to (m+1)th byte in the file.

Q.55 Write a program that reads the following information from the keyboard – student_id, student name and total marks and writer to a file. After taking the information it closes the file and displays the information about the student whose student_id has been entered by the user. **(9)**

Ans:

```
//display records on the basis of student id entered.
#include<stdio.h>
struct student
{
  int stu_id;
  char stu_name[30];
  int marks;
}s[100],b;
void menu();
void create();
void display();
void search();
void heading();
void disp();
int i=0;
FILE *fp,*ft;
void main()
{
```

```
 int first_ch;
 clrscr();
 heading();
 printf("\n\n\n\t\t\t\t    MENU");
 printf("\n\n\n\t\t1.Create a new database.");
 printf("\n\n\t\t2.Continue with the existing database.");
 printf("\n\n\t\t3. Exit.");
 printf("\n\n\n\t\tEnter your choice( 1 - 3): ");
 scanf("%d",&first_ch);
 switch(first_ch)
 {
        case 1:
               create();
               break;
        case 2:
               menu();
                        break;
                      case 3:
              exit();
              break;
        default:
printf("\n\n\n\t\tYou  have  entered  wrong  choice...!!!..The
                     program will exit now.");
 }
 getch();
}
void create()
{
 int ctr=0;
 char ch = 'y';
 fp=fopen("student.c","w");
 while(ch == 'y')
 {
        ctr++;
        clrscr();
        heading();
 printf("\n\n\n\n\tEnter the Employee Records  :\n ");
        printf("\n\n\t\t  Student id : ");
        fflush(stdin);
        scanf("%d",&s[i].stu_id);
        printf("\n\n\t\t        Name : ");
        fflush(stdin);
        scanf("%s",&s[i].stu_name);
        printf("\n\n\t\t Total Marks : ");
        fflush(stdin);
        scanf("%d",&s[i].marks);
        fwrite(&s,sizeof(s[i]),1,fp);
        printf("\n\n\n\t\tDo  you  wish  to  enter  more  records
(y/n) ?");
        ch=getch();
 }
 printf("\n\n\n\t%d   Records   have   been   written   in   the
file.",ctr);
 fclose(fp);
 menu();
}
void menu()
{
        char ch;
        clrscr();
        heading();
        printf("\n\n\n\t\t\t    MENU :-");
        printf("\n\n\t\t1.To DISPLAY all the records");
```

```
        printf("\n\n\t\t2.To SEARCH a record");
        printf("\n\n\t\t3.EXIT");
        printf("\n\n\n\t\tEnter your choice (1 - 3) : ");
        scanf("%d",&ch);
        switch(ch)
        {
               case 1:
                      display();
                      break;
               case 2:
                      search();
                      break;
               case 3:
                      exit();
                      break;
               default:
                       printf("\n\n\tYou have entered a WRONG
                          CHOICE..!!..Please   Re-enter   your
                          choice");
                      menu();
        }
}
void display()
{
 int ctr=0;
 fp=fopen("student.c","r");
 rewind(fp);
 while(fread(&s,sizeof(s[i]),1,fp)==1)
 {
        ctr++;
        clrscr();
        heading();
        printf("\n\n\n\tFollowing are the details :-");
        printf("\n\n\tRecord #%d",ctr);
        printf("\n\n\t\tStudent id : %d",s[i].stu_id);
        printf("\n\n\t\t      Name : %s",s[i].stu_name);
        printf("\n\n\t\tTotal Marks: %d",s[i].marks);
        printf("\n\n\n\n\t\tPlease Press Enter...");
        getch();
 }
 menu();
}
void search()
{
 int f,flag=0;
 char ch='y';
 while(ch=='y')
 {
        clrscr();
        flag=0;
        heading();
 printf("\n\n\n\tEnter the Student id to be searched : ");
        fflush(stdin);
        scanf("%d",&b.stu_id);
        fp=fopen("student.c","r");
        rewind(fp);
        while(fread(&s,sizeof(s[i]),1,fp)==1)
        {
        if(s[i].stu_id == b.stu_id)
               {
                      clrscr();
                      flag=1;
                      heading();
```

```
                                printf("\n\n\n\tThe details of the record
                                       having  Student  id  %d  are  :-
                                       ",b.stu_id);
                                disp();
                              }
                        }
                        fcloseall();
                        if(flag==0)
                        printf("\n\n\n\t\tNo Match found !!");
                printf("\n\n\n\tDo u wish to search for more records  (y/n) ?
                ");
                        ch=getch();
                  }
                menu();
                }
                void heading()
                {
                  printf("\n\n\t\t\tSTUDENT DATABASE MANAGEMENT");
                  printf("\n\n\t\t\t----------------------------");
        }
```

Q.56 Define macros. **(2)**

Ans:

A **macro** is a pre-processor directive which is a program that processes the source code before it passes through the compiler. These are placed in the source program before the main. To define a macro, # define statement is used. This statement, also known as macro definition takes the following general form:

#define identifier string

The pre-processor replaces every occurrence of the identifier in the source code by the string. The preprocessor directive definition is not terminated by a semicolon. For example

#define COUNT 100 will replace all occurrences of COUNT with 100 in the whole program before compilation.

Q.57 Write a C program that reads two strings str1 and str2 and finds the no of occurrence of smaller strings in large string. **(8)**

Ans:

```
#include<conio.h>
#include<string.h>
void main()
{
  char str1[100],str2[100],a[100],b[100],c[100];
  int len1,len2,l1,l2,k=0,i,j,count=0,n;
  clrscr();
  printf("\n Enter the first string : ");
  gets(str1);
  printf("\n Enter the second string : ");
  gets(str2);
  len1=strlen(str1);
  len2=strlen(str2);
  if(len1>len2)
  {
        strcpy(a,str1);
        strcpy(b,str2);
        l1=len1;
        l2=len2;
```

```
  }
  else if(len2>len1)
  {
         strcpy(a,str2);
         strcpy(b,str1);
         l1=len2;
         l2=len1;
  }
  else
         printf("\n\n Please enter one string smaller than the
other!!!");

  for(i=0;i<l1;i++)
  {
         n=i;
         for(j=1;j<=l2;j++)
                c[k++]=a[n++];
         c[k]='\0';
         if(strcmp(b,c)==0)
                ++count;
         k=0;
  }
  printf("\n The number of occurences is : %d",count);
  getch();
}
```

Q.58 Explain path testing. **(6)**

Ans:

Path Testing: Testing in which all paths in the program source code are tested at least once. Path testing has been one of the first test methods, and even though it is a typical white box test, it is nowadays also used in black box tests. First, a certain path through the program is chosen. Possible inputs and the correct result are written down. Then the program is executed by hand, and its result is compared to the predefined. Possible faults have to be written down at once.

Q.59 Distinguish between malloc( ) and calloc( ). **(2)**

Ans:

While malloc allocates a single block of storage space, calloc allocates multiple block of storage, each of the same size, and then sets all bytes to zero. (Explained in Q53)

Q.60 What is a compiler? What type of errors can be detected by it? How does it differ from an interpreter? **(2+3+3)**

Ans:

A **Compiler** is a program that accepts a program written in a high level language and produces an object program. The types of errors detected by a compiler are:

- **Syntax errors** -- The compiler cannot understand a program because it does not follow the **syntax** that is the rules and grammar of a particular language. Common syntax errors include
    - missing or misplaced ; or },
    - missing return type for a procedure,
    - Missing or duplicate variable declaration.
    - Type errors that include

- type mismatch on assignment,
- type mismatch between actual and formal parameters.

An **interpreter** is a program that appears to execute a source program as if it were machine language.

Q.61 What is a subroutine? How do subroutines help in program writing? **(2+2)**

Ans:
A subroutine is a named, independent section of C code that performs a specific task and optionally returns a value to the calling program. Some of the important characteristics of subroutine that help in program writing are:

- A subroutine is named, each have a unique name. By using that name in another part of the program, one can execute the statements contained in the subroutine.
- A subroutine is independent that can perform its task without interference from or interfering with other parts of the program.
- A subroutine performs a specific task. A task is a discrete job that a program must perform as part of its overall operation, such as sending a line of text to a printer, sorting an array into numerical order, or calculating a cube root.
- A subroutine can return a value to the calling program. When a program calls a subroutine, then statements it contains are executed. These statements can pass information back to the calling program.

Q.62 Define the term 'complexity of an algorithm; and explain worst-case and average case analysis of an algorithm. **(2+4)**

Ans:
Complexity of an algorithm is the measure of analysis of algorithm. Analyzing an algorithm means predicting the resources that the algorithm requires such as memory, communication bandwidth, logic gates and time. Most often it is computational time that is measured for finding a more suitable algorithm. This is known as time complexity of the algorithm. The running time of a program is described as a function of the size of its input. On a particular input, it is traditionally measured as the number of primitive operations or steps executed. Worst case analysis of an algorithm is an upper bound on the running time for any input. Knowing that in advance gives us a guarantee that the algorithm will never take any longer. Average case analysis of an algorithm means expected running time of an algorithm on average input data set. The average case is often as bad as the worst case. One problem with performing an average case analysis is that it is difficult to decide what constitutes an "average" input for a particular problem. Often we assume that all inputs of a given size are equally likely.

Q.63 Write a C program to generate a series of Armstrong numbers. A number is an Armstrong number if the sum of the cube of its digit is equal to the number. **(6)**

Ans:
A C program to generate a series of Armstrong numbers is listed below:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
   int n,r,k,sum=0,up;
```

```
    clrscr();
    printf("enter the limit of series ");
    scanf("%d",&n);
    up=n;
    while (up>1)
    {
         k=n;
         sum=0;
         while(n>0)
         {
             r=n%10;
             sum=sum+pow(r,3);
             n=n/10;
          }
         if(sum==k)
         printf("%d\t",k);
         up--;
         n=up;
    }
     getch();
}
```

Q.64 Write a C program to generate the following series:

$$\text{sum} = x + \frac{x^2}{2!} + \frac{x^4}{4!} - - - \frac{x^n}{x!}$$ **(8)**

Ans:

A C program to generate an exponential series is given below:

```
main()
{
  int n,i=0,j=1;
  long int fact=1;
  clrscr();
  printf ("\nEnter the limit for the series:-> ");
  scanf("%d",&n);
  if (n%2!=0)
          n-=1;
    printf("X +");
  for (i=1;i<=n;)
  {
         fact=1;j=1;
         while (j<=i)
         {
                fact*=j;
                j++;
         }
         printf("\tX^%d/%ld\t+",i,fact);
         i*=2;
  }
  getch();
}
```

Q.65 What is bubble sort? Write a program to sort a list of n elements of an array using bubble sort. **(8)**

Ans:

Bubble Sort:

The basic idea in bubble sort is to scan the array to be sorted sequentially several times. Each pass puts the largest element in its correct position by comparing each element in the array with its successor and interchanging the two elements if they are not in proper

order. The first pass put the largest element at (n-1)th position in an array of n elements, next pass put second largest element at (n-2)th position and so on. This way each element slowly bubbles up to its proper position that is why it is called bubble sort. A program sorting array element using bubble sort is given below:

```
#include<stdio.h>
       #include<conio.h>
       void main()
       {
 int a[20],i,j,n,c,flag;
 clrscr();
 printf("how many numbers u want to enter :\n");
 scanf("%d",&n);
 printf("\nenter the numbers :\n");
 for(i=0;i<n;i++)
 scanf("%d",&a[i]);
 for(i=0;i<n-1;i++)
 {
  for(j=0;j<n-1-i;j++)
  {
      if(a[j]>a[j+1])
     {
         c=a[j];
         a[j]=a[j+1];
         a[j+1]=c;
         flag=0;
      }
  }
 if(flag)
  break;
 else
  flag=1;
 }
 printf("sorted elements :\n");
 for(i=0;i<n;i++)
 printf("%d\t",a[i]);
 printf("\n");
 getch();
}
```

Q.66 Explain the difference between a function declaration and function definition. **(4)**

Ans:

Function declaration and Function definition:

A function declaration contains the name of the function, a list of variables that must be passed to it, and the type of variable it returns, if any. For example in the following program in line 2, the function cube is declared. The variables to be passed to the function are called arguments, and they are enclosed in parentheses following the function's name. In this example, the function's argument is long x. The keyword before the name of the function indicates the type of variable the function returns. In this case, a type long variable is returned. The function itself is called the function definition. In the following example, a function cube is defined from line 13 to 18.A function definition has following several parts:

- **Function header**: The function starts out with a function header on line 13. The function header is at the start of a function, and it gives the function's name, the function's return type and describes its arguments. The function header is identical to the function declaration minus the semicolon.

- **Function Body**: The body of the function, lines 14 through 18, is enclosed in braces. The body contains statements that are executed whenever the function is called. Local variables are declared within a function body. Finally, the function concludes with a return statement on line 17, which signals the end of the function and it passes a value back to the calling program.

The following program uses a function to calculate the cube of a number.

```
1: #include <stdio.h>
2: long cube(long x);
3: long a, result;
4: main()
5: {
6: printf("Enter an integer value: ");
7: scanf("%d", &a);
8: result= cube(a);
9: printf("\nThe cube of %ld is %ld.\n", a, result);
10: }
11:
12: /* Function: cube() - Calculates the cube value of a
variable */
13: long cube(long x)
14: {
15: long y;
16: y = x * x * x;
17: return y;
18: }
```

Q.67 What is a pre-processor? Which pre-processor is used to define a macro? **(4)**

Ans:

A pre-processor is a program that processes the source code before it passes through the compiler. It operates under the control of preprocessor directive. These are placed in the source program before the main.

To define a macro, # define statement is used. This statement, also known as macro definition takes the following general form:

```
#define identifier string
```

The pre-processor replaces every occurrence of the identifier in the source code by the string. The preprocessor directive definition is not terminated by a semicolon. For example

`#define COUNT 100` will replace all occurrences of COUNT with 100 in the whole program before compilation.

Q.68 What are the different storage classes in C? **(4)**

Ans:

Storage classes in C

There are four storage classes in C:

a. **Automatic storage class**: The features of variables are as follows

-Storage: Memory
-Default initial value: Garbage value
-Scope: Local to the block in which defined
-Life: till the control remains within the block in which defined.

b. **Register storage class**: The features of variables are as follows

-Storage: CPU registers

-Default initial value: Garbage value
-Scope: Local to the block in which defined
-Life: till the control remains within the block in which defined

c. **Static storage class:** The features of variables are as follows
-Storage: Memory
-Default initial value: Zero
-Scope: Local to the block in which defined
-Life: value of variable persists between different function calls.

d. **External storage class**: The features of variables here are as follows
-Storage: Memory
-Default initial value: Zero
-Scope: global
-Life: As long as program execution does not come to an end.

Q.69 Differentiate between recursion and iteration. **(4)**

Ans:

Recursion and iteration:

The factorial of a number, written as n!, can be calculated as follows:
n! = n * (n-1) * (n-2) * (n-3) * ... * (2) * 1
This approach is called iteration, that is step by step calculation.
However, we can also calculate n! like:
n! = n * (n-1)!
Going one step further, we can calculate (n-1)! using the same procedure:
(n-1)! = (n-1) * (n-2)!
We can continue calculating recursively until the value of n is 1. This approach is called recursion. It refers to a situation in which a function calls itself either directly or indirectly. Indirect recursion occurs when one function calls another function that then calls the first function.

Q.70 Differentiate between pointer (*) and address (&) operator using examples. **(4)**

Ans:

The indirection operator (*) gets the value stored in the memory location whose address is stored in a pointer variable. The address of (&) operator returns the address of the memory location in which the variable is stored. The output of the following example shows the difference between * and &.

```
//difference between * and &.
#include<conio.h>
void main()
{
 int k;
 int *ptr;
 clrscr();
 k=10;
 ptr=&k;
 printf("\n Value of k is %d\n\n",k);
 printf("%d is stored at addr %u\n",k,&k);
 printf("%d is stored at addr %u\n",*ptr,ptr);
 *ptr=25;
 printf("\n Now k = %d\n",k);
 getch();
}
Output: Value of k is 10
```

```
10 is stored at addr 65524
10 is stored at addr 65524
      Now k=25
```

Q.71 Write a program to find the highest of three numbers using pointer to function. **6)**

Ans:

A C program to find the highest of three numbers using pointer to function is listed below:

```
#include<conio.h>
void main()
{
 int x,y,z;
 clrscr();
 printf("\n Enter three numbers : ");
 scanf("%d %d %d",&x,&y,&z);
 printf("\n\n The highest of the three numbers is : ");
 highest(&x,&y,&z);
 getch();
}
highest(a,b,c)
int *a,*b,*c;
{
 if(*a > *b && *a > *c)
        printf("%d",*a);
 else if(*b > *a && *b > *c)
        printf("%d",*b);
 else
        printf("%d",*c);
}
```

Q.72 Differentiate between

i. Static variables and auto variables.
ii. Execution error and compilation error. **(6)**

Ans:

(i) Static variables and auto variables:

Static variables: The features are as follows

Declaration place:-may be declared internally or externally.

Declaration syntax:-we use the keyword static to declare a static variable.

```
Static int age;
```

Default initial value:- Zero

Scope:-in case of internal static variable, the scope is local to the function in which defined while scope of external static variable is to all the functions defined in the program.

Life:- value of variable persists between different function calls.

Automatic variables: The features are as follows

Declaration place:-declared inside a function in which they are to be utilized, that's why referred as local or internal variables.

Declaration syntax:- A variable declared inside a function without storage class specification by default is an automatic variable. However, we may use the keyword auto to declare it explicitly.

```
main()
{
 auto int age;
}
```

Default initial value:-Garbage value

Scope:-created when the function is called and destroyed on exit from the function.

Life:- till the control remains within the block in which defined.

(ii) Execution error and compilation error:

Errors such as mismatch of data types or array out of bound error are known as execution errors or runtime errors. These errors are generally go undetected by the compiler so programs with run-time error will run but produce erroneous results.

Compilation error also known as syntax errors are caused by violation of the grammar rules of the language. The compiler detects, isolate these errors and terminate the source program after listing the errors.

Q.73 Explain the use of structures with pointers. **(5)**

Ans:

C allows a pointer to a structure variable. In fact a pointer to a structure is similar to a pointer to any other variable. The pointer points to the first element of the structure. A structure can have members which points to a structure variable of the same type; such structures are called self-referential structures and widely used in dynamic data structures like trees, linked list etc. A program showing the usage of pointer with structure variable is listed below:

```
#include<stdio.h>
#include<conio.h>
struct student
{
    char name[20];
    int roll_no;
};
void main()
{
   struct student stu[3],*ptr;
   clrscr();
   printf("\n Enter data\n");
   for(ptr=stu;ptr<stu+3;ptr++)
   {  printf ("Name");
       scanf("%s",ptr->name);
       printf("roll_no");
       scanf("%d",&ptr->roll_no);
   }
   printf("\nStudent Data\n\n");
   ptr=stu;
   while(ptr<stu+3)
   {
     printf("%s   %5d\n",ptr->name,ptr->roll_no);
     ptr++;
    }
   getch();
}
```

Q.74 Can a structure be used within a structure? Give appropriate examples to support your answer. **(6)**

Ans:

Yes, a structure can be used within a structure called nesting of structures. For example, inside a structure rectangle, a point (x,y) needs to be represented. So we can first declare a structure point as:

```
struct point
{ int x;
   int y;
};
```

We can now use this definition in defining points in any of geometrical figure. For example

```
struct rectangle
{   struct point pt;
     int  diagonal;
}
```

Q.75 What are the different modes in which a file can be opened? **(5)**

Ans:

Different modes for opening a file are tabulated below:

Modes	Operations
"r"	Open text file for reading
"w"	Open text file for writing, previous content, if any, discarded
"a"	Open or create file for writing at the end of the file
"r+"	Open text file for update
"w+"	Create or open text file for update, previous content lost, if any
"a+"	Open or create text file for update, writing at the end.

Q.76 Write a C program to insert a new node at a specified position in a link list.**(8)**

Ans:

A C program to insert a new node at a specified position in a linked list is listed below:

```
#include<stdio.h>
#include<stdlib.h>
#define NULL 0
struct linked_list
{
  int number;
  struct linked_list *next;
};
typedef struct linked_list node;
void main()
{
  node *head;
  node *n;
  void create(node *p);
  void print(node *p);
  node *insert(node *p);
  clrscr();
  head=(node *)malloc(sizeof(node));
  create(head);
  printf("\n");
  print(head);
  printf("\n");
  n=insert(head);
  printf("\n");
```

```
 print(n);
 getch();
}
void create(node *list)
{
 printf("Input a number\n");
 printf("(type -999 to end) : ");
 scanf("%d",&list->number);
 if(list->number == -999)
        list->next=NULL;
 else
 {
        list->next=(node *)malloc(sizeof(node));
        create(list->next);
 }
 return;
}
void print(node *list)
{
 if(list->next!=NULL)
 {
        printf("%d - ->",list->number);
 if(list->next->next == NULL)
        printf("%d",list->next->number);
 print(list->next);
 }
 return;
}
node *insert(node *head)
{
 node *find(node *p,int a);
 node *new,*tmp;
 node *n1;
 int key;
 int x,i=0;
 printf("\nValue of new item ? ");
 scanf("%d",&x);
 printf("\nValue of key index ? (type -999 if last): ");
 scanf("%d",&key);
 tmp=head;
 while (1)
 {
        if (key==0)
        {
               new=(node *)malloc(sizeof(node));
               new->number=x;
               new->next=head;
               head=new;
        }
        else if(i==key-1)
        {
               new=(node *)malloc(sizeof(node));
               new->number=x;
               new->next=tmp->next;
               tmp->next=new;
        }
        i++;
        tmp=tmp->next;
        if (tmp==NULL)
        {
               printf("\nKey Index out of Reach");
               break;
        }
```

```
  }
  return(head);
}
node *find(node *list,int key)
{
  if(list->next->number==key)
        return(list);
  else
        if(list->next->next==NULL)
              return(NULL);
        else
              find(list->next,key);
}
```

Q.77 Distinguish between the functions islower() and tolower(). **(2)**

Ans:

islower( ) and tolower( ):

islower(c) is a character testing function defined in ctype.h header file. This function determines if the passed argument, in this case c, is lowercase. It returns a nonzero value if true otherwise 0. tolower (c) is a conversion function defined in ctype.h header file that convert argument c to lowercase.

Q.78 Write a C program that reads two strings and copies the smaller string into the bigger string. **(6)**

Ans:

A program to copy smaller string to a bigger string:

```
#include<stdio.h>
#include<conio.h>
void main()
{
   char str1[20],str2[20];
   int m,i,flag=0,j;
   clrscr();
   printf("enter the 1st string");
   gets(str1);
   printf("enter the 2nd string");
   gets(str2);
   if(strlen(str1)<strlen(str2))
   {
       char *str;
        str=str2;
        str2=str1;
        str1=str;
   }
   printf("enter the index after which u want to insert 2nd
   string in 1st : ");
   scanf("%d",&m);
   i=0;
   while(i<=m)
   {
       i++;
   }
   j=0;
   while(str2[j]!='\0')
   {
        str1[i]=str2[j];
        i++;
```

```
            j++;
            if(str1[i]=='\0')
            flag=1;
        }
        if(flag==1)
            str1[i]='\0';
        printf("%s",str1);
        getch();
    }
```

Q.79 Are the following statements valid? Justify your answer

(i) k = (char*)& m

(ii) m= (float*)& p **(4)**

Ans:

Type of m is not known. If it is of type other than char and k is a pointer to char type then statement is valid otherwise it is invalid. Similarly, second statement is valid if m is a pointer to float and p is of type other than float and pointing to more thant four consecutive bytes.

Q.80 Why is C standard library needed? **(4)**

Ans:

C Standard Library:

C would have been a tough programming language in absence of the standard library. The standard library in C includes header files and standard functions in these files. Header files are collection of macros, types, functions and constants. Any program that needs those functions has to include these header files. For example, the standard library functions, included in "stdlib.h" header file, are used for number conversions, storage allocation and other functions on the same line. These functions are called utility functions. C does not have any built-in input/output statements as part of its syntax. All I/O operations are carried out through function calls such as printf and scanf. There are many functions that have become standard for I/O operations in C, collectively known as standard I/O library "stdio.h". Because of this standard input-output header file, standard I/O functions can be used on many machines without any change.

Q.81 What are the functions of the following header files:

(i) ctype.h

(ii) string.h **(4)**

Ans:

(i) ctype.h: It is a header file that contains character testing and conversion functions. For example isalpha (c) returns an int type data and determine if the argument c is alphabetic. It returns nonzero value if true; 0 otherwise.

toascii(c): is a conversion function defined in ctype.h that converts value of argument to ascii.

(ii) string.h: It is also a standard header file that contains string manipulation funcions. For example strcmp(c1,c2) function compares two strings c1 and c2 lexicographically. It returns a negative value if c1<c2; 0 if c1 and c2 are identical; and a positive value if c1>c2.

Q.82 Explain pointers and structures by giving an example of pointer to structure variable? **(5)**

Ans:

We can have a pointer pointing to a structure just the same way a pointer pointing to an int, such pointers are known as structure pointers. For example consider the following example:

```
#include<stdio.h>
#include<conio.h>
struct student
{
     char name[20];
     int roll_no;
};
void main()
{
     struct student stu[3],*ptr;
     clrscr();
     printf("\n Enter data\n");
     for(ptr=stu;ptr<stu+3;ptr++)
    {  printf("Name");
        scanf("%s",ptr->name);
        printf("roll_no");
        scanf("%d",&ptr->roll_no);
    }
    printf("\nStudent Data\n\n");
    ptr=stu;
    while(ptr<stu+3)
   {
        printf("%s   %5d\n",ptr->name,ptr->roll_no);
        ptr++;
    }
getch();
}
```

Here ptr is a structure pointer not a structure variable and dot operator requires a structure variable on its left. C provides arrow operator "->" to refer to structure elements. "ptr=stu" would assign the address of the zeroth element of stu to ptr. Its members can be accessed by statement like "ptr->name". When the pointer ptr is incremented by one, it is made to point to the next record, that is stu[1] and so on.

Q.83 Explain various steps for analysing an algorithm. **(6)**

Ans:

The various steps involved in analysis of an algorithm are:

1. For any algorithm, the first step should be to prove that it always returns the desired output for all legal instances of the problem.
2. Second step to analyze an algorithm is to determine the amount of resources such as time and storage necessary to execute it. Usually the efficiency or complexity of an algorithm is stated as a function relating the input length to the number of steps (time complexity) or storage locations (space complexity). In theoretical analysis of algorithms it is common to estimate their complexity in asymptotic sense, i.e., to estimate the complexity function for reasonably large length of input. Big O notation, omega notation and theta notation are used for this purpose. There are many techniques for solving a particular problem. We must analyze these algorithms and select the one which is simple to follow, takes less execution time and produces required results.

Q.84 What are Translators? Explain its various types. **(3)**

Ans:
A program that translates between high-level languages is usually called a language translator, source to source translator, or language converter.
The various types of translator are:
A compiler converts the source program (user-written program) into an object code (machine language by checking the entire program before execution. If the program is error free, object program is created and loaded into memory for execution. A compiler produces an error list of the program in one go and all have to be taken care even before the execution of first statement begin. It takes less time for execution.
An interpreter is also a language translator that translates and executes statements in the program one by one. It work on one statement at a time and if error free, executes the instruction before going to second instruction. Debugging is simpler in interpreter as it is done in stages. An interpreter takes more time for execution of a program as compared to a compiler.

Q.85 Design an algorithm to generate all the primes in the first *n* positive integers. **(7)**

Ans:
This algorithm will generate all prime numbers less than n.

```
void prime()
{
      int i,n,j;
      printf("enter any number");
      scanf("%d",&n);/*scan the number */
          i=2;
      while(i<n)
        { j=2;
           while(j<=i-1)
      {
        if(i%j==0)
          break;
        else
            j++;
      }
      if(j==i)
       printf("%d\n",i);
      i++;
         }
        }
```

Q.86 Explain various classes of datatypes of C **(4)**

Ans:
Following are the major classes of data types in C language:
1. **Primary data types**: Also known as fundamental/basic data types. All C compliers support four basic data types namely: integer(int), character(char), floating(float), and double precision floating point (double). Then there are extended data types such as long int, double int.
2**. Derived data types**: also known as secondary or user-defined data types. These data types, derived from basic data types, include arrays, pointer, structure, union, enum etc.

Q.87 What are escape sequences characters? List any six of them. **(4)**

Ans:

The characters which when used with output functions like printf( ),putc(),put() etc. helps in formatting the output are known as Escape sequence character. The following is a list of six escape sequences.

\n	Newline
\t	Horizontal Tab
\v	Vertical Tab
\b	Backspace
\r	Carriage Return
\\	Backslash

Q.88 Write a C program to calculate the average of a set of *N* numbers. **(8)**

Ans:

A C program to calculate the average of a set of n numbers:

```
#include<stdio.h>
#include<conio.h>
void main()
{
   int i,a[1000],n,sum;
   float average;
   clrscr();
   printf("how many elements you want to enter");
   scanf("%d",&n);
   printf("enter  values: \n");
   for(i=0;i<n;i++)
   scanf("%d",&a[i]);
   sum=0;
   for(i=0;i<n;i++)
   {
       sum=sum+a[i];
   }
   printf("\n%d",sum);
   average=(float)sum/n;
   printf("\n%f",average);
   getch();
}
```

Q.89 Compare the use of switch statement with the use of nested if-else statement. **(6)**

Ans:

If-else statement: When there are multiple conditional statements that may all evaluate to true, but we want only one if statement's body to execute. We can use an "else if" statement following an if statement and its body; that way, if the first statement is true, the "else if" will be ignored, but if the if statement is false, it will then check the condition for the else if statement. If the if statement was true the else statement will not be checked. It is possible to use numerous else if statements to ensure that only one block is executed.

```
#include <stdio.h>
      void main()
           {
          int age;
          printf( "Please enter your age" );
          scanf( "%d", &age );
          if ( age < 100 ) {
           printf ("You are pretty young!\n" );  }
          else if ( age == 100 ) {
```

```
            printf( "You are old\n" );
              }
          else {
            printf( "You are really old\n" );
          }
        }
```

Switch case statements are a substitute for long if statements that compare a variable to several "integral" values ("integral" values are simply values that can be expressed as an integer, such as the value of a char).The value of the variable given into switch is compared to the value following each of the cases, and when one value matches the value of the variable, the computer continues executing the program from that point. The condition of a switch statement is a value. The case says that if it has the value of whatever is after that case then do whatever follows the colon. The break is used to break out of the case statements. Break is a keyword that breaks out of the code block, usually surrounded by braces, which it is in. In this case, break prevents the program from falling through and executing the code in all the other case statements.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int flag;
    printf( "Enter any value\n" );
    scanf( "%d", &flag );
    switch ( flag ) {
        case 1:
            printf( "It is hot weather!\n" );
            break;
        case 2:
            printf( "It is a stormy weather!\n" );
            break;
        case 3:
            printf( "It is a sticky weather!\n" );
            break;
        default:
            printf( "It is a pleasant weather!\n" );
            break;
    }
    getch();
}
```

Q.90 What do you mean by underflow and overflow of data. **(2)**

Ans:

Underflow and overflow of data:

When the value of the variable is either too long or too small for the data type to hold, the problem of data overflow or underflow occurs. The largest value that a variable can hold depends on the machine. Since floating point values are rounded off to the number of significant digits allowed, an overflow results in the largest possible real value whereas an underflow results in zero. C does not provide any warning or indication of integer overflow, it simply give erroneous result.

Q.91 Write a C program to multiply two matrices (maximum size of the two matrices is 20 x 20 each). **(8)**

Ans:

A C program to multiply two matrices:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[20][20],b[20][20],c[20][20],i,j,k,m,n,p,q;
clrscr();
printf("enter no. of rows and column for 1st matrix\n");
printf("no.of rows\n");
scanf("%d",&m);
printf("no. of columns");
scanf("%d",&n);
printf("\nenter no. of rows and columns for 2nd matrix");
printf("\nno. of rows");
scanf("%d",&p);
printf("\nno. of columns");
scanf("%d",&q);
if(n==p)
{
   printf("enter elements for matrix A");
   for(i=0;i<m;i++)
  {
     for(j=0;j<n;j++)
       scanf("%d",&a[i][j]);
   }
   printf("enter elements for matrix B");
   for(i=0;i<p;i++)
  {
     for(j=0;j<q;j++)
        scanf("%d",&b[i][j]);
   }
    printf("\nMATRIX A:\n");
                for(i=0;i<m;i++)
  {
     for(j=0;j<n;j++)
     {
         printf("%d",a[i][j]);
         printf("\t");
      }
      printf("\n");
    }
    printf("\nMATRIX B:\n");
    for(i=0;i<p;i++)
    {
    for(j=0;j<q;j++)
   {
      printf("%d",b[i][j]);
      printf("\t");
    }
    printf("\n");
}
printf("\n");
printf("MULTIPLICATION :\n");
for(i=0;i<m;i++)
{
    for(j=0;j<q;j++)
   {
       for(k=0;k<p;k++)
      {
           c[i][j]=c[i][j]+(a[i][k]*b[k][j]);
       }
```

```
            printf("%d",c[i][j]);
            c[i][j]=0;
            printf("\t");
          }
          printf("\n");
        }
      }
      else
          printf("multiplication is not possible");

      getch();
      }
```

Q.92 Explain, in brief the purpose of the following string handling functions:
(i) strcat (ii) strcmp (iii) strcpy
Use suitable examples **(6)**

Ans:

(i) strcat() Function concatenates two strings together and has the following form:

```
strcat(string1,string2);
```

When this function is executed, string2 is appended to string1 by removing the null character at the end of string1.
C permits nesting of strcat functions as strcat(strcat(string1,string2),string3);
(ii) strcmp( ) is the string comparison function defined in string.h header file. It has the following form:.

```
int strcmp ( const char *s1, const char *s2 );
```

strcmp will accept two strings. It will return an integer. This integer will either be:
Negative if s1 is less than s2.
Zero if s1 and s2 are equal.
Positive if s1 is greater than s2.
Strcmp performs a case sensitive comparison; if the strings are the same except for a difference in case, then they're countered as being different. Strcmp also passes the address of the character array to the function to allow it to be accessed.
(iii) strcpy ( ) function is just like a string-assignment operator which take the following form:

```
char *strcpy ( char *dest, const char *src );
```

strcpy is short for string copy, which means it copies the entire contents of src into dest. The contents of dest after strcpy will be exactly the same as src such that strcmp (dest, src) will return 0.src may be a character array variable or a string constant.

Q.93 Write a C program to read a line of text containing a series of words from the terminal. **(7)**

Ans:

```
#include<stdio.h>
#include<conio.h>
void main()
{
   char text[100],ch;
   int i=0;
   clrscr();
   printf("Enter text. Press <return> at the end\n");
   do
   { ch=getchar();
     text[i]=ch;
     i++;
   }while(ch!='\n');
```

```
        i=i-1;
        text[i]='\0';
                    printf("The entered text is");
        printf("\n%s\n",text);
        getch();
    }
```

Q.94 Explain the need for user-defined functions. **(3)**

Ans:

The need for user-defined function:

1. A programmer may have a block of code that he has repeated forty times throughout the program. A function to execute that code would save a great deal of space, and it would also make the program more readable.
2. It is easy to locate and isolate a faulty function. Having only one copy of the code makes it easier to make changes.
3. Another reason for functions is to break down a complex program into logical parts. For example, take a menu program that runs complex code when a menu choice is selected. The program would probably best be served by making functions for each of the actual menu choices, and then breaking down the complex tasks into smaller, more manageable tasks, which could be in their own functions. In this way, a program can be designed that makes sense when read. And has a structure that is easier to understand quickly. The worst
programs usually only have the required function, main, and fill it with pages of jumbled code.
4. A function may be used by many other programs. A programmer can use already compiled function instead of starting over from scratch.

Q.95 Write a program in C to find the sum of the first 100 natural numbers
Sum=1+2+3+…….100 **(8)**

Ans:

A program to find the sum of first 100 natural numbers:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
    int i,sum;
    clrscr();
    sum=0;
    for(i=1;i<=100;i++)
    {
        sum=sum+i;
     }
    printf("%d\t",sum);
    getch();
              }
```

Q.96 List any six commonly found programming errors in a C program. **(6)**

Ans:

Six commonly found errors in a C program are:

1. Missing or misplaced ; or }, missing return type for a procedure, missing or duplicate variable declaration.

2. Type mismatch between actual and formal parameters, type mismatch on assignment.
3. Forgetting the precedence of operators, declaration of function parameters.
4. Output errors means the program runs but produces an incorrect result. This indicates an error in the meaning of the program.
5. Exceptions that include division by zero, null pointer and out of memory.
6. Non-termination means the program does not terminate as expected, but continues running "forever."

Q.97 Define a structure in C, which stores subject-wise marks of a student. Using a student array, write a C program to calculate the total marks in each subject for all the students. **(8)**

Ans:

```
#include<stdio.h>
#include<conio.h>
struct marks
{
int sub1,sub2,sub3;
int total;
};
void main()
{
int i,n;
struct marks student[20];
struct marks total;
total.sub1=0;
total.sub2=0;
total.sub3=0;
clrscr();
printf("enter no. of students");
scanf("%d",&n);
for(i=0;i<n;i++)
{
 printf("\nenter marks :");
 scanf("%d%d%d",&student[i].sub1,
&student[i].sub2,&student[i].sub3);
 total.sub1=total.sub1+student[i].sub1;
 total.sub2=total.sub2+student[i].sub2;
 total.sub3=total.sub1+student[i].sub3;
 printf("\n");
}
printf("SUBJECT       TOTAL\n");
printf("Sub1          %d",total.sub1);
printf("\n");
printf("Sub2          %d",total.sub2);
printf("\n");
printf("Sub3          %d",total.sub3);
printf("\n");
getch();
    }
```

Q.98 (**a**) What is dynamic memory allocation? Explain the various memory allocation function with its task. **(3)**

(b) Why a linked list is called a dynamic data structure? What are the advantages of using linked list over arrays? **(6)**

(c) What is a macro? How is it different from a C variable name? What are the advantages of using macro definitions in a program. **(6)**

(d) What are the different modes in which a file can be opened. **(4)**

Ans:

(a) The mechanism of allocating required amount of memory at run time is called dynamic allocation of memory. Sometimes it is required to allocate memory at runtime. When we declare array in any program, the compiler allocates memory to hold the array. Now suppose the numbers of items are larger than the defined size then it is not possible to hold all the elements and if we define an array large enough and data to be stored is less, in that case the allocated memory is wasted leading to a need for dynamic memory allocation. In this mechanism we use a pointer variable to which memory is allocated at run time.

The various memory allocation functions are described below:

(i) malloc( ): It is a memory allocation function that allocates requested size of bytes and returns a pointer to the first byte of the allocated space. The malloc function returns a pointer of type void so we can assign it to any type of pointer. It takes the the following form:

```
ptr= (cast type *) malloc(byte-size);
```

where ptr is a pointer of type cast-type. For example, the statement

`x=(int *) malloc(10 *sizeof(int))` means that a memory space equivalent to 10 times the size of an int byte is reserved and the address of the first byte of memory allocated is assigned to the pointer x of int type.

The malloc function can also allocate space for complex data types such as structures. For example:

`ptr= (struct student*) malloc(sizeof (struct student));` where ptr is a pointer of type struct student.

(ii) calloc( ): It is another memory allocation function that allocates space for an array of elements, initializes them to zero and then returns a pointer to the memory. This function is normally used for requesting memory space at run time. While malloc allocates a single block of storage space, calloc allocates multiple block of storage, each of the same size, and then sets all bytes to zero. It takes the following form:

```
ptr= (cast type *) calloc(n,element-size);
```

This statement allocates contiguous space for n blocks, each of size element-size bytes.

(iii) realloc( ): realloc is a memory allocation function that modifies the size of previously allocated space. Sometime it may happen that the allocated memory space is larger than what is required or it is less than what is required. In both cases, we can change the memory size already allocated with the help of the realloc function known as reallocation of memory. For example, if the original allocation is done by statement

```
ptr= malloc(size);
```

then reallocation is done by the statement

`ptr=realloc(ptr,newsize);` which will allocate a new memory space of size newsize to the pointer variable ptr and returns a pointer to the first byte of the new memory block

(b) A linked list is called a dynamic data structure because it can be used with a data collection that grows and shrinks during program execution. The major advantage of using linked list over arrays is in implementing any data structure like stack or queue. The arrays are fixed in size and so a lot of memory gets wasted if we declare in prior the estimated size and the used space is less. Also it is not time efficient to perform deletion, insertion and updating of information in an array implementation because of

its fixed length memory storage. A linked list allocation storage result in efficient use of memory and computer time. Linked lists are useful over arrays because:
The exact amount of memory space required by a program depends on the data being processed and so this requirement cannot be found in advance. Linked list uses run-time allocation of memory resulting in no wastage of memory space.
Some programs require extensive use of data manipulation like insertion of new data, deletion and modification of old data. Linked lists are self referential structures so provide an efficient time complexity for these operations.
(c) A macro is a pre-processor directive which is a program that processes the source code before it passes through the compiler. These are placed in the source program before the main. To define a macro, # define statement is used. This statement, also known as macro definition takes the following general form:
#define identifier string
The pre-processor replaces every occurrence of the identifier in the source code by the string. The preprocessor directive definition is not terminated by a semicolon. For example
#define COUNT 100 will replace all occurrences of COUNT with 100 in the whole program before compilation.
(d) Different modes for opening a file are tabulated below:

Modes	Operations
"r"	Open text file for reading
"w"	Open text file for writing, previous content, if any, discarded
"a"	Open or create file for writing at the end of the file
"r+"	Open text file for update
"w+"	Create or open text file for update, previous content lost, if any
"a+"	Open or create text file for update, writing at the end.

Q.99 Explain the following directives:
#elif #pragma #error **(6)**

Ans:

(i) '#elif' is a preprocessor directive that provides alternate test facility. `#elif' stands for "else if". Like `#else', it goes in the middle of a conditional group and subdivides it; it does not require a matching `#endif' of its own. Like `#if', the `#elif' directive includes an expression to be tested. The text following the `#elif' is processed only if the original `#if'-condition failed and the `#elif' condition succeeds.

For example, suppose we have following set of statements:

```
#if X == 1
...
#else /* X != 1 */
#if X == 2
...
#else /* X != 2 */
...
#endif /* X != 2 */
#endif /* X != 1 */
```

Using *`#elif'*, we can abbreviate this as follows:

```
#if X == 1
...
#elif X == 2
```

```
...
#else /* X != 2 and X != 1*/
...
#endif /* X != 2 and X != 1*/
```

(ii) #pragma is an implementation oriented directive that specifies various instructions to be given to the compiler.

```
#pragma  name
```

causes compiler to perform "name"

(iii)#error is a preprocessor directive used to produce diagnostic messages during debugging.

```
#error message
```

causes the compiler to display the error message and terminate processing on encountering this directive.

Q.100 Using recursion, write a C program to reverse a given number. **(6)**

Ans:

```
#include<stdio.h>
#include<conio.h>
void main()
{
      int n,r;
      clrscr();
      printf("enter an integer");
      scanf("%d",&n);
      rev(n);
      getch();
}
rev (int n)
{
       if (n>0)
       {
         printf ("%d",n%10);
                rev(n/10);
       }
}
```

Q.101 Write a C function to delete a given item from a single linked list. Check for duplicate elements. **(8)**

Ans:

```
struct node
{
  int data ;
  struct node * link ;
} ;
void delete ( struct node **q, int num )
{
  struct node *old, *temp ;
  temp = *q ;

  while ( temp != NULL )
  {
        if ( temp -> data == num )
        {
/* if node to be deleted is the first node in the linked list */
              if ( temp == *q )
                    *q = temp -> link ;
```

```
            /* deletes the intermediate nodes in the linked list */
                  else
                        old -> link = temp -> link ;
                  /* free the memory occupied by the node */
                  free ( temp ) ;
                  return ;
            }
      /* traverse the linked list till the last node is reached */
            else
            {
                  old = temp ;  /* old points to the previous node */
                  temp = temp -> link ;  /* go to the next node */
             }
      }
      printf ( "\nElement %d not found", num ) ;
   }
```

Q.102 Consider the following:
P_1 is an integer pointer
P_2 is a long integer pointer
P_3 is a character type pointer
The initial value of P_1 is 2800, P_2 is 1411 and P_3 is 1201.
What is the new value of P_1 after $P_1=P_1+1$, P_2 after $P_2=P_2+1$ and P_3 after $P_3=P_3+1$; **(4)**

Ans:
The initial value of P_1 is 2800 which is an integer pointer so new value of P_1 is 2802 after $P_1= P_1+1$
The initial value of P_2 is 1411 which is a long integer pointer so new value of P_1 is 1415 after $P_2= P_2+1$ because long takes 4 bytes of memory.
The initial value of P_3 is 1201 which is a char type pointer so new value of P_3 is 1202 after $P_3= P_3+1$

Q.103 Differentiate between White Box and Black Box Testing. **(4)**

Ans:
White box testing strategy deals with the internal logic and structure of the code. White box testing also known as glass, structural, open box or clear box testing, tests code written, branches, paths, statements and internal logic of the code etc.
In order to implement white box testing, the tester has to deal with the code and hence is needed to possess knowledge of coding and logic i.e. internal working of the code. White box test also needs the tester to look into the code and find out which unit/statement/chunk of the code is malfunctioning. White box testing is applicable at unit, integration and system level however it is mainly done at unit level.
Black box testing Black-box test design treats the system as a "black-box", so it doesn't explicitly use knowledge of the internal structure. It takes an external perspective of the test object to derive test cases. These tests can be functional or non-functional, though usually functional. The test designer selects valid and invalid input and determines the correct output. There is no knowledge of the test object's internal structure.
This method of test design is applicable to all levels of software testing: unit, integration, functional testing, system and acceptance.

Q.104 Write a C program using pointers to compute the sum of all elements stored in an array. **(8)**

Ans:

```
void main()
{
       int i,*p,s;
       static int a[10]={10,20,30,40,50,60,70,80,90,100};
       clrscr();
       i=0;
      p=a;
       s=0;
      while(i<10)
      {
         s=s+ *p;
         i++;
  p++;
}
 printf("sum=%d",s);
 getch();
   }
```

Q.105 Write a C program to create a file contains a series of integer numbers and then reads all numbers of this file and write all odd numbers to other file called odd and write all even numbers to a file called even. **(8)**

Ans:

```
#include<stdio.h>
#include<conio.h>
void main()
   {
  FILE *f1,*f2,*f3;
  int i,j;
  printf("Enter the data\n");
  f1=fopen("file1","w");
  for(i=0;i<=10;i++)
  { scanf("%d",&j);
      if(j== -1) break;
     putw(j,f1);
   }
   fclose(f1);
   f1=fopen("file1","r");
   f2=fopen("od","w");
   f3=fopen("ev","w");

   while((j=getw(f1))!=EOF)
   { if(j%2==0)
      putw(j,f3);
      else
      putw(j,f2);
   }
   fclose(f1);
   fclose(f2);
   fclose(f3);
   f2=fopen("od","r");
   f3=fopen("ev","r");
   printf("\nContents of odd file\n");
   while((j=getw(f2))!=EOF)
      printf("%4d",j);
   printf("\nContents of even file\n");
```

```
    while((j=getw(f3))!=EOF)
       printf("%4d",j);
      fclose(f2);
      fclose(f3);
      getch();
}
```

Q.106 What is structured programming? Explain its advantages. **(8)**

Ans:

Structured Programming: means the collection of principles and practices that are directed toward developing correct programs which are easy to maintain and understand. A structured program is characterized by clarity and simplicity in its logical flow.

The Three important constructs upon which structured programming is built are:

- Sequence-Steps are performed one after the other.
- Selection-Logic is used to determine the execution of a sequence of steps. For example,if-then-else construct.
- Iteration-A set of actions are repeated until a certain condition is met. For example while construct.

The various advantages of structured programming are:

1. Complexity of program reduces.
2. Easy maintenance.
3. Simplified understanding.
4. Reusability of code increases.
5. Testing and debugging become easier.

Q.107 What are the essential features of an algorithm? Write an algorithm to obtain H.C.F. of two given positive integers. **(8)**

Ans:

Essential features of an algorithm:

1.Input: The algorithm should take zero or more input.

2. Output: The algorithm should produce one or more outputs.

3. Definiteness: Each and every step of algorithm should be defined unambiguously.

4. Effectiveness: A human should be able to calculate the values involved in the procedure of the algorithm using paper and pencil.

5. Termination: An algorithm must terminate after a finite number of steps.

Writing algorithms is an art so the language used to write algorithms should be simple and precise. Each and every step of the algorithm should be clear and unambiguous and should not convey more than one meaning to the programmer.

A C language algorithm to obtain H.C.F. of two given positive integers is listed below:

```
void hcf()
{
 int a,b,r,h,k,c;
 clrscr();
 printf("enter two numbers");
 scanf("%d%d",&a,&b);
 h=a;
 k=b;
 while(r!=0)
```

```
  {
     r=a%b;
     a=b;
     b=r;
  }
  printf("H.C.F. of %d and %d = %d",h,k,a);
}
```

Q.108 How do you calculate the complexity of sorting algorithms? Find the complexity of Insertion sort and Bubble Sort. **(8)**

Ans:

The complexity of sorting algorithms depends on the input: sorting a thousand numbers takes longer than sorting three numbers. The best notion of input size depends on the problem being studied. For sorting algorithms, the most natural measure is the number of items in the input that is array size n. We start by associating time "cost" of each statement and the number of times each statement is executed assuming that comments are not executable statements, and so do not take any time.

Complexity of Bubble sort: $O(n^2)$

Complexity of Insertion sort: $O(n^2)$

Q.109 (**a**) Write a C program to compare and copy the structure variables with example. **(8)**

(b) Write a C program using pointers to determine the length of a character string. **(8)**

(**c**) Compare the 2 constructs Arrays of structures and Arrays within structure. Explain with the help of examples. **(8)**

Ans:

(**a**) A C program to compare and copy the structure variables is listed below:

```
#include<stdio.h>
 #include<conio.h>
 struct student
 {
    char name[10];
    int marks;
 };
 void main()
 {
    int i,j;
    static struct student stu1={"Simmi",78};
    static struct student stu2={"Ruci",90};
    static struct student stu3={"Seema",95};
    clrscr();
    stu1=stu3;
    if((stu1.marks==stu3.marks))
    {
        printf(" Student 1 is same as student 3\n");
        printf("%s",stu1.name);
        printf("\t");
        printf("%d",stu3.marks);
        printf("\n");
    }
    else
    printf("\nStudents are different\n");
    getch();
```

```
}
```

(b) A C program to determine the length of a character string:

```
#include<stdio.h>
#include<conio.h>
#include<process.h>
void main()
{  char *a,*ptr;
   int i;
    clrscr();
 printf("enter the string");
 gets(a);
 ptr=a;
 while(*ptr!='\0')
{   ptr++;
  }
  i=ptr-a;
printf("length=%d",i);
            getch();
             }
```

(c) Array of structures and arrays within structures:

Structures variables are just like ordinary variables so we can create an array of structures. However arithmetic operations cannot be performed on structure variables. For example, we can create an array of structure student as shown below:

```
struct student
      {
            char name[30];
            int marks;
      };
```

`struct student list[4]={ { "aaa",90},{"bbb",80},{"ccc",70}};` declare and initialize an array of structure student.

A Structure is a heterogeneous collection of variables grouped under a single logical entity. An array is also a use-defined data type and hence can be used inside a structure just like an ordinary variable. As in the above example, name is a char array within the structure student.

Q.110 Describe the output that will be generated by the following 'C' program. **(4)**

```
#include<stdio.h>
main( )
{
    int i=0, x=0;
    while(i<20)
    { if(i%5 = = 0)
        {x + = i;
          printf("%d", x);
         }
         ++i;
     }
printf("\nx = %d", x);
}
```

Ans:

The output generated by given C program is:

051530

x=30

initially i=0<20, (i%5==0) is true so x=x+i=0+0=**0**, i is incremented to 2, again the next while loop, however condition (i%5==0) is false so i incremented and so on till i=5.

At i=5, (i%5==0) is true so x=x+i=0+5=**5**, i is incremented to 6, again the next while loop, however condition (i%5==0) is false so i incremented and so on till i=10.
At i=10, (i%5==0) is true so x=x+i=5+10=**15**, i is incremented to 11, again the next while loop, however condition (i%5==0) is false so i incremented and so on till i=15.
At i=15, (i%5==0) is true so x=x+i=15+15=**30**, i is incremented to 16, again the next while loop, however condition (i%5==0) is false so i incremented and out of while loop at i=20.
Outer block, it is printing the value of x which is **30**.

Q.111 Write a complete program to create a singly linked list. Write functions to do the following operations

(i) Count the number of nodes
(ii) Add a new node at the end
(iii) Reverse the list. **(10)**

Ans:

A complete program to create a linked list, counting number of nodes, adding new node at the end and reversing the list is as follows:

```
#include <stdio.h>
#include <conio.h>
#include <alloc.h>
/* structure containing a data part and link part */
struct node
{
 int data ;
 struct node *link ;
} ;
void append ( struct node **, int ) ;
void reverse ( struct node ** ) ;
void display ( struct node * ) ;
int count ( struct node * ) ;
void main( )
{
 struct node *p ;
 int n;
 p = NULL ;  /* empty linked list */
 append ( &p, 14 ) ;
 append ( &p, 30 ) ;
 append ( &p, 25 ) ;
 append ( &p, 42 ) ;
 append ( &p, 17 ) ;
 clrscr( ) ;
 display ( p ) ;
 printf ( "\nNo. of elements in the linked list = %d", count (
p ) ) ;
   printf("\n Enter the element you want to insert");
 scanf("%d",&n);
 append (&p,n);
 display ( p ) ;
 printf ( "\nNo. of elements in the linked list = %d", count (
p ) ) ;
 reverse ( &p ) ;
 printf("\n Elements after reversing the linked list\n");
 display (p);
}
/* adds a node at the end of a linked list */
void append ( struct node **q, int num )
{
 struct node *temp, *r ;
```

```
  if ( *q == NULL ) /* if the list is empty, create first node
*/
  {
        temp = malloc ( sizeof ( struct node ) ) ;
        temp -> data = num ;
        temp -> link = NULL ;
        *q = temp ;
  }
  else
  {
        temp = *q ;
        /* go to last node */
        while ( temp -> link != NULL )
              temp = temp -> link ;
        /* add node at the end */
        r = malloc ( sizeof ( struct node ) ) ;
        r -> data = num ;
        r -> link = NULL ;
        temp -> link = r ;
  }
}
void reverse ( struct node **x )
{
  struct node *q, *r, *s ;
  q = *x ;
  r = NULL ;
  /* traverse the entire linked list */
  while ( q != NULL )
  {
        s = r ;
        r = q ;
        q = q -> link ;
        r -> link = s ;
  }
  *x = r ;
}
int count(struct node *list)
{
  if(list->next == NULL)
        return(0);
  else
        return(1+count(list->next));
}
/* displays the contents of the linked list */
void display ( struct node *q )
{
  printf ( "\n" ) ;
  /* traverse the entire linked list */
  while ( q != NULL )
  {
        printf ( "%d ", q -> data ) ;
        q = q -> link ;
  }
}
```

Q.112 What is recursion? Write a recursive program to reverse a string. **(8)**

Ans:

Recursion is a special case of function call where a function calls itself. These are very useful in the situations where solution can be expressed in terms of

successively applying same operation to the subsets of the problem. For example, a recursive function to calculate factorial of a number n is given below:

```
fact(int n)
{
int factorial;
if(n==1||n==0)
      return(1);
else
factorial=n*fact(n-1);
return (factorial);
}
```

Assume n=4, we call fact(4)

Since n≠1or 0, factorial=n*fact(n-1)

Factorial=4*fact(3)	(again call fact function with n=3)
=4*3*fact(2)	(again call fact function with n=2)
=4*3*2*fact(1)	(again call fact function with n=1)
=4*3*2*1	(terminating condition)
=24	

We should always have a terminating condition with a recursive function call otherwise function will never return.

A recursive program to reverse a string is:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    char str[100];
    clrscr();
    printf("enter a string");
    gets(str);
    printf("reverse of string is:\n");
    rev(str);
    getch();
}
rev (char *string)
{
    if (*string)
   {
       rev(string+1);
       putchar(*string);
    }
}
```

Q.113 Distinguish between the following:

(i) Automatic and static variables

(ii) Global and local variables. **(8)**

Ans:

(i) **Automatic and Static variables:**

Automatic variables: The features are as follows

Declaration place:-declared inside a function in which they are to be utilized, that's why referred as local or internal variables.

Declaration syntax:- A variable declared inside a function without storage class specification by default is an automatic variable. However, we may use the keyword auto to declare it explicitly.

```
main()
{
```

```
auto int age;
   }
```

Default initial value:- Garbage value
Scope:-created when the function is called and destroyed on exit from the function.
Life:- till the control remains within the block in which defined.
Static variables: The features are as follows
Declaration place:-may be declared internally or externally.
Declaration syntax:-we use the keyword static to declare a static variable.
Static int age;
Default initial value:- Zero
Scope:-in case of internal static variable, the scope is local to the function in which defined while scope of external static variable is to all the functions defined in the program.
Life:- value of variable persists between different function calls.
(ii) **Global and Local Variables**
Global variables: The features are as follows
Declared outside of all functions or before main.
These can be used in all the functions in the program.
It need not be declared in other functions.
A global variable is also known as an external variable.
Local variables: The features are as follows
Declared inside a function where it is to be used.
These are not known to other function in the program.
These variables are visible and meaningful inside the functions in which they are declared.
For example:

```
#include<stdio.h>
int m;
void main( )
{ int i;
  …..
…..
 Fun1();
}
Fun1( )
{ int j;
  ….
…..
}
```

Here m is a global variable , i is local variable local to main( ) and j is local variable local to Fun1( ).

Q.114 What would be the output of the program given below? **(6)**

```
typedef struct soldier{
char *name;
char *rank;
int serial_number;} SOLDIER;
SOLDIER soldier1, soldier2, soldier3, *ptr;
ptr = &soldier3;
soldier3.name = "Anand Mohanti";
printf("\n%s", (*ptr).name);
printf("\n%c", *ptr->name);
printf("\n%c", *soldier3.name);
```

```
printf("\n%c", *(ptr-> name + 4));
```

Ans:

Output of the program would be:

Anand Mohanti
A
A
d

ptr is pointing to address of soldier3 so (*ptr).name would print the soldier3.name that is Anand Mohanti

*ptr->name would print first character of soldier3 name so printed 'A'.

Similarly 3rd line.

ptr->name means first char of name so ptr->name+4 points to 4th location from beginning so this statement will print d.

Q.115 Define a structure to store the following information about an employee Name, Sex(male, female), Marital_status(single, married, divorced or widowed), age.(using bit fields) **(4)**

Ans:

Definition of a structure to store information of an employee:

```
struct employee
{
              char name[20];
     unsigned sex:  1;
     unsigned martial_status:     1;
     unsigned age:  7;
}emp;
```

Q.116 How is multidimensional arrays defined in terms of an array of pointer? What does each pointer represent? **(6)**

Ans:.

An element in a multidimensional array like two-dimensional array can be represented by pointer expression as follows:

```
*(*(a+i)+j)
```

It represent the element a[i][j]. The base address of the array a is &a[0][0] and starting at this address, the compiler allocates contiguous space for all the element row-wise. The first element of second row is immediately after last element of first row and so on.

Q.117 Write a program to implement the stack using linked list. **(8)**

Ans:

A C program to implement the stack using linked list is given below:

```
/*  Program to implement stack as linked list*/
#include <stdio.h>
#include <conio.h>
#include <alloc.h>
/* structure containing data part and linkpart */
struct node
{
  int data ;
  struct node *link ;
```

```
} ;
void push ( struct node **, int ) ;
int pop ( struct node ** ) ;
void delstack ( struct node ** ) ;
void display ( struct node * ) ;
void main( )
{
 struct node *s = NULL ;
 int i ;
 clrscr( ) ;
 push ( &s, 1 ) ;
 push ( &s, 2 ) ;
 push ( &s, 3 ) ;
 push ( &s, 4 ) ;
 push ( &s, 5 ) ;
 push ( &s, 6 ) ;
 printf("The stack currently contains\n");
       display ( s ) ;
 i = pop ( &s ) ;
 printf ( "\nItem popped: %d", i ) ;
 i = pop ( &s ) ;
  printf ( "\nItem popped: %d", i ) ;
 i = pop ( &s ) ;
 printf ( "\nItem popped: %d", i ) ;
 printf("Stack after deleting three items\n");
       display (s);
 delstack ( &s ) ;
 getch( ) ;
}
/* adds a new node to the stack as linked list */
void push ( struct node **top, int item )
{
 struct node *temp ;
temp = ( struct node * ) malloc ( sizeof ( struct node ) ) ;
 if ( temp == NULL )
       printf ( "\nStack is full." ) ;
 temp -> data = item ;
 temp -> link = *top ;
 *top = temp ;
}
/* pops an element from the stack */
int pop ( struct node **top )
{
 struct node *temp ;
 int item ;
 if ( *top == NULL )
 {
       printf ( "\nStack is empty." ) ;
       return NULL ;
 }
 temp = *top ;
 item = temp -> data ;
 *top = ( *top ) -> link ;
 free ( temp ) ;
 return item ;
}
/* displays the contents of the stack */
void display ( struct node *q )
{
 printf ( "\n" ) ;
 /* traverse the entire stack */
 while ( q != NULL )
 {
```

```
        printf ( "%d ", q -> data ) ;
        q = q -> link ;
  }
}
/* deallocates memory */
void delstack ( struct node **top )
{
  struct node *temp ;
  if ( *top == NULL )
        return ;
  while ( *top != NULL )
  {
        temp = *top ;
        *top = ( *top ) -> link ;
        free ( temp ) ;
  }
}
```

Q.118 Write a C program to calculate the standard deviation of an array of values. The array elements are read from the terminal. Use functions to calculate standard deviations and mean Standard Deviation **(10)**

Ans:

A C program to calculate mean and standard deviation of an array of value:

```
#include<math.h>
#include<conio.h>
#define MAXSIZE 100
void main()
{
  int i,n;
  float
value[MAXSIZE],deviation,sum,sumsqr,mean,variance,stddeviation
;
  sum=sumsqr=n=0;
  clrscr();
  printf("\n Input values : input -1 to end \n");
  for(i=1;i<MAXSIZE;i++)
  {
        scanf("%f",&value[i]);
        if(value[i]==-1)
              break;
        sum+=value[i];
        n+=1;
  }
  mean=sum/(float)n;
  for(i=1;i<=n;i++)
  {
        deviation=value[i]-mean;
        sumsqr+=deviation*deviation;
  }
  variance=sumsqr/(float)n;
  stddeviation=sqrt(variance);
  printf("\n Number of items : %d\n",n);
  printf("              Mean : %f\n",mean);
  printf("Standard Deviation : %f\n",stddeviation);
  getch();
    }
```

Q.119 A file of employees contains data (eno, name and salary) for each employee of a company. Write a program to do the following:

(i) create the file

(ii) insertion in a file
(iii) deletion from a file
(iv) modification in a file **(12)**

Ans:

A program to perform creation, insertion, deletion and modification in a file:

```
#include<stdio.h>
struct rec
{
    int empno;
    char name[25],add[50];
    float salary;
}e;
FILE *fp1,*fp2;
main()
{
   int ch;
    while (1)
    {
clrscr();printf    ("\n1.Insert    a    new    Record\n2.Display
All\n3.Modify One\n4.Delete One\n5.Exit\nEnter your choice:-> ");
    scanf ("%d",&ch);
    switch (ch)
    {
           case 1:      insert();break;
           case 2:      display();break;
           case 3:      modify();break;
           case 4:      del();break;
           case 5:      exit();
           default: printf ("\nOut of Range");
    }
}
getch();
}
insert ()
{
if (fp1==NULL)
       fp1=fopen("Employee.txt","w");
else
       fp1=fopen("Employee.txt","a");
       fseek(fp1,0L,SEEK_END);
       printf ("\nEnter the Employee No.:-> ");
       flushall();
       scanf ("%d",&e.empno);
       printf("\nEnter name of Employee:-> ");
       flushall();
       gets(e.name);
       printf ("\nEnter Address of Employee:-> ");
       flushall();
       gets(e.add);
       printf ("\nEnter Salary of Employee:-> ");
       flushall();
       scanf("%f",&e.salary);
       fwrite (&e,sizeof(e),1,fp1);
       fclose(fp1);
       printf ("\nRecord Added.");
       getch();
}
display()
{
 int i=1;
```

```
 fp1=fopen("Employee.txt","r");
 if (fp1==NULL)
 {
        printf("\nUnable to open file");
        getch();
        return ;
 }
 printf("\nEmp. No\tName\tAddress\tSalary\n");
 while(1)
 {
        i=fread(&e,sizeof(e),1,fp1);
        if (i==0)break;
        printf("\n%d\t%s\t%s\t%f",e.empno,e.name,e.add,e.salary);
 }
 fclose(fp1);
 printf("\nThanks");
 getch();
}
del()
{
 char name[25];
 int flag=0,i=0,sz=0,count=0;
 fp1=fopen("Employee.txt","r");
 fp2=fopen("Temp.Tmp","w");
 if (fp2==NULL)
 {
        printf ("\nMemory Allocation not possible");
        getch();
        return ;
 }
 if (fp1==NULL)
 {
        printf("\nUnable to open file");
        getch();
        return ;
 }
 printf ("\nEnter Name of Employee:-> ");
 flushall();
 gets(name);
 i=0;
 while(1)
 {
        flag=0;
        i=fread(&e,sizeof(e),1,fp1);
        if (i==0)
               break;
        if (strcmp(e.name,name)==0)
        {
               flag=1;
               printf("\nThis is the data u want to modify\n");
               printf
("%d\t%s\t%s\t%f\n",e.empno,e.name,e.add,e.salary);
        }
        else
        {
               fwrite (&e,sizeof(e),1,fp2);
        }
 }
 fclose(fp1);
 fclose(fp2);
 rename(fp2,fp1);
 getch();
}
```

```
modify()
{
 char name[25];
 int flag=0,i=0,sz=0,count=0;
 fp1=fopen("Employee.txt","r");
 fp2=fopen("Temp.Tmp","w");
 if (fp2==NULL)
 {
        printf ("\nMemory Allocation not possible");
        getch();
        return ;
 }
 if (fp1==NULL)
 {
        printf("\nUnable to open file");
        getch();
        return ;
 }
 printf ("\nEnter Name of Employee:-> ");
 flushall();
 gets(name);
 i=0;
 while(1)
 {
        flag=0;
        i=fread(&e,sizeof(e),1,fp1);
        if (i==0)
               break;

        if (strcmp(e.name,name)==0)
        {
               flag=1;
        printf("\nThis is the data u selected for change\n");
               printf
("%d\t%s\t%s\t%f\n",e.empno,e.name,e.add,e.salary);
               printf ("\nEnter new data\n");
               printf ("\nEnter the Employee No.:-> ");
               flushall();
               scanf ("%d",&e.empno);
               printf("\nEnter name of Employee:-> ");
               flushall();
               gets(e.name);
               printf ("\nEnter Address of Employee:-> ");
               flushall();
               gets(e.add);
               printf ("\nEnter Salary of Employee:-> ");
               flushall();
               scanf("%f",&e.salary);
               fwrite(&e,sizeof(e),1,fp2);
               printf("\nRecord Modified");
        }
        else
        {
               fwrite (&e,sizeof(e),1,fp2);
        }
 }
 fclose(fp1);
 fclose(fp2);
 rename(fp2,fp1);
 getch();
}
```

Q.120 What is insertion sort? Write a program to sort the given list of integers using insertion sort. **(4)**

Ans:

Insertion Sort: One of the simplest sorting algorithms is the insertion sort. Insertion sort consists of n - 1 passes. For pass p = 2 through n, insertion sort ensures that the elements in positions 1 through p are in sorted order. Insertion sort makes use of the fact those elements in positions 1 through p - 1 are already known to be in sorted order. To insert a record, we must find the proper place where insertion is to be made.

A C program to sort integers using insertion sort is given below:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[20],i,j,n,val;
clrscr();
printf("how many numbers u want to enter :\n");
scanf("%d",&n);
printf("\nenter the numbers :\n");
for(i=0;i<n;i++)
scanf("%d",&a[i]);
for(i=1;i<n;i++)
     {
      int val=a[i];
      for(j=i-1;j>=0&&val<a[j];j--)
        a[j+1]=a[j];
      a[j+1]=val;
printf("sorted elements :\n");
for(i=0;i<n;i++)
printf("%d\t",a[i]);
printf("\n");
getch();
}
```

Q.121 Define a macro. Write a nested macro that gives the minimum of three values. **(5)**

Ans:

Macro is a preprocessor directive, also known as macro definition takes the following general form:

```
#define identifier string
```

The pre-processor directive replaces every occurrence of the identifier in the source code by the string. The preprocessor directive definition is not terminated by a semicolon. For example

`#define COUNT 100` will replace all occurrences of COUNT with 100 in the whole program before compilation.

A nested macro that gives minimum of three values is listed below:

```
#include<stdio.h>
#include<conio.h>
#define min(a,b) ((a>b)?b:a)
#define minthree(a,b,c) (min(min(a,b),c))
void main()
{
int x,y,z,w;
clrscr();
printf("enter three numbers :\n");
scanf("%d%d%d",&x,&y,&w);
```

```
z=minthree(x,y,w);
printf("Minimum of three value is %d",z);
getch();
}
```

Q.122 Write a program to check whether a box is cube, rectangle or semi-rectangle. Prepare a test procedure to test this program. **(5)**

Ans:

A C program to check if a box is cube, rectangle or semi-rectangle is:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    float length, width, height;
    clrscr();
    printf("enter the dimension of box\n");
    printf("Length=  ");
    scanf("%f",&length);
    printf("width=  ");
    scanf("%f",&width);
    printf("height=  ");
    scanf("%f",&height);
if(length==width)
{  if(length==height)
     printf("\nThe box is a cube\n");
   else
     printf("\nThe box is a rectangle\n");
}
if(width==(0.5*length))
    printf("\nThe box is a semirectangle\n");
getch();
}
```

Q.123 A program has been compiled and linked successfully. When you run this program you face one or more of the following situations

(i) Program executed, but no output
(ii) It produces incorrect answers
(iii) It does not stop running.

What are the possible causes in each case and what steps would you take to correct them? **(6)**

Ans:

A program has been compiled and linked successfully.

(i) Program executed, but no output: this usually happens due to run time errors in the program like referencing an out-of-range array element or mismatch of data types.

(ii) It produces incorrect answers: Then there may be logical errors in the program like failure to consider a particular condition or incorrect translation of the algorithm into the program or incorrect order of evaluation of statements etc.

(iii) It does not stop running: This happens when we make use of correct syntax statement but incorrect logic like `if(code=1) count++;`

Instead of using comparison operator we are using assignment which is syntactically correct so count++ is always executed resulting in infinite loop. Similar mistakes may occur in other control statements, such as for and while loop that causes infinite loops and does not stop running.

Q.124 Design an algorithm to generate n^{th} member of the Fibonacci sequence. **(8)**

Ans:
An algorithm to generate nth member of Fibonacci sequence is:

1. start
2. Scan the number 'n' upto which the series to be generated.
3. Initialize Sum=0, x=0, y=1 and i=3.
4. Print x and y as the part of series.
5. Repeat a-e until i<=n
   a. Calculate Sum=x+y
   b. Print Sum as part of series.
   c. Assign y to x
   d. Assign sum to y
   e. i=i+1
6. Stop.

Q.125 Write an algorithm to sort an arrays of integers using bubble sort technique. **(8)**

Ans:
A C algorithm to sort an array using bubble sort technique:

```
void main()
{
int a[20],i,j,n,c,flag;
printf("how many numbers u want to enter :\n");
scanf("%d",&n);
printf("\nenter the numbers :\n");
for(i=0;i<n;i++)
scanf("%d",&a[i]);
for(i=0;i<n-1;i++)
{
for(j=0;j<n-1-i;j++)
{
if(a[j]>a[j+1])
{
c=a[j];
a[j]=a[j+1];
a[j+1]=c;
flag=0;
}
}
if(flag)
 break;
else
 flag=1;
}
printf("sorted elements :\n");
for(i=0;i<n;i++)
printf("%d\t",a[i]);
printf("\n");
}
```

Q.126 In a company an employee is paid as under:
(i) HRA=10% of Basic salary and DA=55% of Basic salary.

(ii) If his Basic salary is greater than 1500 then HRA=500/- and DA=60% of Basic Salary.
Write a C program to find Gross salary of the employee. **(8)**

Ans:
A C program to find gross salary of an employee is listed below:

```
void main()
 {
 float basic,hra,da,gross;
 clrscr();
 printf("\n Enter the basic salary of the employee : Rs.");
 scanf("%f",&basic);
 if(basic>1500)
 {
        hra=500;
        da=(60.00/100.00)*basic;
 }
 else
 {
        hra=(10.00/100.00)*basic;
        da=(55.00/100.00)*basic;
 }
 gross=basic+hra+da;
 printf("\n The Gross Salary is : Rs. %f",gross);
 getch();
}
```

Q.127 What is looping? **(5)**

Ans:
Loop is a control structure used to perform repetitive operation. Some programs involve repeating a set of instruction either a specified number of times or until a particular condition is met. This is done using a loop control structure. A program loop consists of two parts: Body of the loop and control statement. The control statement tests certain conditions and then decides repeated execution or termination of statements. Most real programs contain some construct that loops within the program, performing repetitive actions on a stream of data or a region of memory.

Q.128 What are Multidimensional Arrays? Using multidimensional array, write a program in C to sort a list of names in alphabetical order. **(11)**

Ans:
Multidimensional array: Multidimensional arrays can be described as "arrays of arrays". For example, a bidimensional array can be imagined as a bidimensional table made of elements, all of them of a same uniform data type.

`int arr[3][5];` represents a bidimensional array of 3 per 5 elements of type `int`.

Similarly a three dimensional array like `int arr[3][4][2];` represent an outer array of three elements , each of which is a two dimensional array of four rows, each of which is a one dimensional array of five elements.
Multidimensional arrays are not limited to two or three indices (i.e., two dimensional or three dimensional). They can contain as many indices as needed. However the amount of memory needed for an array rapidly increases with each dimension. For example:

`char arr [100][365][24][60][60];` declaration would consume more than 3 gigabytes of memory.

Memory does not contain rows and columns, so whether it is a one dimensional array or two dimensional arrays, the array elements are stored linearly in one continuous chain. For example, the multidimensional array is stored in memory just like an one-dimensional array shown below:

```
int arr[3][4][2]= {
                    { {1,2},{3,4},{5,6},{7,8} },
                  { {9,1},{1,2},{3,7},{4,7} },
                   { {6,1},{18,19},{20,21},{22,23} },
                   };
```

1	2	3	4	5	6	7	8	9	1	1	2	3	7	4	7	6	1	18	19	20	21	22	23

101 **117** **133**

Multidimensional arrays are just an abstraction for programmers, since we can obtain the same results with a simple array just by putting a factor between its indices.

A C program to sort a list of names in alphabetical order:

```
#include<stdio.h>
#include<conio.h>
void main()
  {
int i,j,n;
char a[10][20],b[20];
clrscr();
printf("how namy names u want to enter");
scanf("%d",&n);
printf("enter names :\n");
for(i=0;i<n;i++)
scanf("%s",a[i]);
for(i=0;i<=n-2;i++)
    {
     for(j=0;j<n-i-1;j++)
     {
          if(cmp(a[j],a[j+1])>0)
          {
           cpy(b,a[j]);
           cpy(a[j],a[j+1]);
           cpy(a[j+1],b);
            }
     }
    }
    for(i=0;i<n;i++)
    printf("\n%s\n",a[i]);
    getch();
}
void cpy(char *b,char *a)
{
     int i=0;
     while(a[i]!='\0')
     {
      b[i]=a[i];
      i++;
     }
     b[i]='\0';
}
int cmp(char *a,char *b)
{
```

```
        int i=0;
        while(a[i]!='\0'||b[i]!='\0')
        {
          if(a[i]>b[i])
              break;
          if(b[i]>a[i])
              break;
          else
          {
               i++;
          }
        }
        if(a[i]>b[i])
         return(1);
        if(b[i]>a[i])
          return(-1);
}
```

Q.129 What do you understand by scope, lifetime and visibility of the variables? **(6)**

Ans:

Scope, Visibility and Lifetime of variables:

The scope of a variable determines the region of the program in which it is known. An identifier's "visibility" determines the portions of the program in which it can be referenced—its "scope." An identifier is visible only in portions of a program encompassed by its "scope," which may be limited to the file, function or block in which it appears.

File scope: The variables and functions with file scope appear outside any block or list of parameters and is accessible from any place in the translation unit after its declaration. Identifier names with file scope are often called "global" or "external." The scope of a global identifier begins at the point of its definition or declaration and terminates at the end of the translation unit. A function has file scope.

Function scope: A label is the only kind of identifier that has function scope. A label is declared implicitly by its use in a statement. Label names must be unique within a function however a label having the same name in two different functions is allowed.

Block scope: The variables with block scope appear inside a block or within the list of formal parameter declarations in a function definition. It is visible only from the point of its declaration or definition to the end of the block containing its declaration or definition.

Q.130 What is the smallest value of n such that an algorithm whose running time is $50n^3$ runs faster than an algorithm whose running time is 3^n on the same machine? Justify your answer. **(6)**

Ans:

The running time of an algorithm depends upon various characteristics and slight variation in the characteristics varies the running time. The algorithm efficiency and performance in comparison to alternate algorithm is best described by the order of growth of the running time of an algorithm. Suppose one algorithm for a problem has time complexity as c_3n^2 and another algorithm has $c_1n^3 + c_2n^2$ then it can be easily observed that the algorithm with complexity c_3n^2 will be faster than the one with complexity $c_1n^3 + c_2n^2$ for sufficiently larger values of n. Whatever be the value of c_1,

$c_{2 \text{ and }} c_3$ there will be an 'n' beyond which the algorithm with complexity c_3n^2 is faster than algorithm with complexity $c_1n^3 + c_2n^2$, we refer this n as breakeven point.
Here running time of algorithms are $50*n^3$ and 3^n, if we compare both as shown in the following table, we find that 10 is the smallest value of n (9.8) for which $50*n^3$ will run faster than 3^n.

n	**$50*n^3$**	**3^n**
2	400	9
4	3200	81
5	6250	243
9	36450	19683
10	50000	59049

Q.131 Define a sparse matrix. Explain an efficient way of storing sparse matrix in the memory of a computer. Write an algorithm to find the transpose of sparse matrix using this representation. **(10)**

Ans:
Sparse Matrix Definition: - A matrix in which number of zero entries are much higher than the number of non zero entries is called sparse matrix.
An efficient way of storing sparse matrix
The natural method of representing matrices in memory as two-dimensional arrays may not be suitable for sparse matrices. One may save space by storing only nonzero entries. For example matrix A (4*4 matrix) represented below

0	0	0	15
0	0	0	0
0	9	0	0
0	0	4	0

Here the memory required is 16 elements x 2 bytes = 32 bytes
The above matrix can be written in sparse matrix form as:

4	4	3
0	3	15
2	1	9
3	2	4

Here the memory required is 12 elements x 2 bytes = 24 bytes
where first row represent the dimension of matrix and last column tells the number of non zero values; second row onwards it is giving the position and value of non zero number.
An algorithm to find transpose of a sparse matrix is as below:

```
void transpose(x,r,y)
int x[3][3],y[3][3],r;
{
     int i,j,k,m,n,t,p,q,col;
     m=x[0][0];
      n=x[0][1];
      t=x[0][2];
   y[0][0]=n;
   y[0][1]=m;
   y[0][2]=t;
  if(t>0)
{
```

```
     q=1;
      for (col=0;col<=n;col++)
         for(p=1;p<=t;p++)
           if(x[p][1]==col)
           {
                y[q][0]=x[p][1];
     y[q][1]=x[p][0];
     y[q][2]=x[p][2];
        q++;
 }
}
return;
}
```

Q.132 What do you understand by row-major order and column-major order of arrays? Derive their formulae for calculating the address of an array element in terms of the row-number, column-number and base address of array. **(8)**

Ans:

A two dimensional array is declared similar to the way we declare a one-dimensional array except that we specify the number of elements in both dimensions. For example,

```
int grades[3][4];
```

The first bracket ([3]) tells the compiler that we are declaring 3 pointers, each pointing to an array. We are not talking about a pointer variable or pointer array. Instead, we are saying that each element of the first dimension of a two dimensional array reference a corresponding second dimension. In this example, all the arrays pointed to by the first index are of the same size. The second index can be of variable size. For example, the previous statement declares a two-dimensional array where there are 3 elements in the first dimension and 4 elements in the second dimension.

Two-dimensional array is represented in memory in following two ways:

1. **Row major representation**: To achieve this linear representation, the first row of the array is stored in the first memory locations reserved for the array, then the second row and so on.
2. **Column major representation**: Here all the elements of the column are stored next to one another.

In row major representation, the address is calculated in a two dimensional array as per the following formula

```
The address of a[i] [j] =base (a) + (i*m+ j) * size
```

where *base*(a) is the address of a[0][0], *m* is second dimension of array a and *size* represent size of the data type.

Similarly in a column major representation, the address of two dimensional array is calculated as per the following formula

```
The address of a[i] [j] = base (a) + (j*n +i) * size
```

Where *base (a)* is the address of a [0] [0], *n* is the first dimension of array a and *size* represents the size of the data type.

Q.133 Using stacks, write an algorithm to determine whether an infix expression has balanced parenthesis or not. **(8)**

Ans:

The algorithm to determine whether an infix expression has a balanced parenthesis or not.

```
Matching = TRUE
```

```
Clear the stack;
Read a symbol from input string
While not end of input string and matching do
                    {
if (symbol= '(' or symbol = '{' or symbol = '[' )
                  push (symbol, stack)
else (if symbol = ')' or symbol = '}' or symbol = ']'
)
                  if stack is empty
                        matching = FALSE
write ("more right parenthesis than left parentheses")
                  else
                  c=pop (stack)
                  match c and the input symbol
                  if not matched
                  {
                        matching = FALSE
            write ("error, mismatched parentheses")
                  }
            read the next input symbol
      }
      if stack is empty then
      write ( " parentheses are balanced properly")
      else
write ("more left parentheses then right parentheses")
```

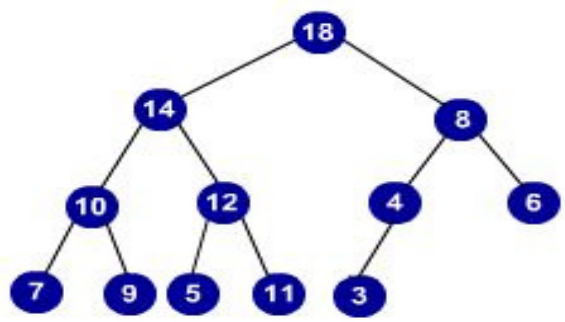
Q.134 How will you represent a max-heap sequentially? Explain with an example. Write an algorithm to insert an element to a max-heap that is represented sequentially. **(8)**

Ans:

Heap:

A binary heap(Minheap) is a complete binary tree in which each node other than root is smaller than its parent. A Maxheap is a complete binary tree in which each node other than root is bigger then its parent.

Heap example: (Maxheap)



Heap Representation:

- A Heap can be efficiently represented as an array
- The root is stored at the first place, i.e. a[1].
- The children of the node *i* are located at 2**i* and 2**i* +1.
- In other words, the parent of a node stored in *i* th location is at [i/2] floor.

- The array representation of a heap is given in the figure below.

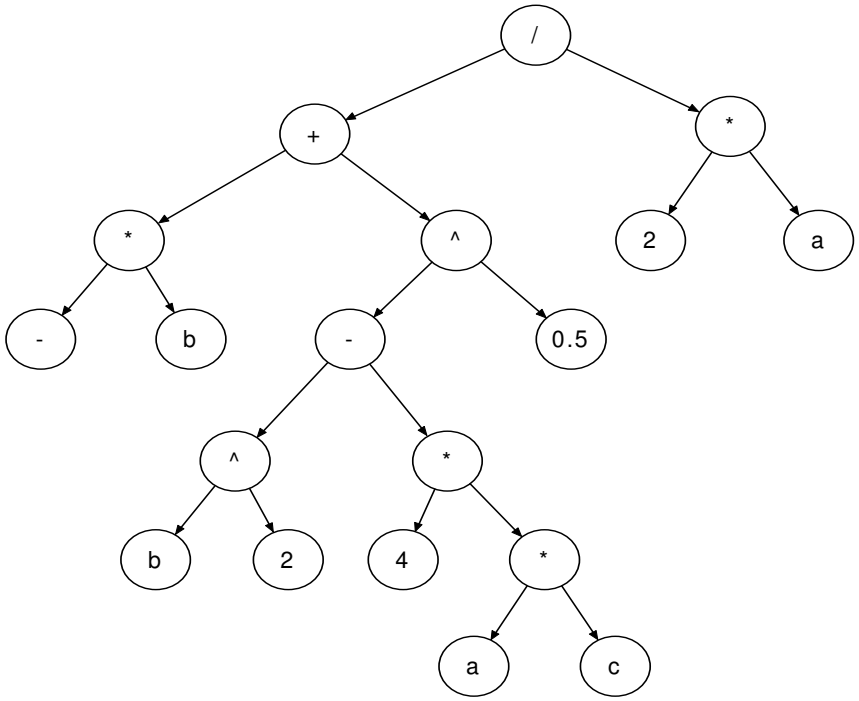
1	2	3	4	5	6	7	8	9	10	11	12
18	14	8	10	12	4	6	7	9	5	11	3

This is how max-heap is represented sequentially. An algorithm to create a heap of size i by inserting a key to a heap of size i-1 where i>=1 is as follows:
s=i;

```
/*find the   parent node of i in the array */
parent = s div 2;
key[s] = newkey;
while (s <> 0 && key [parent] <= key [s])
{
 /* interchange parent and child */
 temp = key [parent];
 key [parent] = key [s];
 key [s] = temp;
 /* advance  one level up in the tree */
 s = parent;
 parent = s div 2;
}
```

Q.135 Construct an expression tree for the expression $\left(-b+\sqrt{b^2-4ac}\right)\Big/2a$. Show all the steps and give pre-order, in-order and post-order traversals of the expression tree so formed. **(8)**

Ans :
The expression tree for the given formula is as follows



The Pre-order traversal of the tree is
/ + * - b ^ - ^ b 2 * 4 * a c 0.5 * 2 a
The Inorder traversal of the tree is
- * b + b ^ 2 - 4 * a * c ^ 0.5 / 2 * a
The Post-order traversal of the tree is
- b * b 2 * 4 a c * * / 0.5 ^ + 2 a * /

Q.136 Write an algorithm to find solution to the Towers of Hanoi problem. Explain the working of your algorithm (with 3 disks) with diagrams. **(10)**

Ans:

The aim of the tower of Hanoi problem is to move the initial n different sized disks from needle A to needle C using a temporary needle B. The rule is that no larger disk is to be placed above the smaller disk in any of the needle while moving or at any time, and only the top of the disk is to be moved at a time from any needle to any needle.

We could formulate a recursive algorithm to solve the problem of Hanoi to move n disks from A to C using B as auxiliary.

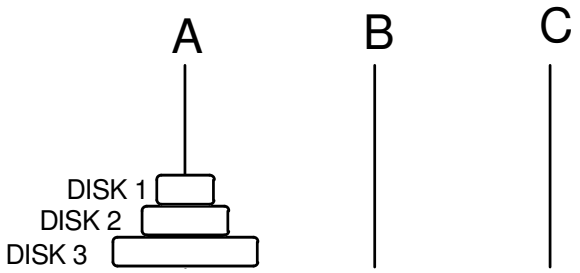
Step1: If n=1, move the single disk from A to C and return,
Step2: If n>1, move the top n-1 disks from A to B using C as temporary.
Step3: Move the remaining disk from A to C.
Step4: Move the n-1 disk disks from B to C, using A as temporary.
If we inplement this algorithm in a C language program, it would be like--

```
void hanoi ( int n, char initial, char final, char temp)
      {
            if ( n==1 )
            {
            printf (" move disk 1 from needle %c to %c\n",
intial,final);
                  return;
            }
            hanoi( n-1, initial ,temp , final);
            printf (“” nove diak %d from %c to %c “, n,
initial,final);
            hanoi(n-1, temp, final, initial);
      }
      void main()
      {
            int n;
            printf (“enter no of disks = “);
            scanf (“%d”,&n);
            hanoi ( n, ‘A’, ‘B’, ‘C’);
}
```

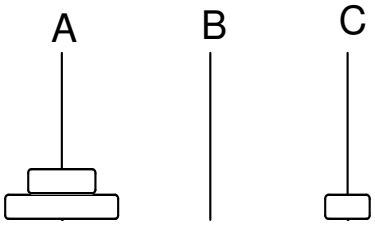
Now let us workout the output when n=3

The output will be as follows:-
Move disk 1 from needle A to needle C
Move disk 2 from needle A to needle B
Move disk 1 from needle C to needle B
Move disk 3 from needle A to needle C
Move disk 1 from needle B to needle A
Move disk 2 from needle B to needle C
Move disk 1 from needle A to needle C

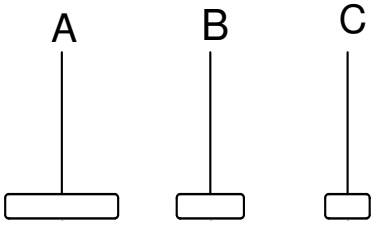
Initially the disk is at needle A



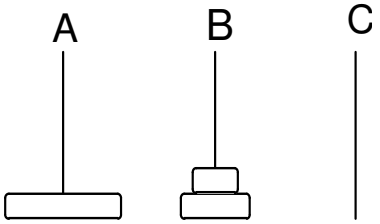
Move disk 1 from needle A to needle C



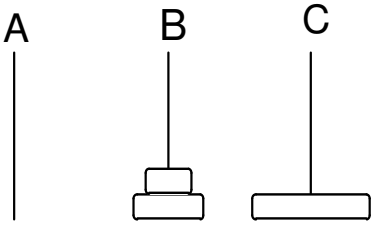
Move disk 2 from needle A to needle B



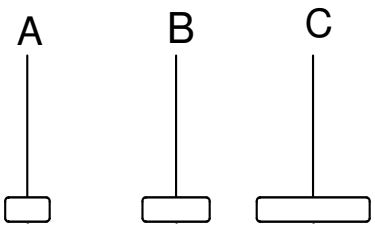
Move disk 1 from needle C to needle B



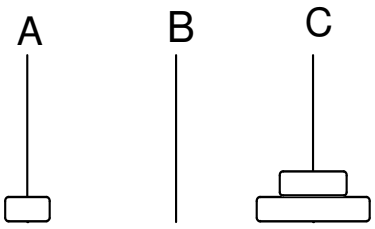
Move disk 3 from needle A to needle C



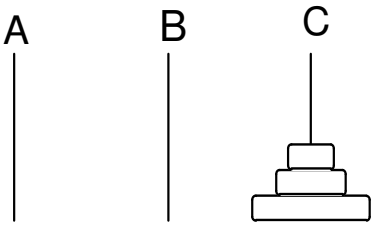
Move disk 1 from needle B to needle A



Move disk 2 from needle B to needle C



Move disk 1 from needle A to needle C



Thus finally all the three disks is in the required position after the completion of the algorithm for n=3 as shown above.

Q.137 What is recursion? A recursive procedure should have two properties. What are they? What type of recursive procedure can be converted in to iterative procedure without using stack? Explain with example. **(6)**

Ans:

Recursion: - Recursion is defined as a technique of defining a set or a process in terms of itself.

There are two important conditions that must be satisfied by any recursive procedure. They are: -

1. A smallest, **base case** that is processed without recursion and acts as decision criterion for stopping the process of computation and
2. A general method that makes a particular case to reach nearer in some sense to the base case.

For eg the problem of finding a factorial of a given number by recursion can be written as

```
Factorial (int n)
     {
int x;
if n = 0
       return (1);
x = n – 1;
return ( n* factorial (x) );
       }
```

in this case the contents of stack for n and x as [n, x] when first pop operation is carried out is as follows

[4, 3] [3, 2] [2, 1] [1, 0] [0, -] →top of stack

Here the relation is simple enough to be represented in an iterative loop, and moreover it can be done without taking the help of stack. So such kind of recursive procedures can always be converted into an iterative procedure.

The iterative solution for the above problem would be

```
Factorial (int n)
{
       int x, fact = 1;
       for ( x = 1 ;  x <= n  ;  x++ )
              fact = fact * x;
       return fact;
}
```

If there are no statements after the recursion call, that recursion can be converted to iterative programme very easily.for example:

```
   void fff (parameters)
            {
             if (condition)
            {
              step1
              step2
              step3
               .
               .
             fff (parameter)
              }
             }
```

whereas, if there are statements after a recursive call, a user stack will be required to convert them into iterative programme. for example:

```
        void fff (parameters)
              {
               if (condition)
              {
                step1
                step2
                fff (parameters)
                step3
                step4
                fff (parameter)
               }
             }
```

Q.138 Give an O(n) time non-recursive procedure that reverses a singly linked list of n nodes. The procedure should not use more than constant storage beyond that needed for the list itself. **(8)**

Ans:

The O (n) time non-recursive procedure that reverses a singly linked list of n nodes is as follows.

```
reverse (struct node **st)
{
struct node *p, *q, *r;
p = *st;
q = NULL;
while (p != NULL)
{
```

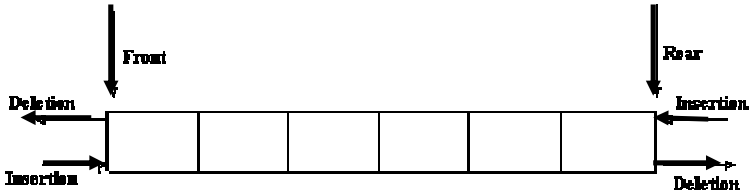
```
            r =q;
            q = p;
            p = p → link;
            q → link = r;
        }
        *st = q;
      }
```

Q.139 Device a representation for a list where insertions and deletions can be made at either end. Such a structure is called a Deque (Double Ended Queue). Write functions for inserting and deleting at either end. **(8)**

Ans :

Dequeue

A Dequeue can be represented as follows



Let the total size of elements be n in this Dequeue represented as an array
int array[n];

```
Then
front = index of the first element
Rear = index of the last element
If front = rear then we suppose that the Dequeue is empty
```

The Dequeue grows from front side till the index of the front becomes 0 and from rear side till the index of the rear equals the array size (n-1).

Now the functions for inserting at the front end of the Dequeue is:-

```
Insert_front ( int data )
{
        if ( front == 0 )
        printf ("Dequeue is full from the front end");
        else
        {
                front--;
                array[front] = data;
        }
   }
 Inserting from the rear end is:-
insert_rear ( int data )
{
        if (rear == n-1)
                printf("Dequeue is full from rear end");
        else
        {
                rear++;
                array [ rear ] = data;
        }
      }
Deleting from the front end: -
 int delete_front ()
 {
       if (front == rear)
              printf ( "the Dequeue is empty");
```

```
                else
                return ( array [ front++ ] );
 }
Deleting from the rear end:-
 int delete_rear ()
 {
        if ( front == rear )
                printf(" the Dequeue is empty ");
        else
                return ( array [ rear-- ] );
}
```

Q.140 The height of a tree is the length of the longest path in the tree from root to any leaf. Write an algorithmic function, which takes the value of a pointer to the root of the tree, then computes and prints the height of the tree and a path of that length. **(8)**

Ans

```
height(Left,Right,Root,height)
1. If Root=Null then height=0;
       return;
2. height(Left,Right,Left(Root),heightL)
3. height(Left,Right,Right(Root),heightR)
4. If heightL≥heightR then
            height=heightL+1;
      Else
            height=heightR+1;
5.  Return
```

Q.141 Two Binary trees are similar if they are both empty or if they are both non-empty and left and right subtrees are similar. Write an algorithm to determine if two binary trees are similar. **(8)**

Ans :

We assume two trees as tree1 and tree2. The algorithm two determine if two Binary trees are similar or not is as follows:

```
similar(*tree1,*tree2)
{
        while ((tree1!=null) && (tree2!=null))
        {
                if ((tree1->root) == (tree2->root))
                        similar (tree1->left,tree2->left);
                        similar (tree1->right,tree2->right);
        }
}
```

Q.142 Write an algorithm for Binary search. What are the conditions under which sequential search of a list is preferred over binary search? **(7)**

Ans :

Algorithm for Binary Search:-

Assuming that a [] is the array of items to be searched, n is the number of items in the array a, and *target* is the value of the item that is to be searched.

```
int binsearch( int target, int a[], int n)
    {
    int low=0;
    int high=1;
    int mid;
    while (low<=high)
    {
```

```
                mid=(low+high)/2;
                if (target==a[mid])
                        return (mid);
                if (target<a[mid])
                        high=mid-1;
                else
                        low=mid+1;
        }
        return (-1);
}
```

Sequential search is preferred over binary search in the following conditions:-
i). If the list is short sequential search is easy to write and efficient than binary search.
ii) If the list is unsorted then binary search cannot be used, in that case we have to use sequential search.
iii) If the list is unordered and haphazardly constructed, the linear search may be the only way to find anything in it.

Q.143 Work through Binary Search algorithm on an ordered file with the following keys {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15}. Determine the number of key comparisons made while searching for keys 2, 10 and 15. **(9)**

Ans:
The given list is 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15.
where n=14
initially low=0, high=14, mid=(low+high)/2=7

Searching for 2

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
(low) (mid) (high)

Comparison 1: 2< a[mid]
so high=mid-1=6 and mid=(0+6)/2=3

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
(low) (mid) (high)

Comparison 2: 2< a[mid]
so high=mid-1=2 and mid=(0+2)/2=1

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
(low)(mid)(high)

comparison 3: 2=a[mid]
So by the **third** comparison, we find the element in the list.

Searching for 10

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
(low) (mid) (high)

Comparison 1: 10 > a[mid]
so low=mid+1=8 and mid=(8+14)/2=11

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
(low) (mid) (high)

Comparison 2: 10< a[mid]
so high=mid-1=10 and mid=(8+10)/2=9

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
(low)(mid)(high)

comparison 3: 10=a[mid]
So by the **third** comparison, we find the element in the list.

Searching for 15

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
(low) (mid) (high)

Comparison 1: 15 > a[mid]
so low=mid+1=8 and mid=(8+14)/2=11

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
(low) (mid) (high)

Comparison 2: 15> a[mid]
so low=mid+1=12 and mid=(12+14)/2=13

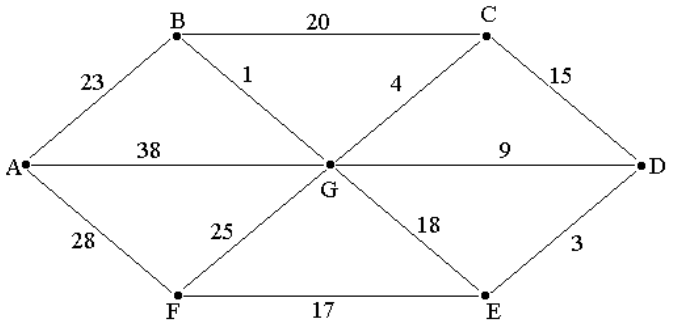
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
(low)(mid)(high)

comparison 3: 15>a[mid]
so low=mid+1=14 and mid=(14+14)/2=14

comparison 4: 15=a[mid]

So by the **fourth** comparison, we find the element in the list.

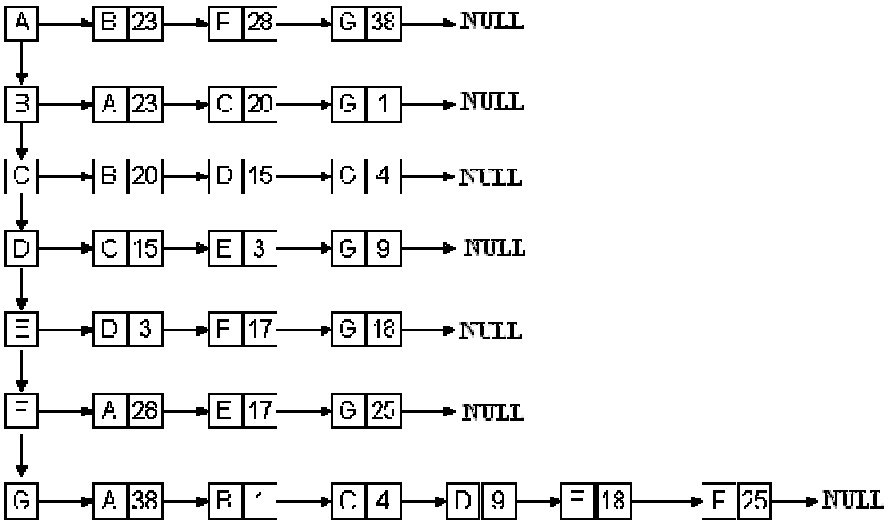
Q.144 Consider the following undirected graph:



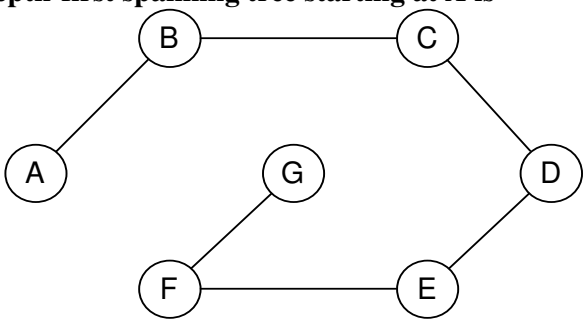
(i) Find the adjacency list representation of the graph.
(ii) Find a depth-first spanning tree starting at A.
(iii) Find a breadth-first spanning tree starting at A.
(iv) Find a minimum cost spanning tree by Kruskal's algorithm.**(4 x 3 = 12)**

Ans :

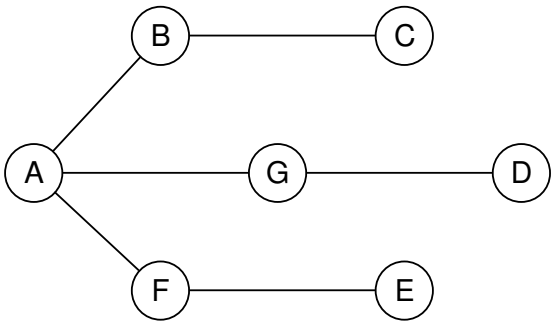
(i) The adjacency list representation of the above graph is



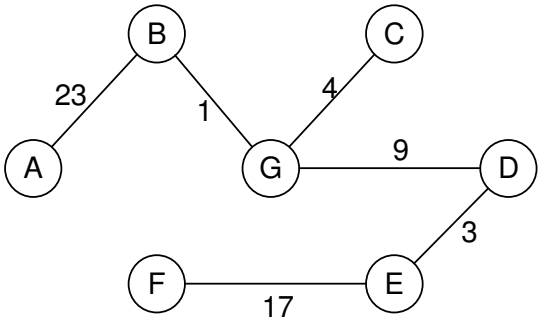
(ii) The depth-first spanning tree starting at A is



(iii) The breadth-first spanning tree starting at A is



(iv) The minimum cost spanning tree using Kruskal's Algo is:-



Q.145 Suppose that a graph has a minimum spanning tree already computed. How quickly can the minimum spanning tree be updated if a new vertex and incident edges are added to G? **(4)**

Ans : If the new vertex and the new edges are not forming a cycle on the MST, pick up the smallest edge from the set of new edges. If the new vertex and the corresponding edges are forming cycle on the MST break the cycle by removing the edge with largest weight. This will update the minimum spanning tree.

Q.146 Write code to implement a queue using arrays in which insertions and deletions can be made from either end of the structure (such a queue is called deque). Your code should be able to perform the following operations

(i) Insert element in a deque
(ii) Remove element from a deque
(iii) Check if deque is empty
(iv) Check if it is full

(v) Initialize it **(12)**

Ans : Here there are 4 operations

1. q insert front
2. q insert rear
3. q delete front
4. q delete rear

when front = rear = -1, queue is empty when (rear+1) mod size of queue = front queue is full. Here the queue is to be used circularly.

(1) q insert front

```
if (front==rear==-1)
then front=rear=0
q (front)=item
else if(rear+1)mod sizeof q=front
q is full, insertion not possible
else front = (front + sizeof q-1) mod sizeof q
```

(2) q insert rear

```
if (front==rear==-1)
front=rear=0
q(rear)=item
else if (rear+1) mod sizeof queue + front
queue is full, insertion not possible
else rear = (rear+1) mode sizeof queue
q(rear) = item
```

(3) q delete front

```
if (front==rear==-1)
queue is empty
else if (front==rear)
k = q(front), front = rear= -1
return(k)
else k=q(front)
front = (front+1) mod sizeof queue
return (k)
```

(4) q delete rear

```
if (front==rear==-1)
queue is empty
else if (front==rear)
k = q(rear), front=rear=-1
return(k)
else
k=q(rear)
rear = (rear +size of queue -1) mod sizeof q
return (k)
```

Q.147 What are stacks? List 6 different applications of stacks in a computer system. **(4)**

Ans :

A **stack** is an abstract data type in which items are added to and removed only from one end called TOP. For example, consider the pile of papers on your desk. Suppose you add papers only to the top of the pile or remove them only from the top of the pile. At any point in time, the only paper that is visible is the one on the top. This structure is called *stack*. The six various application of **stack in computer application are:**

1. Conversion of infix to postfix notation and vice versa.
2. Evaluation of arithmetic expression.

3. Problems involving recursion are solved efficiently by making use of stack. For example-Tower of Hanoi problem
4. Stack is used to maintain the return address whenever control passes from one point to another in a program.
5. To check if an expression has balanced number of parenthesis or not.
6. To reverse string.

Q.148 Write a recursive code to compute the sum of squares as shown in the series $m^2 + (m+1)^2 + \ldots + n^2$ for m, n integers $1 \le m \le n$ **(8)**

Ans :

A recursive code to compute sum of squares from m to n.

Make int n a global variable so that it is not passed in every recursive cell.

```
int  n;
void main()
{
    int m,sum;
    clrscr();
    printf (" Value of m : ");
    scanf ("%d",&m);
    printf (" Value of n : ");
    scanf ("%d",&n);
    sum=sum_recursive(m);
    printf ("\n\n\n\n%d",sum);
}
int sum_recursive(int m)
{
    int sum;
    if (m == n)
          return (n*n);
    else
          sum = (m*m) + sum_recursive(m+1);
    printf ("\t%d",sum);
    return sum;
}
```

Q.149 Give mathematical recursive definition of an AVL tree. **(4)**

Ans :

AVL tree definition:

1. An empty binary tree a is an AVL tree
2. If 'B_t' is the non-empty binary tree with 'B_{tL}' and 'B_{tR}' as its left and right-subtrees, then 'B_t' is an AVL tree if only if:

- 'B_{tL}' and 'B_{tR}' are AVL trees and
- $|h_L - h_R| \le 1$ where h_L and h_R are the heights of 'B_{tL}' and 'B_{tR}' respectively. $|h_L - h_R|$ is called balance factor of a node, its value can be {-1, 0, 1}

Q.150 Given the following recursive code. Point out the possible errors in the code.

```
//num is a positive integer
int fac(int num)
{
if (num≤1)
      return 1;
else {
      return num*fac(num+1);
        }
```

```
}
```
(4)

Ans : There is no misplaced else.Two round brackets are missing for the return statements.it should have been

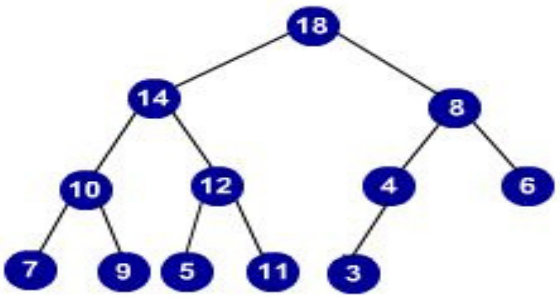
```
return (1)
return (num* fac (num-1))
```

Note :- it is not (num+1) but (num-1)

Q.151 What are the properties of a *heap*? Distinguish between a *max heap* and a *min heap*.)

Ans :

A complete binary tree, each of whose elements contains a value that is greater than or equal to the value of each of its children is called a Heap

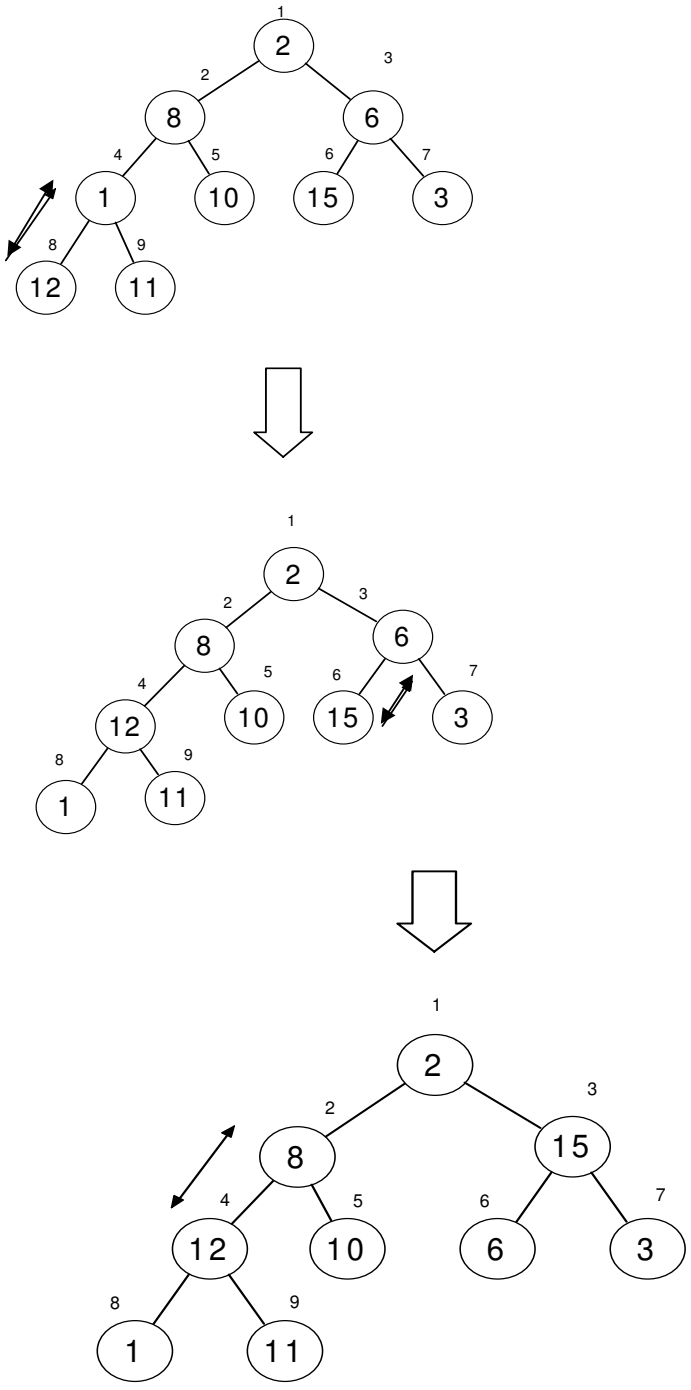


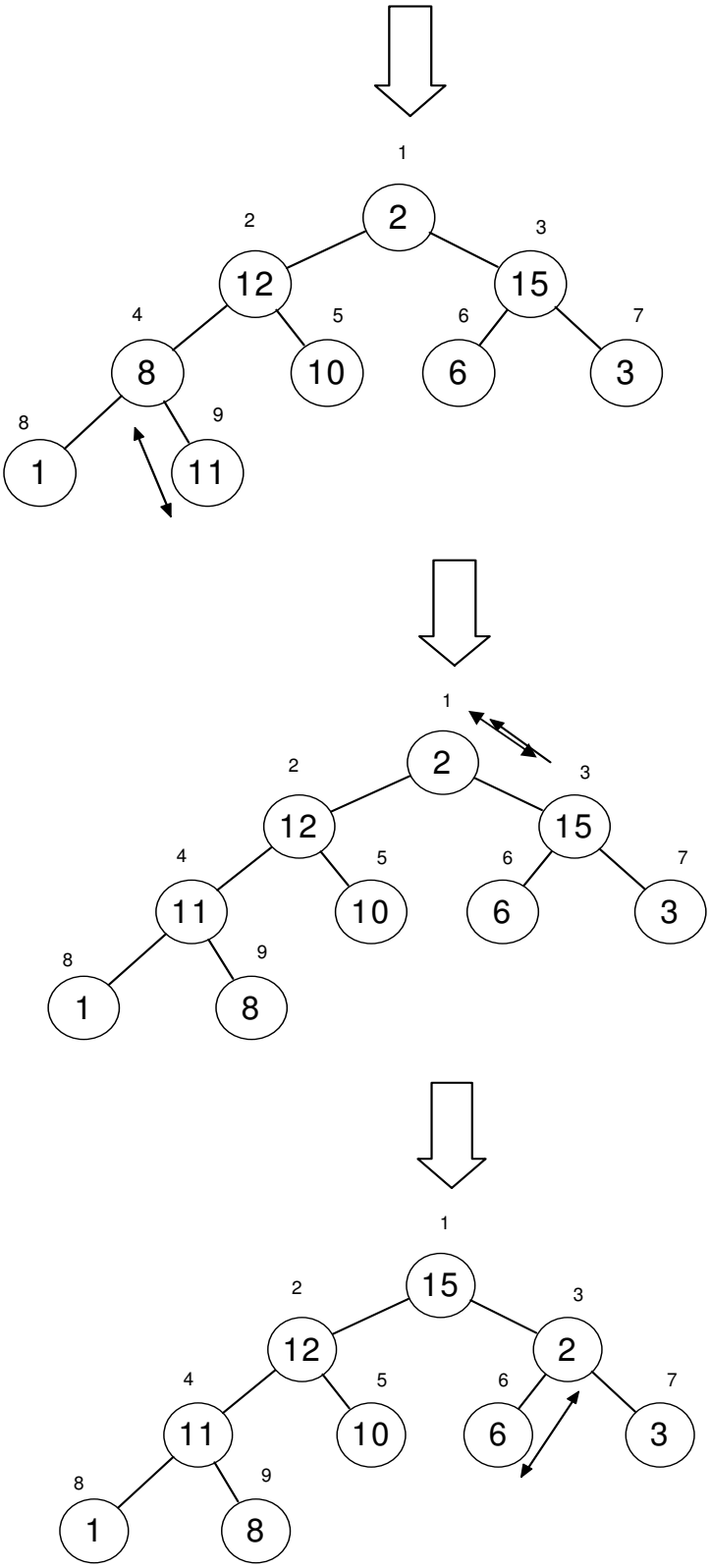
For example, above is a heap of size 12. Because of the order property of heap, the root node always contains the largest value in the heap. When we refer to such a heap, it is called a "maximum heap (*max heap*)," because the root node contains the maximum value in the structure. Similarly we can also create a "minimum heap," each of whose elements contains a value that is *less* than or equal to the value of each of its children

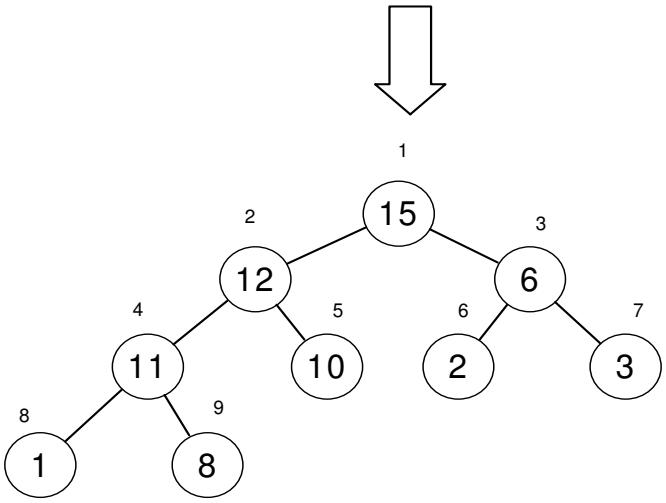
Q.152 Transform the array 2 8 6 1 10 15 3 12 11 into a heap with a bottom up method **(8)**

Ans :

Given array is 2, 8, 6, 1, 10, 15, 3, 12, 11


1
2
2
8
3
6
4
1
5
10
6
15
7
3
8
12
9
11
1
2
2
8
3
6
4
12
5
10
6
15
7
3
8
1
9
11
1
2
2
8
3
15
4
12
5
10
6
6
7
3
8
1
9
11


1
2
2
12
3
15
4
8
5
10
6
6
7
3
8
1
9
11
1
2
2
12
3
15
4
11
5
10
6
6
7
3
8
1
9
8
1
15
2
12
3
2
4
11
5
10
6
6
7
3
8
1
9
8



Final heap

Q.153 What are threaded binary trees? Explain inorder threading using an example. **(4)**

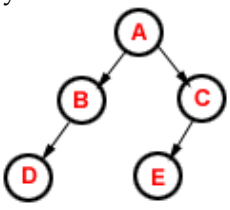
Ans :

Threaded Binary Tree: If a node in a binary tree is not having left or right child or it is a leaf node then that absence of child node is represented by the null pointers. The space occupied by these null entries can be utilized to store some kind of valuable information. One possible way to utilize this space is to have special pointer that point to nodes higher in the tree that is ancestors. These special pointers are called threads and the binary tree having such pointers is called threaded binary tree. There are many ways to thread a binary tree each of these ways either correspond either in-order or pre-order traversal of T.A Threaded Binary Tree is a binary tree in which every node that does not have a right child has a THREAD (in actual sense, a link) to its INORDER successor. By doing this threading we avoid the recursive method of traversing a Tree, which makes use of stacks and consumes a lot of memory and time.

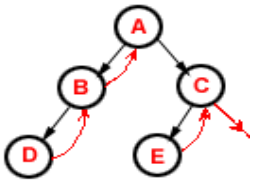
The node structure for a threaded binary tree varies a bit and its like this

```
struct NODE
{
 struct NODE *leftchild;
 int node_value;
 struct NODE *rightchild;
 struct NODE *thread;
}
```

Let's make the Threaded Binary tree out of a normal binary tree...



The INORDER traversal for the above tree is -- D B A E C. So, the respective **Threaded Binary tree will be --**



B has no right child and its inorder successor is A and so a thread has been made in between them. Similarly, for D and E. C has no right child but it has no inorder successor even, so it has a hanging thread.

Q.154 Write an algorithm to implement Depth-first search? How is Depth-first search different from Breadth-first search? **(10)**

Ans :

Depth First Search Algorithm

An algorithm that does depth first search is given below:

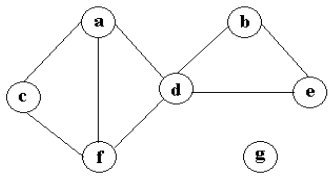
```
struct node
{
  int data ;
  struct node *next ;
} ;
int visited[MAX] ;
void dfs ( int v, struct node **p )
{
  struct node *q ;
  visited[v - 1] = TRUE ;
  printf ( "%d\t", v ) ;
  q = * ( p + v - 1 ) ;
  while ( q != NULL )
  {
        if ( visited[q -> data - 1] == FALSE )
              dfs ( q -> data, p ) ;
        else
              q = q -> next ;
  }
}
```

Depth-first search is different from Breadth-first search in the following ways:

A depth search traversal technique goes to the deepest level of the tree first and then works up while a breadth-first search looks at all possible paths at the same depth before it goes to a deeper level. When we come to a dead end in a depth-first search, we back up as *little* as possible. We try another route from a recent vertex-the route on top of our stack. In a breadth-first search, we want to back up as *far* as possible to find a route originating from the earliest vertices. So the stack is not an appropriate structure for finding an early route because it keeps track of things in the order opposite of their occurrence-the latest route is on top. To keep track of things in the order in which they happened, we use a FIFO queue. The route at the front of the queue is a route from an earlier vertex; the route at the back of the queue is from a later vertex.

Q.155 Represent the following graph as

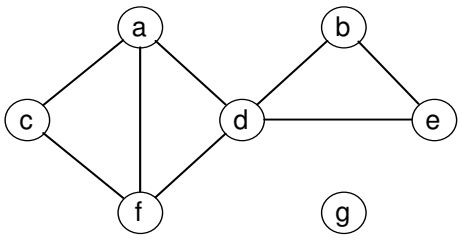
(i) Adjacency list
(ii) Adjacency matrix
(iii) Incidence matrix



(6)

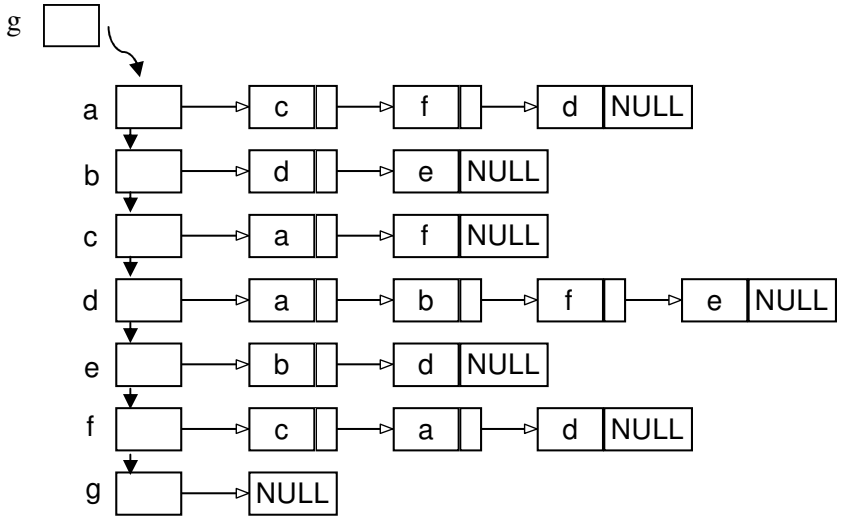
Ans :

The given graph is –



The representation of the above graph is as follows: -

(i) Adjacency List : -



(ii) Adjacency Matrix : -

	a	b	c	d	e	f	g
a			1	1		1	
b				1	1		
c	1					1	
d	1	1			1	1	
e		1		1			
f	1		1	1			
g							

(iii) **Incidence Matrix** : - This is the incidence matrix for an undirected group. For directed graphs, the vertex from where an edge is originating will have +1 and the vertex where the edge is reaching will have a value -1.

	C1	C2	C3	C4	C5	C6	C7	C8
a	1	1	1					
b				1	1			
c	1					1		
d		1		1			1	1
e					1		1	
f			1			1		1
g								

Rest of the entries are zero.

Q.156 Define B-tree of order m? When is it preferred to use B-trees? **(4)**

Ans :

A B-Tree of order M is either the empty tree or it is an M-way search tree T with the following properties:

(i) The root of T has at least two subtrees and at most M subtrees.

(ii) All internal nodes of T (other than its root) have between [M / 2] and M subtrees.

(iii)All external nodes of T are at the same level.

Just as AVL trees are balanced binary search trees, B-trees are balanced M-way search trees. By imposing a balance condition, the shape of an AVL tree is constrained in a way which guarantees that the search, insertion, and withdrawal operations are all O(log n), where n is the number of items in the tree. The shapes of B-Trees are constrained for the same reasons and with the same effect.

Q.157 Write an algorithm to search a key in a B-tree? What is the worst case of searching in a B-tree? List the possible situations that can occur while inserting a key in a B-tree? **(8)**

Ans :

B-tree search makes an n-way choice. By performing the linear search in a node, correct child Ci of that node is chosen. Once the value is greater than or equal to the desired value is obtained, the child pointer to the immediate left of the value is followed. Otherwise rightmost child pointer is followed. If desired value is obtained, the search is terminated.

B-tree search (x, k) /*This function searches the key from the given B-tree*/

The Disk_Read operation indicates that all references to a given node be preceded by a read operation.

1. Set i ← 1
2. While ($i < n[x]$ and $k > key_i [x]$)
   Set i = i + 1
3. If ($i <= n[x]$ and $k = key_i [x]$) then
   { return (x, i) }
4. If (leaf [x]) then
   return NIL
   else Disk_read ($c_i [x]$)
   return B-tree search (ci [x], k)

The worst case of searching is when a B-tree has the smallest allowable number of pointers per non-root made, q=[m/2], the search has to reach a leaf (either for a successful or an unsuccessful search)

There are three common similarities that are encountered when inserting a key in a B-tree.

1) A key is placed in a leaf that still has some room.
2) The leaf in which the key should be placed is full. In this case, the leaf is split, creating a new leaf and half of the keys are moved from the full leaf to the new leaf. But the new leaf has to be incorporated in to the B-tree. The last key of the old leaf is moved to the parent and a pointer to the new leaf is placed in the parent as well. The same procedure can be repeated for each internal node of the B-tree. Such a split ensures that each leaf never has less than [m/2]-1 keys.
3) A special case arises if the root of the B-tree is full. In this case, a new root and a new sibling of the existing root has to be created. This split results in two new nodes in the B-tree.

Q.158 What is the complexity of the following code?

```
int counter = 0;
for (i=0; i<n; i++)
        for (j=0; j<n*n; j++)
                counter++;
```

(4)

Ans :

The complexity of given code is $O(n^2)$.

Q.159 Show under what order of input, the insertion sort will have worst-case and best-case situations for sorting the set {142, 543, 123, 65, 453, 879, 572, 434}. **(8)**

Ans :

The given list is

142 543 123 65 453 879 572

As we know that insertion sort is based on the idea of inserting records into an existing sorted file. To insert a record, we must find the proper place for the record to be inserted and this requires searching the list.

An insertion sort will have the worst-case when the list is reversely sorted i.e. in a decreasing order. So for the above given list the insertion sort exhibits its **worse case** when the list is in following order

879 572 543 453 434 142 123 65

Again since it is the property of the insertion sort that it searches for the correct position in the list for an element, so this technique exhibits its **best case** computing when the list is already sorted in ascending order as given below.

65 123 142 434 453 543 572 879

Q.160 Explain how Merge Sort sorts the following sequence of numbers using a diagram {142, 543, 123, 65, 453, 879, 572, 434}. **(8)**

Ans :

Sorting using Merge Sort

The given sequence of numbers is:-

Original Sequence [434]	[142]	[543]	[123]	[65]	[453]	[879]	[572]
Pass one [572	[142	543]	[65	123]	[453	879]	[434
Pass two 879]	[65	123	142	543]	[434	453	572
Pass three [879	[65	123	142	434	453	543	572

This number sequence is sorted in three passes using merge sort.

Q.161 Given a stack *s* and a queue *q*, show the contents of each after the indicated operations. The starting contents of *s* and *q* are shown. If an operation would result in an error, write "error" and assume the contents of *s* and *q* do not change. If there is no change otherwise, leave the cell blank or use ditto marks ("). **(8)**

Operation	**Contents of stacks Top bottom**	**Contents of queue q front rear**
Start	*empty*	2, 4
s.pop()		
s.push(3)		
q.add(5)		
s.push(q.peek())		
q.add(s.peek())		
q.add(s.pop())		
s.push(s.pop())		
q.add(q.remove())		

Ans :

pop(): Removing an element from top of the satck
push(element): inserting element on the top of stack
add(element): inserting element from rear of the queue
remove(): removing element from front of the queue
peek():reading from the front of the queue.

We have left the cells blank where there is no change from the previous state.

Operation	Contents of Stack Top bottom	Content of queue q Front rear
start	empty	2,4
s.pop()	error	-do-
s.push(3)	3	-do-
q.add(5)	-do-	2,4,5
s.push(q.peek())	2,3	-do-
q.add(s.peek())	-do-	2,4,5,2
q.add(s.pop())	3	2,4,5,2,2
s.push(s.pop())	-do-	-do-
q.add(q.remove())	-do-	4,5,2,2,2

Q.162 If x is a pointer to a node in a doubly linked list, then x-> prev is the pointer to the node before it and x->next is the pointer to the node after it.
What does this pair of statements, executed in order, do?

```
(x->next)->prev = x->prev;
(x->prev)->next = x->next;
```

If we then execute this pair of statements (after executing the first pair of statements), what happens?

```
(x->next)->prev = x;
(x->prev)->next = x;
```

(4)

Ans :

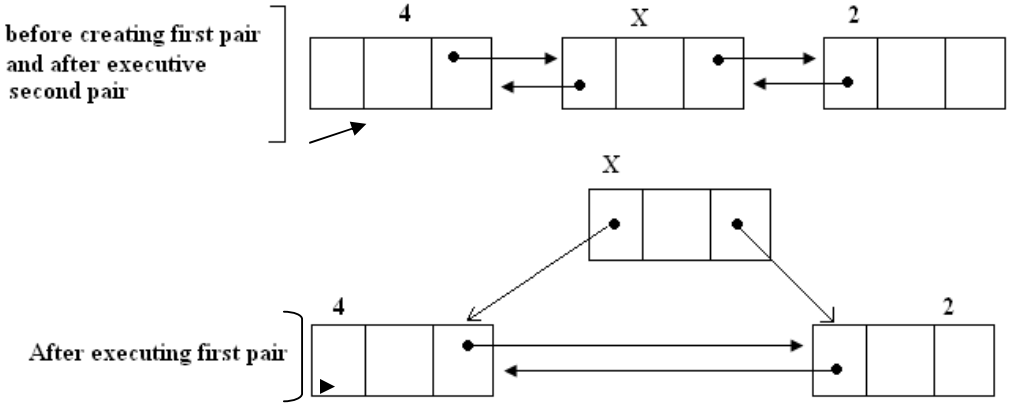
In the above given condition, if the following

```
(x->next)->prev=x->prev;
(x->prev)-next=x->next;
```

statements are executed then the resulting linked list will no longer have x as one of its elements i.e. the element x gets deleted from the doubly linked list.
Now, after executing the above pair of statements; if we execute the following pair i.e.-

```
(x->next)->prev=x;
(x->prev)->next=x;
```

then this causes the element x to become a part of the linked list again(i.e. it gets inserted again) and at the same position as before.



Q.163 Consider an insertion of the key = 222 into the hash table shown in the figure. Calculate and indicate the sequence of table entries that would be probed and the final location of the insertion for the key for linear probing and quadratic probing method. The hash function is h(y) = y mod 10, so for the key = 222 , h(x) = 2 mod 10. In both cases start with the same initial table. Example calculation given at index 2 **(8)**

0		
1		
2	152	h(222) = 2 mod 10
3	53	
4		
5	75	
6	136	
7	27	
8		
9	999	

Ans :

Linear Probing:

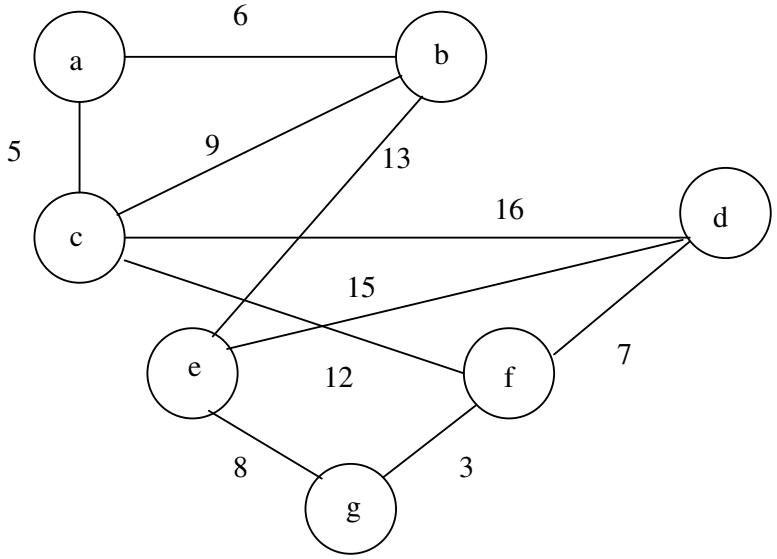
i. Initially all locations marked 'empty'
ii. When new items are stored, filled locations marked 'full'
iii. In case of collisions, newcomers are stored at the next available location, found via probing by incrementing the pointer (mod n) until either an empty location is found or starting point is reached.
So in the given case we have 222 mod 10=2, index 2 is probed. Since it is already filled so newcomer will go to the next available location that is index 4.

Rehashing:

i. The collision key is move to the considerable distance from the initial collision when the hashing function compute the value then the quadratic probing work as follows
(Hash value+1^2) mod table size
ii. If there is a collision, take the second hash function as
(Hash value+2^2) mod table size and so on.
iii. The probability that two key values will map to the same address with two different hash functions is very low.

So in the given case, we see $222+1^2$ mod 10=3 there is collision; Use next hash function.
$222+2^2$ mod 10=222+4=226 mod 10=6 , there is collision
$222+3^2$ mod 10=222+9=231 mod 10=1 so newcomer can be inserted at index 1.

Q.164 Using Dijkstra's method find a spanning tree of the following graph. **(8)**



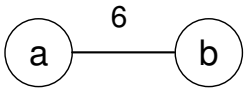
Ans :

Disjkstra algorithm Method is never used for finding Minimum-spanning tree
Minimum-spanning tree

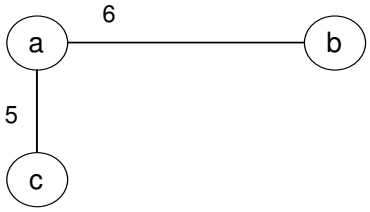
tree=null;
edges=an unsorted sequence of all edges of graph;
for j=1 to $|E|$
add e_j to tree;
If there is cycle in tree

Remove an edge with maximum weight from this only cycle.
The spanning tree for given graph is given below:

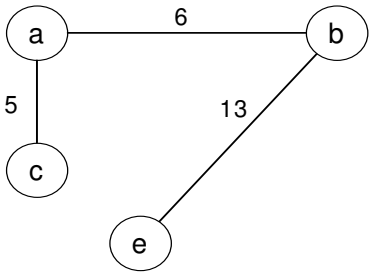
Step 1



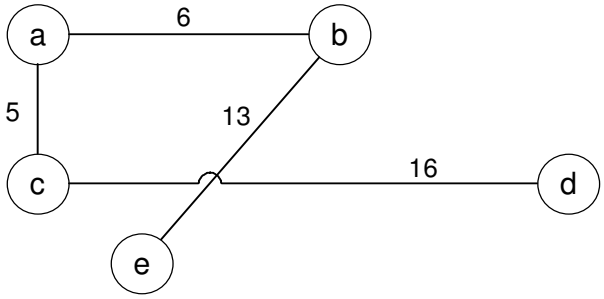
Step 2



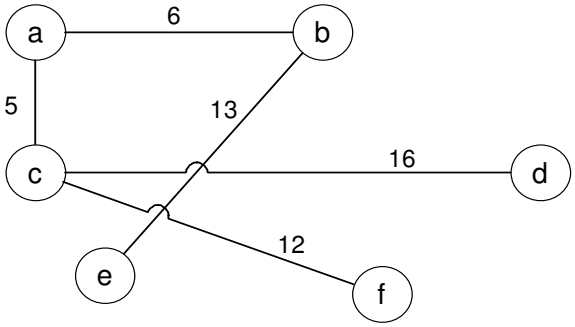
Step 3



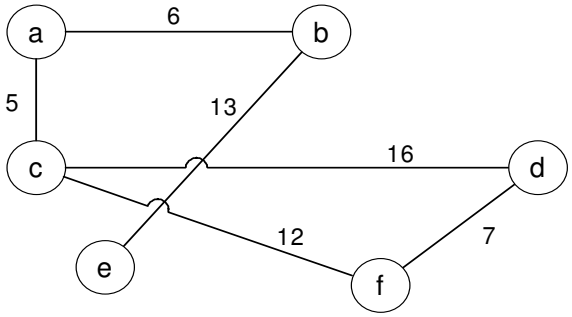
Step 4



Step 5

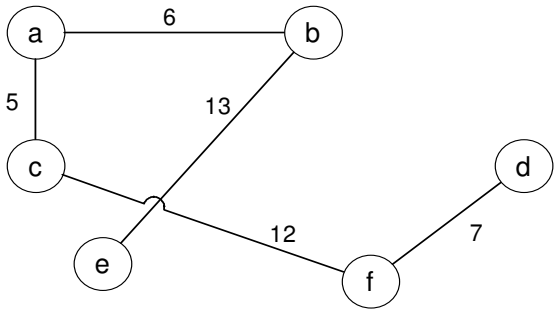


Step 6

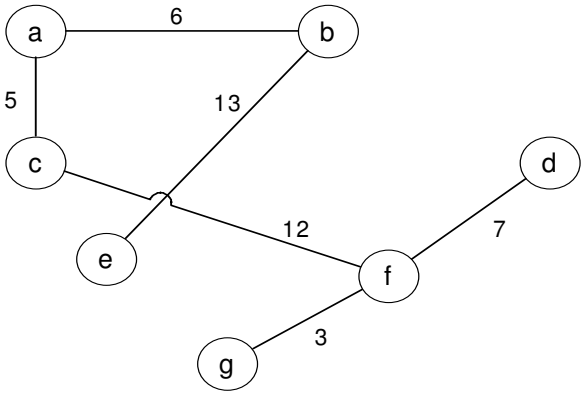


Since the above representation creates a cycle so we delete the edge having the highest weight in the cycle and the new representation would be as follows: -

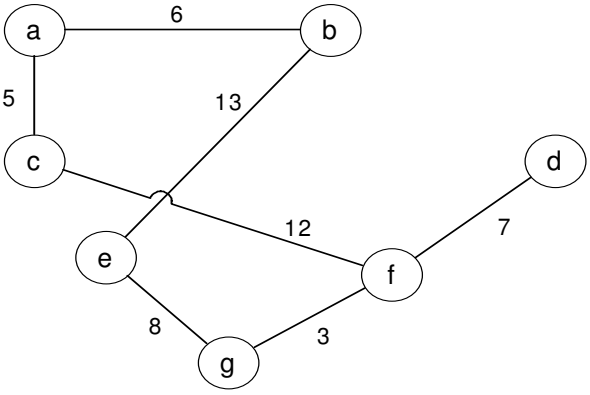
Step 7



Step 8

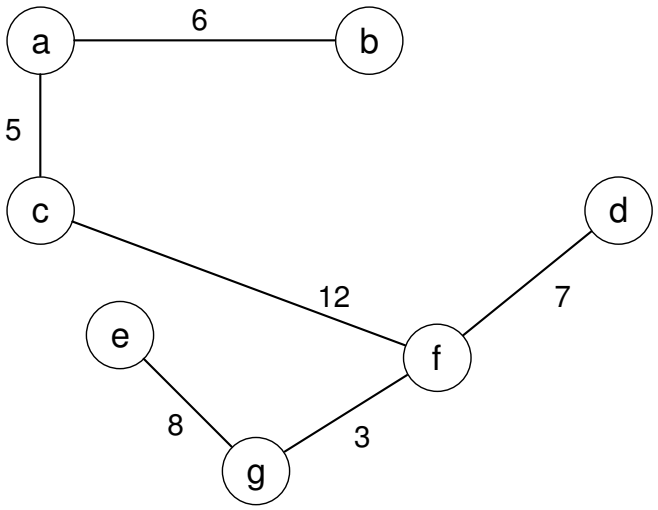


Step 9



The above representation creates cycle so we'll delete the edge having the highest weight as shown below.

Step 10



Q.165 What are the three different ways of representing a polynomial using arrays? Represent the following polynomials using any three different methods and compare their advantages and disadvantages.

(i) $7x^5 - 8x^4 + 5x^3 + x^2 + 2x + 15$

(ii) $3x^{100} - 5x^{55} - 10$ **(8)**

Ans :

A general polynomial A(x) can be written as $a_nx^n+a_{n-1}x^{n-1}+\ldots+a_1x+a_0$ where $a_n \neq 0$ then we can represent A(x)

1. As an ordered list of coefficients using a one dimensional array of length n+2, A=(n, a_n, a_{n-1},…,a_1, a_0) , the first element is the degree of A followed by n+1 coefficients in order of decreasing exponent. So the given polynomial can be represented as

i. (5,7, -8,5,1,2,15)

ii. (100,3,0,0,0,0…(45 times), -5, 0,0,0…(55 times),-10)

Advantage of this method is simple algorithms for addition and multiplication as we have avoided the need to explicitly store the exponent of each term. However there is a major disadvantage of this method and that is wastage of memory for certain polynomial like example (ii) because we are storing more zero values than non-zero values.

2. As an ordered list where we keep only non-zero coefficient along with their exponent like $(m,e_{m-1},b_{m-1},e_{m-2},b_{m-2},\ldots,e_0,b_0)$ where first entry is the number of nonzero terms, then for each term there are two entries representing an exponent-coefficient pair. So the given polynomial can be represented as

i. (6,5,7,4,-8,3,5,2,1,1,2,0,15)

ii. (3,100,3,55,-5,0,-10)

This method is solves our problem of wasted memory like what happened in example 2.

Q.166 Write an algorithm to evaluate the following polynomial. The number of additions and multiplications taken should be 5.

$9x^5 - 10x^4 + 6x^3 + 5x^2 - 10x + 15$ **(8)**

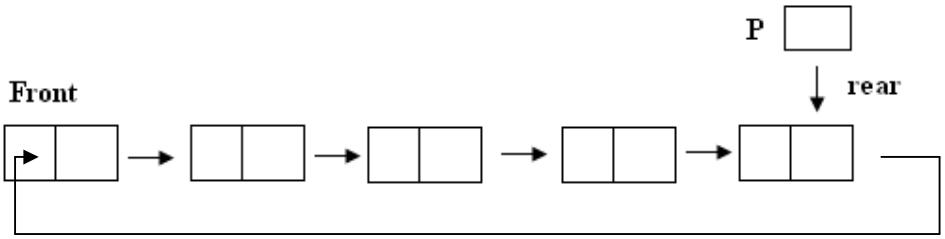
Ans :

The given polynomial can be evaluated using Hormer's rule as

$9x^5-10x^4+6x^3+5x^2-10x+15$
$\equiv x^4[9x-10]+6x^3+5x^2-10x+15$
$\equiv x^3[x[9x-10]+6x]+5x^2-10x+15$
$\equiv x^2[x[x[9x-10]+6]+5]-10x+15$
$\equiv x[x[x[9x-10]+6]+5]-10x+15$
$\equiv x[x[x[x][9x-10]+6]+5]-10]+15$
$\equiv 0[5]$

Q.167 Let P be a pointer to a circularly linked list. Show how this list may be used as a queue. That is, write algorithms to add and delete elements. Specify the value for P when the queue is empty. **(10)**

Ans : External pointer should always point to the last node of the circularly linked list. Then both the insertion and deletion from the queue can be done in O(1) time.



To add a new node x (insert a new element in the queue)

```
x →     next = P  →   next
P →     next = x
P=x
```

To delete a node (Deleting front element from queue)

```
x = P  →  next
P →    next = x →    next
return (x)
```

Q.168 Give an algorithm for a singly linked list, which reverses the direction of the links. **(6)**

Ans :

The algorithm to reverse the direction of a singly link list is as follows:

```
reverse (struct node **st)
{
      struct node *p, *q, *r;
      p = *st;
      q = NULL;
      while (p != NULL)
      {
        r =q;
        q = p;
        p = p → link;
        q → link = r;
      }
      *st = q;
}
```

Q.169 Suppose that we have numbers between 1 and 1000 in a Binary Search Tree and want to search for the number 363. Which of the following sequences could not be the sequence of nodes examined? Explain your answer.

(i) 2, 252, 401, 398, 330, 344, 397, 363
(ii) 924, 220, 911, 244, 898, 258, 362, 363
(iii) 925, 202, 911, 240, 912, 245, 363
(iv) 2, 399, 387, 219, 266, 382, 381, 278, 363
(v) 935, 278, 347, 621, 299, 392, 358, 363 **(10)**

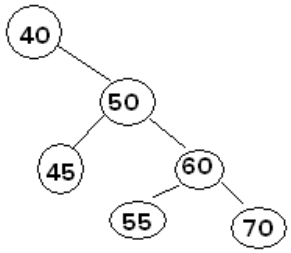
Ans :

(i) This is a valid representation of sequence.
(ii) This is also a valid representation of sequence.
(iii)This could not be the sequence since 240 is encountered after 911 so it must be its left child, and any other node henceforth must be less than 911(parent). But the node 912 breaks this sequence, which is greater than 911 and lies on the left subtree of 911, which violates the basic rule of a Binary Search Tree.
(iv)This is also a valid representation of sequence.
(v)This could not be a valid sequence since 621 is encountered after 347 so 621 must be its right child, and any other node henceforth must be greater than 347(parent). But this sequence is broken by the node 299 < 347 and lies on the right subtree of 347 which violates the basic rule of a Binary Search Tree.

Q.170 Professor Banyan thinks he has discovered a remarkable property of binary search trees. Suppose that the search for key k in a BST ends up in leaf. Consider three sets (1) set A, the keys to the left of the search path, (2) set B the keys on the search path and (3) set C, the keys on the right of the search path. Professor Banyan claims that any three keys $a \in A$, $b \in B$ and $c \in C$ must satisy $a \le b \le c$. Is his claim true? Otherwise give a counter example to the professor's claim. **(3)**

Ans :

Professor Banyan claim is completely wrong.
The claim of professor Banyan is wrong. Counter example is given below.

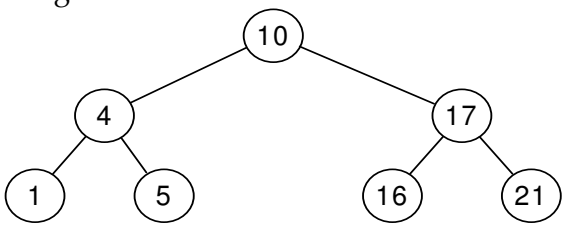


Let the key to be searched is 70 then:
Set A = {45, 55}, B = {40, 50, 60, 70} and C = { }
Here is very clear that $45 \nleq 40$ OR $55 \nleq 50$

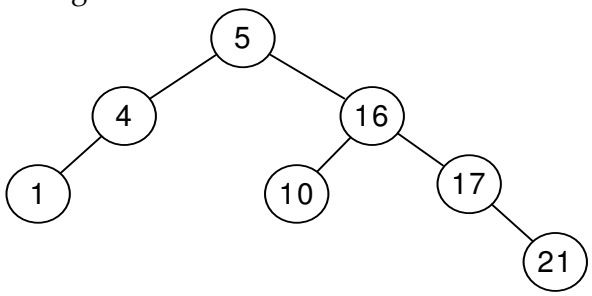
Q.171 Draw BSTs of height 2 and 3 on the set of keys {1, 4, 5, 10, 16, 17, 21}. **(3)**

Ans :

The BST of height 2 is:



The BST of height 3 is:



Q.172 Why don't we allow a minimum degree of t=1 for a B-tree? **(2)**

Ans :

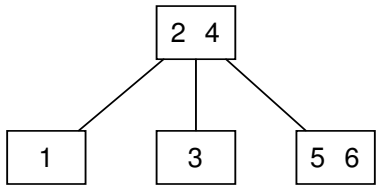
According to the definition of B-Tree, a B-Tree of order n means that each node in the tree has a maximum of n-1 keys and n subtrees. Thus a B-Tree of order 1 will have maximum 0 keys, i.e. no node can have any keys so such a tree is not possible and hence we can't allow a minimum degree of t=1 for a B-Tree.

Q.173 Show all legal B-Trees of minimum degree 3 that represent {1, 2, 3, 4, 5, 6}. **(4)**

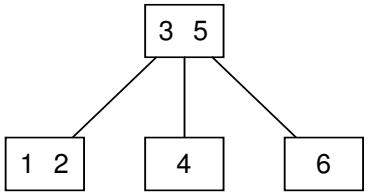
Ans :

B-Trees of minimum degree 3 are as follows:

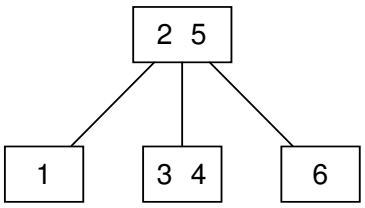
[1]



[2]



[3]

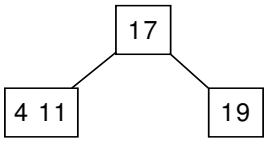


Q.174 Show the result of inserting the keys 4,19, 17, 11, 3, 12, 8, 20, 22, 23, 13, 18, 14, 16, 1, 2, 24, 25, 26, 5 in order in to an empty B-Tree of degree 3. Only draw the configurations of the tree just before some node must split, and also draw the final configuration. **(10)**

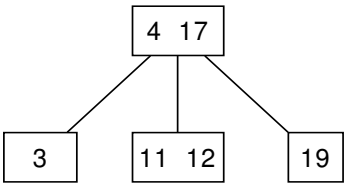
Ans :

4, 19

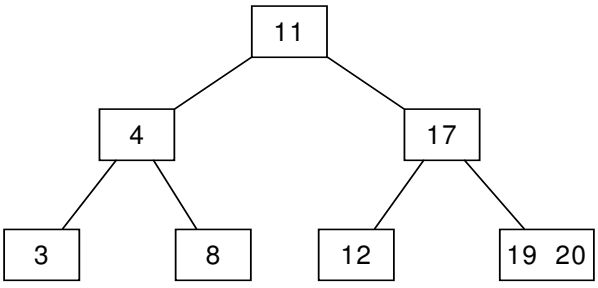
17, 11



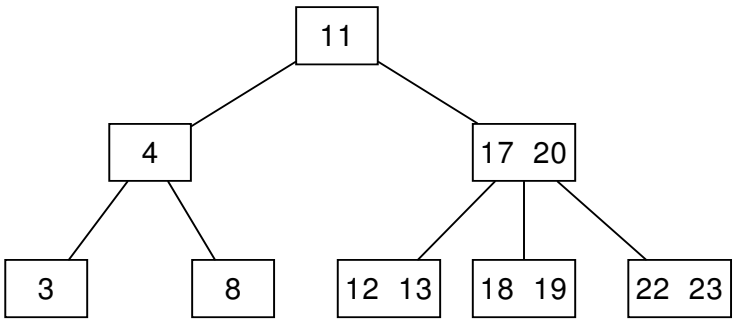
3, 12



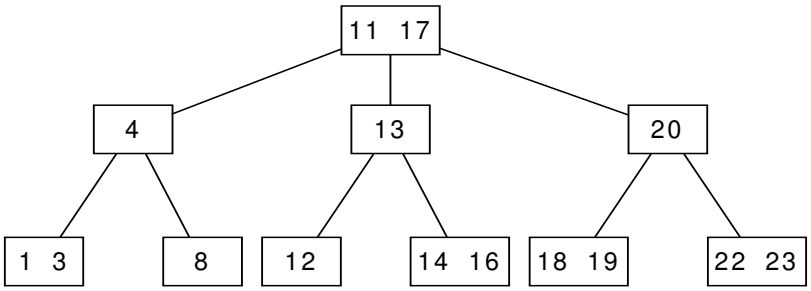
8, 20



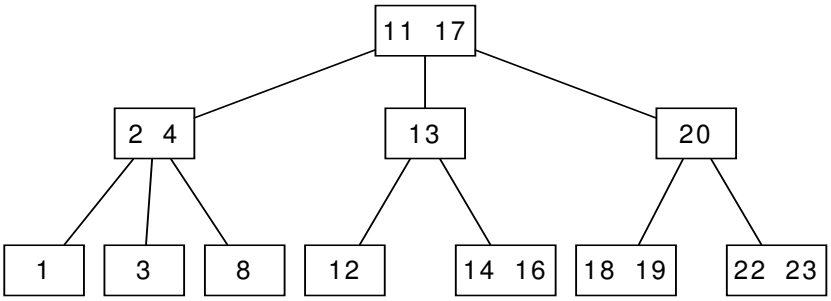
22, 23, 13, 18



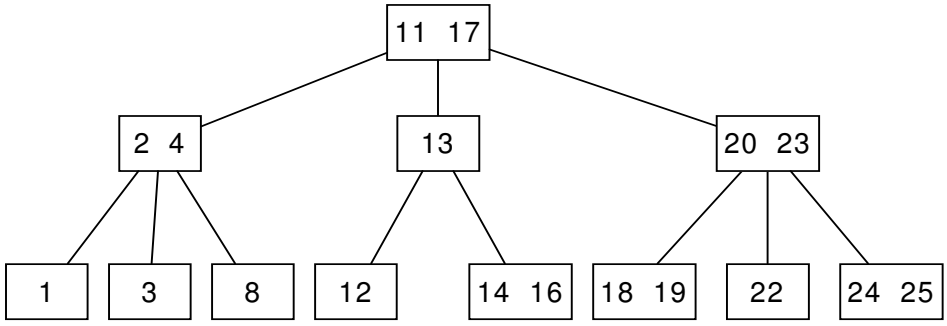
14, 16, 1



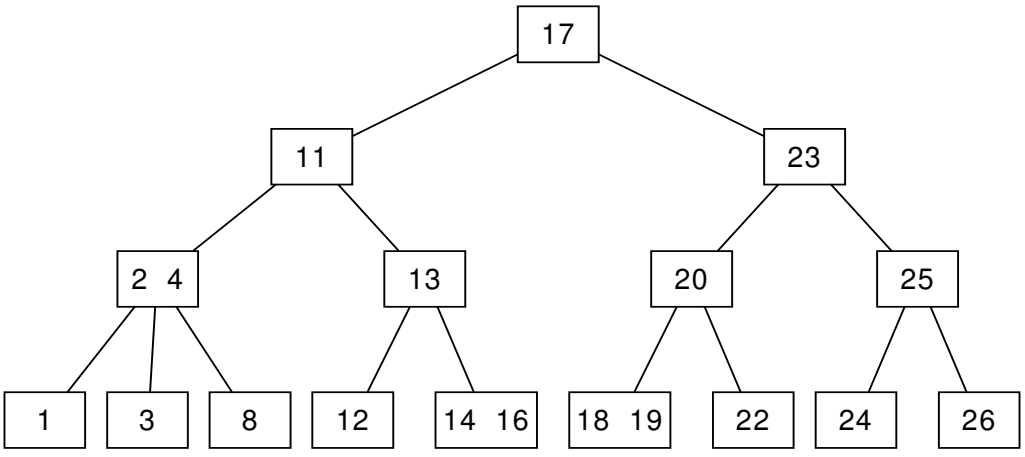
2



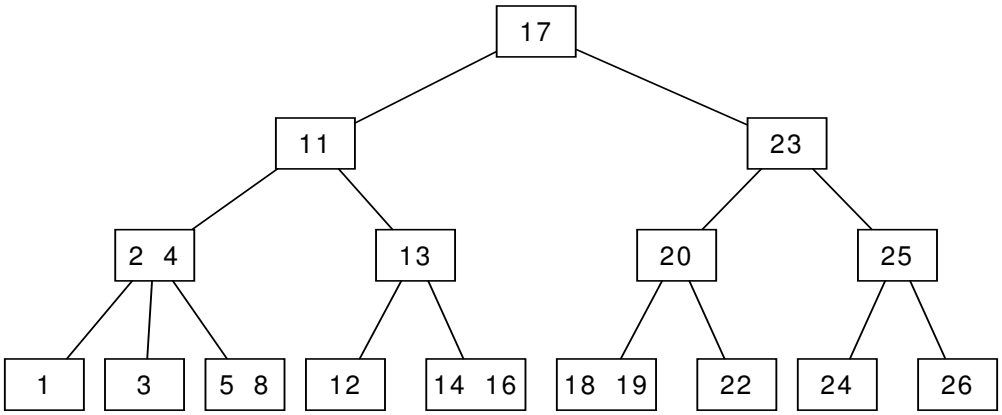
24, 25



26



Finally 5 is inserted



The above is the final tree formed after inserting all the elements given in the question.

Q.175 Write down the algorithm for quick sort.**(8)**

Ans :

The algorithm for quick sort is as follows:

```
void quicksort ( int a[ ], int lower, int upper )
{
  int i ;
  if ( upper > lower )
  {
        i = split ( a, lower, upper ) ;
        quicksort ( a, lower, i - 1 ) ;
        quicksort ( a, i + 1, upper ) ;
  }
}
int split ( int a[ ], int lower, int upper )
{
  int i, p, q, t ;
  p = lower + 1 ;
  q = upper ;
  i = a[lower] ;
  while ( q >= p )
  {
        while ( a[p] < i )
              p++ ;
        while ( a[q] > i )
              q-- ;
        if ( q > p )
        {
              t = a[p] ;
              a[p] = a[q] ;
              a[q] = t ;
        }
  }
  t = a[lower] ;
  a[lower] = a[q] ;
  a[q] = t ;
  return q ;
}
```

Q.176 Show how quick sort sorts the following sequences of keys:
5, 5, 8, 3, 4, 3, 2 **(7)**

Ans :

The given data element to be sorted using quick sort is

5 5 8 3 4 3 2

Choosing pivot 5 (Pass1)

(2) 5 8 3 4 3 (5)

2 5 (5) 3 4 3 (8)

2 5 (3) 3 4 (5) 8

choosing 2 as pivot (pass2)

(2) 5 3 3 4 (5 8)

choosing 5 as pivot (pass 3)

(2) (3 3 4 5) (5 8)

Pass (4), (5) and (6) will not change the sub list

Q.177 On which input data does the algorithm quick sort exhibit its worst-case behaviour? **(1)**

Ans :

The Quick Sort technique exhibits its worst-case behavior when the input data is "Already Completely Sorted".

Q.178 What do you mean by hashing? Explain any five popular hash functions. **(5)**

Ans :

Hashing

Hashing provides the direct access of record from the file no matter where the record is in the file. This is possible with the help of a hashing function H which map the key with the corresponding key address or location. It provides the key-to-address transformation.

Five popular hashing functions are as follows:-

Division Method: An integer key x is divided by the table size m and the remainder is taken as the hash value. It can be defined as

H(x)=x%m+1

For example, x=42 and m=13, H(42)=45%13+1=3+1=4

Midsquare Method: A key is multiplied by itself and the hash value is obtained by selecting an appropriate number of digits from the middle of the square. The same positions in the square must be used for all keys. For example if the key is 12345, square of this key is value 152399025. If 2 digit addresses is required then position 4^{th} and 5^{th} can be chosen, giving address 39.

Folding Method: A key is broken into several parts. Each part has the same length as that of the required address except the last part. The parts are added together, ignoring the last carry, we obtain the hash address for key K.

Multiplicative method: In this method a real number c such that 0<c<1 is selected. For a nonnegative integral key x, the hash function is defined as

H(x)=[m(cx%1)]+1

Here,cx%1 is the fractional part of cx and [] denotes the greatest integer less than or equal to its contents.

Digit Analysis: This method forms addresses by selecting and shifting digits of the original key. For a given key set, the same positions in the key and same rearrangement pattern must be used. For example, a key 7654321 is transformed to

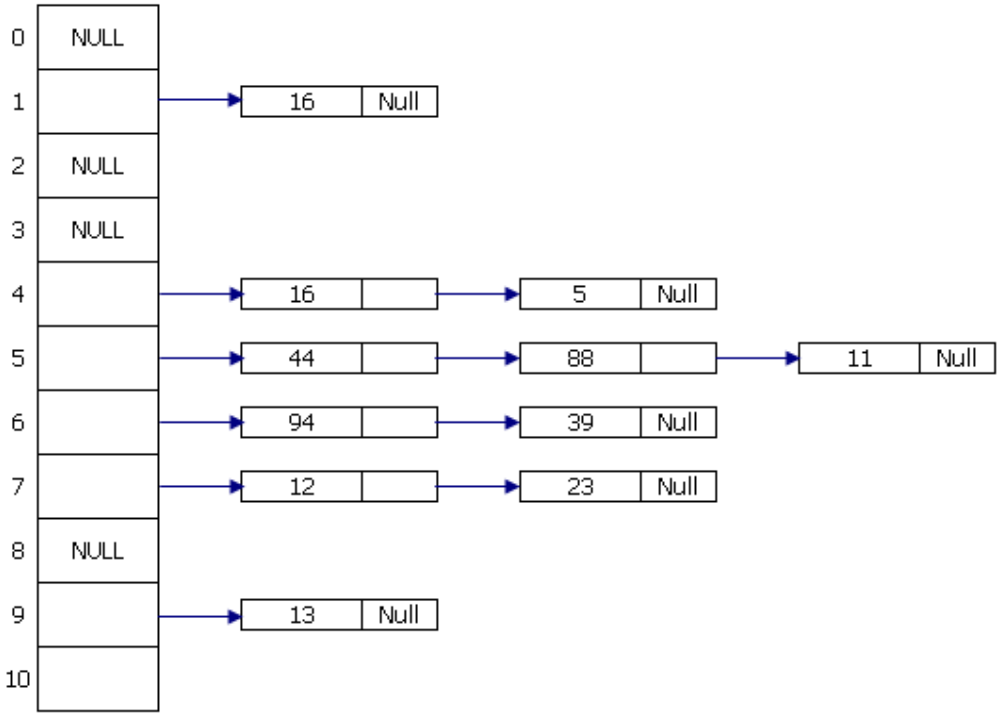
the address 1247 by selecting digits in position 1,2,4 and 7 then by reversing their order.

Q.179 Draw the 11-item hash table resulting from hashing the keys 12, 44, 13, 88, 23, 94, 11, 39, 20, 16 and 5 using the hash function h (i) = (2i+5) mod 11 **(7)**

Ans :

Table size is 11

h(12) = (24+5) mod 11 = 29 mod 11 = 7
h(44) = (88+5) mod 11 = 93 mod 11 = 5
h(13) = (26+5) mod 11 = 31 mod 11 = 9
h(88) = (176+5)mod 11= 181mod 11 = 5
h(23) = (46+5) mod 11 = 51 mod 11 = 7
h(94) = (188+5)mod 11= 193mod 11 = 6
h(11) = (22+5) mod 11 = 27 mod 11 = 5
h(39) = (78+5) mod 11 = 83 mod 11 = 6
h(20) = (40+5) mod 11 = 45 mod 11 = 1
h(16) = (24+5) mod 11 = 29 mod 11 = 4
h(5) = (10+5) mod 11 = 15 mod 11 = 4



Q.180 Explain any two methods to resolve collision during hashing. **(4)**

Ans :

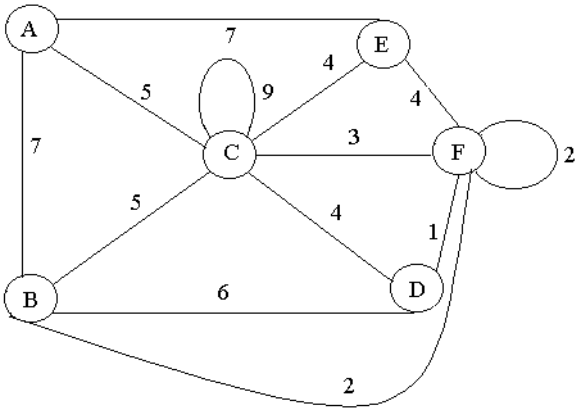
The two methods to resolve collision during hashing are:

Open addressing and Chaining.

Open addressing: The simplest way to resolve a collision is to start with the hash address and do a sequential search through the table for an empty location. The idea is to place the record in the next available position in the array. This method is called linear probing. An empty record is indicated by a special value called null. The major drawback of the linear probe method is clustering.

Chaining: In this technique, instead of hashing function value as location we use it as an index into an array of pointers. Each pointer access a chain that holds the element having same location.

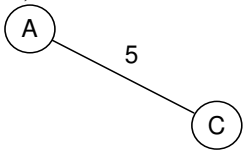
Q.181 Find out the minimum spanning tree of the following graph. **(8)**



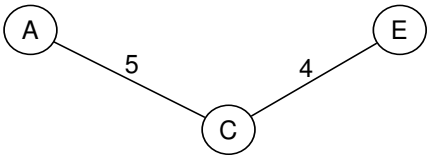
Ans :

The minimum spanning tree of given graph: -

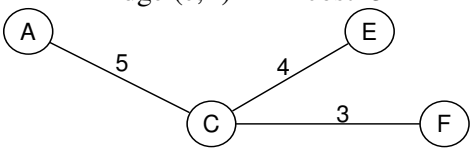
Step 1: Edge (a, c) cost=5 add to tree T



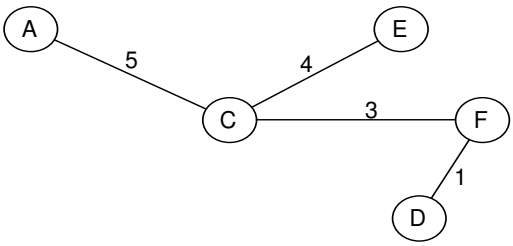
Step 2: Edge (c, e) cost=4 add to T



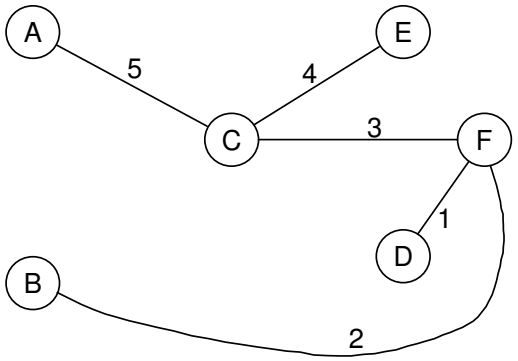
Step 3: Edge (c, f) cost=3 add to T



Step 4: Edge (f, d) cost=1 add to T



Step 5: Edge (f, b) cost=2 add to T



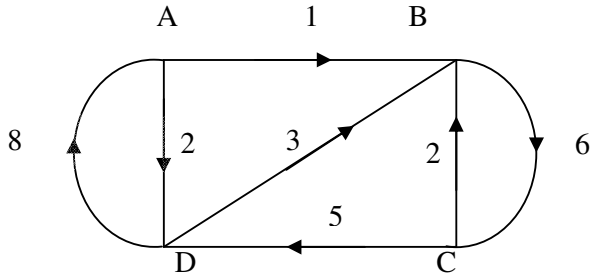
Thus the above tree is the minimum spanning tree of the graph.

Q.182 Give an algorithm to compute the second minimum spanning tree of a graph. **(8)**

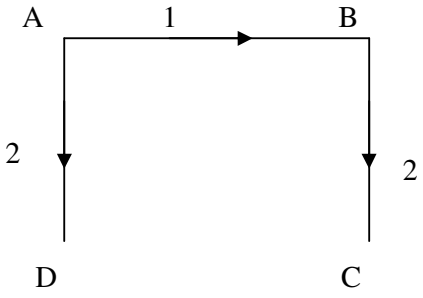
Ans : Find out the minimum spanning tree using Prim`s or Kruskal`s Algorithm.
Remove one edge from MST. One vertex will become isolated. Add an edge, which is not in the MST, such that the vertex gets connected. Find out the increase in the total weight of the spanning tree.
Repeat this process with every edge of the MST.
Remove that edge, whose removal and addition of some other edge, will increase the total weight of MST by minimum. This will give the second minimum spanning tree.

For example:-

Let G be as follows:-



Now MST:-



Remove AB from MST
Now to get a spanning tree, either DB is to be added or CD is to be added. Addition of DB will increase the weight by 2 units (i.e. 3 – 1) and addition of CD will increase the weight by 4 units (5-1). Therefore removal of AB will increase the weight of the next spanning tree by 2.

Now remove edge CB then CD or BC is to be added, edge CVD will increase the weight of the tree by 3 units (5-2)
Similarly if AD is to be removed, it will be replaced by CD which will increase the weight of the tree by 3. Thus the second minimum spanning tree is obtained by remaining AB and adding DB.

Q.183 Define a threaded binary tree. Write an algorithm for inorder traversal of a threaded binary tree. **(6)**

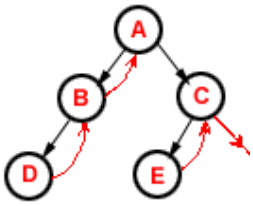
Ans :

Threaded Binary Tree:-

If a node in a binary tree is not having left or right child or it is a leaf node then that absence of child node is represented by the null pointers. The space occupied by these null entries can be utilized to store some kind of valuable information. One possible way to utilize this space is to have special pointer that point to nodes higher in the tree that is ancestors. These special pointers are called threads and the binary tree having such pointers is called threaded binary tree.
There are many ways to thread a binary tree each of these ways either correspond either in-order or pre-order traversal of T.A Threaded Binary Tree is a binary tree in which every node that does not have a right child has a THREAD (in actual sense, a link) to its INORDER successor. By doing this threading we avoid the recursive method of traversing a Tree, which makes use of stacks and consumes a lot of memory and time.
The node structure for a threaded binary tree varies a bit and its like this

```
struct NODE
{
 struct NODE *leftchild;
 int node_value;
 struct NODE *rightchild;
 struct NODE *thread;
}
```

The INORDER traversal for the above tree is -- D B A E C. So, the respective Threaded Binary tree will be --



B has no right child and its inorder successor is A and so a thread has been made in between them. Similarly, for D and E. C has no right child but it has no inorder successor even, so it has a hanging thread.
The algorithm for doing the above task is as follows;

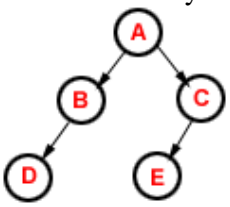
```
Void inorder (NODEPTR root)
{
 NODEPTR p, trail;
 p = root;
 do
       {
        trail = null;
        while ( p!= null)
        {
               trail = p;
               p = p -> left;
```

```
}
if (trail != null)
{
        printf (" %d\n", trail->info);
        p = trail -> right;
        while ( trail -> rt && p != NULL)
        {
                printf ("%d/x", p -> info);
                trail =p;
                p = p -> right;
        }
}
} while (trail != NULL);
  }
```

Q.184 Give an algorithm to sort data in a doubly linked list using insertion sort. **(10)**

Ans :

Simple ***Insertion sort*** is easily adaptable to doubly linked list. In this Let's make the Threaded Binary tree out of a normal binary tree...



method there is an array *link* of pointers, one for each of the original array elements. Initially link[i] = i + 1 for 0 < = I < n-1 and link [n-1] = -1. Thus the array can be thought of as a doubly link list pointed to by an external pointer *first* initialized to 0. To insert the kth element the linked list is traversed until the proper position for x[k] is found, or until the end of the list is reached. At that point x[k] can be inserted into the list by merely adjusting the list pointers without shifting any elements in the array. This reduces the time required for insertion but not the time required for searching for the proper position. The number of replacements in the link array is O (n).

```
void insertion_sort( input type a[ ], unsigned int n )
{
 unsigned int j, p;
 input type tmp;
        a[0] = MIN_DATA;          /* sentinel */
 for ( p=2; p <= n; p++ )
 {
              tmp = a[p];
              for ( j = p; tmp < a[j-1]; j-- )
                   a[j] = a [j-1];
              a[j] = tmp;
 }
}
```

Q.185 What is an Abstract Data Type (ADT)? Explain with an example. **(4)**

Ans :

Abstract data types or **ADTs** are a mathematical specification of a set of data and the set of operations that can be performed on the data. They are abstract in the sense that the focus is on the definitions of the constructor that returns an abstract handle

that represents the data, and the various operations with their arguments. The actual implementation is not defined, and does not affect the use of the ADT.
For **example**, rational numbers (numbers that can be written in the form a/b where a and b are integers) cannot be represented natively in a computer. A Rational ADT could be defined as shown below.
Construction: Create an instance of a rational number ADT using two integers, a and b, where a represents the numerator and b represents the denominator.
Operations: addition, subtraction, multiplication, division, exponentiation, comparison, simplify, conversion to a real (floating point) number.
To be a complete specification, each operation should be defined in terms of the data. For example, when multiplying two rational numbers a/b and c/d, the result is defined as ac/bd. Typically, inputs, outputs, preconditions, postconditions, and assumptions to the ADT are specified as well.
When realized in a computer program, the ADT is represented by an interface, which shields a corresponding implementation. Users of an ADT are concerned with the interface, but not the implementation, as the implementation can change in the future.
ADTs typically seen in textbooks and implemented in programming languages (or their libraries) include:

- String ADT
- List ADT
- Stack (last-in, first-out) ADT
- Queue (first-in, first-out) ADT
- Binary Search Tree ADT
- Priority Queue ADT
- Complex Number ADT (imaginary numbers)

There is a distinction, although sometimes subtle, between the abstract data type and the data structure used in its implementation. For example, a List ADT can be represented using an array-based implementation or a linked-list implementation. A List is an abstract data type with well-defined operations (add element, remove element, etc.) while a linked-list is a pointer-based data structure that can be used to create a representation of a List. The linked-list implementation is so commonly used to represent a List ADT that the terms are interchanged and understood in common use.
Similarly, a Binary Search Tree ADT can be represented in several ways: binary tree, AVL tree, red-black tree, array, etc. Regardless of the implementation, the Binary Search Tree always has the same operations (insert, remove, find, etc.)

Q.186 Suppose we wish to multiply four matrices of real numbers M1*M2*M3*M4 were M1 is 10*20, M2 is 20*50, M3 is 50*1 and M4 is 1*100. Assume that the multiplication of a p*q matrix by q*r matrix requires pqr scalar operations, find the optimal order in which to multiply the matrices so as to minimize the total number of scalar operations. **(5)**

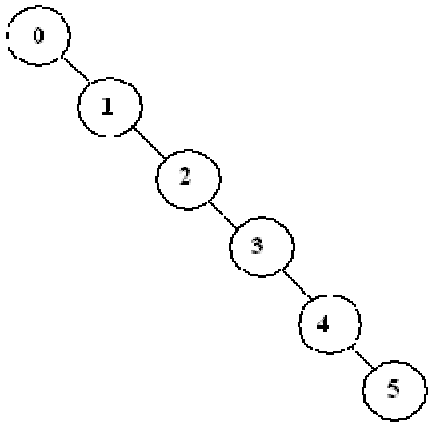
Ans :
Here M1 = 10x20
M2 = 20x50
M3 = 50x1 and
M4=1x100
Now optimal order is
1. M2*M3 = [20x50]*[50x1] = [20x1]
And no of scalar operations are 20*50*1 = 1000

2. M1*M2M3 = [10x20]*[20x1] = [10x1]
   And no of scalar operations are 10*20*1 = 200
3. M1M2M3*M4 = [10x1]*[10x100] = [10x100]
   And no of scalar operations are 10*1*100=1000
   So total no of scalar operations require are 1000+200+1000=2200 .

Q.187 Show a tree (of more than one node) for which the preorder and inorder traversals generate the same sequence. Is this possible for preorder and post order traversals? If it is, show an example. **(5)**

Ans :

A binary tree with 5 nodes 0 , 1, 2, 3 and 4 whose inorder and post order sequences are the same .



This is not possible for preorder and post order traversals.

Q.188 Write modules to do the following operations on a Binary Tree.

(i) Count the number of leaf nodes.
(ii) Count the number of nodes with two children.

(9)

Ans :

```
(i ) Leafcount (T)
        {
         static int n=0;
         if (T!= NULL)
        {leaf count (T→ left);
        if (T→left == NULL && T→right = NULL)
        n++
        leafcount (T→right)
        }
        return (n);}
  (ii) Leafcount (T)
        {
        static int n=0;
         if (T! = NULL)
        {leaf count (T→ left);
        if (T→left!= NULL && T→right! = NULL)
        n++
        leafcount (T→right)
        }
        return (n);}
```

Q.189 If n >=1, prove that for any n-key B-tree T of height h and minimum degree t >= 2, h <= $\log_t (n+1)/2$. **(7)**

Ans :

In the case of a B-tree with a minimum no of keys , there is one key in the root , 2 nodes at depth 1 , $2t^i-1$ nodes at depth i .

Let h be the height of the tree and n be the no of nodes . Then

$$n >= 1+(t-1) \sum_{i=1}^{h} 2t^i-1$$

which works out to $t^h <= (n+1) / 2$

So we take the $\log_t$ of both sides

$h <= \log_t(n+1)/2$

Q.190 A cocktail shaker sort designed by Donald Kunth is a modification of bubble sort in which the direction of bubbling changes in each iteration: in one iteration, the smallest element is bubbled up; in the next, the largest is bubbled down; in the next, the second smallest is bubbled up; and so forth. Write an algorithm to implement this and explore its complexity. **(14)**

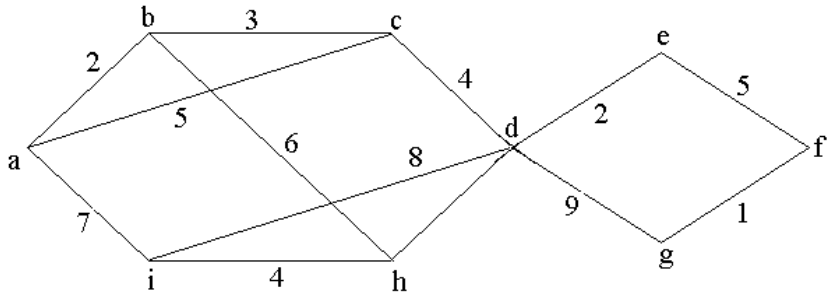
Ans :

Cocktail sort, also known as **bidirectional bubble sort**, **cocktail shaker sort**, **shaker sort**, **ripple sort**, or **shuttle sort**, is a stable sorting algorithm that varies from **bubble sort** in that instead of repeatedly passing through the list from top to bottom, it passes alternately from top to bottom and then from bottom to top. It can achieve slightly better performance than a standard bubble sort.

Complexity in Big O notation is $O(n^2)$ for a worst case, but becomes closer to O(n) if the list is mostly ordered at the beginning.

```
void cocktail_sort (int A[], int n)
 {
   int left = 0, right = n;
   bool finished;
   do
   {
    finished = true;
    --right;
    for (int i = left; i < right; i++)
     if (A[i] > A[i+1])
        {
      std::swap(A[i], A[i+1]);
      finished = false;
     }
   if (finished) return; finished = true;
  for (int i = right; i > left; i--)
   if (A[i] < A[i-1])
    {
     std::swap(A[i], A[i-1]);
     finished = false;
    }
  ++left;
 } while (!finished);
}
```

Q.191 Consider the following undirected graph:



(i) Find a minimum cost spanning tree by Kruskal's algorithm.
(ii) Find a depth-first spanning tree starting at ***a*** and at ***d***.
(iii) Find a breadth-first spanning tree starting at ***a*** and at ***d***.
(iv) Find the adjacency list representation of the graph. **(3.5 x 4 = 14)**

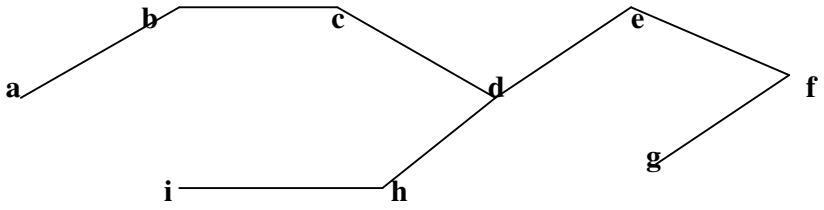
Ans :

(i) Find a minimum cost spanning tree by kruskal's algorithm .

Assume dh = 3

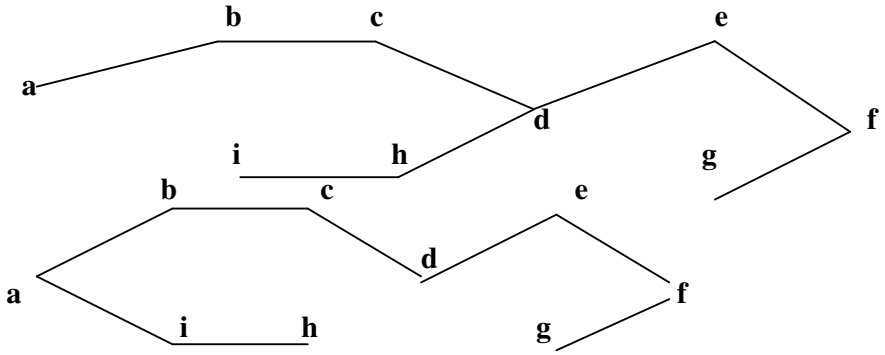
Edges	(g f)	=	1	Accept
	(a b)	=	2	Accept
	(d e)	=	2	Accept
	(b c)	=	3	Accept
	(d h)	=	3	Accept
	(c d)	=	4	Accept
	(i h)	=	4	Accept
	(e f)	=	5	Accept
	(a c)	=	5	Reject
	(b h)	=	6	Reject
	(a i)	=	7	Reject
	(i d)	=	8	Reject
	(d g)	=	9	Reject

So minimum cost spanning tree is :

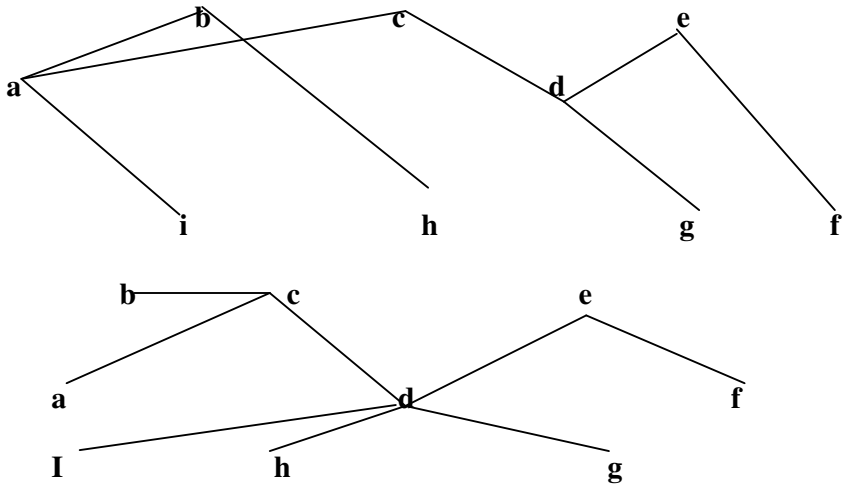


And Cost is 2+3+4+2+5+1+3+4 = 24

(ii) Find a Depth first spanning tree starting at a and d .



(iii) Find a breadth first spanning tree starting at a and d .



(iv) Find the adjacent list representation of the graph .

a	→	b,c,i
↓	→	
b	→	a,c,h
↓	→	
c	→	a,b,d
↓	→	
d	→	c,e,g,h,i
↓	→	
e	→	d,f
↓	→	
f	→	e,g
↓	→	
g	→	d,f
↓	→	
h	→	b,d,i
↓	→	
i	→	a,d,h

Q.192 Work through Binary Search algorithm on an ordered file with the following keys: {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16}. Determine the number of key comparisons made while searching for keys 2, 10 and 15. **(6)**

Ans :

Here List={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16}

Binary Search for key 2

(1) Here bottom =1 Top =16 and middle =(1+16)/2 = 8
Since 2 < list(8)

(2) bottom = 1 Top = middle-1=7 and middle=(1+7)/2 =4
2 < list(4)

(3) bottom = 1 Top = middle-1=3 and middle=(1+3)/2 = 2
2 = List(2)

So total number of comparisons require = 3

Binary Search for key = 10

(1) Here bottom=1 Top=16 and middle = 8
10 > List(8)

(2) bottom = middle+1=9 Top=16 middle=(9+16)/2=12
10 < List(12)

(3) bottom =9 Top=middle-1=11 middle=(9+11)/2=10
10 = List(10)

So total no of comparisons = 3

Binary Search for key = 15

(1) Here bottom=1 Top=16 and middle = 8
15 > List(8)

(2) bottom = middle+1=9 Top=16 middle=(9+16)/2=12
15 > List(12)

(3) bottom =middle+1=13 Top=16 middle=(13+16)/2=14
15 > List(14)

(4) bottom = middle+1 =15 Top=16 middle=(15+16)/2=15
15=List(15)

So total no of comparisons = 4

Q.193 What are threaded binary trees? What are the advantages and disadvantages of threaded binary trees over binary search trees? **(4)**

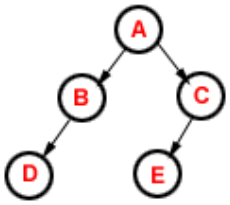
Ans :

A Threaded Binary Tree is a binary tree in which every node that does not have a right child has a THREAD (in actual sense, a link) to its INORDER successor. By doing this threading we avoid the recursive method of traversing a Tree, which makes use of stacks and consumes a lot of memory and time.

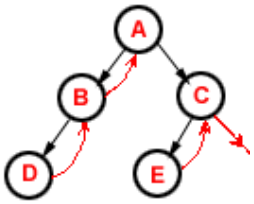
The node structure for a threaded binary tree varies a bit and its like this --

```
struct NODE
{
 struct NODE *leftchild;
 int node_value;
 struct NODE *rightchild;
 struct NODE *thread;
}
```

Let's make the Threaded Binary tree out of a normal binary tree...



The INORDER traversal for the above tree is -- D B A E C. So, the respective Threaded Binary tree will be --



B has no right child and its inorder successor is A and so a thread has been made in between them. Similarly, for D and E. C has no right child but it has no inorder successor even, so it has a hanging thread.

Advantage

1. By doing threading we avoid the recursive method of traversing a Tree , which makes use of stack and consumes a lot of memory and time .
2 . The node can keep record of its root .

Disadvantage

1. This makes the Tree more complex .
2. More prone to errors when both the child are not present & both values of nodes pointer to their ansentors .

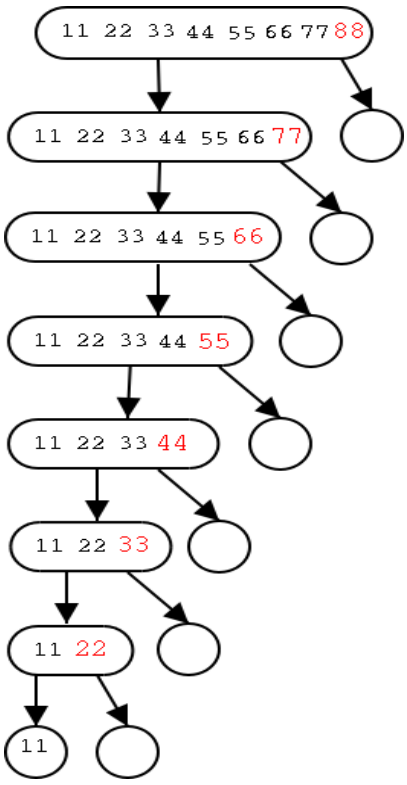
Q.194 Show that quick algorithm takes O $\left(n^2\right)$ time in the worst case. **(4)**

Ans :

The Worst Case for Quick-SortThe tree illustrates the worst case of quick-sort, which occurs when the input is already sorted!

The height of the tree is N-1 not O(log(n)). This is because the pivot is in this case the largest element and hence does not come close to dividing the input into two pieces each about half the input size. It is easy to see that we have the worst case. Since the pivot does not appear in the children, at least one element from level i does not appear in level i+1 so at level N-1 you can have at most 1 element left. So we have the highest tree possible. Note also that level i has at least i pivots missing so cna have at most N-i elements in all the nodes. Our tree achieves this maximum. So the time needed is proportional to the total number of numbers written in the diagram which is N + N-1 + N-2 + ... + 1, which is again the one summation we know N(N+1)/2 or $\Theta(N^2$.

Hence the ***worst case*** complexity of quick-sort is quadratic .



Q.195 Write an algorithm to convert an infix expression to a postfix expression. Execute your algorithm with the following infix expression as your input. $(m+n)*(k+p)/(g/h)\uparrow(a\uparrow b/c)$ **(8)**

Ans :

Algorithm: Polish (Q,P)

Suppose Q is an arithmatic expression written in fix notation. This algorithm finds the equivalent postfix expression P.

1. Push " (" onto STACK, and add ")" to the end of Q
2. Scan Q from left to right and repeat Steps 3 to 6 for each element of Q until the STACK is empty:
3. If an operand is encountered, add it to P
4. If a left parenthesis is encountered, push it onto STACK.
5. If an operator [x] is encountered, then:

(a) repeatedly pop from STACK and add to p each operator (on the top of stack) has the same precedence as or higher precedence than [x]

(b) add [x] to STACK.

[End of If structure]

6. If a right parenthesis is encountered, then:

(a) Repeatedly pop from STACK and add to P each operator (on the top of STACK) until a left parenthesis is encountered.

(b) Remove the left parenthesis. [Do not add the left parenthesis to P]

[End of If structure.]

[End of Step 2 loop.]

7. Exit

Example :
Input (m+n)*(k+p)/(g/h)^(a^b/c)

Step 1 : (| (| Output : nil

Step 2 : m | (| Output : m

Step 3 : + | + (| Output : m

Step 4 : n | + (| Output : mn

Step 5 :) | | Output : mn+

Step 6 : * | * | Output : mn+

Step 7 : (| (* | Output : mn+

Step 8 : k | (* | Output : mn+k

Step 9 : + | + (* | Output : mn+k

Step 10 : p | + (* | Output : mn+kp

Step 11 :) | * | Output : mn+kp+

Step 12 : / | / | Output : mn+kp+*

Step 13 :(| (/ | Output : mn+kp+*

Step 14 :g | (/ | Output : mn+kp+*g

Step 15 :/ | / (/ | Output : mn+kp+*g

Step 16 : h | / (/ | Output : mn+kp+*gh

Step 17 :^ | ^ / | Output : mn+kp+*gh/

Step 18 :(| (^ / | Output : mn+kp+*gh/

Step 19 : a | (^ / | Output : mn+kp+*gh/a

Step 20 : ^ | ^ (^ / | Output : mn+kp+*gh/a

Step 21 : b | ^ (^ / | Output : mn+kp+*gh/ab

Step 22 : / | / (^ / | Output : mn+kp+*gh/ab^

Step 23 : c | / (^ / | Output : mn+kp+*gh/ab^c

Step 24 :) | ^ / | Output : mn+kp+*gh/ab^c/

Step 25 : End | | Output : **mn+kp+*gh/ab^c/^/**

Q.196 What is recursion? A recursive procedure should have two properties. What are they? **(4)**

Ans :

Recursion means function call itself repeately .

Properties :

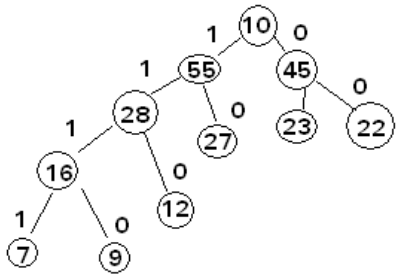
(1) There must be some **base case** where the condition end .

(2) Each recursive call to the procedure involve a smaller case of the problem .

Q.197 Suppose characters a,b,c,d,e,f have probabilities 0.07, 0.09, 0.12, 0.22, 0.23, 0.27 respectively. Find an optimal Huffman code and draw the Huffman tree. What is the average code length? **(10)**

Ans :

Huffman Tree :



Huffman Code are a=1111
b =1110
c=110
d=00
e=01
f=10

Average code length = P(a).4+P(b).4+P(c).3+P(d).2+P(e).2+P(f).2
= 0.07*4+0.09*4+0.12*3+0.22*2+0.23*2+0.27*2
= .28+.36+.36+.44+.46+.54 =2.44

Q.198 Prove that the number of nodes with degree 2 in any Binary tree is 1 less than the number of leaves. **(4)**

Ans :

The proof is by induction on the size n of T .

Let L(T) = No of leaves & $D_2(T)$= No of nodes of T of degree 2

To Prove D2(T)=L(T)-1

Basic Case : n=1 , then T consists of a single node which is a leaf .

L(T)=1 and D2(T)=0

So $D_2(T)$=L(T)-1

Induction Step : Let n > 1 and assume for all non empty trees T1 of size k<n that $D_2(T`)$=L(T`)-1

Since n >1 , at least one of x=left(T) or y=right(T) is non empty .

Assume x is non empty ; the other case is symetric

By the induction hypothesis ,

$D_2(x)$ = L(y)-1

IF y is empty , then again by the induction hypothesis ,

$D_2(y)$ =L(y)-1 and

$D_2(T)$ =$D_2(x)$+$D_2(y)$+1

=L(x)-1+L(y)-1+1

= L(x)+L(y)-1

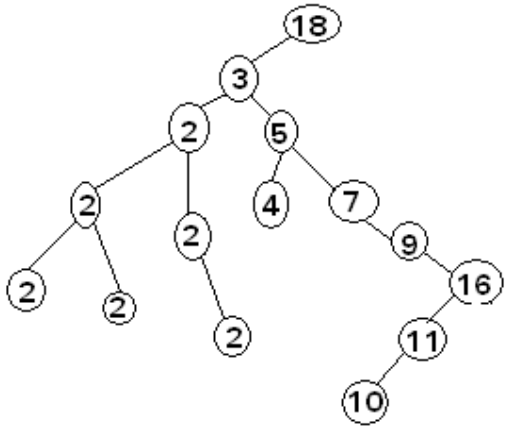
$D_2(T)$ = L(T)-1 Hence Proved

Q.199 Equal keys pose problem for the implementation of Binary Search Trees. What is the asymptotic performance the usual algorithm to insert n items with identical keys into an initially empty Binary Search Tree? Propose some technique to improve the performance of the algorithm. **(7)**

Ans :

A sympotic performance will be linear in O(n). That is the tree created will be skew symmetric. Identical keys should be attached either as left or right child of the parent, wherever positions are available. For eg. 18,3,5,2,4,7,2,9,2,16,2,11,2,10,2

The creation of the tree should be as follows:



i.e. similar keys should not skew to the BST. Then performance will become inferior.

Q.200 Consider a list of numbers 9, 20, 6, 10, 14, 8, 60, 11 given. Sort them using Quick Sort. Give step wise calculation. **(8)**

Ans:

9,20,6,10,14,8,60,11

choosing 9 as Pivot(Pass—1)

6,8,9,10,14,20,60,11

The sublist on the left side of 9 is sorted therefore, no more sorting on left sublist

Now choosing 10 as Pivot (pass – 2)

(10 is already in the position)

6,8,9,10,14,20,60,11

Choosing 14 as Pivot (pass –3)

6,8,9,10,11,14,60,20

Choosing 60 as Pivot [pass – 4)

6,8,9,10,11,14,20,60

Q.201 What is a sparse matrix? Explain an efficient way of storing a sparse matrix in memory. **(4)**

Ans:

Sparse Matrix

A matrix in which number of zero entries are much higher than the number of non zero entries is called sparse matrix. The natural method of representing matrices in memory as two-dimensional arrays may not be suitable foe sparse matrices. One may save space by storing for only non zero entries. For example matrix A (4*4 matrix) represented below

where first row represent the dimension of matrix and last column tells the number of non zero values; second row onwards it is giving the position and value of non zero number.

0	0	0	15
0	0	0	0
0	9	0	0
0	0	4	0

Here the memory required is 16 elements X 2 bytes = 32 bytes

The above matrix can be written in sparse matrix form as follows:

4	4	3
0	3	15
2	1	9
3	2	4

Here the memory required is 12elements X 2 bytes = 24 bytes

Q.202 Evaluate the following prefix expressions **(9)**

(i) + * 2 + / 14 2 5 1

(ii) – * 6 3 – 4 1

(iii) + + 2 6 + – 13 2 4

Ans:

Evaluating a prefix expression.

(i)	+	*	2	+	/	14	2	5	1
	=	+	*	2	+	7	5	1	
	=	+	*	2	12	1			
	=	+	24	1					
	=	25							
(ii)	-	*	6	3	-	4	1		
	=	-	18	-	4	1			
	=	-	18	3					
	=	15							
(iii)	+	+	2	6	+	-	13	2	4
	=	+	8	+	-	13	2	4	
	=	+	8	+	11	4			
	=	+	8	15					
	=	23							

Q.203 (i) Give four properties of Big – O Notations. **(4)**

(ii) Give the graphical notations of Big-O estimations for the following functions:

$\log_2 n$, n , $n \log_2 n$, n^2, n^3, 2^n **(3)**

Ans:

(i) Four properties of Big-O notations are:

1. Reflexivity
2. Symmetry
3. Transpose Symmetry
4. Transitivity

Reflexivity:

$f(n)=O(f(n))$

Symmetry and Transpose Symmetry

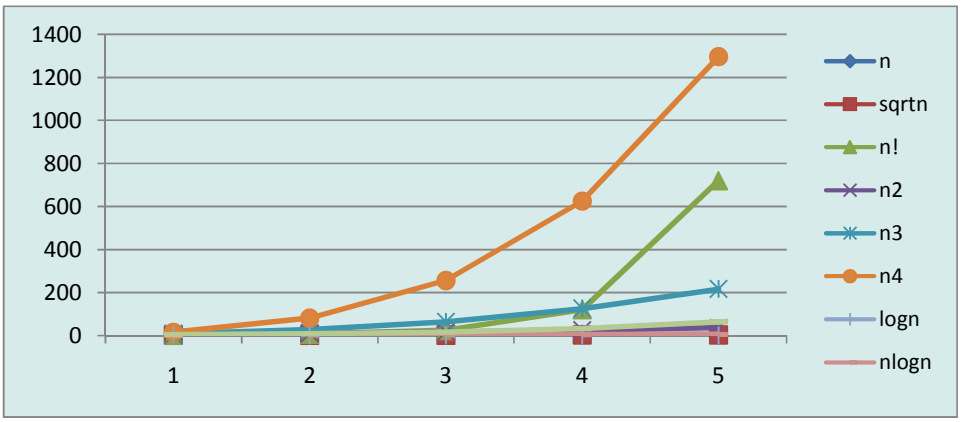
$f(n)=O(g(n))$ iff $g(n)=\Omega(f(n))$

Transitivity

$f(n)=O(g(n))$ and $g(n)=O(h(n))$ implies $f(n)= O(h(n))$

(ii)

n	Sqrt n	n!	n2	n3	n4	logn	nlogn	2n
2	1.41	2	4	8	16	0.30103	0.60206	4
3	1.73	6	9	27	81	0.477121	1.431364	8
4	2.00	24	16	64	256	0.60206	2.40824	16
5	2.24	120	25	125	625	0.69897	3.49485	32
6	2.45	720	36	216	1296	0.778151	4.668908	64
7	2.65	5040	49	343	2401	0.845098	5.915686	128
8	2.83	40320	64	512	4096	0.90309	7.22472	256
9	3.00	362880	81	729	6561	0.954243	8.588183	512
10	3.16	3628800	100	1000	10000	1	10	1024



Q.204 Write an algorithm to insert an element *k* in double linked list at **(8)**

(i) Start of linked list
(ii) After a given position *P* of list
(iii) End of linked list.

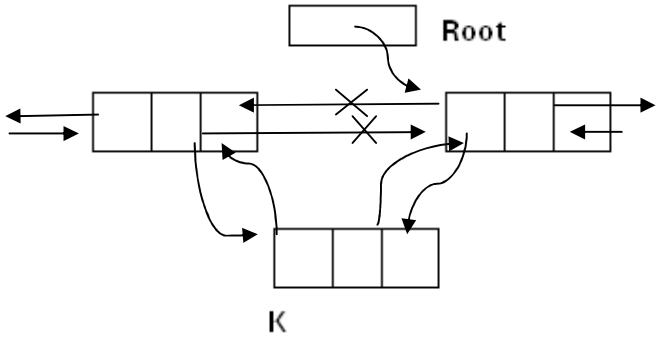
Ans:

Algorithm to insert an element k in double linked list at a position which is:-

(i) Start of a linked list

while inserting at the start of the linked list we have to change the root node and so we need to return the new root to the calling program.

```
nodeptr insertatstart ( nodeptr root, nodeptr k)
{
k->next = root;
k->back = root->back;
root->back->next = k;
root->back=k;
return k;
}
```

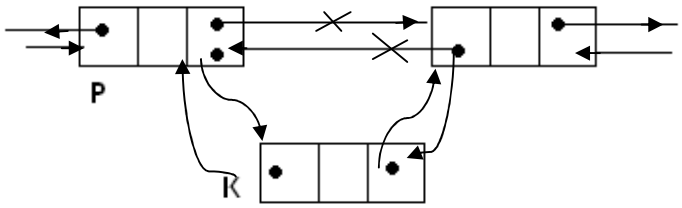


(ii) After a given position p of list

Here we assume that p points to that node in the doubly linked list after which insertion is required.

```
insertafter ( nodeptr p , nodeptr k )
{
        if (p == null)
        {
                printf ("void insertion /n");
                return;
        }
        k->back = p;
```

```
            k->next = p->next;
            p->next->back = k;
            p->next = k;
            return;
      }
```



(iii) End of linked list

Here k is the element to be inserted and root is the pointer to the first node of the doubly linked list.

```
      insertatend ( nodeptr root, nodeptr k )
      {
            nodeptr trav;
            while (trav->next != root)
                  trav = trav->next;
            k->next=root;
            root->back=k;
          trav->next = k;
            k->back=trav;
            return;
      }
```

Q.205 Write an algorithm to add two polynomials using linked list. **(8)**

Ans:

Algorithm to add two polynomials using linked list is as follows:-

```
Let p and q be the two polynomials represented by the linked list.
1. while p and q are not null, repeat step 2.
2. If powers of the two terms ate equal
 then if the terms do not cancel
 then insert the sum of the terms into the sum Polynomial
              Advance p
              Advance q
 Else if the power of the first polynomial> power of second
 Then insert the term from first polynomial into sum polynomial
              Advance p
 Else insert the term from second polynomial into sum polynomial
              Advance q
3. copy the remaining terms from the non empty polynomial into the
sum polynomial.
```

The third step of the algorithm is to be processed till the end of the polynomials has not been reached.

Q.206 Write modules to perform the following operation on Binary tree **(8)**

(i) count number of leaf nodes

(ii) find height of tree

Ans:

(i) count number of leaf nodes

```
int leaf count (node  *t)
{
static int count = 0;
      if (t! = NULL)
{
     leaf count (t→ left)
    if (t→left == NULL && t→right == NULL)
       count ++;
      leaf count (t→right);
}
       return (count);
}
```

(ii) Module to find height of tree

```
height(Left,Right,Root,height)
  1.  If Root=Null then height=0;
      return;
  2.  height(Left,Right,Left(Root),heightL)
  3.  height(Left,Right,Right(Root),heightR)
  4.  If heightL≥heightR then
            height=heightL+1;
       Else
            height=heightR+1;
  5.  Return;
```

Q.207 Create B-Tree of order 5 from the following list of data items: **(8)**
20, 30, 35, 85, 10, 55, 60, 25

Ans:

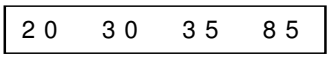
For the given list of elements as

20 30 35 85 10 55 60 25

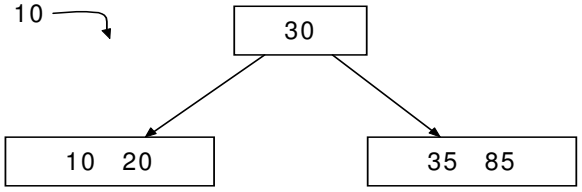
The B-Tree of order 5 will be created as follows

Step 1:

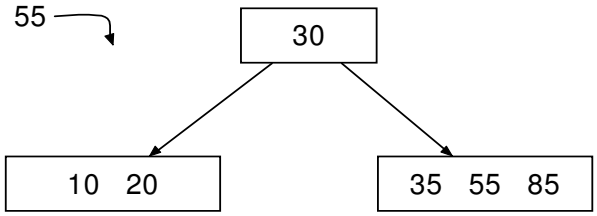
Insert 20, 30, 35 and 85



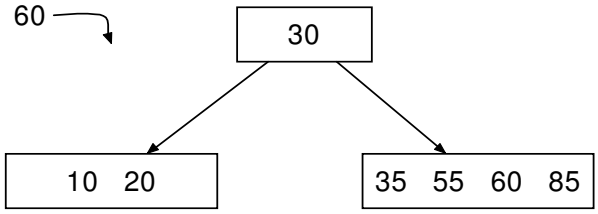
Step 2:



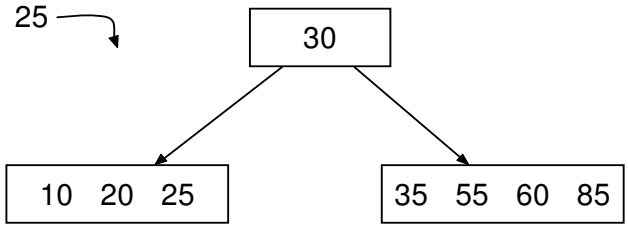
Step 3:



Step 4:



Step 5:



This is the final B-tree created after inserting all the given elements.

Q.208 Write an algorithm to insert an element *k* into binary search tree. Give the analysis and example. **(8)**

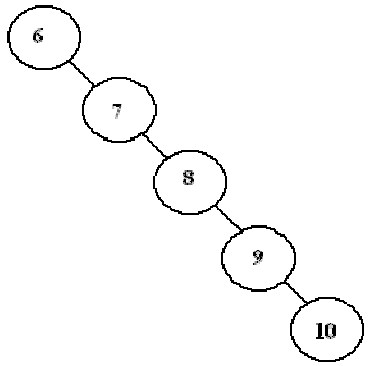
Ans:

The algorithm to insert an element k into binary search tree is as follows:-

```
/* Get a new node and make it a leaf*/
getnode (k)
left(k) = null
right(k) = null
info (k) = x
/* Initialize the traversal pointers */
p = root
trail = null
/* search for the insertion place */
while p <> null do
begin
trail = p
if info (p) > x
       then p = left (p)
else
       p = right (p)
end
/* To adjust the pointers */
If trail = null
Then root = k /* attach it as a root in the empty tree */
      else
             if info (trail) > x
             then
                    left (trail) = k
             else
                    right (trail) = k
```

Analysis : - We notice that the shape of binary tree is determined by the order of insertion. If the values are sorted in ascending or descending order, the resulting tree will have maximum depth equal to number of input elements. The shape of the tree is important from the point of view of search efficiency. The depth of the tree determines the maximum number of comparisons. Therefore we can maximize search efficiency by minimizing the height of the tree.

eg. For the values as 6, 7, 8, 9, 10
the binary tree formed would be as follows



Q.209 Fill in the following table, showing the *number of comparisons* necessary to either find a value in the array DATA or determine that the value is not in the array. **(8)**

DATA

26	42	96	101	102	162	197	201	243
[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

Number of Comparisons

Value	**Sequential Ordered Search**	**Binary Search**
26		
2		
103		
244		

Ans:

Value	Sequential Ordered Search	Binary Search
26	1	4
2	1	4
103	6	3
244	9	4

Q.210 Explain the functionality of linear and quadratic probing with respect to hashing technique. **(8)**

Ans:

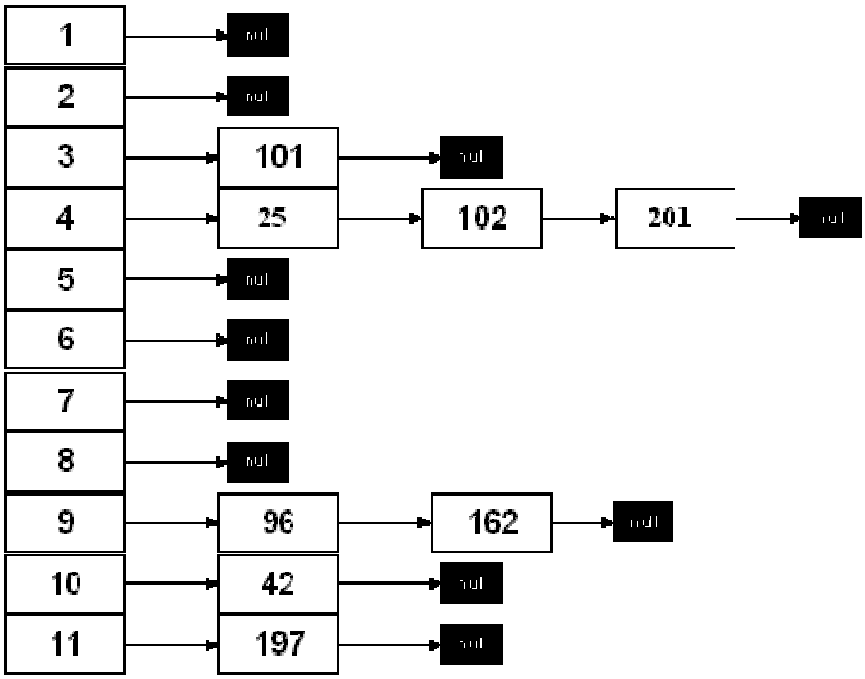
Linear Probing: The simplest way to resolve a collision is to start with the hash address and do a sequential search through the table for an empty location. The idea is to place the record in the next available position in the array. This method is called linear probing. An empty record is indicated by a special value called null. The array should be considered circular, so that when the last location is reached the search proceeds to the first record of the array. An unoccupied record location is always found using this method if atleast one is available; otherwise, the search halts unsuccessfully after scanning all locations. The major drawback of the linear probe method is that of clustering.

When the table is initially empty, it is equally likely that a new record will be inserted in any of the empty position but when the list becomes half full, records start to appear in long strings of adjacent positions with gaps between the strings. Therefore the search to find an empty position becomes longer.

Quadratic probing: In the above case when the first insertion is made the probability of new element being inserted in a particular position is 1/n where n is the size of the array. At the time of second insertion the probability becomes 2/n and so on for the k^{th} insertion the probability is k/n, which is k times as compared to any other remaining unoccupied position. Thus to overcome the phenomenon of long sequence of occupied positions to become even longer we use *quadratic rehash,* in this method if there is a collision at hash address h, this method probes the array at locations h+1, h+4, h+9, ... That is (h(key) $+i^2$ % hash size) for i=1,2,... gives the i^{th} hash of h.

Q.211 The following values are to be stored in a hash table
25, 42, 96, 101, 102, 162, 197, 201
Use division method of hashing with a table size of 11. Use sequential method of resolving collision. Give the contents of array. **(8)**

Ans:
Table size is given as 11
H (25) = (25) mod 11 + 1 = 3 + 1 = 4
H (42) = (42) mod 11 + 1 = 9 + 1 = 10
H (96) = (96) mod 11 + 1 = 8 + 1 = 9
H (101) = (101) mod 11 + 1 = 2 + 1 = 3
H (102) = (102) mod 11 + 1 = 3 + 1 = 4
H (162) = (162) mod 11 + 1 = 8 + 1 = 9
H (197) = (197) mod 11 + 1 = 10 + 1 = 11
H (201) = (201) mod 11 + 1 = 3 + 1 = 4



Q.212 Write Kruskal's algorithm and give the analysis. Compare it with Dijkstra's method. **(8)**

Ans:

Kruskal's Algorithm for minimum cost spanning tree

```
1. Make the tree T empty.
2. Repeat steps 3, 4 and 5 as long as T contains less
than n-1 edges and E not empty; otherwise proceed to step 6.
3. Choose an edge (v,w) from E of lowest cost.
4. Delete (v,w) from E.
5. If (v,w) does not create a cycle in T then add(v,w) to T
else dicard(v,w).
6. If T contains fewer than n-1 edges then print ('no spanning
tree').
```

The complexity of this algorithm is determined by the complexity of sorting method applied; for an efficient sorting it is $O(|E|\log|E|)$.It also depends on the complexity of the methods used for cycle detection which is of $O(|V|)$ as only $|V|$-1 edges are added to tree which gives a total of O(E logV) time. So complexity of Kruskal's algorithm is is $O(|V+E|\log|V|)$. As V is asymptotically no larger than E, the running time can also be expressed as O(E log V).

Dijkstra has not developed any algorithm for finding minimum spanning tree. It is Prim's algorithm. Dijkstra's algorithm to find the shortest path can be used to find MST also, which is not very popular.

Q.213 Write algorithm for Breadth First Search (BFS) and give the complexity. **(8)**

Ans:

This traversal algorithm uses a queue to store the nodes of each level of the graph as and when they are visited. These nodes are then taken one by one and their adjacent nodes are visited and so on until all nodes have been visited. The algorithm terminates when the queue becomes empty.

Algorithm for Breadth First Traversal is as follows:

```
clearq (q)
visited (v) = TRUE
while not empty (q) do
       for all vertices w adjacent to v do
       if not visited then
       {
              insert (w , q)
              visited (w) = TRUE
       }
delete (v, q);
```

Here each node of the graph is entered in the queue only once. Thus the while loop is executed n times, where n is the order of the graph. If the graph is represented by adjacency list, then only those nodes that are adjacent to the node at the front of the queue are checked therefore, the for loop is executed a total of E times, where E is the number of edges in the graph. Therefore, breadth first algorithm is O (N*E) for linked expression. If the graph is represented by an adjacency matrix, the for loop is executed once for each other node in the graph because the entire row of the adjacency matrix must be checked. Therefore, breadth first algorithm is $O(N^2)$ for adjacency matrix representation.

Q.214 A binary tree T has 10 nodes. The inorder and preorder traversals of T yield the following sequence of nodes:

Inorder : D B H E A I F J C G
Preorder : A B D E H C F I J G

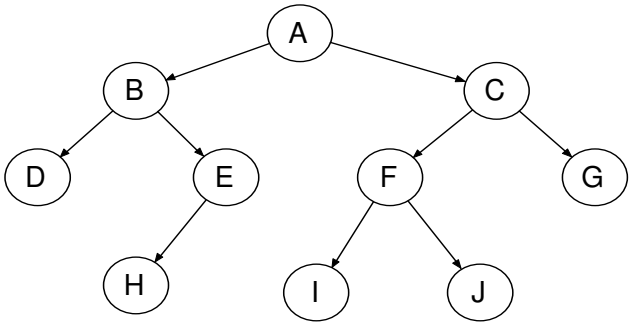
Draw the tree T. **(8)**

Ans:

Given of a tree, the

INORDER: D B H E A I F J C G
PREORDER: A B D E H C F I J G

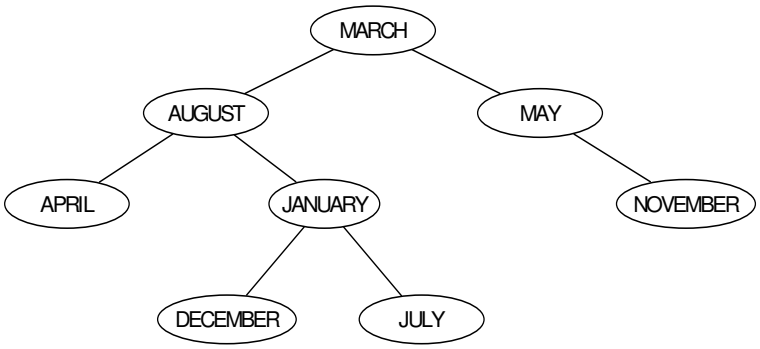
Then the tree of whose traversal is given is as follows



Q.215 Construct AVL tree from the following set of elements
{ March, May, November, August, April, January, December, July } **(8)**

Ans:

The AVL tree from the given set of elements is as follows:
{March, May, November, August, April, January, December, July}



Q.216 Evaluate the following postfix-expression using stack. Show the content of the stack in each step.

6 2 3 + - 3 8 2 / + * 2 $ 3 + **(6)**

Ans:

The content of stack in each iteration will be as follows:

Symbol	Opnd1	Opnd2	Value	Stack
6				6
2				6,2
3				6,2,3
+	2	3	5	6,5
-	6	5	1	1
3	6	5	1	1,3
8	6	5	1	1,3,8
2	6	5	1	1,3,8,2
/	8	2	4	1,3,4
+	3	4	7	1,7
*	1	7	7	7
2	1	7	7	7,2
$	7	2	49	49
3	7	2	49	49,3
+	49	3	52	52

The value returned is 52.

Q.217 Write an algorithm that will split a circularly linked list into two circularly linked lists. **(7)**

Ans:

A function that split a circular linked list into two linked list is as follows:

```
nodeptr splitlist(nodeptr p)
{
nodeptr p1, p2, p3;
p1 = p2 = p3 = p;
if (not p) return NULL;
do {
p3 = p2;
p2 = p2->next; /* advance 1 */
p1 = p1->next;
if (p1) p1 = p1->next; /* advance 2 */
} while (p1);
/* now form new list after p2 */
p3->next = NULL; /* terminate 1st half */
return p2;
} /* splitlist */
```

Q.218 Let A[n] be an array of n numbers. Design algorithms to perform the following operations:

Add (i, y): Add the value y to the i^{th} number in the array

Partial-sum(i) : returns the sum of the first i numbers in the array **(4)**

Ans:

Operation 1

```
add (i ,y)
{
        A[i] = A[i] + y;
 }
```

Operation 2

```
partialsum (i, y)
{
        int sum=0;
        for ( ;  i > 0 ; i--)
                sum = sum + A[i];
```

```
        return sum;
    }
```

Q.219 Calculate the efficiency of Quick sort. **(4)**

Ans:

Efficiency of Quick sort

Assume that the file size n is a power of 2

say $n=2^m$

$m=\log_2 n$ (taking log on both sides)..... (A)

Assume also that the proper position for the pivot always turns out to be exact middle of the sub array. In that case, there will be approx n comparisons (actually n-1) on the first pass, after which the file is split into two sub files each of size n/2. Each of them will have n/2 comparisons and file is again split into 4 files each of size n/4 and so on halving it m times.

Proceeding in this way:-

	No. of comparisons
1. file of size n	1*n=n
2. file of size n/2	2*n/2=n
3. file of size n/4	4*n/4=n
...	
...	
n file of size 1	n*1=n

Total no. of comparisons for entire sort

= n+n+n+.....+n (m times)

= nm

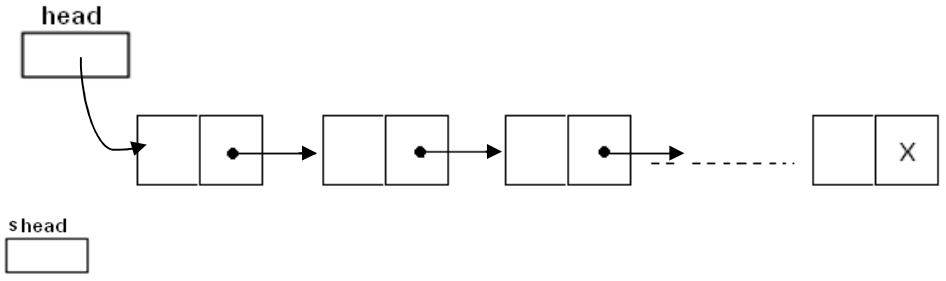
$=n \log_2 n$ (from A)

Therefore the efficiency of Quick sort is **O ($n \log_2 n$).**

Q.220 Formulate an algorithm to perform insertion sort on a linked list. **(8)**

Ans:

Simple ***Insertion sort*** is easily adaptable to singly linked list. In this method there is an array *link* of pointers, one for each of the original array elements. Initially link[i] = i + 1 for 0 < = I < n-1 and link [n-1] = -1. Thus the array can be thought of as a linear link list pointed to by an external pointer *first* initialized to 0. To insert the kth element the linked list is traversed until the proper position for x[k] is found, or until the end of the list is reached. At that point x[k] can be inserted into the list by merely adjusting the list pointers without shifting any elements in the array. This reduces the time required for insertion but not the time required for searching for the proper position. The number of replacements in the link array is O (n).



Each node from the unsorted linked list is to be detached one by one from the front and attached to the new list pointed by s head. While attaching, the value got stored in each node is considered so that it is inserted at the proper position.

```
S head = head
head=head    →next
S head→ next = NULL
While (head!=NULL)
{
P=head
head= head →next
q=S head
while (q→info<P→info & q!NULL)
{
r=q
q=q→next
}
P→next=q
r→next=P
}
```

Q.221 The following values are to be stored in a Hash-Table:
25, 42, 96, 101, 102, 162, 197, 201
Use the Division method of Hashing with a table size of 11. Use the sequential method for resolving collision. **(6)**

Ans:

The given values are as follows: -
25 42 96 101 102 162 197 201
Table size is = 11
Division method of Hashing: -
H (k) = {Hash address range from 0 to m-1. Key (mod) table-size.}
H (25)= 3
H (42) = 9
H (96) = 8
H (101) = 2
H (102) = 3
H (162) = 8
H (197) =10
H (201) = 3
The Hash table is as follows
Hash [0] = [197]
Hash [1] = [NULL]
Hash [2] = [101]
Hash [3] = [25] ← Collision Occurred
Hash [4] = [102] ← Inserted in next available slot
Hash [5] = [201]
Hash [6] = [NULL]
Hash [7] = [NULL]
Hash [8] = [96]
Hash [9] = [42]
Hash [10] = [162]

0	197	
1		
2	101	
3	25	---102(collision)
4	102	---201
5	201	
6		
7		
8	96	
9	42	
10	162	

Q.222 Rewrite your solution to part **(b)** using rehashing as the method of collision resolution. Use [(key+3) mod table-size] as rehash function. **(5)**

Ans: Rehash function [(key+3) mod table size]

H(25) = 28 mod 11
H(42) = 45 mod 11
H(96) = 99 mod 11
H(101) = 104mod 11
H(102) = 105mod 11
H(162) = 165 mod 11
H(197) = 200mod 11
H(201) = 204mod 11
The newly created hash function is as follows:-
Hash[0] = [96]
Hash[1] = [42]
Hash[2] = [162]
Hash[3] = [197]
Hash[4] = [NULL]
Hash[5] = [101]
Hash[6] = [25]
Hash[7] = [102]
Hash[8] = [201]
Hash[9] = [NULL]
Hash[10] = [NULL]

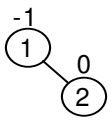
Q.223 How many AVL trees of 7 nodes with keys 1, 2, 3, 4, 5, 6, 7 are there? Explain your answer. **(6)**

Ans:

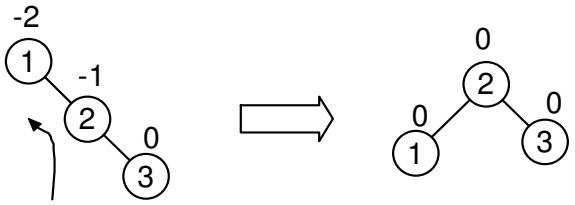
Step 1

No rebalancing required

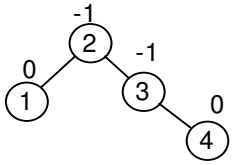
Step 2



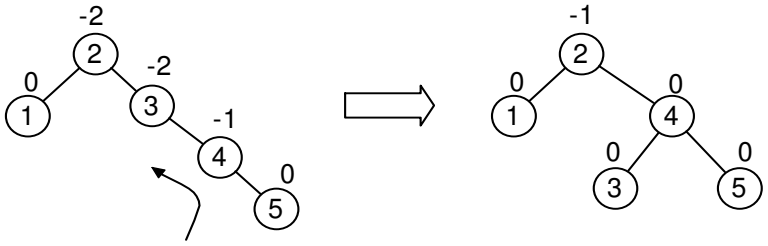
Step 3



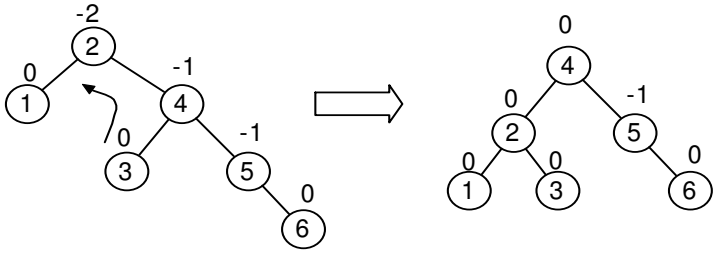
Step 4



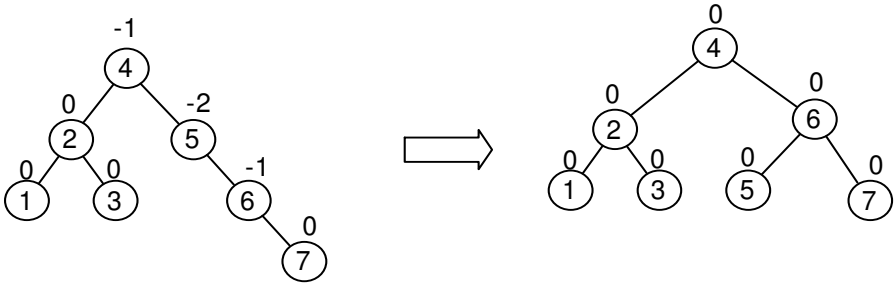
Step 5



Step 6



Step 7



Q.224 Describe the Dijkstra's algorithm for finding a shortest path in a given graph. **(8)**

Ans:

Let G = (v,e) be a simple graph. Let a & z be any two vertices of the graph. Suppose L(x) denotes the label of the vertex z which represents the length of the

shortest path from the vertex a to the vertex z. W_{ij} denotes the weight of the edge $e_{ig} = (V_i, V_j)$

1. let P = Q where p is the set of those vertices which have permanent labels &
T = {all vertices of the graph G}
Set L (a) = 0, L (x) = ∞for all x €T & x≠a

2. Select the vertex v in T which has the smallest label. This label is called the permanent label of v.
Also set P = P U {v} and T = T-{v}
if v=z then L (z) is the length of the shortest path from the vertex a to z and stop.

3. If v ≠z then revise the labels of vertices of T i.e. the vertices which do not have permanent labels. The new label of a vertex x in T is given by
L (x) = min{old L(x),L(v)+w(y,x)}
where w (v,x) is the weight of the edge joining the vertex v & x
If there is no direct edge joining v & x then take w(v,x) = ∞

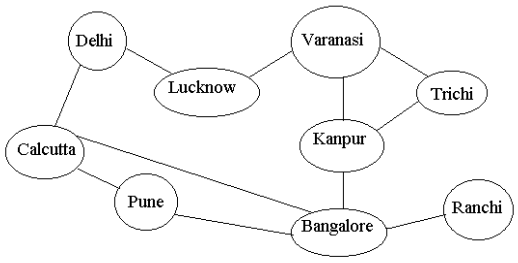
4. Repeat steps 2 and 3 until z gets the permanent label.

Q.225 Explain the difference between depth first and breadth first traversing techniques of a graph. **(4)**

Ans:

Depth-first search is different from Breadth-first search in the following ways:
A depth search traversal technique goes to the deepest level of the tree first and then works up while a breadth-first search looks at all possible paths at the same depth before it goes to a deeper level. When we come to a dead end in a depth-first search, we back up as *little* as possible. We try another route from a recent vertex-the route on top of our stack. In a breadth-first search, we want to back up as *far* as possible to find a route originating from the earliest vertices. So the stack is not an appropriate structure for finding an early route because it keeps track of things in the order opposite of their occurrence-the latest route is on top. To keep track of things in the order in which they happened, we use a FIFO queue. The route at the front of the queue is a route from an earlier vertex; the route at the back of the queue is from a later vertex.

Q.226 Consider the following undirected graph and answer the following questions.



Assume that the edges are ordered alphabetically (i.e. when facing with alternatives, choose the edges in alphabetical order)

(i) List the nodes (cities) of the graph by depth first search starting from **Varanasi.**

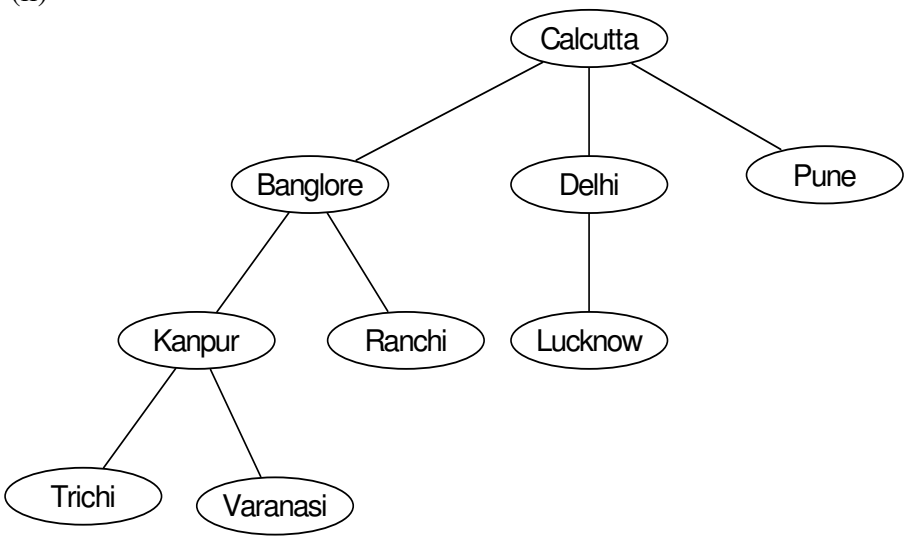
(ii) List the nodes (cities) of the graph by breadth first search starting from **Calcutta**. **(8)**

Ans:

(i) The list of nodes from the graph by performing Depth-First-Search starting from Varanasi is as follows

Varanasi → Kanpur → Bangalore → Calcutta → Delhi → Pune → Ranchi → Lucknow→Trichi

(ii)



The list of nodes from the graph by performing Breadth-First-Search starting from Calcutta is as follows

Calcutta → Bangalore → Delhi → Pune → Kanpur → Ranchi → Lucknow →Trichi →Varanasi

Q.227 Consider the following function

F(int n, int m)

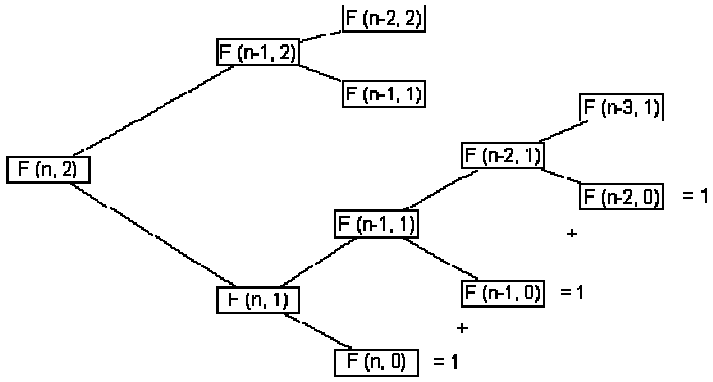
{if n < = 0 OR m<= 0 then return 1 else return (F(n-1, m) + F(n, m-1));}

Now answer the following questions assuming that n and m are positive integers.

(i) What is the value of F(n, 2), F(5, m), F(3, 2), F(n, m)?

(ii) How many recursive calls are made to the function F, including the original call when evaluating F(n, m)? **(4)**

Ans:



The recursion continues till every function call gets its absolute value. Order is

$$\frac{n\,(n-1)}{2} = O(n)$$

(i) The value of
   a. F (n,2) = O(n)
   b. F (5, m) = O (m)
   c. F (3, 2) = O (1).
   d. F (n, m) = O (n). O(m) = O(n.m).

(ii) While evaluating F(n, m) , since it is of order O (n, m) so the number of recursive calls required is of the order n x m.

Q.228 Write a recursive function that has one parameter which is an integer value called x. The function prints x asterisks followed by x exclamation points. Do not use any loops. Do not use any variables other than x. **(4)**

Ans:

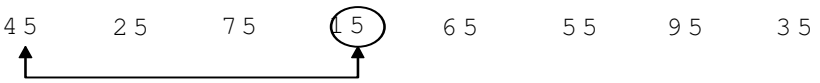
The recursive function printast () for the required task is as follows:-

```
void printast (int a)
{
 if (a > 0)
 {
        printf ("*");
        a--;
        printast (a);
 }
 if (a!=0)
        printf ("!");
}
```

Q.229 Sort the following list using selection Sort. Show each pass of the sort.
45, 25, 75, 15, 65, 55, 95, 35 **(7)**

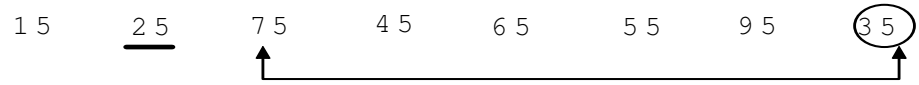
Ans:

The original array is 45, 25, 75, 15, 65, 55, 95, 35



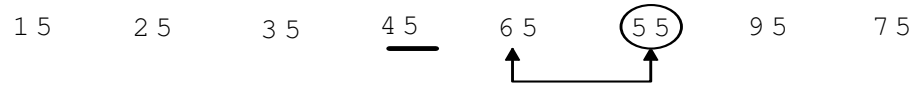
After Pass 1

After Pass 2



After Pass 3

After Pass 4



After Pass 5

After Pass 6

After Pass 7

The array is sorted after 7 passes of selection sort.

Q.230 What is Hashing? Can a perfect Hash function be made? Justify your answer. Explain in brief the various methods used to resolve collision. **(9)**

Ans:

Hashing : Hashing provides the direct access of record from the file no matter where the record is in the file. This is possible with the help of a hashing

function H which map the key with the corresponding key address or location. It provides the key-to-address transformation.

No a perfect hash function cannot be made because hashing is not perfect.

Occasionally, a collision occurs when two different keys hash into the same hash value and are assigned to the same array element. Programmers have come up with various techniques for dealing with this conflict.Two broad classes of collision resolution techniques are: open addressing and chaining.

Open addressing: The simplest way to resolve a collision is to start with the hash address and do a sequential search through the table for an empty location. The idea is to place the record in the next available position in the array. This method is called linear probing. An empty record is indicated by a special value called null. The major drawback of the linear probe method is clustering.

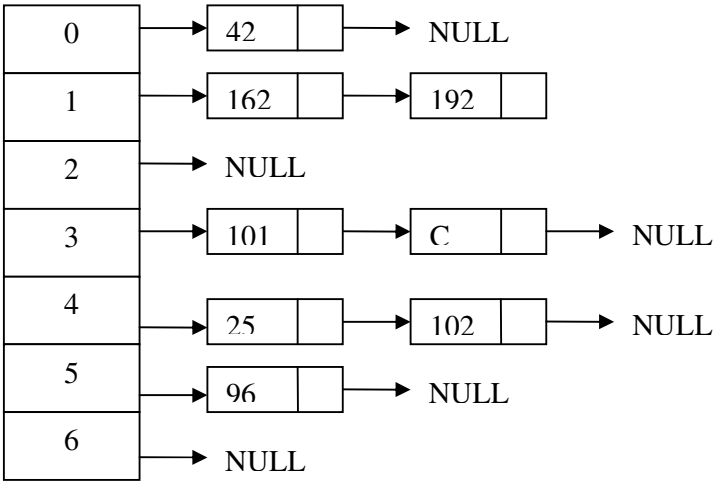
Chaining: In this technique, instead of hashing function value as location we use it as an index into an array of pointers. Each pointer access a chain that holds the element having same location.

Because two entries cannot be assigned the same array element, the programmer creates a linked list. The first user-defined structure is assigned to the pointer in the array element. The second isn't assigned to any array element and is instead linked to the first user-defined structure, thereby forming a linked list.

For example: in a table size of 7

42, 56 both are mapped to index 0 as 42%7=0 and 56%7=0.

25, 42, 96, 101, 102, 162, 197 can be mapped as follows:



Q.231 Define a B tree of order m.
Write algorithms to
(i) Search for a key in B-tree.
(ii) Insert a key in a B-tree.
(iii) Delete a key from a B-tree. **(10)**

Ans:

B Tree of order m

A balanced multiway search tree of order m in which each nonroot node contains at least m/2 keys is called a B-Tree of order m. where order means maximum number of sub-trees.

A B-Tree of order m is either the empty tree or it is an m-way search tree T with the following properties:

(i) The root of T has at least two subtrees and at most m subtrees.
(ii) All internal nodes of T (other than its root) have between [m / 2] and m subtrees.
(iii) All external nodes of T are at the same level.

Algorithm to search for a key in B-tree is as follows:

B-tree search makes an n-way choice. By performing the linear search in a node, correct child Ci of that node is chosen. Once the value is greater than or equal to the desired value is obtained, the child pointer to the immediate left of the value is followed. Otherwise rightmost child pointer is followed. If desired value is obtained, the search is terminated.

B-tree search (x, k) *This function searches the key from the given B-tree*

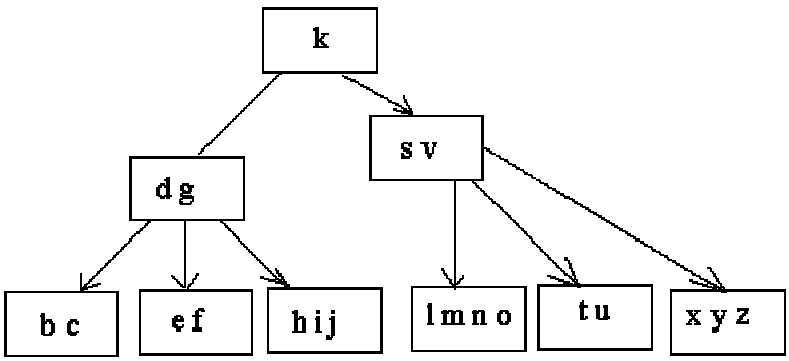
The Disk_Read operation indicates that all references to a given node be preceded by a read operation.

1. i ← 1
2. While ($i \leq n[x]$ and $k > key_i [x]$)
   Set i ← i + 1
3. If ($i <= n[x]$ and $k = key_i [x]$) then
   { return (x, i) }
4. If (leaf [x]) then
   return NIL
   else Disk_read ($c_i [x]$)
   return B-tree search (ci [x], k)

Algorithm to insert a key in B-tree is as follows:

1. First search is done for the place where the new record must be put. As the keys are inserted, they are sorted into the proper order.
2. If the node can accommodate the new record, insert the new record at the appropriate pointer so that number of pointers remains one more than the number of records.
3. If the node overflows because there is an upper bound on the size of a node, splitting is required. The node is split into three parts; the middle record is passed upward and inserted into parent, leaving two children behind. If n is odd (n-1 keys in full node and the new target key), median key int(n/2)+1 is placed in parent node, the lower n/2 keys are put in the left leaf and the higher n/2 keys are put in the right leaf. If n is even, we may have left biased or right biased means one key may be more in left child or right child respectively.
4. Splitting may propagate up the tree because the parent, into which splitted record is added, may overflow then it may be split. If the root is required to be split, a new record is created with just two children.

Q.232 Given the following tree

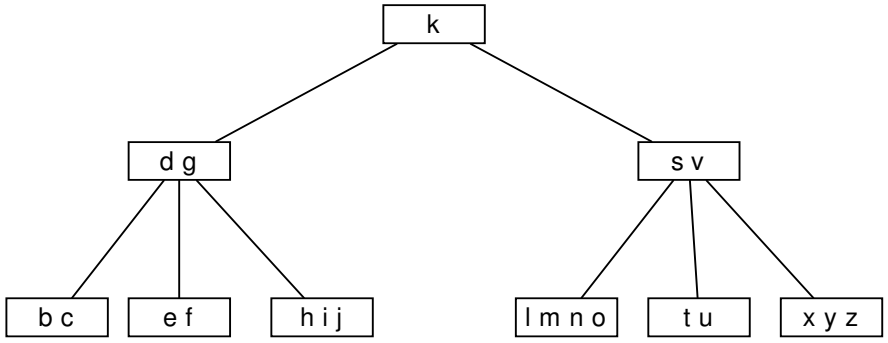


Show how the tree will change when the following steps are executed one after another

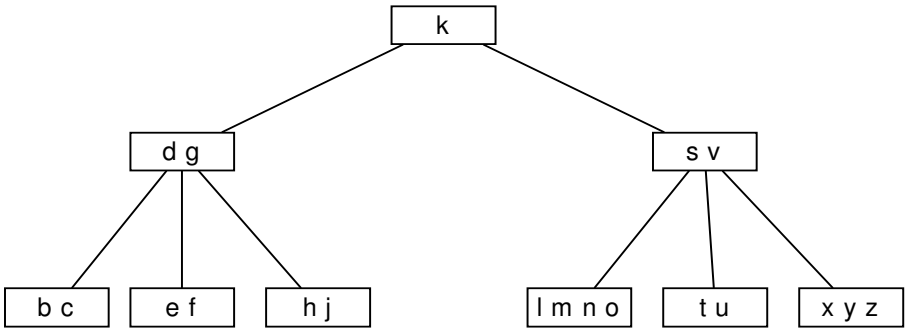
(i) Key 'i' is deleted
(ii) Key 'v' is deleted
(iii) Key 't' is deleted **(6)**

Ans:

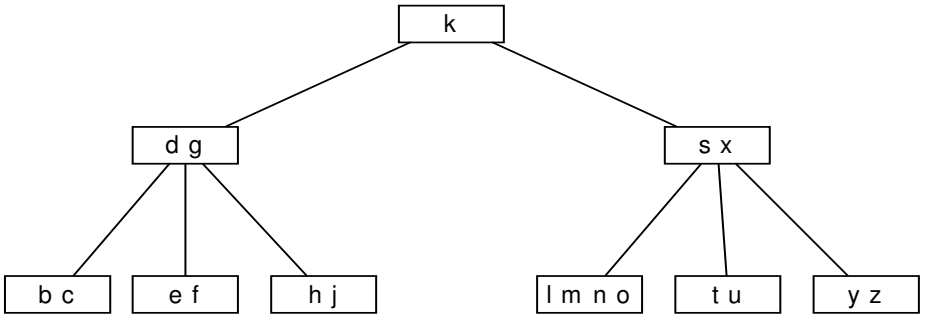
The given tree is:-



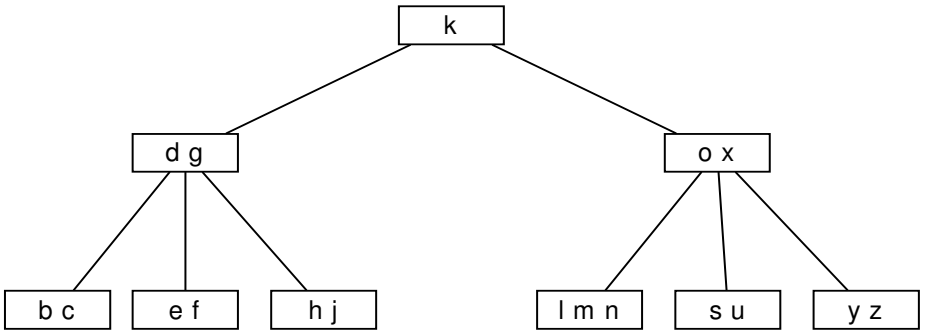
(i) **Key 'i' is deleted**



(ii) **Key 'v' is deleted**



(iii) **Key 't' is deleted**



Q.233 Write an algorithm to reverse a singly linked list. **(4)**

Ans:

The algorithm to reverse a singly link list is as follows:-

```
reverse (struct node **st)
{
      struct node *p, *q, *r;
      p = *st;
      q = NULL;
      while (p != NULL)
      {
        r =q;
        q = p;
        p = p → link;
        q → link = r;
      }
      *st = q;
}
```

Q.234 Consider a 2D array as char A[5] [6] stored in contiguous memory locations starting from location 1000" in column major order. What would the address of a[i] [d]? **(2)**

Ans:

Considering the given char array A[5][6] stored in contiguous memory in column major order; the address of a[i] [d] would be calculated as
base (a) + (d*n1 +i) * size ; where base(a) =1000 (given).
Since it is a char array so size=1 and n1 is the total no of rows
here n1=6 (0,1,2,3,4,5,) Substituting values we get the address of

A[i] [d] = 1000 + d*6 + i

Q.235 What are priority Queues? How can priority queues be implemented? Explain in brief. **(4)**

Ans:

Priority Queues:-

There are some queues in which we can insert items or delete items from any position based on some property. Now, those queues based on the property of priority of the task to be processed are referred to as **priority queues**.

To implement a priority queue we first need to identify different levels of priority and categorize them as the least priority through the most priority level. After having identified the different levels of priority required, we then need to classify each incoming item (job) as to what priority level should be assigned to it. While creating the queues, we need to create n queues if there are n priority levels defined initially. Let suppose there are three priority levels identified as P1, P2, and P3. In this case we would create three queues as Q1, Q2 and Q3 each representing priority levels P1, P2, and P3 respectively. Now every job would have a priority level assigned to them. Jobs with priority level as P1 will be inserted in Q1. Jobs with priority level P2 are inserted in Q2 and so on.

Each queue will follow FIFO behavior strictly. Jobs are always removed from the front end of any queue. Elements in the queue Q2 are removed only when Q1 is

empty and the elements of the queue Q3 are only removed when Q2 is empty and so on.

Q.236 Consider a linked list with n integers. Each node of the list is numbered from '1' to 'n'. Write an algorithm to split this list into 4 lists so that
first list contains nodes numbered 1,5, 9, 13- - -
second list contains nodes numbered 2, 6, 10, 14- - -
third list contains nodes numbered 3, 7, 11, 15- - -
and fourth list contains nodes numbered 4,8, 12, 16- - - **(8)**

Ans:

Algorithm for the above said task is as follows: -

Let us denote the pointer to an item in the original linked list as P
Then the data and consequent pointers will be denoted as
P->data and P->next respectively.

We assume that the functions list1.add (x), list2.add (x), list3.add (x) and list4.add (x) is defined to insert the "item x" into the linked list list1, list2, list3 and list4 respectively.

```
start
set i = 1
while ( P->next != NULL )
 Do
 If (i % 4 = 1)
 List1.add (P->data)
 Else if (i % 4 = 2)
        List2.add (P->data)
        Else if (i % 4 = 3)
               List3.add (P->data)
        Else
               List4.add (P->data)
 P = P->next //increment P to point to its next location
 i++ // increment i to count the next node in sequence.
 Done
 End
```

Q.237 Write a recursive algorithm to find Greatest Common Divisor of two integers with the help of Euclid's algorithm, as per the following formula

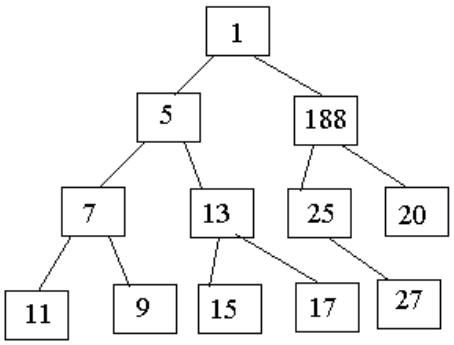
$$GCD(n,m) = \begin{cases} GCD(m,n) \text{ if } n < m \\ m \text{ if } n \geq m \text{ and } n \bmod m = 0 \\ GCD(m, n \bmod m) \text{ otherwise} \end{cases} \quad \textbf{(4)}$$

Ans:

Recursive algorithm to find the GCD of two numbers using Euclid's algorithm is as follows;

```
int gcd (n, m)
{
 if (n < m)
        return (gcd (m, n));
 else if ( n >= m  &&  n % m == 0)
        return (m);
 else
        return (gcd (m, n % m));
}
```

Q.238 Traverse the following binary tree in inorder, preorder and postorder. **(3)**



Ans:

The binary tree traversal is as follows:-

Preorder
1,5,7,11,9,13,15,17,188,25,27,20
Inorder
11,7,9,5,15,13,17,1,25,27,188,20
Postorder
11,9,7,15,17,13,5,27,25,20,188,1

Q.239 Define the following:

(i) Tree (recursive defination)
(ii) Level of a node.
(iii) Height of a tree.
(iv) Complete Binary tree.
(v) Internal Path length. **(2+1+1+2+1)**

Ans:

(i) **Tree (recursive definition)**

A tree is a finite set of one or more nodes such that.
(1) There is a specially designated node called the root.
(2) The remaining nodes are partitioned into n>=0 disjoint sets T1, T2 ... Tn where each of these sets is a tree. T1, T2 ... Tn are called the subtree of the root.

(ii) **Level of a node**

The root is at level 0(zero) and the level of any node is 1 more than the level of its parent.

(iii) **Height of a tree**

The length of the longest path from root to any node is known as the height of the tree.

(iv) **Complete Binary tree**

A complete binary tree can be defined as a binary tree whose non leaf nodes have nonempty left and right sub tree and all leaves are at the same level.

(v) **Internal Path length**

It is defined as the number of node traversed while moving through one particular node to any other node in the tree.

Q.240 What is an AVL tree? Explain how a node can be inserted into an AVL tree. **(6)**

Ans:

AVL Tree

An AVL tree is a binary tree in which the difference in heights between the left and the right subtree is not more than one for every node.

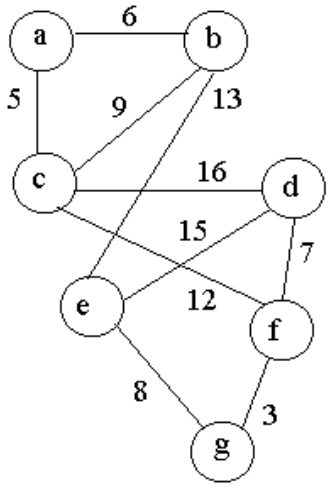
We can insert a node into an AVL tree by using the insertion algorithm for binary search trees in which we compare the key of the new node with that in the root and then insert the new node into the left subtree or right subtree depending on whether it is less than or greater than that in the root. Insertion may or may not result in change in the height of the tree. While inserting we must take care that the balance factor of any node not changes to values other than 0, 1 and -1. A tree becomes unbalanced if and only if

(i) The newly inserted node is a left descendant of a node that previously had a balance of 1.

(ii) The newly inserted node is a right descendent of a node that previously had a balance of -1.

If the balance factor of any node changes during insertion than one of the subtree is rotated one or more times to ensure that the balance factor is restored to correct values.

Q.241 Describe Kruskal's algorithm to find minimum spanning trees.



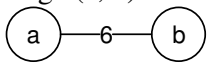
Apply Kruskal's algorithm for the above graph. **(10)**

Ans:

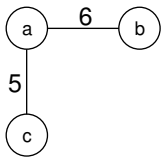
Applying Kruskal's Algorithm starting from the node a

Step 1: Start from node a, Add to tree T

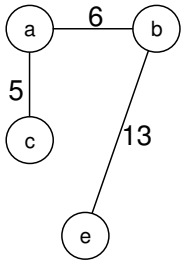
Step 2: Edge (a, b) cost=6 add to T



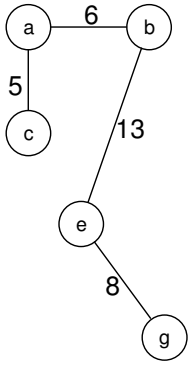
Step 3: Edge (a, c) cost=5 add to T



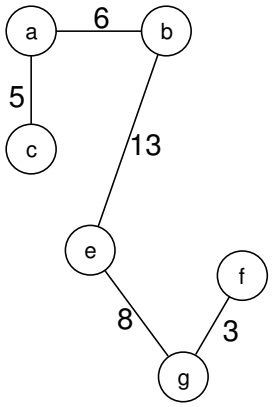
Step 4: Edge (b,e) cost=13 add to T



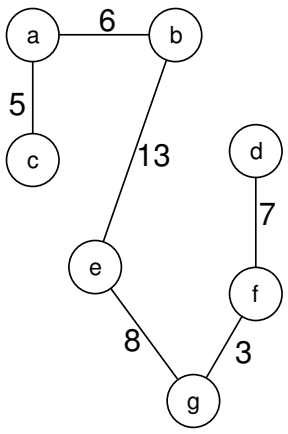
Step 5: Edge (e, g) cost=8 add to T



Step 6: Edge (g, f) cost=3 add to T



Step 7: Edge (f, d) cost=7 add to T



This is the minimum spanning tree of the given graph.

Q.242 Differentiate between
(i) Top-down and Bottom-up programming?
(ii) Structured and Modular programming. **(6)**

Ans:

(i) Top-down and Bottom up programming.

Top down method is also known as step-wise refinement. Here the problem is first divided into main tasks. Each of these is divided into subtasks and so on. These subtasks should more precisely describe how the final goal is to be reached. Whereas the Bottom-up programming is just the opposite of the top-down approach. In this technique, all small related tasks are grouped into a major task and the main task becomes the main program.

(ii) Structured and Modular programming.

Structured programming means the collection of principles and practices that are directed toward developing correct programs which are easy to maintain and easy to understand. It should read like a sequence from beginning to end instead of branching from later statements to earlier ones and back again. Whereas in modular programming the modularity of a system can be represented by a hierarchical structure which has a single main module at level 1 and gives a brief description of the system. At level 2 the main module refers to a number of sub program modules the give a more detailed description of the system and so on. An important aspect in this hierarchical structuring process is the desire to understand a module at a certain level independently of all other modules at the same level.

Q.243 How is recursion handled internally. Explain with the help of a suitable example. **(4)**

Ans:

Internally, each recursive call to a function needs to store the intermediate values of the parameters and local variables in a run time stack. The general algorithm for any recursive procedure contains the following steps:

1. Save the parameters, local variables and return address.

2. If the base criterion has been reached, then perform the final computation and go to step 3, otherwise perform the partial computation and go to step 1 with reduced parameter values (initiate a recursive call).

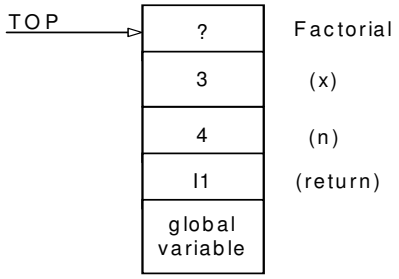
3. Restore the most recently saved parameters, local variables and return address. Go to this return address.

All parameters is accessed by their positions relative to the top of the stack. Whenever a subroutine is started the variables are pushed onto the stack and whenever it completes execution, the variables are popped off the stack. For ex. consider the factorial function.

```
Factorial (int n)
                {
  int x;
  if n = 0
         return (1);
  x = n - 1;
  return (n * factorial (x));
}
```

let the initial call is Y = factorial (4)

At first time, 4 locations are put in the run time stack



And on each subsequent calls the intermediate values of all these variables are pushed till the condition reaches where n=0. and then the contents are popped one by one and is returned to the place of the precious call. This continues until the control is back to the first call.

Q.244 Write an algorithm to add two polynomials using linked list. **(12)**

Ans:

Algorithm to add two polynomials using linked list is as follows:-

Let p and q be the two polynomials represented by linked lists

```
1. while p and q are not null, repeat step 2.
2. If powers of the two terms ate equal
       then if the terms do not cancel
       then insert the sum of the terms into the sum Polynomial
                    Advance p
                    Advance q
   Else if the power of the first polynomial> power of second
   Then insert the term from first polynomial into sum polynomial
                    Advance p
   Else insert the term from second polynomial into sum polynomial
                    Advance q
3. copy the remaining terms from the non empty polynomial into the
   sum polynomial.
```

The third step of the algorithm is to be processed till the end of the polynomials has not been reached.

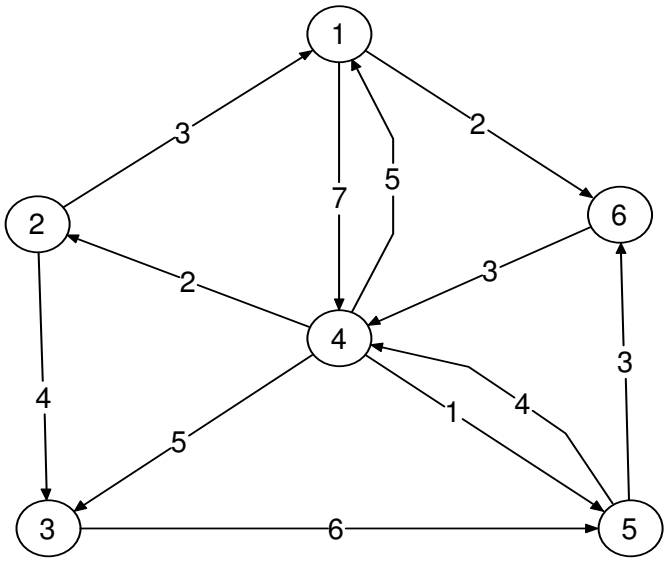
Q.245 Execute Dijkstra's algorithm on the following graph with vertices numbered 1 to 6. The graph is given in the form of an adjacency list.

i.	: (6, 2), (4, 7)
ii.	: (1,3), (3,4)
iii.	: (5,6)
iv.	: (2,2) (3,5), (1,5), (5,1)
v.	: (4,4), (6,3)
vi.	: (4,3)

Here (3,4) in vertex 2's adjacency list means that vertex 2 has a directed edge to vertex 3 with a weight of 4. Assuming the source vertex to be 2, find the shortest distance and the path to all the remaining vertices. **(16)**

Ans:

The source vertex is 2 (given).



Step 1

P=Q T = {1, 2, 3, 4, 5, 6}
L(2) = 0Starting Vertex
L(x) = ∞ for all x Є T, x≠2

Step 2

V=2 (vertex 2 has smallest label)
P = {2} T = {1, 3, 4, 5, 6}
Calculate new labels for all vertices of T.
L (1) = min (∞, 0+3) = 3
L (3) = min (∞, 0+4) = 4
L (4) = min (∞, 0+∞) = ∞
L (5) = ∞
L (6) = ∞

Step 3

V=1 (Permanent label L (1) = 3)
P = {2, 1} T = {3, 4, 5, 6}
Calculate new labels for all vertices of T.

L (3) = min (4, 3+∞) = 4
L (4) = min (∞, 3+7) = 10
L (5) = min (∞, 3+∞) =∞
L (6) = min (∞. 3+2) = 5

Step 4

V=3 (Permanent label L (3) = 4)
P = {2, 1, 3} T = {4, 5, 6}
Calculate new labels for all vertices of T.
L (4) = min (10, 4+∞) = 10
L (5) = min (∞, 4+6) = 10
L (6) = min (5. 4+∞) = 5

Step 5

V=6 (Permanent label L (6) = 5)
P = {2, 1, 3, 6} T = {4, 5}
Calculate new labels for all vertices of T.
L (4) = min (10, 5+3) = 8
L (5) = min (10, 5+∞) = 10

Step 6

V=4 (Permanent label L (4) = 8)
P = {2, 1, 3, 6, 4} T = {5}
Calculate new labels for all vertices of T.
L (5) = min (10, 8+1) = 9

The vertex 5 got its permanent label and all the vertices got their respective permanent labels as shown in the table below.

Vertex (v)	Permanent Label L (v)
1	3
3	4
4	8
5	9
6	5

Now, assuming the source vertex to be 2, the shortest distance and the path to all remaining vertices are as follows: -

Vertex 1

Shortest distance = 3
Path is 2→1

Vertex 3

Shortest distance = 4
Path is 2 → 3

Vertex 4

Shortest distance = 8
Path is 2→1 → 6→4

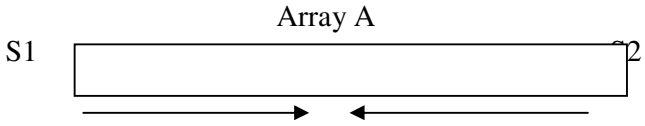
Vertex 5

Shortest distance = 9
Path is 2→1→ 6→4→5

Vertex 6

Shortest distance = 5
Path is 2→1→ 6

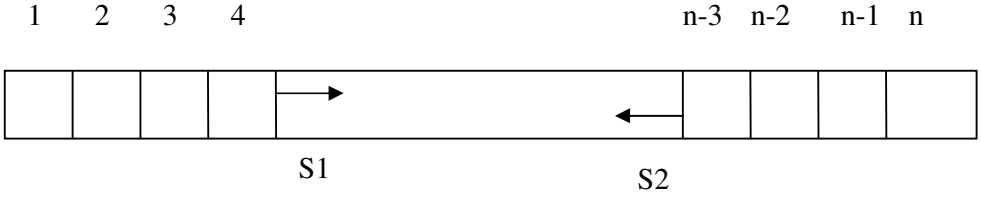
Q.246 Two stacks are implemented using an array. What should be the initial values of top for the two stacks? Arrows show the direction of growth of the stack.



How can we implement m stacks in a contiguous a memory. Explain how the stacks will grow and indicate the boundary condition. Write push and pop programs for such a scenario. **(10)**

Ans:

Two stacks s1 and s2 can be implemented in one array A [1, 2... N] as shown in the following figure



We define A[1] as the bottom of stack S1 and let S1 "grow" to the right and we define A[n] as the bottom of the stack S2 and S2 "grow" to the left. In this case, overflow will occur only S1 and S2 together have more than n elements. This technique will usually decrease the number of times overflow occurs. There will separate push1, push2, pop1 and pop2 functions to be defined separately for two stacks S1 and S2. These will be defined as follows:-

```
initial volume of top for
stack S1 is top1=0 and
for stack S2 is top 2=n+1
S1. Push 1(x)
{
if (top 1+1==top2)
print if ("Stack S1 is overflowing")
Else
{
top 1++
A[top1]= x
}
}
S1.POP()
{
if (top1==0)
print if ("stack S1 is empty")
return (-1)
}
return (A[top--])
}
S2.push2(x)
}
if (top1+1==top2)
print if ("S2 is over flowing")
else
{
top2–
A [top2]=x
```

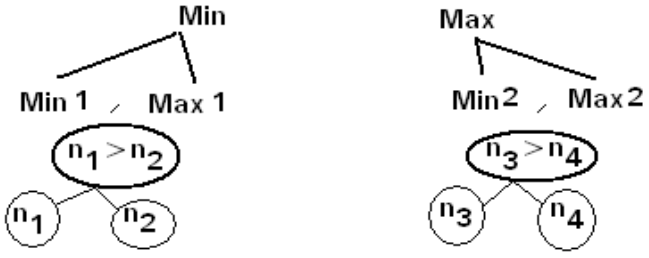
```
}
}
S2.POP2()
{
if (top2==n+1)
}
print if ("stack is empty")
return(-1)
}
else
return( A[top2++])
}
```

Q.247 How can we modify almost any algorithm to have a good best case running time? **(4)**

Ans: Guess an answer
Verify that it is correct, in that case stop.
Otherwise run the original algorithm.
OR
Check whether the input constitutes an input at the very beginning
Otherwise run the original algorithm.

Q.248 Sketch an algorithm to find the minimum and the maximum elements from a sequence of n elements. Your algorithm should not take more than (3n/2)-2 number of comparisons where n is a power of 2. **(6)**

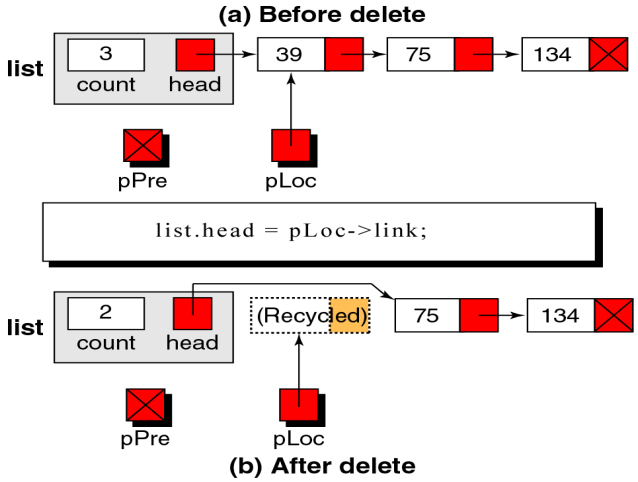
Ans:
Rather than processing each element of the input by comparing it against the correct minimum and maximum, at the cost of 2 comparisons per element, we process elements in pairs. We compare pairs of elements from the input first with each other, and then we compare the smaller to the current minimum and the larger to the current maximum, at the cost of 3 comparisons for every 3 elements.
If n is off, we set both the minimum and maximum to the value of first element and the process the rest of the elements in pairs.
If n is even, we perform 1 comparison on the first 2 elements to determine the initial values of minimum and maximum, and the process the rest of the elements in pairs.So if n is odd, we perform 3[n/2] comparisons.
If n is even, we perform one initial comparison followed by 3(n-2)/2 comparisons for a total of 3n/2-2. Thus in either, number of comparisons are almost 3[n/2]



Q.249 Write an O(1) algorithm to delete a node p in a singly linked list. Can we use this algorithm to delete every node? Justify. **(7)**

Ans:

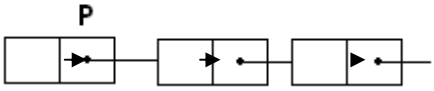
One deletion operation consists in deleting a node at the beginning of the list and returning the value stored in it. This operation is implemented by a member function deleteFromHead() given below. In this operation, the information from the first node is temporarily stored in local variable and then head is reset so that what was the second node becomes the first node. In this way, the former first node can be deleted in constant O(1) time.



Deleting the first node, from a linked list pointed by 'head' can be done is O(i) time

head = head →next

Deleting a note that follows a node with address P can also be done in O(i) time



P →next = (P →next) → next

But a node with address P cannot be deleted in O(i) time. It takes linear time.

```
//Delete the first node
deleteFromHead(struct node *q,int num)
{
        struct node *temp;
        temp=*q;
        *q=temp→link;
        free(temp);
}
```

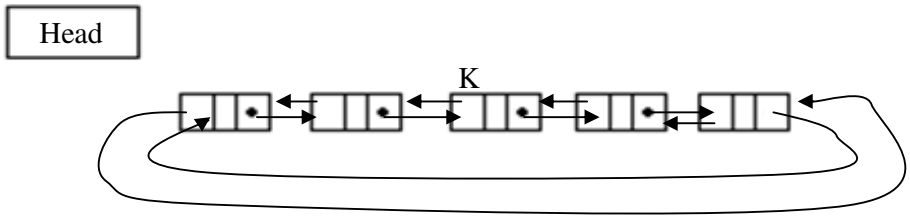
We can use this algorithm to delete every node by calling it recursively. The best case is when the head node is to be deleted, which takes O(1) time to accomplish. The worst case is when the last node needs to be deleted because in that case it will take O(n) performance.

Q.250 Suggest an efficient sorting algorithm to sort an array of n large records, while minimizing number of exchanges. **(2)**

Ans:

Selection sort

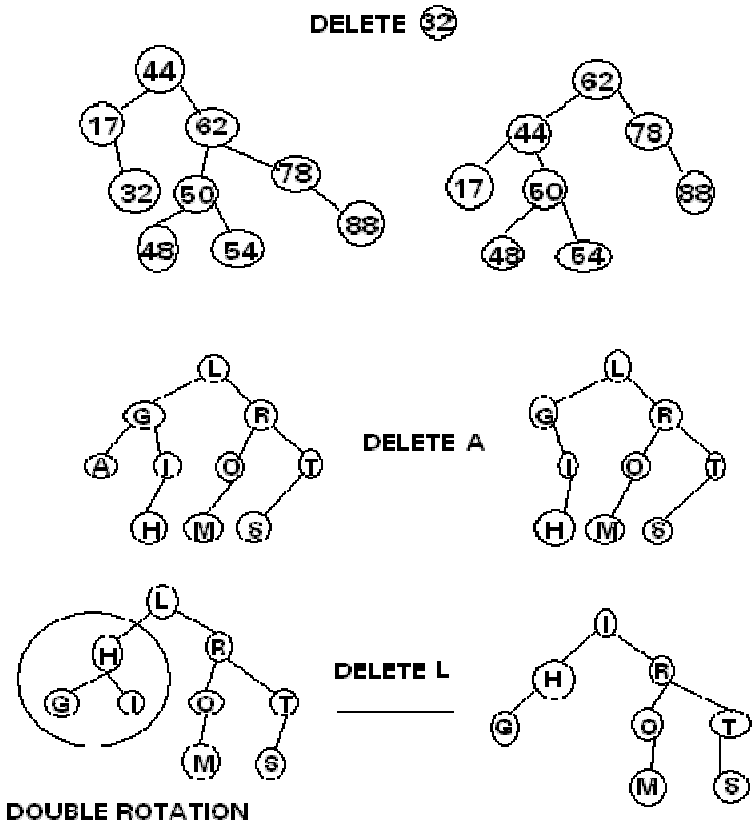
In all cases: $n^2/2$ comparisons and n exchanges execution time is not sensitive to original ordering of Input data.



Suppose K is the node after which the list is to be split. Then
head2 = k →front
k→front = head
head → back = k
P = head2
while (P→front:=head)
P=P→ front
P→front=head2
head2→back=P

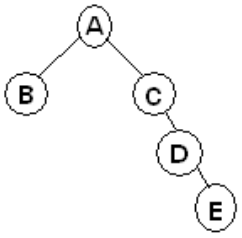
Q.251 Give an AVL tree for which the deletion of a node requires two double rotations. Draw the tree and explain why two rotations are needed? **(10)**

Ans:



Q.252 A funny tree is a binary tree such that, for each of its nodes x, the number of nodes in each sub tree of x is at most 2/3 the number of nodes in the tree rooted at x. Draw the tallest funny tree of 5 nodes. **(7)**

Ans:



Q.253 Explain the term step-wise refinement. (3)

Ans:

Step **Wise Refinement**

Refinement is a process of elaboration. Here one begins with a statement of function that is defined at a high-level abstraction. That is the statement describes the program/function conceptually but provides no information about internal working. Refinement causes the programmer to elaborate on the original statement providing more and more detail. In a stepwise refinement, at each step of refinement one or several instructions of the given program are decomposed into more detailed instructions. The successive refinement terminates when all the instructions are expressed in terms of atomic expressions of the language.

Q.254 What is the difference between top-down and bottom-up, programming? (4)

Ans: Top-down and Bottom-up Approaches

In Top-down programming approach, we start by identifying the major modules of the program, decomposing them into their lower modules and iterating until the desired level of detail is achieved. This is also termed as Step Wise Refinement; starting from an abstract design, in each step the design is refined to a more concrete level until we reach a level where no more refinement is needed and the design can be implemented directly, whereas in Bottom-up programming approach, we identify modules that are required by programs and decide how to combine these modules to provide larger ones; to combine those to provide even larger ones, and so on, till we arrive at one big module which is the whole of the desired program.

In Bottom-up approach, we need to use a lot of intuition to decide exactly what functionality a module should provide. This is not the case in Top-down approach.

Q.255 What do you mean by complexity of an algorithm? Derive the asymptotic time complexity of a non recursive, binary search algorithm. (7)

Ans:

The term complexity is used to describe the performance of an algorithm. Typically performance is measured in terms of time or space. Space complexity of an algorithm is the amount of memory is needed to run the algorithm. Time complexity of an algorithm is the amount of computer time it needs to run the algorithm. Most

common notation for describing time complexity 0(f(n)), where n is the input size and f is a function of n. An algorithm ~ 0(f(n)) means there exists N such that for all n > N the run time of the algorithm is less than c.f(n), where c is a constant.

Binary search can be applied on a sorted array to search for a particular item. Given array positions A[l], the search is conducted in the following way.

```
Binsearch (x, l,n)
{
L=l;
U = n;
While (L <= U)
{ mid = (L + U) / 2;
If (A[mid] = x)
Return.True;
        Else
If (A[mid ] > x)
L = mid + 1;
    Else
U = mid - 1;
  }
return False;
  }
```

Thus at each state the array is halved into two equal parts. In the worst case the algorithm terminates when the portion of the array considered is of length 1, Thus if n = 2 , in the worst case the while loop will have to repeat k times. Where k = log n. Thus asymptotic complexity is O(log n).

Q.256 Assume the declaration of multidimensional arrays A and B to be,
A (-2:2, 2:22) and B (1:8, -5:5, -10:5)
(i) Find the length of each dimension and the number of elements in A and B. **(3)**
(ii) Consider the element B[3,3,3] in B. Find the effective indices E_1, E_2, E_3 and the address of the element, assuming Base (B) = 400 and there are W = 4 words per memory location. **(4)**

Ans:
(i) A (-2:2,2:22), B (1:8, -5:5, -10:5) length of 1st dimension in A= 2-(-2)+1
= 2+2+1=5
length of 2nd dimension **in A** =22-2+1=21
LI = length of 1st dimension in B=8-l+l=8
L2 = length of 2nd dimension in B= 5-(-5)+1=11 L3= length of 3rd dimension in B=5-(-10)+l=16 No. of elements in *A*= 5*21=105
No. of elements in B=8*11*16=1408

(ii) B[3,3,3], Base(B)=400, W=4
El= 3-1=2
E2=3-(-5)=8
E3=3-(-10)=13
Address of B[3,3,3]=400+4(493)
=2372

E1L2=2*11=22
E1L2+E2=22+8=30 (E1L2+E2)L3=30*16=480
(E1L2+E2)L3+E3=480+13= 493

Q.257. (i) what is meant by the terms 'row-major order' and 'column-major order'? (2)
(ii) Can the size of an array be declared at runtime? (2)
(iii) The array DATA [10, 15] is stored in memory in 'row - major order'. If base address is 200 and element size is 1. Calculate the address of element DATA [7, 12]. (3)

Ans: (i) Storing the array column by column is known as column-major order and storing the array row by row is known as row-major-order. The particular order used depends upon the programming language, not the user. For example consider array A (3,4) and how this array will be stored in memory in both the cases is shown below

	(1,1)			(1,1)	
	(2,1)	Column1		(1,2)	Row1
	(3,1)			(1,3)	
	(1,2)			(1,4)	
	(2,2)	Column2		(2,1)	Row2
	(3,2)			(2,2)	
	(1,3)			(2,3)	
	(2,3)	Column3		(2,4)	
	(3,3)			(3,1)	Row3
	(1,4)			(3,2)	
	(2,4)	Column4		(3,3)	
	(3,4)			(3,4)	

(ii) No, the size of an array can't be declared at run time, we always need to mention the dimensions of an array at the time of writing program i.e before run time
(iii) Base address=200
Element Size=l
Address of element DATA [7,12]
= 200+[(7-l)*15+(12-l)]*l
= 200+[6*15+11]
= 200+[90+11]=301
Address of DATA [7,12]=301

Q.258 Write an algorithm to insert a node after a given node in a linear linked list. **(7)**

Ans:
Suppose we are given a value of LOC where LOC is the location of the node 'A' in the linked list.
The following algorithm inserts an 'ITEM' into LIST (given Linked list) so that 'TTEM' follows **node** 'A'.

```
1. If AVAIL = NULL , Then write overflow and exit
2. set NEW = AVAIL and AVAIL=:link[AVAIL] [remove first node from
AVAIL LIST]
3. set INFO[NEW]=ITEM [copies new data into new node]
4. if LOC=NULL then [insert as first node]
     set LINK[NEW]= START and START = NEW

     else [insert after node with

     location LOC]

     set LINK[NEW]=LINK[LOC] and LINK[LOC]=NEW

     [end of if structure]
5.    exit
```

Q.259 Show how the following polynomial can be represented using a linked list. **(4)**

$$7x^2y^2-4x^2y+5xy^2-2$$

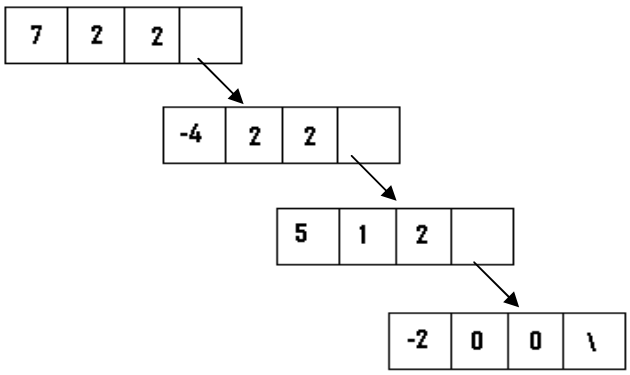
Ans:

Representation of Polynomial using Linked List

Each node will have four parts(as shown below)

Coeff	Power of'x'	Power of 'y'	Next address

The polynomial is



Q.260. What is the advantage of doubly linked list over singly linked list? **(3)**

Ans:

Advantages of the doubly linked list over singly linked list

1 A doubly linked list can be traversed in two directions; in the usual forward direction from the beginning of the list to the end, or in the backward direction from the end of the list to the beginning of the list.

2 Given the location of a node 'N' in the list, one can have immediate access to both the next node and the preceding node in the list.

3 Given a pointer to a particular node 'N', in a doubly linked list, we can delete the Node 'N' without traversing any part of the list. Similarly, insertion can also be made before or after 'N' without traversing the list.

Q.261 What is a binary tree? Write an algorithm for the preorder traversal of a binary tree using stacks. , (7)

Ans:

A binary tree 'T' is defined as

A finite set of elements, called nodes, such that: either (i) or (ii) happens:

(i) T is empty (called the 'null tree' or 'empty tree')

(ii) T contains a distinguished node R, called the 'root' of T and the remaining nodes of 'T' form an ordered pair of the disjoint binary tree Tl and T2.

If 'T' contains a root 'R' then the two trees Tl and T2 are called, the 'left' and 'right' sub trees of R**,** respectively.

Algorithm for the pre order traversal of a binary tree using a stack

A binary tree T is in memory, an array 'STACK' is used to temporarily hold the addresses of the nodes. "**PROCESS**" is the operation that will be applied to each node of the tree.

```
(1) Set TOP=1, STACK [1] =
 NULL and PTR= ROOT
 [initially push null onto
 Stack and initialize PTR]
(2) repeat steps (3) to (5) while PTR != NULL
(3) apply PROCESS to INFO[PTR]
(4)  [right child ?]
if RIGHT[PTR] != NULL then[push on STACK]
Set TOP=TOP+1 band STACK[TOP]=RIGHT[PTR]
[End of IF Structure]
(5)  [Left Child ?]
if LEFT[PTR] != NULL then:
set PTR= LEFT[PTR]
else [POP from STACK]
set PTR=STACK[TOP] and TOP=TOP+1
[End of IF structure]
(6)  Exit
```

Q.262. A binary tree T has 9 nodes. The in order and preorder traversals of T yield the following sequences of nodes:

In Order: EACKFHDBG

Pre order:FAEKCDHGB

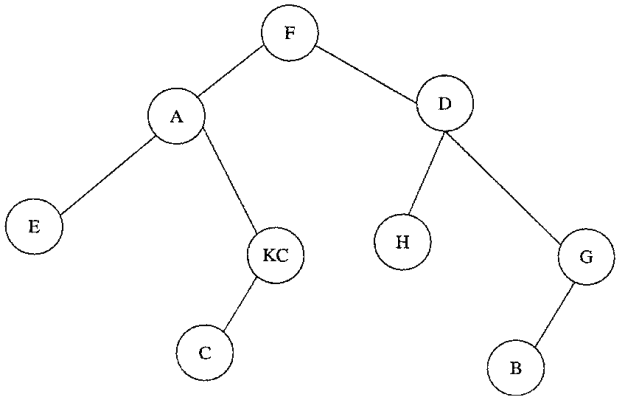
Draw the tree T. Also give the yield of the post order traversal. (7)

Ans:

Inorder :EACKFHDBG

Preocder : E A E K C D H G B

Tree 'T' is



Postorder Traversal **-.ECKAHBGDF**

Q.263 Write short notes on the following:

(i) Threaded binary tree.
(ii) Buddy systems.
(iii) B - trees.
(iv) Minimum spanning tree. (14)

Ans:

(i) **Threaded binary tree** : Consider the linked representation of a binary tree 'T. Approximately half of the entries is the pointer fields 'LEFT' and 'RIGHT' will contain null elements. This space may be more efficiently used by replacing the null entries by some other type of information. Specially we will replace certain null entries by special pointers which points to nodes in the tree. These special pointers 4 are called 'threads' and binary trees with such pointers are called 'threaded trees'.

There are two major types of threading: - one way threading and two way threading.

In the one way threading of T, a thread will appear in the right field of a node and will point to the next node in the inorder traversal of T and in the two way threading of T, a thread will also appear in tl left field of a node and will point to the preceding node in the inorder traversal of T. There is another kind of one way threading which corresponds to the preorder traversal of T.

(ii) **Buddy systems**: A method of handling the storage management problem is kept separate free lists for blocks of different sizes. Each list contains free blocks of only one specific size. For example, memory contains 1024 words then it might be divided into fifteen blocks, one block of 256 words, t blocks of 128 words, four blocks of 64 words, and 8 blocks of 32 words. Whenever sport is requested the smallest block is whose size is greater than or equal to the size needed to reserve for example a block of 97 words is field by a block of size of 128 but in this method there are several limitation, first, space is wasted due to internal fragmentation, second, a request of block size 300 cannot be filled. The largest size is maintained is 256, the source of this problem is that free space are never combined

A variation to this scheme is buddy system and is quite useful. In Buddy systems several free list constituets of various sized blocks are maintained. Adjacent free blocks are smaller size may be removed from list combined into free blocks of larger size, and placed on the larger size free list. These larger blocks can be used intact to satisfy a request for a large amount of memory or they can be split once into their smallest constituents to satisfy several smaller request.

(iii) **B-tree :** A B-tree is a balanced m-way tree. A node of the tree may contain many records or key and pointers to children. It is also known as the balanced sort tree. It finds its use in external sorting. It is not a binary tree.

B-tree of order m has following properties:

(1) each node has a maximum of m children and minimum of m/2 children or any number from 2 to maximum.

(2) The no. of keys in a node is one less than its no of children. The arrangement

$P_O\ K_I\ P_I$ .... $K_n\ P_n$

Δ Δ Δ

$T_o\ T_I\ T_n$ each T_I is a m-way tree

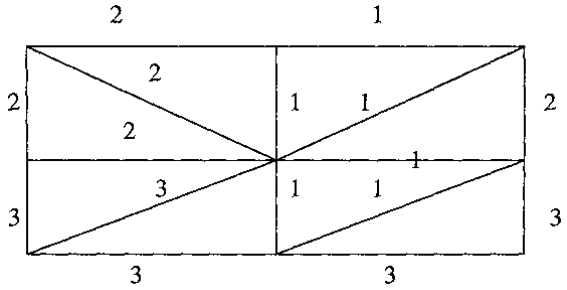
(3) The keys 'in a node Ki — K_n are arranged in sorted order $K_1 < K_2 — < K_n$. All the keys present in the subtree pointed to by a pointer P_i $K_i.P_i.K_{i+1}$ are greater than

(4) When a new key is to be inserted into a full node, the key is split into two nodes and the key with the median value is inserted in the parent node. In case the parent node is a root, a new root is created.

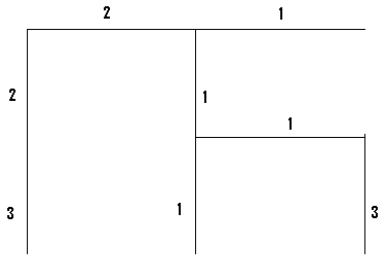
(5) All the leaves are on the same level i.e. there is no empty subtree above the level of the leaves. All the normal nodes of B-tree (Except 'root' and terminal nodes) have between m/2 and m children.

(iv) **Minimum Spanning Tree :** Given a weighted graph G, it is often desired to create a spanning tree T for G, such that the sum of weights of the edges in T is the least. Such a tree is called a minimum spanning tree and represents the cheapest way of connecting all the nodes in G . There are no. of techniques to create a minimum spanning tree for a weighted graph. For ex. Kruskal's algo. Prim's algorithm.

Example :-The given connected weighted graph G is



One of the minimum spanning trees of the graph G is:-



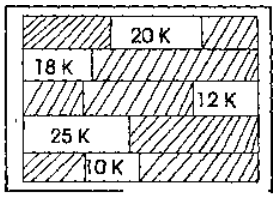
Minimum cost of the spanning tree = 14

Q.264. What is memory allocation? Explain the first fit and best fit algorithms for storage allocation with suitable examples. (7)

Ans:

Memory Allocation : It is the technique of allocating memory storage to program in order that the program can be run.

First fit algorithm: The question asked for examples not algorithms the first fit algorithm may be explained.



Suppose the memory allocation is as follows: suppose ▨ indicates allocated block & ▭ denotes a free block. The size of the free blocks are indicated. Suppose the current requirement is 8K. Then the first fit algorithm will allocate it from the block of 20K.

The advantage ; it is part disadvantage : fragmentation.

```
P=freeblock
aloc= null
 q=null;
while (p!=null && size (p) < n )
{
Q=p;
 p=next(P)
}/* end while */
if (p!=null){/* there is block large enough*/
s=size(p);
alloc=p+s-n; /* alloc contain the address of the designed
block*/
if(s==n)
/* remove the block from the free list*/
if (q==null)
freeblock=next(p);
else
 next(q)=next(p);
else
/* adjust the size of tile remaining free block*/
size(p)=s-n;
}/* end if*/
```

The Best fit algorithm : The best fit method obtains the smallest free block whose size is greater than or equal to n. An algorithm to obtain such a block by traversing the entire free list follows. We assume that the memsize is the total no. of words in memory.

The Best fit algorithm allocates the memory from the block that fits the requirement best i.e. the block that has size greater than the requirement, but no other block of lesser size can meet the requirement. Thus for the above problem the allocation will be made from block of size 10 K. Advantage : fragmentation is reduced

Disadvantage : more time.

```
p=freeblock;  /*p is used to traverse the free list */
 q==null;    /* q is one block behind p*/
 r=null;    /* r points to the desired block*/
rq=null;   /* rq is one block behind r */
 rsize=memsize+l; /* rsize is the size of the block at r */
alloc = null; /* alloc will point to block selected */
while(!=null)
{
 if(size(p)>= n && size(p)<rsize)
```

```
{
/* we have found a free block closer in size */
r=p;
rq=q;
rsize=size(p);
}
/* END IF */
/* continue traversing the free list */
q=P;
p=next (p);
}
/* end while */
if{r!=null)
{
/* there is block of sufficient size */
 alloc = r + r size - n ;
if (r size = =n){
/* remove the block from the free
list */ if(rq= = null )
freeblock = next (r );
else
next(rq)= next( r );
else size(r) = rsize- n ;
} /* end if */
```

Q.265. Consider the following graph

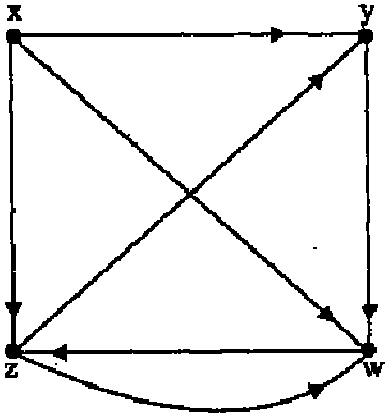


Fig 2

Let the nodes be stored in memory in an array G as :G; X, Y, Z, W

(i) Find the adjacency matrix A of the graph G.
(ii) Find the path matrix P of G using powers of the adjacency matrix -A.
(iii) Is G strongly connected? (3+3+1)

Ans:

Adjacency Matrix 'A' =

(i) =

	X	Y	Z	W
X	0	1	1	1
Y	0	0	0	1
Z	0	1	0	1
W	0	0	1	0

(ii) =

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$A^2 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$A^2 = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

$$A^3 = A^2.A = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 & 2 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$A^4 = A^3.A = \begin{bmatrix} 0 & 1 & 2 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$A^4 = \begin{bmatrix} 0 & 2 & 2 & 3 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

$A+A^2+A^3+A^4 =$ $\begin{bmatrix}0&1&1&1\\0&0&0&1\\0&1&0&1\\0&0&1&0\end{bmatrix}$ + $\begin{bmatrix}0&1&1&2\\0&0&1&0\\0&0&1&1\\0&1&0&1\end{bmatrix}$

+ $\begin{bmatrix}0&1&2&2\\0&1&0&1\\0&1&1&1\\0&0&1&1\end{bmatrix}$ + $\begin{bmatrix}0&2&2&3\\0&0&1&1\\0&1&1&2\\0&1&1&1\end{bmatrix}$

$= \begin{bmatrix}0&5&6&8\\0&1&2&3\\0&3&3&5\\0&2&3&3\end{bmatrix}$

Path matrix 'P' = $\begin{bmatrix}0&1&1&1\\0&1&1&1\\0&1&1&1\\0&1&1&1\end{bmatrix}$

(iii) No, G is not strongly connected.

Q.266. What is a hash function? Describe any three hash functions. (7)

Ans:

Hash function : This is the function from the set 'K' of keys into the set 'L' of memory addresses.

H : K→ L

These are used to determine the address of a record with the use of some key. Such a function 'H' may not yield distinct values: it is possible that two diff keys Kl and K2 will yield the same hash address;
This situation is called collision and some method must be used to resolve it.

Examples

1. Division method: Choose a number 'm' larger than the number 'n' of keys in K: (The number m is usually chosen to be a prime number). The hash function 'H' is defined by
H(K)= k (mod m)
Where K(mod m) denotes the remainder when 'K' is divided by m.

2. Midsquare **method:** The key 'K' is squared, then the hash function 'H' is defined by
H (K) =1 Where 1 is obtained by deleting digits from both ends of K^2.

3. Folding Method: The key **'K'** is partitioned into a no. of parts $k_1, k_2, \ldots k_T$, Where each part except , possibly the last, has the same number of digits as the required address. Then the parts are added together, ignoring the last carry, that is H (k) = $k_1, k_2, \ldots k_T$,. Where the leading digits carries, if any, are ignored.

Q.267 Write an algorithm for bubble sort. What is the asymptotic time complexity of bubble sort in the worst case. (7)

Ans:

Algorithm for bubble sort:

Let 'A' be a linear array of elements and temp be the variable for interchanging the position of the elements.

```
1.   Input 'n' elements of an array 'a'.
2.   initialize i=0
3.   repeat trough step 6 while (i<n)
4.   set j=0
5.   repeat through step 6 while (j<n-i-l)
6.   if (a[j]>a|j+l])
       (i) temp=a[j]
       (ii) a[j]=a[j+l]
       (iii) a[j+l]=temp
7.   display the sorted
elements of array 'a'
8.   Exit.
```

In this sorting technique, each element is compared with its adjacent element. If the first element is larger than the second one then the position of the elements are interchanged, otherwise it is not changed. Then next element are compared with its adjacent element and the same process is repeated for all the elements in the array. During the first pass the same process is repeated leaving the last element. During the second pass the second last element occupies the second last position. The same process is repeated until no more elements are left for the comparison. Finally the array is sorted. One may use a Hag to check whether there is any interchange in a particular pass. If there is no interchange in a pass, the array has already got sorted, and the algorithm can terminate.

Time complexity in worst case: in worst case, the array will be in reverse sorted order and the no. of comparisons are calculated as given below

$F(n) = l+2+3+4+$——— $+ (n-l)$

$= n (n-l)/2 = O (n^2)$

Q.268 Apply Kruskal's algorithm to find a minimum spanning tree of the graph in the following figure-Display execution of each step of the algorithm. (8)

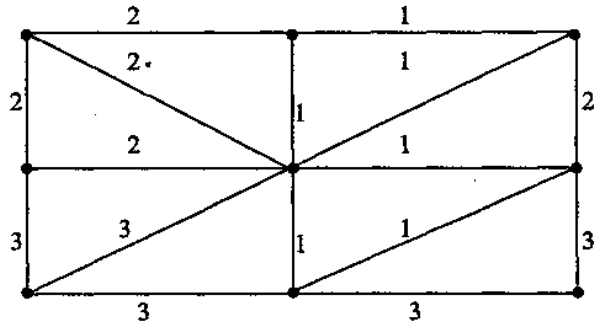
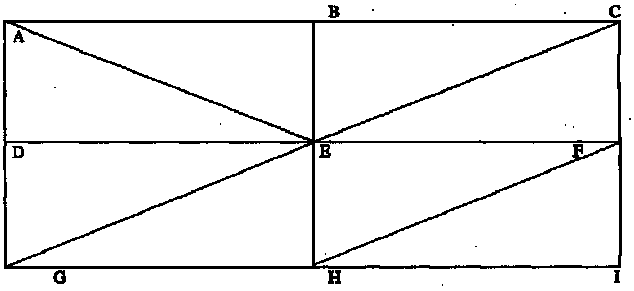


Fig 3

Describe the insertion sort algorithm. Compute its asymptotic time complexity for worst case and average case. (6)

Ans:

The given graph is name the vertices as shown



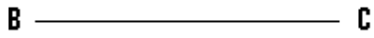
To find minimum spanning tree of the given graph :-

Edges :	AB	BC	AD	BE	CF	AE	CE	DE	EF
Weight :	2	1	2	1	2	2	1	2	1
Edges :	DG	EH	FI	GH	HI	GE	HE		
Weight :	3	1	3	3	3	3	1		

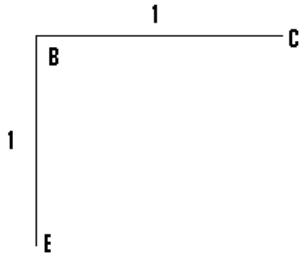
Edges in increasing order of weights

Edges:	BC	BE	CE	EF	EH	HF	AB	AD	CF	AE	DE	DG	FI	GH	HI	GE
Weight:	1	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3
Add:	yes	yes	no	yes	yes	no	yes	yes	no	no	no	yes	yes	no	no	no

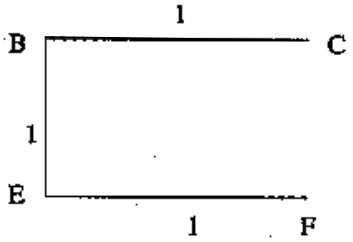
Step 1 Add 'BC'



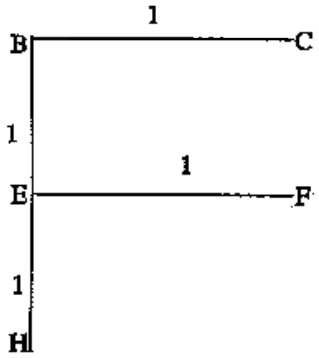
Step 2 Add 'BE'



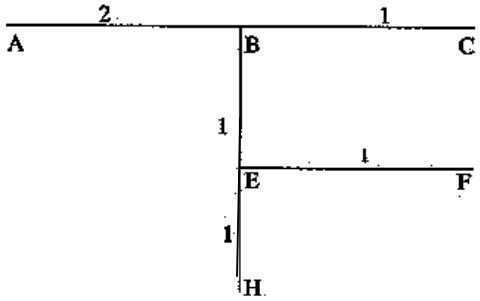
Step 3 Add 'EF'



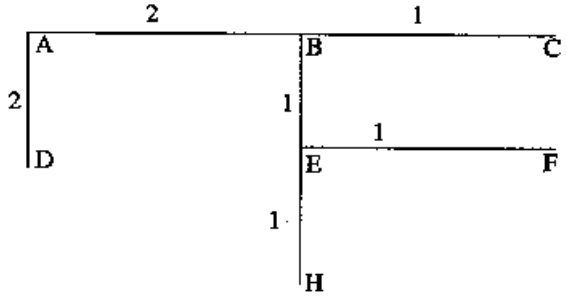
Step 4 Add 'EH'



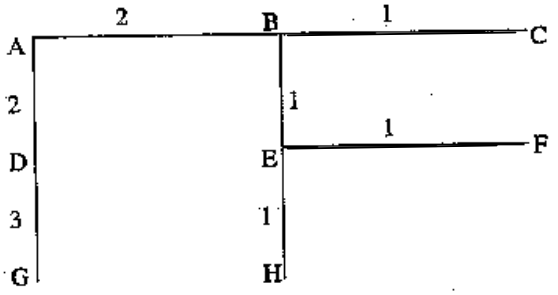
Step 5 Add 'AB'



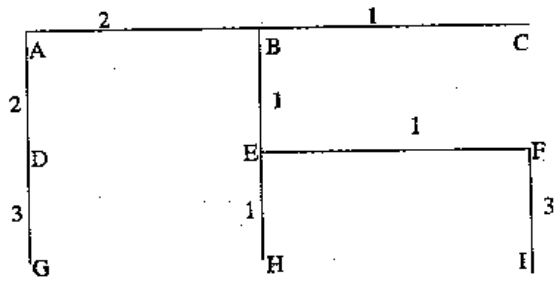
Step 6 Add 'AD'



STEP 7 Add **'DG'**



STEP 8 Add 'FI'



Cost of the spanning Tree= 1+2+2+1+3+1+3+1=14 This is the required spanning tree of the given graph.

Q.269 Define a heap. Describe the algorithm to insert an element into the heap. (2+5)

Ans:

Insertion Sort Algorithm:

```
       Consider an array 'A' with 'N'
elements
1.   set A[0]=-infinity [initialize sentinel element]
2.   Repeat steps 3 to 5 for K=2,3..,.......N
3.  set TEMP=A[K] and PTR=K-1 4   Repeat while TEMP <A[PTR]
(a) set A[PTR+1]=A[PTR] [moves element forward]
(b) set PTR=PTR-1 [End of loop]

5.   set A[PTR+1]=TEMP [inserts elements in proper place]
[End of step2 loop]

6.   Return
```

Complexity of insertion sort

Worst case: The worst case occurs when the -array 'A' is in reverse order and the inner loop must use the max number K-l of comparisons. Hence

$F(n)=l+2+.....+(n-l)=n(n-l)/2=0(n^2)$

Average case: In Average case there will be approximately (K-l)/2 comparisons in the inner loop. Accordingly, for the average case,

$F(n)=l/2+2/2+..... ...+(n-l)/2=n(n-l)/4=0(n^2)$

Q.270 Construct the heap showing the insertion of each of the following elements in separate figures. 44, 30, 50, 22, 60; 50 (7)

Ans:

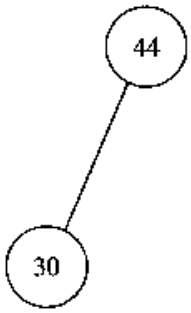
Heap : Suppose 'H' is a complete binary tree with 'n' elements. Then 'H' is called a Heap, as a 'maxheap', if each node 'N' of 'H' has the following property:
The value at 'N' is greater than or equal to the value at each of the children of N. Accordingly, the value at N is greater than or equal to the value at any of the descerdants of N.
Algorithm to insert an element in to the heap: Consider a heap 'H' with 'N' elements is stored in the array TREE and an ITEM of information is given. This procedure inserts 'ITEM' as a new element of 'H'.'PTR' gives the location of ITEM as it rises in the tree, and 'PAR' denotes the location of the parent of 'ITEM'.

```
1.   [ Add new node to 'H' and initialize PTR ] set N=N+1 and
PTR=N
2.   [ Find location
to insert ITEM ]
Repeat steps 3 to 6
while PTR>1
3.   set PAR= [PTR/2] (floor operation)
[Location of parent node] If ITEM<= TREE
[PAR] then step 4 else go to step 5.
4.   set TREE
[PTR]=ITEM and
return End of If
structure
5.  set TREE [PTR]= TREE [PAR]        [moves node down]
6   set PTR
=PAR//» updates
PTR
      //» End of
step2 loop
7.   //» Comments Assign
ITEM as the root of H] set
TREE [1] = ITEM
8.   Return
```

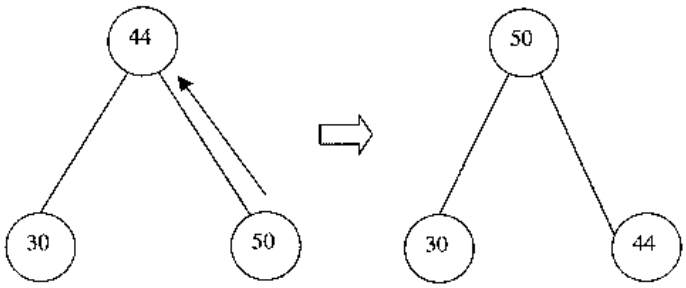
10. (b) Construction of heap :

Step 1 inserting 44

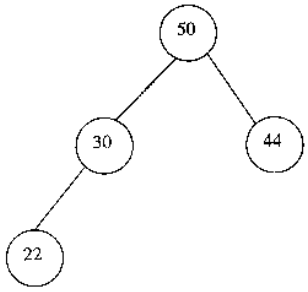
Step 2 Inserting 30



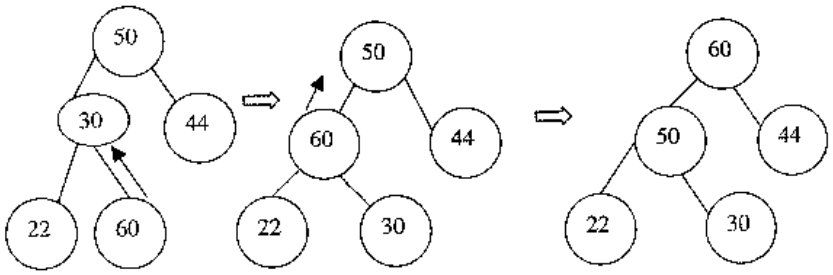
Step 3 Inserting 50



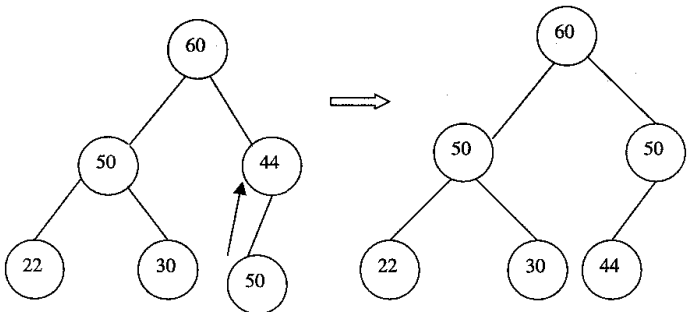
Step 4 inserting 22



Step 5 inserting 60



step 6 inserting 50



This is the required heap.

Q.271 Give the equivalent postfix expression for the following expressions:

(i) (A-B)/ ((D+E)*F) (ii) ((A+B)/D)↑((E-F)*G) (3+4)

Ans:

Convert the following infix expressions into postfire using a stack. Show at each step the stack contents and the output string.

(i) ((A-B)/((B+E)*F)) Add "(" to the starting and")"at the end of expr.

	Symbols scanned	Stack	Expression P
1	(	(	
2	(	((	
3	A	((	A
4	-	((-	A
5	B	((-	AB
6	)	(	AB-
7	/	(/	AB-
8	(	(/(	AB-
9	(	(/((	AB-
10	D	(/((	AB-D
11	+	(/((+	AB-D
12	E	(/((+	AB—DE
13	)	(/(	AB-DE+
14	*	(/(*	AB-DE+
15	F	(/(*	AB-DE+F
16	)	(/	AB-DE+F*
17	)		AB-DE+F*/

The postfix expression is P=AB-DE+F*/

(ii) The given expression is(((A+B)/D) ^ ((E-F)*G)) Add "(' to the start and ") at the end of the expression.

	Symbols scanned	Stack	Expression P
1	(	(	
2	(	((	
3	(	(((	
4	A	(((	A
5	+	(((+	A
6	B	(((+	AB
7	)	((	AB+
8	/	((/	AB+
9	D	((/	AB+D
10	)	(	AB+D/
11	↑	(↑	AB+D/
12	(	(↑(	AB+D/
13	(	(↑ ((	AB+D/
14	E	(↑((	AB+D/E
15	-	(↑ ((-	AB+D/E
16	F	(↑ ((-	AB+D/EF
17	)	(↑(	AB+D/EF-
18	*	(↑ (*	AB+D/EF-
19	G	(↑(*	AB+D/EF-G
20	)	(↑	AB+D/EF-G*
21	)		AB+D/EF-G*
The post fix expression 'P'= AB+D/EF-G*↑			

Q.272 Suppose a queue is maintained by a circular array QUEUE with N = 12 memory cells. Find the number of elements in QUEUE if

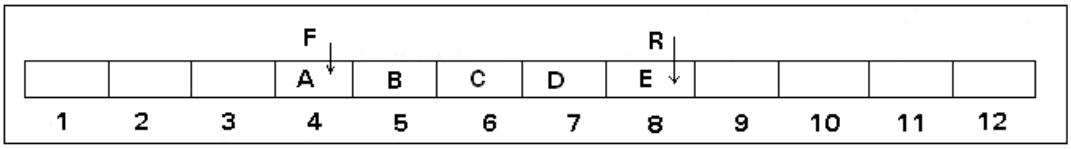
(i) Front = 4, Rear =8.
(ii) Front = 10, Rear = 3.
(iii) Front = 5, Rear = 6 and then two elements are deleted. (2+2+3)

Ans:

N=12

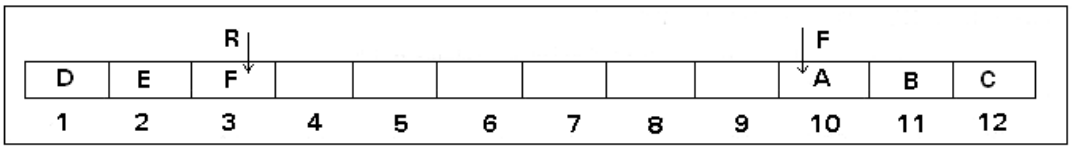
(i) Front=4, Rear=8

No of elements = 8-4 + 1 = 5

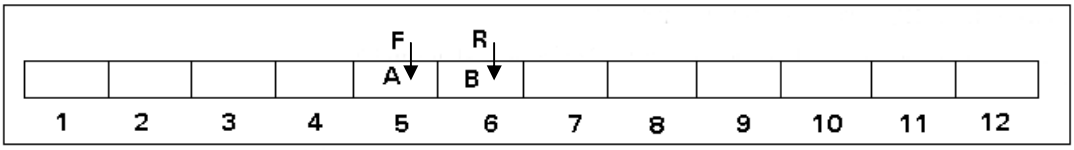


(ii)Front = 10, Rear = 3

No of elements = 6



(iii) Front = 5, Rear = 6



After deletion of two elements in the queue = 0

Q.273 Suppose we wish to partition the square roots of the integers from 1 to 100 in to two piles of fifty numbers each, such that the sum of the numbers in the first pile is as close as possible to the sum of the numbers in the second pile. If you could use minimum computer time to answer this question, what computations would you perform on the computer in that time? **(5)**

Ans :

According to question, sum of square roots in two piles should be as close as possible. For that we can add square roots of odd numbers. In one pile and square roots of even numbers in another pile. Since for natural numbers also if we want to divide into two piles according to nearest equal sum requirement above solution will work i.e.1+3+……+99, 2+4+ ……..+100.Square root is a strictly increasing function for positive numbers. So result holds for square root also.

Computations

1. Check whether no. is odd or even dividing by two, module is zero or not.
2. Add the variable sum computation time for n nos. It will require θ(n) time.

Q.274 Farely fractions of level one is defined as sequence (0/1, 1/1). This sequence is extended in level two to form a sequence (0/1, ½, 1/1), sequence (0/1, 1/3, ½, 2/3, 1/1) at level three, sequence (0/1, ¼, 1/3, ½, 2/3, ¾, 1/1) at level four, so that at each level n, a new fraction (a+b) / (c+d) is inserted between two neighbour fractions a/c and b/d only if c+d <= n. Devise a procedure, which for a number n entered by the user creates – by constantly extending it – a linked list of fractions at

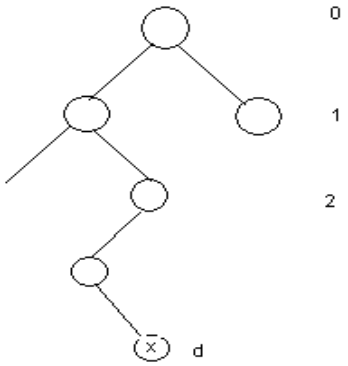
level n. **(8)**

Ans :

```
List{int info;
     int info;
     list*next;};
Farely fractions(n){
listI0,I1,x;
info1(I1)=00;
info2(I1)=1;
next(I1)=I2;
info1(I2)=1;
info(I2)=1;
next(I2)=NULL;
for(i=2;i<=n;i++){
   x=head[L];
   while(x!=nil&&info1(x)+info1(next)(xL)<=i){
      listI;
info(Ij)=info2(xo)+info2(next,x0));
info(Ij)=info1(x)+info1(next);
list insert(L,x);
x=next(x)   }}}
```

Q.275 Suppose that a Binary Search Tree is constructed by repeatedly inserting distinct values in to the tree. Argue that the number of nodes examined in searching for a value in the tree is one plus the number of nodes examined when the value was first inserted in to the tree. **(7)**

Ans :
Let us consider an element x to insert in a binary search tree. So for inserting the



element x first at level 0,then level 1 and suppose upto level (d-1). While examine at (d-1) level, x might have less or more in comaprison to element at (d-1). Then we insert x either left or right. In both cases no of examined code will be d. Now suppose we want to search for x, this time again traverses the same path as we traverse. While inserting the element, we stop at (d-1) the level but for searching we examine node at dth level also i.e. the node containing x. Thus number of node examine while inserting are d while incase of searching it is d+1 i.e. one more than while inserting, hence the result.

TYPICAL QUESTIONS & ANSWERS

PART -I

OBJECTIVE TYPE QUESTIONS

Each Question carries 2 marks.

Choosethe correct or the best alternative in the following:

Q.1 Which of the following relational algebra operations do not require the participating tables to be union-compatible?

(A) Union **(B)** Intersection
(C) Difference **(D)** Join

Ans: (D)

Q.2 Which of the following is not a property of transactions?

(A) Atomicity **(B)** Concurrency
(C) Isolation **(D)** Durability

Ans: (B)

Q.3 Relational Algebra does not have

(A) Selection operator. **(B)** Projection operator.
(C) Aggregation operators. **(D)** Division operator.

Ans: (C)

Q.4 Checkpoints are a part of

(A) Recovery measures. **(B)** Security measures.
(C) Concurrency measures. **(D)** Authorization measures.

Ans: (A)

Q.5 Tree structures are used to store data in

(A) Network model. **(B)** Relational model.
(C) Hierarchical model. **(D)** File based system.

Ans: (C)

Q.6 The language that requires a user to specify the data to be retrieved without specifying exactly how to get it is

(A) Procedural DML. **(B)** Non-Procedural DML.
(C) Procedural DDL. **(D)** Non-Procedural DDL.

Ans: (B)

Q.7 Precedence graphs help to find a

(A) Serializable schedule. **(B)** Recoverable schedule.
(C) Deadlock free schedule. **(D)** Cascadeless schedule.

Ans: (A)

Q.8 The rule that a value of a foreign key must appear as a value of some specific table is called a
(A) Referential constraint. **(B)** Index.
(C) Integrity constraint. **(D)** Functional dependency.

Ans: (A) The rule that a value of a foreign key must appear as a value of some specific table is called a referential constraint. (Referential integrity constraint is concerned with foreign key)

Q.9 The clause in SQL that specifies that the query result should be sorted in ascending or descending order based on the values of one or more columns is
(A) View **(B)** Order by
(C) Group by **(D)** Having

Ans: (B) The clause in SQL that specifies that the query result should be sorted in ascending or descending order based on the values of one or more columns is ORDER BY. (ORDER BY clause is used to arrange the result of the SELECT statement)

Q.10 What is a disjoint less constraint?
(A) It requires that an entity belongs to no more than one level entity set.
(B) The same entity may belong to more than one level.
(C) The database must contain an unmatched foreign key value.
(D) An entity can be joined with another entity in the same level entity set.

Ans: (A) Disjoint less constraint requires that an entity belongs to no more than one level entity set. (Disjoint less constraint means that an entity can be a member of at most one of the subclasses of the specialization.)

Q.11 According to the levels of abstraction, the schema at the intermediate level is called
(A) Logical schema. **(B)** Physical schema.
(C) Subschema. **(D)** Super schema.

Ans: According to the levels of abstraction, the schema at the intermediate level is called ***conceptual schema***.
(**Note:** All the options given in the question are wrong.)

Q.12 It is an abstraction through which relationships are treated as higher level entities
(A) Generalization. **(B)** Specialization.
(C) Aggregation. **(D)** Inheritance.

Ans: (C) It is an abstraction through which relationships are treated as higher level entities Aggregation. (In ER diagram, aggregation is used to represent a relationship as an entity set.)

Q.13 A relation is in _____ if an attribute of a composite key is dependent on an attribute of other composite key.

(A) 2NF **(B)** 3NF
(C) BCNF **(D)** 1NF

Ans: (B) A relation is in 3 NF if an attribute of a composite key is dependent on an attribute of other composite key. (If an attribute of a composite key is dependent on an attribute of other composite key then the relation is not in BCNF, hence it has to be decomposed.)

Q.14 What is data integrity?

(A) It is the data contained in database that is non redundant.
(B) It is the data contained in database that is accurate and consistent.
(C) It is the data contained in database that is secured.
(D) It is the data contained in database that is shared.

Ans: (B) (Data integrity means that the data must be valid according to the given constraints. Therefore, the data is accurate and consistent.)

Q.15 What are the desirable properties of a decomposition

(A) Partition constraint. **(B)** Dependency preservation.
(C) Redundancy. **(D)** Security.

Ans: (B) What are the desirable properties of a decomposition – dependency preserving. (Lossless join and dependency preserving are the two goals of the decomposition.)

Q.16 In an E-R diagram double lines indicate

(A) Total participation. **(B)** Multiple participation.
(C) Cardinality N. **(D)** None of the above.

Ans: (A)

Q.17 The operation which is not considered a basic operation of relational algebra is

(A) Join. **(B)** Selection.
(C) Union. **(D)** Cross product.

Ans: (A)

Q.18 Fifth Normal form is concerned with

(A) Functional dependency. **(B)** Multivalued dependency.
(C) Join dependency. **(D)** Domain-key.

Ans: (C)

Q.19 Block-interleaved distributed parity is RAID level

(A) 2. **(B)** 3
(C) 4. **(D)** 5.

Ans: (D)

Q.20 Immediate database modification technique uses

(A) Both undo and redo. **(B)** Undo but no redo.
(C) Redo but no undo. **(D)** Neither undo nor redo.

Ans: (A)

Q.21 In SQL the statement **select * from R, S** is equivalent to

(A) Select * from R natural join S. **(B)** Select * from R cross join S.
(C) Select * from R union join S. **(D)** Select * from R inner join S.

Ans: (B)

Q.22 Which of the following is not a consequence of concurrent operations?

(A) Lost update problem. **(B)** Update anomaly.
(C) Unrepeatable read. **(D)** Dirty read.

Ans: (B)

Q.23 As per equivalence rules for query transformation, selection operation distributes over

(A) Union. **(B)** Intersection.
(C) Set difference. **(D)** All of the above.

Ans: (D)

Q.24 The metadata is created by the

(A) DML compiler **(B)** DML pre-processor
(C) DDL interpreter **(D)** Query interpreter

Ans: (C)

Q.25 When an E-R diagram is mapped to tables, the representation is redundant for

(A) weak entity sets **(B)** weak relationship sets
(C) strong entity sets **(D)** strong relationship sets

Ans: (B)

Q.26 When $R \cap S = \phi$, then the cost of computing $R \bowtie S$ is

(A) the same as $R \times S$ **(B)** greater the $R \times S$
(C) less than $R \times S$ **(D)** cannot say anything

Ans: (A)

Q.27 In SQL the word 'natural' can be used with

(A) inner join **(B)** full outer join
(C) right outer join **(D)** all of the above

Ans: (A)

Q.28 The default level of consistency in SQL is

(A) repeatable read **(B)** read committed
(C) read uncommitted **(D)** serializable

Ans: **(D)**

Q.29 If a transaction T has obtained an exclusive lock on item Q, then T can

(A) read Q **(B)** write Q
(C) both read and write Q **(D)** write Q but not read Q

Ans: (C)

Q.30 Shadow paging has

(A) no redo **(B)** no undo
(C) redo but no undo **(D)** neither redo nor undo

Ans: (A)

Q.31 If the closure of an attribute set is the entire relation then the attribute set is a

(A) superkey **(B)** candidate key
(C) primary key **(D)** not a key

Ans: (A)

Q.32 DROP is a _____ statement in SQL.

(A) Query **(B)** Embedded SQL
(C) DDL **(D)** DCL

Ans: (C)

Q.33 If two relations R and S are joined, then the non matching tuples of both R and S are ignored in

(A) left outer join **(B)** right outer join
(C) full outer join **(D)** inner join

Ans: (D)

Q.34 The keyword to eliminate duplicate rows from the query result in SQL is

(A) DISTINCT **(B)** NO DUPLICATE
(C) UNIQUE **(D)** None of the above

` **Ans: (C)**

Q.35 In 2NF

(A) No functional dependencies (FDs) exist.
(B) No multivalued dependencies (MVDs) exist.
(C) No partial FDs exist.
(D) No partial MVDs exist.

Ans: (C)

Q.36 Which one is correct statement?
Logical data independence provides following without changing application programs:
(i) Changes in access methods.
(ii) Adding new entities in database
(iii) Splitting an existing record into two or more records
(iv) Changing storage medium
(A) (i) and (ii) **(B)** (iv) only, **(C)** (i) and (iv) **(D)** (ii) and (iii)

Ans: (D)

Q.37 In an E-R, Y is the dominant entity and X is a subordinate entity. Then which of the following is incorrect :
(A) Operationally, if Y is deleted, so is X
(B) existence is dependent on Y.
(C) Operationally, if X is deleted, so is Y.
(D) Operationally, if X is deleted, & remains the same.

Ans: (C)

Q.38 Relational Algebra is
(A) Data Definition Language .
(B) Meta Language
(C) Procedural query Language
(D) None of the above

Ans: (C)

Q.39 Which of the following aggregate functions does not ignore nulls in its results?.
(A) COUNT . **(B)** COUNT (*)
(C) MAX **(D)** MIN

Ans: (B)

Q.40 R (A,B,C,D) is a relation. Which of the following does not have a lossless join dependency preserving BCNF decomposition
(A) A→B, B→CD **(B)** A→B, B→C, C→D .
(C) AB→C, C→AD **(D)** A→BCD

Ans: (D)

Q.41 Consider the join of relation R with a relation S. If R has m tuples and S has n tuples, then the maximum and minimum size of the join respectively are
(A) m+n and 0 **(B)** m+n and |m-n|
(C) mn and 0 **(D)** mn and m+n

Ans: (C)

Q.42 Maximum height of a B+ tree of order m with n key values is

(A) $Log_m(n)$ **(B)** (m+n)/2

(C) $Log_{m/2}(m+n)$ **(D)** None of these

Ans: (D)

Q.43 Which one is true statement :

(A) With finer degree of granularity of locking a high degree of concurrency is possible.

(B) Locking prevents non – serializable schedules.

(C) Locking cannot take place at field level.

(D) An exclusive lock on data item X is granted even if a shared lock is already held on X.

Ans: (A)

Q.44 Which of the following statement on the view concept in SQL is invalid?

(A) All views are not updateable

(B) The views may be referenced in an SQL statement whenever tables are referenced.

(C) The views are instantiated at the time they are referenced and not when they are defined.

(D) The definition of a view should not have GROUP BY clause in it.

Ans: (D)

Q.45 Which of the following concurrency control schemes is not based on the serializability property?

(A) Two – phase locking **(B)** Graph-based locking

(C) Time-stamp based locking **(D)** None of these .

Ans: (D)

Q.46 Which of the following is a reason to model data?

(A) Understand each user's perspective of data

(B) Understand the data itself irrespective of the physical representation

(C) Understand the use of data across application areas

(D) All of the above

Ans: (D)

Q.47 If an entity can belong to only one lower level entity then the constraint is

(A) disjoint **(B)** partial

(C) overlapping **(D)** single

Ans: (B)

Q.48 The common column is eliminated in

(A) theta join **(B)** outer join

(C) natural join **(D)** composed join

Ans: (C)

Q.49 In SQL, testing whether a subquery is empty is done using
(A) DISTINCT **(B)** UNIQUE
(C) NULL **(D)** EXISTS

Ans: (D)

Q.50 Use of UNIQUE while defining an attribute of a table in SQL means that the attribute values are
(A) distinct values **(B)** cannot have NULL
(C) both **(A)** & **(B)** **(D)** same as primary key

Ans: (C)

Q.51 The cost of reading and writing temporary files while evaluating a query can be reduced by
(A) building indices **(B)** pipelining
(C) join ordering **(D)** none of the above

Ans: (B)

Q.52 A transaction is in _____ state after the final statement has been executed.
(A) partially committed **(B)** active
(C) committed **(D)** none of the above

Ans: (C)

Q.53 In multiple granularity of locks SIX lock is compatible with
(A) IX **(B)** IS
(C) S **(D)** SIX

Ans: (B)

Q.54 The statement that is executed automatically by the system as a side effect of the modification of the database is
(A) backup **(B)** assertion
(C) recovery **(D)** trigger

Ans: (D)

Q.55 The normal form that is not necessarily dependency preserving is
(A) 2NF **(B)** 3NF
(C) BCNF **(D)** 4NF

Ans: (A)

Q.56 A functional dependency of the form $x \rightarrow y$ is trivial if

(A) $y \subseteq x$ **(B)** $y \subset x$
(C) $x \subseteq y$ **(D)** $x \subset y$

Ans: (A)

Q.57 The normalization was first proposed by _____.
(A) Code **(B)** Codd
(C) Boyce Codd **(D)** Boyce

Ans: (B)

Q.58 The division operator divides a dividend A of degree m+n by a divisor relation B of degree n and produces a result of degree
(A) m – 1 **(B)** m + 1
(C) m * m **(D)** m

Ans: (D)

Q.59 Which of the following is not a characteristic of a relational database model?
(A) Table **(B)** Tree like structure
(C) Complex logical relationship **(D)** Records

Ans: (B)

Q.60 Assume transaction A holds a shared lock R. If transaction B also requests for a shared lock on R.

(A) It will result in a deadlock situation.
(B) It will immediately be rejected.
(C) It will immediately be granted.
(D) It will be granted as soon as it is released by A .

Ans: (C)

Q.61 In E-R Diagram total participation is represented by

(A) double lines **(B)** Dashed lines
(C) single line **(D)** Triangle

Ans: (A)

Q.62 The FD $A \rightarrow B$, $DB \rightarrow C$ implies
(A) $DA \rightarrow C$ **(B)** $A \rightarrow C$
(C) $B \rightarrow A$ **(D)** $DB \rightarrow A$

Ans: (A)

Q.63 The graphical representation of a query is _____.
(A) B-Tree **(B)** graph
(C) Query Tree **(D)** directed graph

Ans: (C)

Q.64 Union operator is a :

(A) Unary Operator **(B)** Ternary Operator
(C) Binary Operator **(D)** Not an operator

Ans: (C)

Q.65 Relations produced from an E-R model will always be

(A) First normal form. **(B)** Second normal form.
(C) Third normal form. **(D)** Fourth normal form.

Ans: (A)

Q.66 Manager salary details are hidden from the employee .This is

(A) Conceptual level data hiding.
(B) External level data hiding.
(C) Physical level data hiding.
(D) None of these.

Ans: (A)

Q.67 Which of the following is true for network structure?

(A) It is a physical representation of the data.
(B) It allows many to many relationship.
(C) It is conceptually simple.
(D) It will be the dominant database of the future.

Ans: **(A)**

Q.68 Which two files are used during operation of the DBMS?

(A) Query languages and utilities
(B) DML and query language
(C) Data dictionary and transaction log
(D) Data dictionary and query language

Ans: (C)

Q.69 A list consists of last names, first names, addresses and pin codes. If all people in the list have the same last name and same pin code a useful key would be

(A) the pin code
(B) the last name
(C) the compound key first name and last name
(D) Tr from next page

Ans: (C)

Q.70 In b-tree the number of keys in each node is ____ than the number of its children.

(A) one less **(B)** same
(C) one more **(D)** half

Ans: (A)

Q.71 The drawback of shadow paging technique are

(A) Commit overhead **(B)** Data fragmentation
(C) Garbage collection **(D)** All of these

Ans: (D)

Q.72 Which normal form is considered adequate for normal relational database design?

(A) 2NF **(B)** 5NF
(C) 4NF **(D)** 3NF

Ans: (D)

Q.73 Which of the following addressing modes permits relocation without any change over in the code?

(A) Indirect addressing **(B)** Indexed addressing
(C) PC relative addressing **(D)** Base register addressing

Ans: (B)

Q.74 In a multi-user database, if two users wish to update the same record at the same time, they are prevented from doing so by

(A) jamming **(B)** password
(C) documentation **(D)** record lock

Ans: (D)

Q.75 The values of the attribute describes a particular_____

(A) Entity set **(B)** File
(C) Entity instance **(D)** Organization

Ans: (C)

Q.76 Which of the following relational algebraic operations is not from set theory?

(A) Union **(B)** Intersection
(C) Cartesian Product **(D)** Select

Ans: (D)

Q.77 Which of the following ensures the atomicity of the transaction?

(A) Transaction management component of DBMS
(B) Application Programmer
(C) Concurrency control component of DBMS
(D) Recovery management component of DBMS

Ans: (A)

Q.78 If both the functional dependencies : $X \rightarrow Y$ and $Y \rightarrow X$ hold for two attributes X and Y then the relationship between X and Y is

(A) M:N **(B)** M:1
(C) 1:1 **(D)** 1:M

Ans: (C)

Q.79 What will be the number of columns and rows respectively obtained for the operation, A-B, if A B are Base union compatible and all the rows of a are common to B? Assume A has 4 columns and 10 rows; and B has 4 columns and 15 rows
(A) 4,0 **(B)** 0,0
(C) 4,5 **(D)** 8,5

Ans: (A)

Q.80 For correct behaviour during recovery, undo and redo operation must be
(A) Commutative **(B)** Associative
(C) idempotent **(D)** distributive

Ans: (C)

Q.81 Which of the following is not a consequence of non-normalized database?
(A) Update Anomaly **(B)** Insertion Anomaly
(C) Redundancy **(D)** Lost update problem

Ans: (D)

Q.82 Which of the following is true for relational calculus?
(A) $\forall x(P(x)) \equiv \neg(\exists x)(\neg P(x))$ **(B)** $\forall x(P(x)) \equiv \neg(\exists x)(P(x))$
(C) $\forall x(P(x)) \equiv (\exists x)(\neg P(x))$ **(D)** $\forall x(P(x)) \equiv (\exists x)(P(x))$

Ans: (A)

Q.83 The part of a database management system which ensures that the data remains in a consistent state is
(A) authorization and integrity manager
(B) buffer manager
(C) transaction manager
(D) file manager

Ans: (C)

Q.84 Relationships among relationships can be represented in an-E-R model using
(A) Aggregation **(B)** Association
(C) Weak entity sets **(D)** Weak relationship sets

Ans: (A)

Q.85 In tuple relational calculus P1 AND P2 is equivalent to
(A) $(\neg P1 OR \neg P2)$. **(B)** $\neg(P1 OR \neg P2)$.
(C) $\neg(\neg P1 OR\ P2)$. **(D)** $\neg(\neg P1 OR\ \neg P2)$.

Ans: (D)

Q.86 If $\alpha \rightarrow \beta$ holds then so does

(A) $\gamma\alpha \rightarrow \gamma\beta$ **(B)** $\alpha \rightarrow \rightarrow \gamma\beta$

(C) both (A) and (B) **(D)** None of the above

Ans: (A)

Q.87 Cascading rollback is avoided in all protocol except

(A) strict two-phase locking protocol.

(B) tree locking protocol

(C) two-phase locking protocol

(D) validation based protocol.

Ans: (D)

Q. 88 Wait-for graph is used for

(A) detecting view serializability. **(B)** detecting conflict serializability.

(C) deadlock prevention **(D)** deadlock detection

Ans: (D)

Q.89 The expression $\sigma_{\theta 1}(E1 \bowtie_{\theta 2} E2)$ is the same as

(A) $E1 \bowtie_{\theta 1 \wedge \theta 2} E2$ **(B)** $\sigma_{\theta 1}\, E1 \wedge \sigma_{\theta 2}\, E2$

(C) $E1 \bowtie_{\theta 1 \vee \theta 2} E2$ **(D)** None of the above

Ans: (A)

Q.90 The clause **alter table** in SQL can be used to

(A) add an attribute

(B) delete an attribute

(C) alter the default values of an attribute

(D) all of the above

Ans: (D)

Q. 91 The data models defined by ANSI/SPARC architecture are

(A) Conceptual, physical and internal

(B) Conceptual, view and external

(C) Logical, physical and internal

(D) Logical, physical and view

Ans: (D)

Q.92 Whenever two independent one-to-many relationships are mixed in the same relation, a _____ arises.

(A) Functional dependency **(B)** Multi-valued dependency

(C) Transitive dependency **(D)** Partial dependency

Ans:(B)

Q.93 A table can have only one

(A) Secondary key **(B)** Alternate key
(C) Unique key **(D)** Primary key

Ans: (D)

Q.94 Dependency preservation is not guaranteed in

(A) BCNF **(B)** 3NF
(C) PJNF **(D)** DKNF

Ans: (A)

Q.95 Which is the best file organization when data is frequently added or deleted from a file?

(A) Sequential **(B)** Direct
(C) Index sequential **(D)** None of the above

Ans: (B)

Q.96 Which of the following constitutes a basic set of operations for manipulating relational data?

(A) Predicate calculus **(B)** Relational calculus
(C) Relational algebra **(D)** SQL
Ans: (C)

Q.97 An advantage of views is

(A) Data security **(B)** Derived columns
(C) Hiding of complex queries **(D)** All of the above

Ans: (A)

Q.98 Which of the following is not a recovery technique?

(A) Deferred update **(B)** Immediate update
(C) Two-phase commit **(D)** Shadow paging

Ans: (C)

Q.99 Isolation of the transactions is ensured by

(A) Transaction management **(B)** Application programmer
(C) Concurrency control **(D)** Recovery management

Ans: (C)

Q.100 _____ operator is used to compare a value to a list of literals values that have been specified.

(A) Like **(B)** COMPARE
(C) BETWEEN **(D)** IN
Ans: (A)

PART- II

DESCRIPTIVES

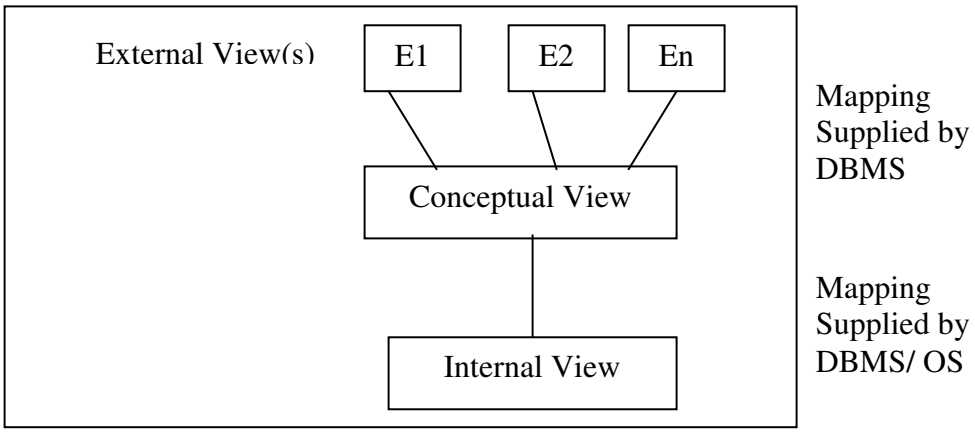
Q.1 Briefly discuss the different layers of ANSI SPARC architecture. Define physical and logical data independence. How does this architecture help in achieving these? **(7)**

Ans: The three layers of ANSI SPARC architecture are as follows:

- **Internal view** is at the lowest level of abstraction, closest to the physical storage method used. It indicates how the data will be stored and describes the data structures and access methods to be used by the database. There is one internal view for the entire database.
- **Global or Conceptual View :** At this level of abstraction all the database entities and the relationships among them are included. There is one conceptual view for the entire database.
- **External or User View:** The external or user view is at the highest level of database abstraction where only those portions of the database concern to a user or application programme are included. Any number if external or user views may exists for a given global or conceptual view.

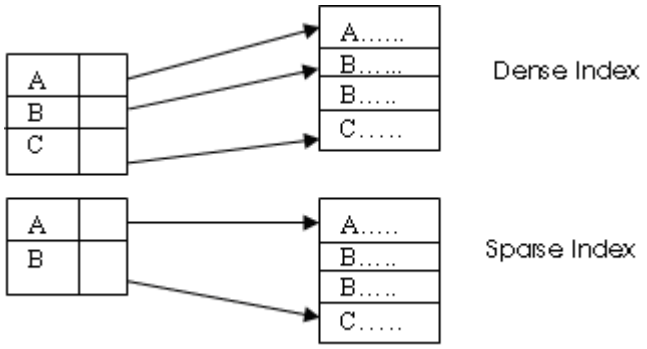
Data independence implies that change in one view must not require a change in the view(s) above. There are two types of data independence : logical and physical. **Logical data independence** means that the conceptual view can be changed without effecting the existing external view, i.e., a given record may be spilt or combined with other records but the external views need not be changed to reflect this. Physical data independence means that the physical storage structures or devices used to store the data can be changed without effecting the existing conceptual view or external view, i.e., if earlier indexed sequential files are used to store data and then the B- trees are used, even then the upper layers should not be effected.

There are two different mappings between the layers as shown below in the diagram. The mapping between external and conceptual levels is responsible to provide logical data independence and the mapping between internal and conceptual levels is responsible to provide physical data independence.



Q.2 Describe the storage structure of indexed sequential files and their access method. **(7)**

Ans: Storage structure of Indexed Sequential files and their access : To gain fast random access to records in a file, we can use an index structure. An index record consists of a search key value and pointers to data records, which is associated with a particular search key. An ordered index stores the values of the search keys in sorted order. A file may have several indices on different search keys. If the files containing the records is sequentially ordered, a primary index is an index whose search key also defines the sequential order of the file. Such files are known as index sequential files. There are two types of ordered indices : dense and sparse. In dense index, and index record appears only for some of the search-key in the files as shown below.



To access a particular record with search key value, K, using dense index we search index record with search key value, K, from which reach the first entry of data record with search key value K. Then the data records are searched linearly to obtain the required record. If either the index record is not there or the linear search reaches the data record with different search key value, then the record is not there. Now for sparse index, we search index record with search key value, K or the index record with highest search key value less than K, from which reach the first entry of data record with search key value K or highest value less than K. Then the data records are searched linearly to obtain the required record. If the linear searches the data record with search key value greater than K, then the record is not there.

Q.3 Define the terms entity, attribute, role and relationship between the entities, giving examples for each of them. **(4)**

Ans: Entity: An entity is a "thing" or "object" in the real world that is distinguishable from other objects. For example, a person and bank account can be considered as entities.
Attribute: Entities are described in a database by a set of attributes, i.e., the characteristics of an entity are known as attributes. For example, name, age, date of birth, etc are attributes of the entity person. Similarly, account number, balance, nature of account, etc are attributes of the entity bank account.
Relationship: A relationship is an association among the several entities. For example, a depositor relationship associates the entity person with a bank account.

Role: The function that an entity plays in a relationship is called that entity's role. For example, in the relationship depositor mentioned above, the entity person plays the role of a customer in the relationship.

Q.4 What are the three data anomalies that are likely to occur as a result of data redundancy? Can data redundancy be completely eliminated in database approach? Why or why not? **(5)**

Ans: The three type of anomalies that can arise in the database because of redundancy are insertion, deletion and modification/updation anomalies. Consider a relation emp_dept with attributes: E#, Ename, Address, D#, Dname, Dmgr# with the primary key as E#.

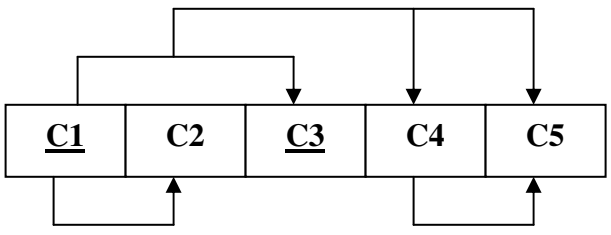
Insertion anomaly: Let us assume that a new department has been started by the organization but initially there is no employee appointed for that department, then the tuple for this department cannot be inserted into this table as the E# will have NULL, which is not allowed as E# is primary key. This kind of a problem in the relation where some tuple cannot be inserted is known as insertion anomaly.

Deletion anomaly: Now consider there is only one employee in some department and that employee leaves the organization, then the tuple of that employee has to be deleted from the table, but in addition to that the information about the department also will get deleted. This kind of a problem in the relation where deletion of some tuples can lead to loss of some other data not intended to be removed is known as deletion anomaly.

Modification /update anomaly: Suppose the manager of a department has changed, this requires that the Dmgr# in all the tuples corresponding to that department must be changed to reflect the new status. If we fail to update all the tuples of the given department, then two different records of employee working in the same department might show different Dmgr# leading to inconsistency in the database. This is known as modification/update anomaly.

The data redundancy. Cannot be totally removed from the database, but there should be controlled redundancy, for example, consider a relation student_report(S#, Sname, Course#, SubjectName, marks) to store the marks of a student for a course having some optional subjects, but all the students might not select the same optional papers. Now the student name appears in every tuple, which is redundant and we can have two tables as students(S#, Sname, CourseName) and Report(S#, SubjectName, Marks). However, if we want to print the mark-sheet for every student using these tables then a join operation, which is a costly operation, in terms of resources required to carry out, has to be performed in order to get the name of the student. So to save on the resource utilization, we might opt to store a single relation, students_report only.

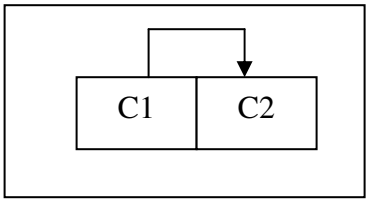
Q.5 Given the dependency diagram shown in the following figure, (the primary key attributes are underlined)



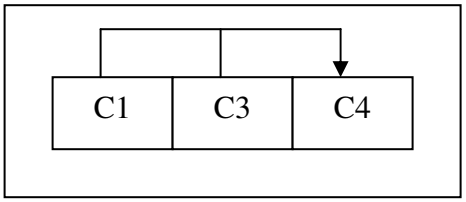
(i) Identify and discuss each of the indicated dependencies?
(ii) Create a database whose tables are atleast in 3NF, showing dependency Diagram for each table? **(4+5)**

Ans:
(i) The *FDS* for the given relation are:
1. C1→ C2
2. C1, C3 → C2, C4, C5
3. C4 → C5

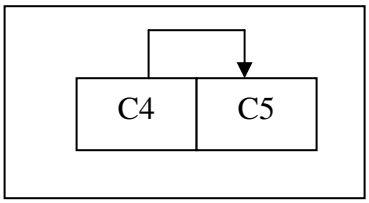
The Primary key of the given relation is (C1, C3), so C1 → C2 is a partial *FD* and there is a transitive *FD* also : C1, C3 → C4 →C5 in the relation.
(ii) According to the algorithm to decompose the relation schema into 3*NF* as follows:
R1 (C1, C2) with *FD* {C1 → C2} and functional dependency diagram.



R2(C1, C3, C4) with *FD* {C1, C3 → C4} and functional dependency diagram



R3(C4, C5) with *FD* {C4 → C5} and functional dependency diagram.



We cannot have a single relation for the *FD* C1, C3 → C2, C4, C5 as then there will be partial and transitive *FD*s in the relation.

Q.6 Consider the following relations with underlined primary keys.
Product(P_code, Description, Stocking_date, QtyOnHand, MinQty, Price, Discount, V_code)
Vendor(V_code, Name, Address, Phone)
Here a vendor can supply more than one product but a product is supplied by only one vendor. Write SQL queries for the following :
(i) List the names of all the vendors who supply more than one product.
(ii) List the details of the products whose prices exceed the average product price.

(iii) List the Name, Address and Phone of the vendors who are currently not supplying any product. **(3 x 3)**

Ans:

(i) Select Name from Vendor
Where V_code in (Select V_code from Product
group by V_code having count (V_code) > 1)

(ii) Select * from Product
Where Price > (Select avg (Price)from Product)

(iii) Select Name, Address, Phone From Vendor
Where V_code not in (Select V_code from Product)

Q.7 Define the domain relational calculus. **(5)**

Ans: An expression in the domain relational calculus is of the form
$\{< x_1, x_2, \ldots\ldots x_n > 1\ P(x_1, x_2 \ldots.., x_n)\} \ldots\ldots\ldots\ldots 1$ Where $x_1, x_2 \ldots\ldots\ldots x_n$ represent the domain variables. *P* represents a formula composed of atoms. An atom in the domain relational calculas has one of the following forms:

- $\{< x_1, x_2), \ldots.. x_n > \in r$, where r is a relation on *n* attributes and $x_1, x_2, \ldots.. x_n$ are domain variables or domain constants.
- x Θ y, where a and y where r domain variables and Θ is a comparison operator (<, ≤, =, ≠, ≥, >). We require that attributes x and y have domains that can be compared by Θ.
- x Θ c, where x is a domain variable, Θ is a comparison operator, and c is a constant in the domain of the attribute for which x is a domain variable.

We build up formulae from atoms by using the following rules :

- An atom is a formula.
- If P_1 is a formula, then so are $\neg p_1$ and (P_1)

If P_1 and P_2 are formulae, then so are $p_1 \Lambda\ p_2$ and $p_1 \vee p_2$ and $p_1 \Rightarrow P_2$

- If p_1 (x) is a formula in x, where x is a domain variable, then so are $\exists x(p_1(x))$ and $\forall x(p_1(x))$.

Q.8 Given $R(A, B, C, D, E)$ with the set of FDs, $F\{AB \rightarrow CD, A \rightarrow E, C \rightarrow D\}$. Is the decomposition of R into $R_1(A, B, C), R_2(B, C, D)$ and $R_3(C, D, E)$ lossless? Prove. **(5)**

Ans: To find whether the decomposition of R(A, B,C,D,E) with the set of functional dependencies, F={AB → CD, A → E, C → D} into R_1(A,B,C) R_2(B,C, D) and R_3(C,D,E) is lossless or not, we have the following initial table.

	A	B	C	D	E
R1	α_A	α_B	α_C	α_{1D}	β_{1E}
R2	β_{2A}	α_B	α_C	α_D	β_{2E}
R3	β_{3A}	β_{3B}	α_C	α_D	α_E

There is no change in table above for the functional dependencies AB → CD and A →E, but for the functional dependency C → D, all the rows of C have α_C so and the corresponding rows of D can be updated to α_D as there is an α_D in two of such rows. Here the new table is as follows:

	A	B	C	D	E
R1	α_A	α_B	α_C	α_D	β_{1E}
R2	β_{2A}	α_B	α_C	α_D	β_{2E}
R3	β_{3A}	β_{3B}	α_C	α_D	α_E

No change can be made in the table further and from this table we can see that there is no row with all the α's, hence the decomposition is a lossy one.

Q.9 Given R(A,B,C,D,E) with the set of FDs,
F{AB→CD, ABC → E, C → A}
(i) Find any two candidate keys of R
(ii) What is the normal form of R? Justify. **(9)**

Ans:

(i) To find two candidate keys of R, we have to find the closure of the set of attributes under consideration and if all the attributes of R are in the closure then that set is a candidate key. Now from the set of FD's we can make out that B is not occurring on the RHS of any FD, therefore, it must be a part of the candidate keys being considered otherwise it will not be in the closure of any attribute set. So let us consider the following sets AB and BC.

Now $(AB)^+$= ABCDE, CD are included in closure because of the FD AB →CD, and E is included in closure because of the FD ABC → E.

Now $(BC)^+$= BCAED, A is included in closure because of the FD C → A, and then E is included in closure because of the FD ABC → E and lastly D is included in closure because of the FD AB → CD.

Therefore two candidate keys are : AB and BC.

(ii) The prime attributes are A, B and C and non-prime attributes are D and E.

A relation scheme is in 2NF, if all the non-prime attributes are fully functionally dependent on the relation key(s). From the set of FDs we can see that the non-prime attributes (D,E) are fully functionally dependent on the prime attributes, therefore, the relation is in 2NF.

A relation scheme is in 3NF, if for all the non-trivial FDs in F^+ of the form X → A, either X is a superkey or A is prime. From the set of FDs we see that for all the FDs, this is satisfied, therefore, the relation is in 3NF.

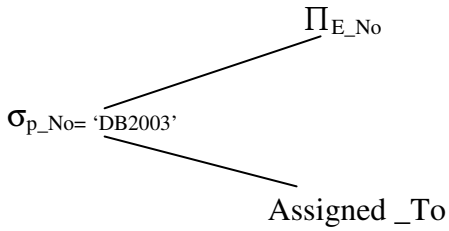
A relation scheme is in BCNF, if for all the non-trivial FDs in F^+ of the form X→ A, X is a superkey. From the set of FDs we can see that for the FD C → A, this is not satisfied as LHS is not a superkey, therefore, the relation is not in BCNF.

Hence, the given relation scheme is in 3NF.

Q.10 How does a query tree represent a relational algebra expression? Discuss any three rules for query optimisation, giving example as to when should each rule be applied. **(8)**

Ans: A query tree is a tree structure that corresponds to a relational algebra expression. It represents the input relations as leaf nodes of the tree, and represents the relational algebra operations as internal nodes. An execution of the query tree consists of executing an internal node operation whenever its operands are available and then replacing that internal node by the relation that results from executing the operation. The execution terminates when the root node is executed and produces the result relation for the query. For example, the query tree for the relational algebra expression

$\prod_{E\_No} (\sigma_{p\_No = \text{'DB2003'}} (\text{Assigned\_To}))$:



The three rules for query optımızatıon are as follows:

- In a cascade (sequence) of $\prod$ operations, all but the last one can be ignored :
  $\prod_1(\prod_2(\ldots\ldots\prod_n (R))) \equiv \prod_1(R)$
- If a selection condition c involves only those attributes A1, A2, …. An in the projection list, the two operations can be commuted :
  $\prod_{A1, A2, \ldots\ldots An} (\sigma_C (R) \equiv \sigma_C (\prod_{A1, A2, \ldots\ldots An} (R)) \ldots\ldots))$
- A conjunctive selection condition can be broken up into a cascade of individual σ operations :
  $\sigma_{C1 \text{ AND } C2 \text{ AND } \ldots\ldots \text{AND } Cn} (R) \equiv \sigma_{C1} (\sigma_{C2}(\ldots.(\sigma_{Cn} (R))\ldots.))$

Q. 11 Consider the following database with primary keys underlined

Project(<u>P_No</u>, P_Name, P_Incharge)

Employee(<u>E_No</u>, E_Name)

Assigned_To(<u>P_No</u>, <u>E_No</u>)

Write the relational algebra for the following :

(i) List details of the employees working on all the projects.
(ii) List E_No of employees who do not work on project number DB2003. **(6)**

Ans:

(i) Employee $\bowtie$(Assigned_To) $\div \prod_{p\_no}$ (Project))
(ii) $\prod_{E\_No}$(Assigned_To) - $\prod_{E\_No}(\sigma_{p\_No = \text{'DB2003'}}$ (Assigned _To))

Q.12 What is a hashing function? What are the properties of a good hashing function? Describe the folding technique for hashing functions. **(7)**

Ans: Hashing function is a technique to store a file on the disk. A hash function is a function which when applied to a value of a record (hash field) gives a hash value, which is the address of the disk block in which the record is stored.

The properties of a good hash function are as follows:

- It should be easy to evaluate.
- The hash value obtained should be within the range of valid addresses for a given file.
- The hash values should be uniformly distributed over the range of addresses so that the collisions are minimum.

Folding technique for hashing functions: It involves applying an arithmetic function such as addition or a logical function like exclusive OR to different parts of a hash field to calculate the hash addresses. For example, the numeric hash field can be split into start, middle and end regions, such that the sum of the lengths of the start and end regions equals the length of the middle region. The start, middle and end regions' digits are added to get a new value, x. Then x mod s, where s is the upper limit for the hash function, gives the hash value.

Q.13 Describe the problems of lost update, inconsistent read and phantom phenomenon which arise as a result of concurrency. **(7)**

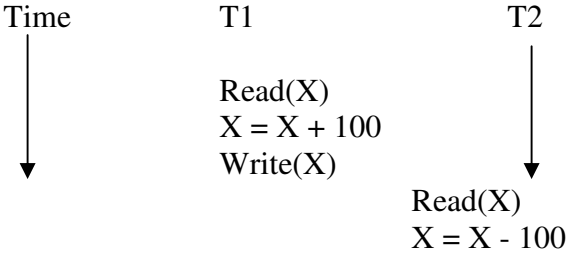
Ans: **Lost update problem:** The problem occurs when two transactions that access the same database items have their operations interleaved in a way that makes the value of some database item incorrect. For example, consider the following interleaved schedule of two transactions:

Time	T1	T2
	Read(X)	
	X = X+ 100	
		Read(X)
		X = X – 100
	Write(X)	
		Write(X)

From the schedule, it can be seen that the updation of x by T1 will be overwritten by T2. If these transactions execute in serial order then the value of x should not change, but with this schedule the value of x will be reduced by 100, which is incorrect.

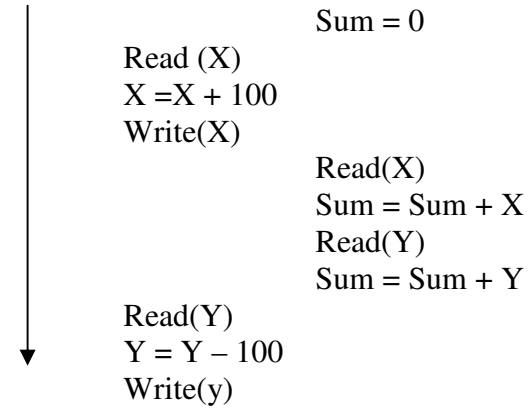
Inconsistent Read or Temporary Update or Dirty Read problem: problem occurs when one transaction updates a database item and then the transaction fails for some reason. The updated value is accessed by another transaction before it is changed back to

the original value. For example, consider the following interleaved schedule of two transactions :



From the schedule, it can be seen that T2 will read the value written by T1, however if T1 fails for some reason before commit then T2 will use the value which is not valid.

Phantom Phenomenon : If a transaction is calculated an aggregate summary function on a number of records while other transactions are updating some of records, the aggregate function may calculate some values before they are updated and others after they are updated. For example, consider the following interleaved schedule of two transactions:



From the schedule, it can be seen that T2 will read the new value of X written by T1 but not that of Y, so the value of Sum will be off by 100 from the actual valid value.

Q.14 Define

(i) Shared locks.
(ii) Serializable schedule.
(iii) Thomas write rule.
(iv) Two phase commit. **(4)**

Ans:

(i) **Shared lock:** During concurrent execution of transactions, before a transaction can access a data item, it has to acquire a lock on it. Now if the transactions only want to read the data item then every transaction can lock the data item in shared mode, such a lock is known as shared lock.

(ii) **Serializable schedule:** An interleaved schedule of more than one transactions is called a serializable schedule, if it is equivalent to some serial schedule of those transactions.

(iii) **Thomas' write rule:** The Thomas' write rule is a modification of timestamp-ordering protocol for concurrency control. Suppose that transaction T_i issues write (Q) and $TS(T_i)$ is its timestamp then the following actions are taken depending on $TS(T_i)$ and the read-write timestamps of the data item Q:

- If $TS(T_i)$<R-timestamp (Q), then the value of Q that T_i is producing was previously needed, and it had been assumed that the value would never be produced. Hence, the system rejects the write operation and rolls T_i back.
- If $TS(T_i)$ < W-timestamp (Q), then T_i is attempting to write an obsolete value of Q. Hence, this write operation can be ignored otherwise, the system executes the write operation and sets W-timestamp(Q) to $Ts(T_i)$

(iv) Two phase commit: To ensure atomicity, all the sites in which a transaction is being executed must agree on the final outcome of the execution. The transaction must either commit at all sites or it must abort at all sites. For this simplest protocol is two-phase commit, which has the voting phase and the decision phase, In the voting phase the sub-transactions are requested to vote on their readiness to commit or abort. In the decision phase the decision as to whether all sub-transactions should commit or abort is made and carried out. The transactions at a site interact with the transaction manager of the site, cooperating in the exchange of messages through which the two –phase commit is executed.

Q.15 Let transactions T1, T2 and T3 be defined to perform the following operations :

T1 : Add one to A
T2 : Double A
T3 : Display A on the screen and then set A to one.
(where A is some item in the database)

Suppose transactions T1, T2 and T3 are allowed to execute concurrently. If A has initial value zero, how many possible correct results are there? Enumerate them. **(10)**

Ans: The following three transactions, manipulating the contents of A, are to execute concurrently:
T1 : Add one to A ie. A = A + 1
T2 : Double A ie. A + 2 * A
T3 : Display A on screen and then set it to 1 ie. A = 1
Assuming A has an initial value zero. Now three transactions can execute concurrently in the following different ways. The tables are showing the different transaction schedules along with the changes they make in the value of A and the value of A displayed by T3.

Transactions Schedule	Current value of A	Display
T1	1	
T2	2	
T3	1	2

Transactions Schedule	Current value of A	Display
T1	1	
T3	1	1
T2	2	

Transactions Schedule	Current value of A	Display
T2	0	
T1	1	
T3	1	1

Transactions Schedule	Current value of A	Display
T2	0	
T3	1	0
T1	2	

Transactions Schedule	Current value of A	Display
T3	1	0
T1	2	
T2	4	

Transactions Schedule	Current value of A	Display
T3	1	0
T2	2	
T1	3	

Q.16 Define and differentiate between the following :-

(i) Deadlock prevention.

(ii) Deadlock detection.

(iii) Deadlock avoidance. **(6)**

Ans:

(i) Deadlock prevention: These protocols ensure that the system will never enter a deadlock state. There are two approaches to deadlock prevention. One approach ensures that no cyclic waits can occur by ordering the requests for locks, or requiring all locks to be acquired together. The other approach is closer to deadlock recovery, and performs transaction rollback instead of waiting for a lock, whenever the wait could potentially result in a deadlock. In this approach two timestamp based techniques are there : wait – die and wound – wait . In wait –die, the older transaction is allowed to wait if it needs data from older transaction. In wound - wait, the younger transaction is rolled back if it needs data from older transaction if older transaction needs data currently held by a younger transaction, however younger transaction is allowed wait if it needs data from older transaction.

(ii) Deadlock detection: If a system does not employ some protocol that ensures deadlock freedom, then a detection and recovery scheme must be used. An algorithm that examines the state of the system is invoked periodically to determine whether a deadlock has occurred. If one has, then the system must attempt to recover from the deadlock.

(iii) Deadlock avoidance : These protocols also ensure that the system will never enter a deadlock state but the way it is done is different from deadlock prevention. In this whenever a transaction requests for some data, the system consider the resources currently allocated to each process and the future requests and releases of each process currently allocated to each process and the future requests and releases of each process, to decide whether the current request can be satisfied or must wait to avoid a possible future deadlock. In this the transactions have to give additional information about how the data will be requested in future.

Q.17 Write short notes on any FOUR of the following:

(i) Two phase locking protocol.
(ii) Audit Trails.
(iii) Query Processing.
(iv) Disadvantages of file based systems.
(v) Query-by-Example.
(vi) B-tree. **(3.5×4=14)**

Ans:

(i) Two Phase Locking Protocol : This is a protocol which is used to ensure serializability of transactions. This protocol requires that each transaction issue lock and unlock requests in two phases :

Growing Phase : A transaction may obtain locks, but may not release any lock.

Shrinking Phase: A transaction may release locks, but may not obtain any new locks. Initially, a transaction is in growing phase. The transaction acquires locks as needed. Once the transaction releases a lock, it enters the shrinking phase, and it can issue no more lock requests. Two-phase locking ensures conflict serializability, but does not ensure freedom from deadlock.

(ii) Audits trails : Audit trail is maintained by the databases for security. An audit trail is a log of all changes (inserts /deletes /updates) to the database, along with information such as which user performed the change and when the change was performed. The audit trails help in security in several ways. For example, if the balance on an account is found to be incorrect, the bank may wish to trace all the updates performed on the account, to find out incorrect (or fraudulent) updates, as well as the persons who carried out the updates. The bank could then also use the audit trails to trace all the updates performed by these persons, in order to find other incorrect or fraudulent updates.

An audit trail is a series of records of computer events, about an operating system, an application, or user activities. It is generated by an auditing system that monitors system activity. Audit trails have many uses in the realm of computer security :

Individual Accountability : An individual's actions are tracked in an audit trail allowing users to be personally accountable for their actions. This deters the users from circumventing security policies. Even if they do, they can be held accountable.

Reconstructing Events : Audit trails can also be used to reconstruct events after a problem has occurred. The amount of damage that occurred with an incident can be assessed by reviewing audit trails of system activity to pinpoint how, when, and why the incident occurred.

Problem Monitoring : Audit trails may also be used as on-line tools to help monitor problems as they occur. Such real time monitoring helps in detection of frauds etc.

Intrusion Detection : Intrusion detection refers to the process of identifying attempts to penetrate a system and gain unauthorized access. Audit trails can help in intrusion detection if they record appropriate events. Determining what events to audit so that audit trails can be used in an effective manner to aid intrusion detection is one of the present research issues being looked into by the research community.

(iii) QueryProcessing: query processing is the procedure of selecting the best plan or strategy to be used in responding to a database request. The plan is then executed to generate a response. The component of the DBMS responsible for generating this strategy is called a query processor. It is a stepwise process. The first step is to transform the query into a standard from. For example, a query in QBE looked into by the research community. **Query** may be transformed into a relational algebraic expression, the

optimization is performed by substituting equivalent expressions for those in the query. In the next step a number of strategies called access pans are generated for evaluating the transformed query. The physical characteristics of the data and any supporting access method are taken into account in generating alternate access pans. The cost of each access plan is estimated and the optimal one is chosen and executed.

(iv) Disadvantages of file based systems : Some of the disadvantages of file based systems are as follows:

- **Data redundancy and inconsistency :** the different application programs may require the same data but may store it in separate files to be used by them only. For example, in a college environment, the address, contact no. etc., may be maintained both by the office staff and the library staff to sent appropriate reminders to the fee defaulters or to the students who have not returned the books on due dates, respectively. If a student changes the residence and that is reflected in only one of the system then it leads to inconsistency.
- **Difficulty in accessing data :** Even if the data is there in the file but if a new type of details are required from the data, they cannot be retrieved unless and until a new application is written for it.
- **Data isolation :** Because data are scattered in various files, and files may be in different formats, writing new application programs to retrieve the appropriate data is difficult.
- **Integrity problems:** The integrity constraint can be applied through the application program, but not directly to the data file. For example, there is a constraint to insure that the balance of an account must not be below Rs. 1000, then through application program this can be done but someone may directly make the changes to the data file and violate this constraint.

(v) Query-By-Example (QBE) : QBE is a query language based on domain calculus and has two dimensional syntax. The queries are written in the horizontal and vertical dimensions of table. Queries are formed by entering an example of a possible answer in a skeleton table as shown in the following diagram:

Relation on Name	**Attribute_name**	**Atrtribute_name 2**		**Attribute_name**
Tuple Operations	Domain var., Constants and Predicates	Domain Var., Constants and predicates	…… ……..	Domain Var., Constants and predicates

Conditions specified in a single row are ANDed and conditions specified in separate rows are ORed. The tuple operations are P., I ., etc., for PRINT, insert commands, resp. The variable names are specified by preceding their name with an underscore. QBE also provides a "conditions" box to specify additional constraints so that the conditions that cannot be added in a Skelton table can be entered in the conditions table. The aggregation operations like max., min, sum. etc. are also provided by QBE.

(vi) B-tree : To gain fast random access to record in a file, we can use an index structure. B-trees take the form of a balanced tree in which every path from the root of the tree to a leaf of the tree is of same length. Each nonleaf node in the tree has between *n/2* and *n* children, where *n* is fixed for a particular tree. A generalized leaf (a) and nonleaf (b) nodes of a B-tree are shown below:

P1	K1	P2	…….	P_{n-1}	K_{n-1}	P_n

(a)

P_1	B_1	K_1	P_2	B_2	K_2	…..	P_{m-1}	B_{m-1}	K_{m-1}	P_m

(b)

In the leaf nodes, for i =1,2,….n-1, pointer P_i points to either a file record with the search key value K_i or to a bucket of pointers, each of which points to a file record with search key value K_i . The bucket key structure is used only if the search key does not from a primary key, and if the file is not sorted on the search key value order. The pointer P_i is used to chain together the leaf nodes in the search key order. In nonleaf nodes, the pointer P_i are the tree pointers that are similar to leaf node pointers, P_i. The pointers *B*-tree, they are used to store the indices. How ever, a variation of them, B^+-trees, are nowadays more frequently used by the databases as they are more easier to implement.

Q.18 Differentiate between

(i) Single value and multiple valued attribute.
(ii) Derived and non-derived attributes.
(iii) Candidate and super key.
(iv) Partial key and primary key.

(8)

Ans:

(i) Single Value and Multivalued attribute – Single value attribute has a single atomic value for an entity and multivalued attribute may have more than one value for an entity. For example, PreviousDegrees of a STUDENT.

(ii) Derived and Non-Derived Attribute – In some cases, two or more attribute values are related, for example, Age and BirthDate attributes of a person. For particular person entity, the value of Age can be determined from the current date and the value of that person's BirthDate. Hence, the attribute Age is called as derived attribute and the attribute BirthDate is called as non-derived or stored attribute.

(iii) Candidate Key and Super Key – A super key is a set of one or more attributes that, taken collectively, allows us to identify uniquely a tuple in the relation. While a candidate key is a minimal superkey, which can be used to uniquely identify a tuple in the relation. In superkey, there may some extra attributes.

(iv) Partial Key and Primary Key – A partial key, also called as discriminator, is the set of attributes that can uniquely identify weak entities that are related to the same owner entity. Primary key is a unique key but not have ***null values***. An entity set can have ***at most one primary key***.

Q.19 Define the basic operations of the relational algebra? **(4)**

Ans: Basic operators of relational algebra are:

1. **Union (∪) -** Selects tuples that are in either P or Q or in both of them. *The duplicate tuples are eliminated.*

$$R = P \cup Q$$

2. **Difference or Minus (-)** - Removes common tuples from the first relation.

$$R = P - Q$$

3. **Cartesian Product or Cross Product (×) -** The cartesian product of two relations is the concatenation of tuples belonging to the two relations and consisting of all possible combination of the tuples.

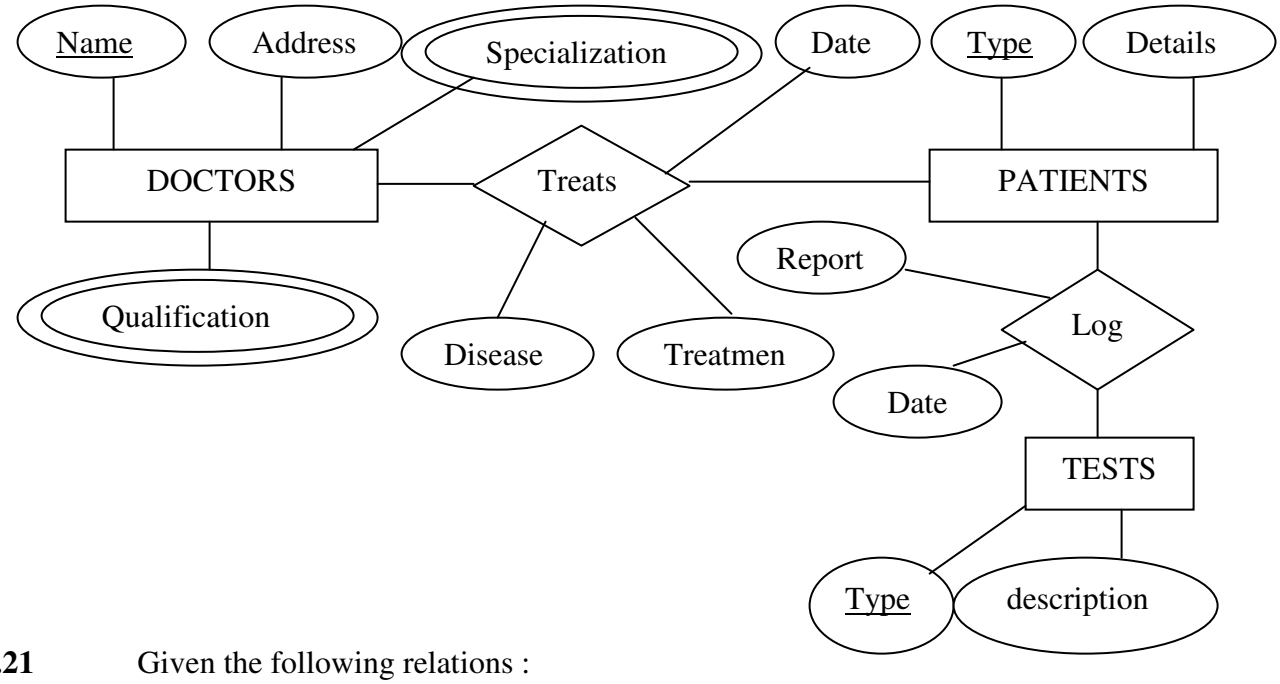
$$R = P \times Q$$

4. **Projection (π) -** The projection of a relation is defined as a projection of all its tuples over some set of attributes, i.e., it yields a ***vertical subset*** of the relation. The projection of a relation T on the attribute A is denoted by T[A] or $\pi_A(T)$,.
5. **Selection (σ) -** Selects only some of the tuples, those satisfy given criteria, from the relation. It yields a ***horizontal subset*** of a given relation.

$$R = \sigma_B(P)$$

Q.20 Construct an ER diagram for a hospital with a set of patients and a set of medical doctors. Associate with each patient a log of the various tests and examinations conducted. **(6)**

Ans:



Q.21 Given the following relations :

vehicle (reg_no, make, colour)
Person (eno, name, address)
Owner (eno, reg_no)

Write expressions in relational algebra to answer the following queries :

(i) List the names of persons who do not own any car.
(ii) List the names of persons who own only Maruti Cars. **(6)**

Ans:

(i) $\pi_{name}(PERSON \bowtie \pi_{eno}(PERSON) - \pi_{eno}(OWNER))$

(ii) $\pi_{name}(((\pi_{eno}(OWNER \bowtie \sigma_{make='maruti'}(VEHICLE))$

$- \pi_{eno}(OWNER \bowtie \sigma_{make\neq'maruti'}(VEHICLE))) \bowtie PERSON)$

Q.22 Define Join and Outer Join and differentiate between them. **(4)**

Ans: Join – It produces all the combinations of tuples from two relations that satisfy a join condition. **Outer Join** - If there are any values in one table that do not have corresponding value(s) in the other, in an equi-join that will not be selected. Such rows

can be forcefully selected by using the *outer join. The corresponding columns for that row will have NULLs.* There are actually three forms of the outer-join operation: left outer join (⟕), right outer join (⟖) and full outer join (⟗).

Q.23 Consider the following relations
Physician (rgno, phyname, addr, phno)
Patient (ptname, ptaddr)
Visits(rgno, ptname, dateofvisit, fees-charged)
Answer the following in SQL :

(i) Define the tables. Identify the keys and foreign keys.

(ii) Create an assertion that the total fees charged for a patient can not be more than Rs.1000/- assuming that patients can visit the same doctor more than once.

(iii) Create a view Patient_visits(name, times) where name is the name of the patient and times is the number of visits of a patient.

(iv) Display the ptname, ptaddr of the patient(s) who have visited more than one physician in the month of May 2000 in ascending order of ptname. **(3.5 x 4 = 14)**

Ans:

(i)
```
CREATE TABLE PHYSICIAN
( RGNO          VARCHAR2(5) PRIMARY KEY,
  PHYNAME       VARCHAR2(15),
  ADDR          VARCHAR2(25),
  PHNO          VARCHAR2(10));
CREATE TABLE PATIENT
(       PTNAME      VARCHAR2(15) PRIMARY KEY,
        PTADDR      VARCHAR2(25));
CREATE TABLE VISITS
( RGNO          VARCHAR2(5) REFERENCES PHYSICIAN(RGNO),
  PTNAME        VARCHAR2(15) REFERENCES, PATIENT(PTNAME),
  DATEOFVISIT   DATE,
  FEE-CHARGED   NUMBER(8,2),
  CONSTRAINT VISITS_PK PRIMARY KEY(RGNO, PTNAME));
```

(ii)
```
CREATE ASSERTION
CHECK ((SELECT SUM(FEES_CHARGED) FROM VISITS
WHERE PTNAME = :NEW.PTNAME AND RGNO = :NEW.RGNO) <= 1000)
```

(iii)
```
CREATE VIEW PATIENT_VISITS (NAME, TIMES) ASSELECT PTNAME,
  COUNT(PTNAME) FROM VISITS GROUP BY PTNAME;
```

(iv)
```
SELECT * FROM PATIENT
    WHERE PTNAME IN (SELECT PTNAME FROM VISITS A, VISITS B
    WHERE A.PTNAME = B.PTNAME AND A.RGNO <> B.RGNO
    AND A.DATEOFVISIT BETWEEN TO_DATE('01-MAY-2000') AND
    TO_DATE('31-MAY-2000')) ORDER BY PTNAME
```

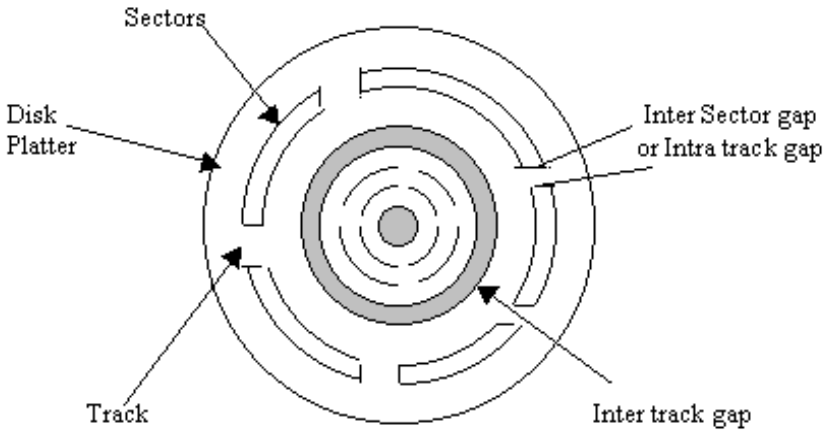
Q.24 What are the advantages of having an index structure? **(2)**

Ans: The index structures typically provide secondary access paths, which provide alternative ways of accessing the records without affecting the physical placement of records on disk. They enable efficient access to records based on the indexing fields that

are used to construct the index. Indexes are used to improve the efficiency of retrieval of records from a data file.

Q.25 What are the physical characteristics of magnetic disks? **(4)**

Ans: A magnetic disk is a circular plate or platter of plastic or metal which is coated with magnetizable material such as iron-oxide on both sides. Data are recorded on the disk surface in the form of tiny invisible magnetized spots (representing 1s) or non-magnetized spots (representing 0s). A conducting coil, named as Head, performs the job of reading and writing on the surface of magnetic disk. Generally, the head remains stationary while the disk rotates below it for reading or writing operation. The surface of the disk is divided into a number of concentric circles called as tracks. Number of tracks varies from disk to disk. Each track is further subdivided into sectors. Each sector of a disk is assigned a unique number. Each sector having the same capacity.

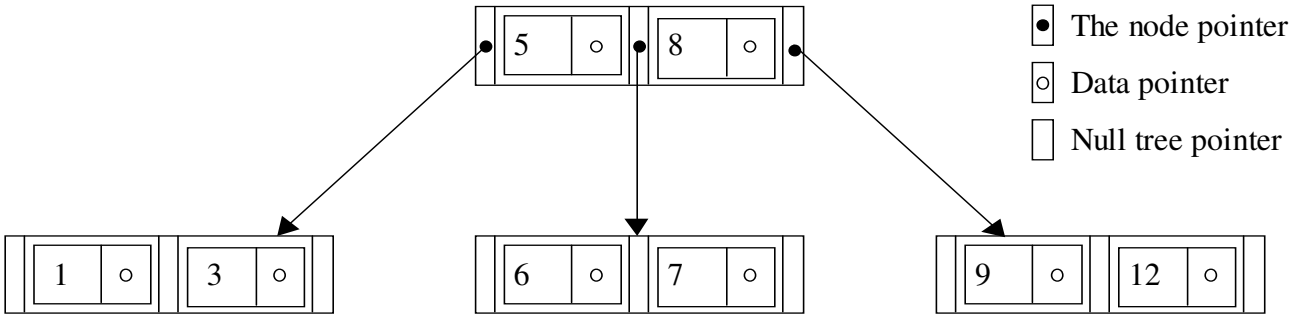


Q.26 Differentiate between B-tree and B^+ tree **(4)**

Ans: A B-tree (or Balanced Tree) is a search tree with additional constraints that ensures that the tree is always balanced and that the space wasted by deletion, if any, never becomes excessive. In a B-tree, every value of the search field appears once at some level in the tree, along with a data pointer. In B^+-tree, data pointers are stored only at the leaf nodes of the tree; hence, the structure of leaf nodes differs from the structure of internal nodes. The leaf nodes of the B^+-tree are usually linked together to provide ordered access on the search field to the records. Some search field values from the leaf nodes are repeated in he internal nodes of the B^+-tree to guide the search.

Q.27 Draw an index structure for B-tree. **(4)**

Ans:

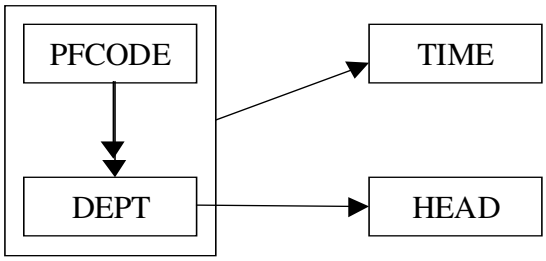


An index structure for B-tree

Q.28 Consider the following relation
Professor (Pfcode, dept, head, time)
It is assumed that
(i) A professor can work in more than one dept.
(ii) The time he spends in each dept is given.
(iii) Each dept has only one head

Draw the dependency diagram for the above relation by identifying the dependencies. **(2)**

Ans:



Dependency Diagram

Q.29 Normalize the relation given Q.28. Justify each step. **(6)**

Ans:
The key of the relation is {PFCODE, DEPT}. Partial functional dependency because DEPT → HEAD. So,
R_1(PFCODE, DEPT, TIME)
R_2(DEPT, HEAD)
R_1 is not in 4NF because of multivalued dependency PFCODE →→ DEPT.
So, final decompositions are:
R_1(PFCODE, DEPT) F_1={PFCODE →→ DEPT}
R_2(PFCODE, TIME)
R_3(DEPT, HEAD) F_2={DEPT → HEAD}

Q.30 Is the decomposition dependency preserving? **(2)**

Ans: We can see that PFCODE, DEPT → TIME is neither in F_1 nor F_2 in the given Q.28 and $(F_1 \cup F_2)^+ \neq F^+$
So, it is not dependency preserving.

Q.31 Define multivalued dependency and 4NF. **(4)**

Ans: **Multivalued Dependency** – Let R be a relation schema and let $\alpha \subseteq R$ and $\beta \subseteq R$. The multivalued dependency

$$\alpha \rightarrow\rightarrow \beta$$

holds on R if, in any legal relation r(R), for all pairs of tuples t_1 and t_2 in r such that $t_1[\alpha] = t_2[\alpha]$, there exist tuples t_3 and t_4 in r such that

$$t_1[\alpha] = t_2[\alpha] = t_3[\alpha] = t_4[\alpha]$$
$$t_3[\beta] = t_1[\beta]$$
$$t_3[R - \beta] = t_2[R - \beta]$$
$$t_4[\beta] = t_2[\beta]$$
$$t_4[R - \beta] = t_1[R - \beta]$$

Fourth Normal Form (4 NF) – A relation schema R is in 4NF with respect to a set D of functional and multivalued dependencies if, for all multivalued dependencies in D^+ of the form $\alpha \rightarrow\rightarrow \beta$, where $\alpha \subseteq R$ and $\beta \subseteq R$, at least one of the following holds:

- $\alpha \rightarrow\rightarrow \beta$ is a trivial multivalued dependency,
- α is a superkey for schema R.

Q.32 What do you understand by transitive dependencies? Explain with an example any two problems that can arise in the database if transitive dependencies are present in the database. **(7)**

Ans: Suppose X, Y, and Z are the set of attributes. If X → Y and Y → Z then the functional dependency X → Z is a transitive functional dependency. In other words, if a nonkey attribute is also functionally dependent on the other nonkey attribute(s), then there is a transitive functional dependency. Transitive functional dependencies are also cause the data redundancies and the other consequences. For example, consider the following relation:

```
CREATE TABLE CONTACTS
( CONTACT_ID          NUMBER(5) PRIMARY KEY,
  L_NAME              VARCHAR(20),
  F_NAME              VARCHAR(20),
  COMPANY_NAME        VARCHAR(20),
  COMPANY_LOCATION    VARCHAR(50))
```

In the above relation, all the nonkey attributes are functionally dependent on the primary key, i.e., CONTACT_ID. But, there is also a transitive functional dependency exists in the relation – CONTACT_ID → COMPANY_LOCATION because CONTACT_ID → COMPANY_NAME and COMPANY_NAME → COMPANY_LOCATION. As a result of transitive dependency, there are anomalies in the above relation as follows:

- **Insertion Anomaly** – A new company cannot be inserted until a contact person has been assigned to that company.

- **Deletion Anomaly** – If a company has only one contact person and is deleted from the table, we will loose the information about that company, as the company information is associated with that person.
- **Update Anomaly** – If a company changes its location we will have to make the changes in all the records where the company name appears.

Q.33 Given R {ABCD} and a set F of functional dependencies on R given as $F = \{AB \rightarrow C, AB \rightarrow D, C \rightarrow A, D \rightarrow B\}$. Find any two candidate keys of R. Show each step. In what normal form is R? Justify. **(7)**

Ans: To find out the candidate keys of a relation schema, R, based on given set of FDs, F, we can compute the closure of X (X^+), where X is the set of attributes. If the X^+ have all the attributes of the relation schema then X is the candidate key of R. With the given set of FDs, it is not possible to derive or access all the other attributes based on single attribute (e.g., $\{A\}^+ = \{A\}$, $\{B\}^+ = \{B\}$, $\{C\}^+ = \{CA\}$, $\{D\}^+ = \{DB\}$). Therefore, we have to chose the composite keys:

1. If we assume X = {AB} then X^+ = {ABCD} because we can determine the attributes C and D from AB as per F.
2. If we assume X = {CD} then X^+ = {CDAB} because we can determine the attributes A and B from C and D respectively as per F.

The other possible candidate keys are: {CB} and {AD}.
If we choose, {CD} as the primary key of the relation R, then FDs C→A and D→B are partial functional dependencies as CD is the key of the relation. Therefore, the given relation R is not in 2NF and therefore first normal form (1 NF).

Q.34 What is a query tree? **(2)**

Ans: A query tree, also called as operator graph, is a tree data structure that corresponds to a relational algebra expression. It represents the input relations of the query as leaf nodes of the tree, and represents the relational algebra operations as internal nodes or root node.

Q.35 What is meant by heuristic optimisation? Discuss the main heuristics that are applied during query optimisation. **(6)**

Ans: In heuristic optimization, heuristics are used to reduce the cost of optimization instead of analyzing the number of different plans to find out the optimal plan. For example, A heuristic optimizer would use the rule 'Perform selection operation as early as possible' without finding out whether the cost is reduced by this transformation. Heuristics approach usually helps to reduce the cost but not always. The main heuristics that are applied during query optimization are:

- Pushes the selection and projection operations down the query tree
- Left-deep join trees – convenient for pipelined evaluation
- Non-left-deep join trees

Q.36 Consider the relations of Q.21. For the query

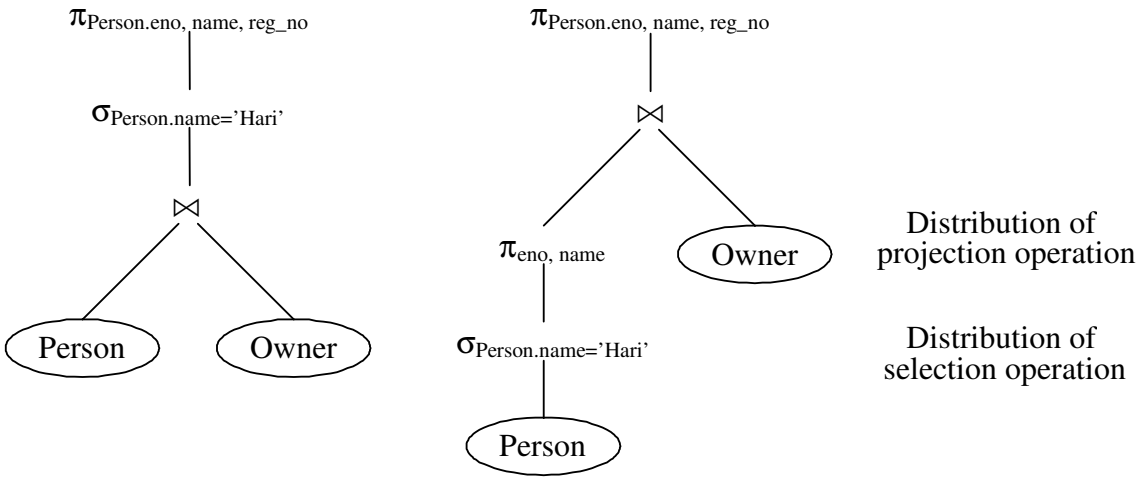
```
Select eno, name, reg_no
From Person, Owner
Where Person.eno = Owner.eno and Person.name = 'Hari'
```

Draw the initial query tree.
Optimise the query and draw the optimised query tree. **(6)**

Ans:

Assuming eno in the select statement refers to persons, the SQL statement will be:
Select Person.eno, name, reg_no From Person, Owner
Where Person.eno = Owner.eno and Person.name = 'Hari'
To draw the query tree the SQL expression is to transformed into relational algebraic expression:

$\pi_{Person.eno,\ name,\ reg\_no}(\sigma_{Person.name='Hari'}(Person \bowtie Owner))$



The initial query tree **The optimised query tree**

Q.37 How does the two phase protocol ensure serializability in database schedules? **(4)**

Ans: A transaction is said to follow the two-phase locking protocol if all locking operations precede the first unlock operation in the transaction. Such transaction can be divided into two phases: and expanding or growing (first) phase and a shrinking (second) phase. The two-phase locking protocol ensures serializability in database schedules. Consider any transaction. The point in the schedule where the transaction has obtained its final lock (the end of its growing phase) is called the lock point of the transaction. Now, transactions can be ordered according to their lock points – this ordering is, in fact, a serializability ordering for the transactions

Q.38 Give a schedule for three transactions T1, T2 and T3 such that the schedule
(i) observes two phase and

(ii) is not deadlock free.

Ans: **(4)**

Transaction T1	Transaction T2	Transaction T3
Write_lock(A)		
Read_item(A)		
	Write_lock(B)	
	Read_item(B)	
Read_Lock(B)		
Read_item(B)		
A := A * B		
Write_item(A)		
Unlock(A)		
Unlock(B)		
	Read_lock(A)	
	Read_item(A)	
	B := A / B	
	Write_item(B)	
	Unlock(B)	
	Unlock(A)	
		Sum := 0
		Read_lock(A)
		Read_lock(B)
		Read_item(A)
		Read_item(B)
		Sum := A + B
		Unlock(A)
		Unlock(B)

Time ↓

Q.39 Describe the algorithm to draw the dependency graph? **(4)**

Ans: **Algorithm to draw Precedence Graph is as follows:**

1. For each transaction T_i participating in schedule S, create a node labeled T_i in the precedence graph.
2. For each case in S where T_j executes a read_item(X) after T_i executes a write_item(X), create an edge ($T_i \rightarrow T_j$) in the precedence graph.
3. For each case in S where T_j executes a write_item(X) after T_i executes a read_item(X), create an edge ($T_i \rightarrow T_j$) in the precedence graph.
4. For each case in S where T_j executes a write_item(X) after T_i executes a write_item(X), create an edge ($T_i \rightarrow T_j$) in the precedence graph.
5. The schedule S is serializable if and only if the precedence graph has no cycles.

Q.40 What is system log? What are the entries? **(3)**

Ans: The system log, which is usually written on stable storage, contains the redundant data required to recover from volatile storage failures and also from errors discovered by the transaction or the database system. System log is also called as log and

has sometimes been called the DBMS journal. It contains the following entries (also called as log records):

- [start_transaction, T]: Indicates that transaction T has started execution.
- [write_item, T, X, old_value, new_value]: Indicates that transaction T has changed the value of database item X from old_value to new_value.
- [read_item, T, X]: Indicates that transaction T has read the value of database item X.
- [commit, T]: Indicates that transaction T has completed successfully, and affirms that its effect can be committed (recorded permanently) to the database.
- [abort, T]: Indicates that transaction T has been aborted.

Q.41 Discuss deferred update technique of recovery. What are the advantages? **(6)**

Ans: The deferred update techniques do not physically update the database on disk until after a transaction reaches its commit point; then the updates are recorded in the database. In other words, deferred update techniques postpone any actual updating of the database on disk until a transaction reaches its commit point. The transaction force-writes the log to disk before recording the updates in the database.

Advantages:

• If a transaction fails before reaching its commit point, it will not have changed the database in any way, so UNDO is not needed. Deferred update can lead to a recovery algorithm known as NO-UNDO/REDO.

• This approach, when used with certain concurrency control methods, is designed never to require transaction rollback, and recovery simply consists of redoing the operations of transactions committed after the last checkpoint from the log.

Q.42 How does the system cope up with a record crash when recovery is going on after the first crash. **(2)**

Ans: In a system the undo and redo operations are required to be idempotent to cope up with a record crash when recovery is going on after the crash. Due to the idempotent property, recovery process, while in the process of undoing or redoing the actions of a transaction, may fail without a trace, and this type of failure can occur any number of times before the recovery is completed successfully.

Q.43 Define check point and its impact on data base recovery. **(3)**

Ans: In a large on-line database system there could be hundreds of transactions handled per minute. The log for this type of database contains a very large volume of information. A scheme called checkpoint is used to limit the volume of log information that has to be handled and processed in the event of a system failure involving the loss of volatile information. The checkpoint scheme is an additional component of the logging scheme. A checkpoint operation, performed periodically, copies log information onto stable storage. For all transactions active at checkpoint, their identifiers and their database modification actions, which at that time are reflected only in the database buffers, will be propagated to the appropriate storage.

Q.44 Write short notes on

a Wait for graph.

b Direct file organization.

c Canonical cover.
d. Timestamp ordering. **(3.5 x 4)**

Ans:

a The wait-for-graph is a directed graph and contains nodes and directed arcs; the nodes of the graph are active transactions. An arc of the graph is inserted between two nodes if there is a data-item is required by the node at the tail of the arc, which is being held by the node at the head of the arc. If there is a transaction T_i, waiting for a data-item that is currently allocated and held by transaction T_j, then there is a directed arc from the node for transaction T_i, to the node for transaction T_j. The wait-for-graph is used for detection of deadlock in the system. If there is cycle exits in the graph, all the transaction (represented as nodes) are in deadlock, which are in the cycle.

b. The direct file organization is used to improve the performance in accessing the records as compared to sequential file organization. In direct file organization, the key value is mapped directly to the storage location. The usual method of direct mapping is by performing some arithmetic manipulation of the key value. This process is called hashing. When using this organization, an application such as on-line transaction processing system can be designed so that centralized data are not only instantly accessible but are always up-to-date.

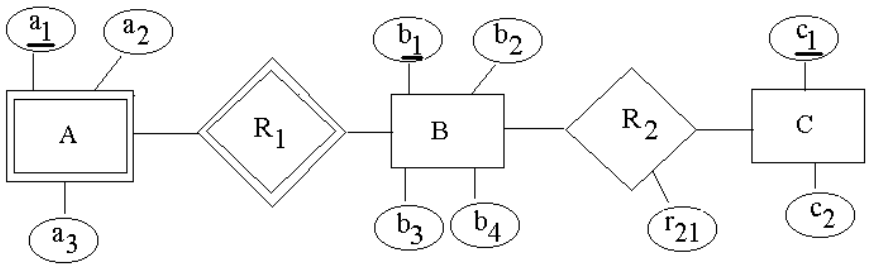
c. A canonical cover F_c for F is a set of dependencies such that F logically implies all dependencies in F_c, and F_c logically implies all dependencies in F. F_c must have the following properties:

- No functional dependency in Fc contains extraneous attribute
- Each left side of a functional dependency in F_c is unique. That is, there are no two dependencies $\alpha_1 \rightarrow \beta_1$ and $\alpha_2 \rightarrow \beta_2$ in F_c such that $\alpha_1 = \alpha_2$.

F_c is minimal in certain sense – it does not contain extraneous attributes, and it combines functional dependencies with the same left side. It is cheaper to test Fc than it is to test F itself.

d. In timestamp-based method, a serial order is created among the concurrent transaction by assigning to each transaction a unique nondecreasing number. The usual value assigned to each transaction is the system clock value at the start of the transaction, hence the name timestamp ordering. This value then can be used in deciding the order in which the conflict between two transactions is resolved. A transaction with a smaller timestamp value is considered to be an "older" transaction than another transaction with a larger timestamp value.

Q.45 Convert the following ER – diagram into a relational database (the primary keys are underlined): **(8)**



Ans: The relational database schema for the given ER diagram is as follows:
$A(\underline{a_1, b_1}, a_2, a_3)$
$B(\underline{b_1}, b_2, b_3, b_4)$
$C(\underline{c_1}, c_2)$
$D(\underline{b_1, c_1}, r_{21})$

Q.46 List two restrictions that are applied on the modification (updation, insertion or deletion) of base tables through view. With the help of examples, explain why those restrictions are required. **(6)**

Ans: The two restrictions are:

- If a view does not contain primary key and/or at least one of the attributes with the NOT NULL constraint that do not have default values specified, is not updatable. This restriction is required because if we are inserting any row in the view than all attributes not included in view of the base relation are will be represented by the NULLs if any default value is not specified. In that case, if any attribute is set as NOT NULL or primary key, then it must not be NULL, and the insertion will be violating the constraints.
- If a view defined using grouping and/or aggregate functions, is not updatable. This restriction is also required because, if we want to modify the base relation through view then we are not in the position to identify which tuple(s) will be modified in the group in what way.

For Example:
Employee:

EmpNo	Name	Job	Salary
1	Ramesh	Salesman	5000
2	John	Clerk	3000
3	Sanjeet	Manager	10000
4	Praveen	Salesman	4500
5	Rakesh	Manager	12000

View1: Create View Name_Job as Select Name, Job from Employee
View2: Create View Job, Sum(salary) from Employee Group By Job
In the view1 if we want to insert a new record, then the primary key constraint violates because primary key does not accept NULL. In the view2 if we want to make any modification (insertion, updation, or deletion) then it is difficult to identify the corresponding tuples of base relation from the view for the modification.

Q.47 Consider the following relations with key underlined
Customer (C#, Cname, Address)
Item (I#, Iname, Price, Weight)
Order (O#, C#, I#, Quantity)
Write SQL queries for the following:

a. List the names of customers who have ordered items weighing more than 1000 and only those.
b. List the names of customers who have ordered atleast one item priced over Rs.500.
c. Create a view called "orders" that has the total cost of every order. **(9)**

Ans:

(a)
```
SELECT CNAME FROM CUSTOMER WHERE C# IN
        (SELECT C# FROM ORDER
         MINUS
         SELECT C# FROM ORDER, ITEM
           WHERE ORDER.I# = ITEM.I# AND WEIGHT <= 1000)
```

(b)
```
SELECT CNAME FROM CUSTOMER WHERE C# IN
        (SELECT C# FROM ORDER WHERE I# IN
          (SELECT I# FROM ITEM WHERE PRICE > 500))
                    OR
SELECT DISTINCT CNAME FROM CUSTOMER, ORDER, ITEM
WHERE CUSTOMER.C# = ORDER.C# AND ORDER.I# = ITEM.I#
      AND PRICE > 500
```

(c)
```
CREATE VIEW ORDERS (O#, TOTALCOST) AS
SELECT O#, SUM(QUANTITY * PRICE)
FROM ORDER, ITEM WHERE ORDER.I# = ITEM.I#
GROUP BY O#
```

Q.48 List the Armstrong's axioms for functional dependencies. What do you understand by soundness and completeness of these axioms? **(5)**

Ans: **The Armstrong's axioms are:**

- F1: Relexivity: If X is a set of attributes and $Y \subseteq X$ then $X \rightarrow Y$ holds.
- F2: Augmentation: If $X \rightarrow Y$ holds and Z is a set of attributes then $XZ \rightarrow YZ$.
- F3: Transitivity: $\{X \rightarrow Y, Y \rightarrow Z\} \models \{X \rightarrow Z\}$

Soundness: By sound, we mean that a given set of functional dependencies F specified on a relation schema R, any dependency that we can infer from F by using F1 through F3 holds in every relation state r of R that satisfies the dependencies in F.

Completeness: By complete, we mean that using F1 through F3 repeatedly to infer dependencies until no more dependencies can be inferred results in the complete set of all possible dependencies that can be inferred from F.

Q.49 Consider the following relations

Employee (E#, Ename, salary, Bdate, D#)
Department (D#, Dname, mgremp#, Location)
Dependent (E#, DependentName)

a. Write tuple calculus queries for the following:

(i) List the names of mangers who have at least one dependent.

(ii) List the names of employees working for 'research' department. **(6)**

b. Write QBE queries for the following:

(i) List all the employees who earn more than the average salary of all employees.

(ii) Increase the salary of managers by 10%.

(iii) List the names of all employees working in Delhi.

(iv) Change the location of all departments to "Mumbai" which have location as "Bombay" **(8)**

Ans:

(a) (i) List names of managers who have at least one dependent.

{m[Ename] | m ∈ EMPLOYEE ∧ ∃u, t (t ∈ DEPENDENT
∧ u ∈ DEPARTMENT ∧ t[E#] = u[mgremp#] ∧ m[E#] = u[mgremp#])}

(ii) List the names of employees working for 'research' department.

{e[Ename] | e ∈ EMPLOYEE ∧ ∃d (d ∈ DEPARTMENT
∧ d[Dname] = 'research' ∧ e[D#] = d[D#])}

(b) (i) List all the employees who earn more than the average salary of all employees.

	E#	Ename	Salary	Bdate	D#
P.	EX	NX	SX AVG.ALL.SY	BX	DX

Conditions
SX > AVG.ALL.SY

(ii) Increase the salary of managers by 10%.

DEPARTMENT	D#	Dname	mgremp#	Location
			MX	

EMPLOYEE	E#	Ename	Salary	Bdate	D#
U.	MX		SX SX * 1.1		

(iii) List the names of all employees working in Delhi

DEPARTMENT	D#	Dname	mgremp#	Location
	DX			Delhi

EMPLOYEE	E#	Ename	Salary	Bdate	D#
		P.NX			DX

(iv) Change the location of all departments to "Mumbai" which have locations as "Bombay."

DEPARTMENT	D#	Dname	mgremp#	Location
U.				Bombay Mumbai

Q.50 Consider the following relation:

A	B	C
10	b1	c1
10	b2	c2
11	b4	c1
12	b3	c4
13	b1	c1
14	b3	c4

Given the above state, which of the following dependencies hold in the above relation at this point of time? If the dependency does not hold, explain why, by specifying the tuples that cause the violation

(i) $A \rightarrow B$ (ii) $B \rightarrow C$
(iii) $C \rightarrow B$ (iv) $B \rightarrow A$
(v) $C \rightarrow A$.

Does the above relation have a potential candidate key? If it does, what is it? If it does not, why not? **(7)**

Ans:

S. No.	FD	Valid / Invalid	Tuples that cause the violation
(i)	A → B	Invalid	1st and 2nd
(ii)	B → C	Valid	
(iii)	C → B	Invalid	3rd and 1st
(iv)	B → A	Invalid	1st and 5th, 4th and 6th
(v)	C → A	Invalid	1st, 3rd, and 5th

Yes, the given relation has the potential candidate keys. The candidate keys are: {AB}, and {BC} because $\{AB\}^+ = \{ABC\}$ and $\{BC\}^+ = \{BCA\}$.

Q.51 Find 3NF decomposition of the relation scheme,
R = {Faculty, Dean, Dept, Chairperson, Professor, Rank, Student} with the set of functional dependencies,

F = {Faculty → Dean
Professor → Rank, Chairperson
Student → Department, Faculty,Dean
Professor, Rank → Department, Faculty}
Department → Chairperson
Department → Faculty
Dean → Faculty

(7)

Ans: To find out 3NF decomposition of a relation we use canonical cover (F_c).

Requirements for canonical cover (F_c):

1. Each FD in Fc must be simple.

F' = {Faculty → Dean, Department → Chairperson, Professor → Rank, Professor → Chairperson, Department → Faculty, Student → Department, Student → Faculty, Student → Dean, Dean → Faculty, ProfessorRank → Department, ProfessorRank → Faculty}

2. Each FD in Fc must be left reduced.

As given in F, the attribute Rank is functionally dependent on the Professor. Therefore the FDs ProfessorRank → Department and ProfessorRank → Faculty can be left reduced to Professor → Department and Professor → Faculty respectively based on the Augmentation axiom.

F'' = {Faculty → Dean, Department → Chairperson, Professor → Rank, Professor → Chairperson, Department → Faculty, Student → Department, Student → Faculty, Student → Dean, Dean → Faculty, Professor → Department, Professor → Faculty}

3. Each FD in Fc must be non-redundant.

The FDs Professor → Chairperson, Student → Faculty, Student → Dean and Professor → Faculty are redundant and have to be removed based on the Transitivity axiom.

F_c = {Faculty → Dean, Department → Chairperson, Professor → Rank, Department → Faculty, Student → Department, Dean → Faculty, Professor → Department}

To make lossless join decompositions, it is required to preserve the key of the original relation. The key of the relation R is {Professor, Student}. The 3NF decompositions of the given relation are:

R_1 = {Faculty, Dean} F_1= {Faculty → Dean, Dean → Faculty}

R_2 = {Department, Chairperson, Faculty} F_2 = {Department → Chairperson,Faculty}

R_3 = {Professor, Rank, Department} F_3 = {Professor → Rank,Department}

R_4 = {Student, Department} F_4 = {Student → Department }

R_5 = {Professor, Student} F_5 = {ProfessorStudent → ϕ}

Q.52 Set R(A,B,C) and S(B,C,D) be the relations:

R:

A	**B**	**C**
a	c	c
a	e	c
a	c	d
b	d	d

S:

B	**C**	**D**
c	c	a
d	c	a
e	d	b

Compute the following for the relations above:

(i) $R \div \pi_c(s)$ **(ii)** $\pi_{R:B,\,S:C}(\sigma_{A=D}(R \times S))$

(iii) $R \cup S$. **(8)**

Ans: (i) $R \div \pi_c(S)$

$\pi_c(S)$

C
c
d

$R \div \pi_c(S)$

A	**B**
a	c

(ii) $\pi_{R.B,\,S.C}(\sigma_{A=D}(R \times S))$

R.B	**S.C**
c	c
e	c
d	d

(iii) $R \cup S$

A	**B**	**C**
a	c	c
a	e	c
a	c	d
b	d	d
c	c	a
d	c	a
e	d	b

Q.53 Consider the decomposition of relation scheme, shipping = (Ship, Capacity, Date, Cargo, Value) with the set of functional dependencies, F = {Ship →Capacity

Ship, Date $\rightarrow$Cargo, Cargo, Capacity $\rightarrow$Value} into R_1={Ship, Capacity} with F_1 = {Ship $\rightarrow$Capacity} and R_2={Ship, Date, Cargo, Value} with F_2={Ship, Date $\rightarrow$Cargo, Value}.
Is this decomposition in BCNF? Is this decomposition lossless and dependency preserving? Justify your answers. **(6)**

Ans: The given decomposition of the given relation is in BCNF because there is no overlapped composite keys in the relation. The decomposition is lossless because the key of the original relation {Ship,Date} is preserved in the decomposition R2. But the decomposition is not dependency preserving because the FD Cargo,Capacity → Value is not implied by the set of FDs {Ship → Capacity, Ship,Date → Cargo,Value}

Q.54 What are deferred modification and immediate modification technique for recovery? How does recovery takes place in case of a failure in these techniques? **(7)**

Ans: **Deferred Update** – The deferred update techniques do not physically update the database on disk until after a transaction reaches its commit point; then the updates are recorded in the database. In other words, deferred update techniques postpone any actual updating of the database on disk until a transaction reaches its commit point. The transaction force-writes the log to disk before recording the updates in the database.
Immediate Update – The immediate update techniques may apply changes to the database on disk before the transaction reaches a successful conclusion. However, these changes are typically recorded in the log on disk by force writing before they are applied to the database, making recovery still possible.
Recovery – In deferred update technique, if a transaction fails before reaching its commit point, it will not have changed the database in any way, so UNDO is not needed. Deferred update can lead to a recovery algorithm known as NO-UNDO/REDO. This approach, when used with certain concurrency control methods, is designed never to require transaction rollback, and recovery simply consists of redoing the operations of transactions committed after the last checkpoint from the log. In immediate update technique, if a transaction fails after recording some changes in the database but before reaching its commit point, the effect of its operations on the database must be undone; that is, the transaction must be rolled back. This technique requires both operations and is used most often in practice. Therefore, it is also known as UNDO/REDO.

Q.55 What are the two integrity rules? Explain with examples how these rules are important to enforce consistent database states. **(7)**

Ans: The two integrity rules are: Entity Integrity Rule and Referential Integrity Rule.
Entity Integrity Rule – If the attribute A of relation R is a prime attribute of R then A cannot accept null values.
Referential Integrity Rule – In referential integrity, it is ensured that a value that appears in one relation for a given set of attributes also appears for a certain set of attributes in another relation.

For example:

STUDENT

Enrl No	Roll No	Name	City	Mobile
11	17	Ankit Vats	Delhi	9891663808
15	16	Vivek Rajput	Meerut	9891468487
6	6	Vanita	Punjab	
33	75	Bhavya	Delhi	9810618396

GRADE

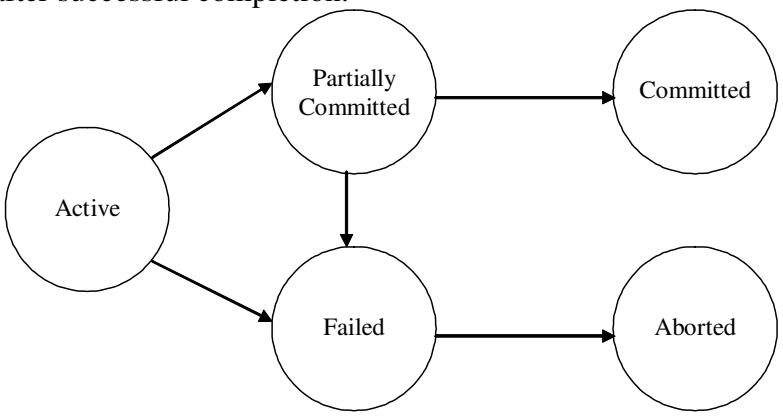
Roll No	Course	Grade
6	C	A
17	VB	C
75	VB	A
6	DBMS	B
16	C	B

- **Roll No** is the primary key in the relation STUDENT and **Roll No + Course** is the primary key of the relation GRADE - (Entity Integrity). Primary keys ensure that any prime attribute must not be NULL.
- **Roll No** in the relation GRADE (child table) is a foreign key, which is referenced from the relation STUDENT (parent table) - (Referential Integrity). Referential integrity here ensures that the values in the foreign key must belongs to it parent key or it may be entirely NULL.

Q.56 What are the various states through which a transaction passes through in its lifetime? Briefly discuss all the events that causes transition from one state to another. **(7)**

Ans: A transaction can be considered to be an atomic operation by the user, but actually it goes through a number of states during its lifetime as given follows:

- **Active**, the initial state; the transaction stays in this state while it is executing
- **Partially Committed**, after the initial statement has been executed
- **Failed**, after the discovery that normal execution can no longer proceed
- **Aborted**, after the transaction has been rolled back and the database has been restored to its state prior to the start of the transaction
- **Committed**, after successful completion.



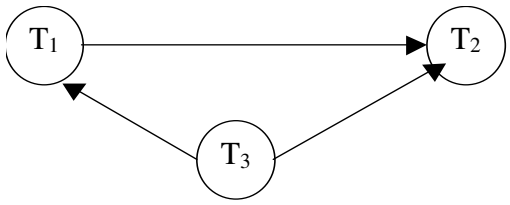
State Diagram of a Transaction

It is assumed that before a transaction starts, the database is in a consistent state. A transaction starts when the first statement of the transaction is executed; it becomes active. After any outcome, the database will be considered in consistent state.

Q.57 Consider three transactions : T_1, T_2 and T_3. Draw the precedence graph for the following schedule consisting of these three transactions and determine whether it is serializable. If so, give its serial order(s). **(7)**

Time	T_1	T_2	T_3
t_1 :			read(Y)
t_2 :			read(Z)
t_3 :	read(X)		
t_4 :	write(X)		
t_5 :			write(Y)
t_6 :			write(Z)
t_7 :		read(Z)	
t_8 :	read(Y)		
t_9 :	write(Y)		
t_{10} :		read(Y)	
t_{11} :		write(Y)	
t_{12} :		read(X)	
t_{13} :		write(X)	

Ans:



Precedence Graph

The given schedule is serializable. The serial orders of the transactions are: T_3, T_1, and T_2.

Q.58 Which sorting technique is used to sort databases, whose sizes are very big? Give one such algorithm. Why do sorting techniques like quicksort, insertion sort, etc. not applied on very big databases? **(7)**

Ans: To sort a large database or file, the sort-merge (or k-way merge) algorithm is widely used. This is one of the external sorting algorithms. External sorting algorithm refers to sorting algorithms that are suitable for large file of records stored on disk that do not fit entirely in main memory, such as most database files. The sort-merge algorithm starts by sorting small subfiles -–called runs – of the main file and then merges the sorted runs, creating larger sorted subfiles that are merged in turn. Like other database algorithms, it requires buffer space in main memory, where the actual sorting and

merging of the runs is performed. This algorithm consists of two phases: (1) the sorting phase, and (2) the merging phase.

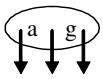
The sorting techniques like quick sort, insertion sort, etc. are not applied on large databases because these are internal sorting algorithms, which requires buffer space in main memory. The space complexity of these algorithms are O(n) and require a large buffer space in main memory to sort the large database. Therefore, internal sorting algorithms are not applied on large databases.

Q.59 Insert the keys below in the order stated into an initially empty B-tree of order 3. Show all intermediate steps

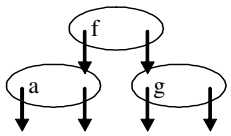
a g f b k d h m j e s i r **(7)**

Ans:

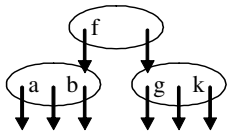
After inserting a and g



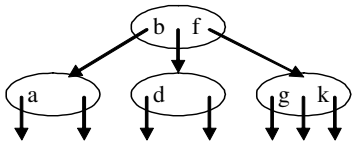
After inserting f



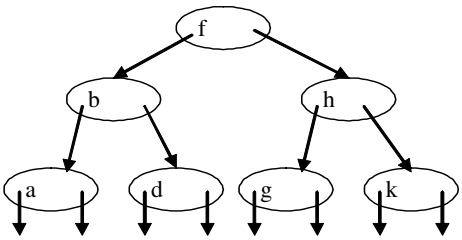
After inserting b and k



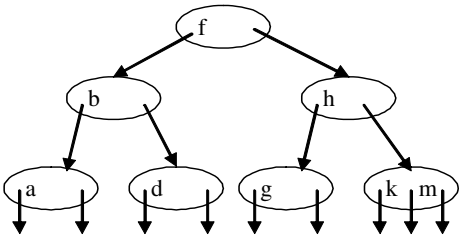
After inserting d



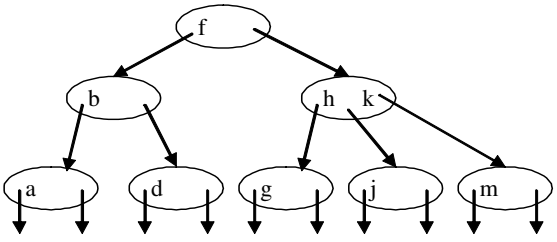
After inserting h



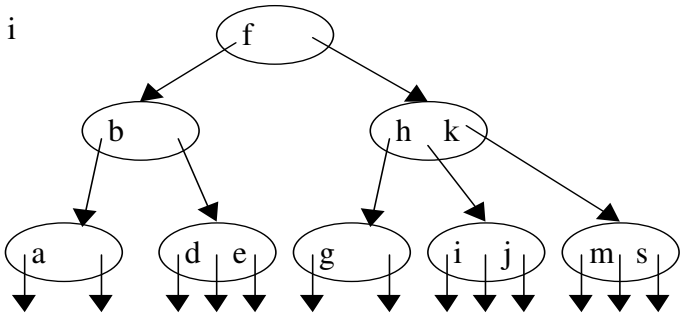
After inserting m



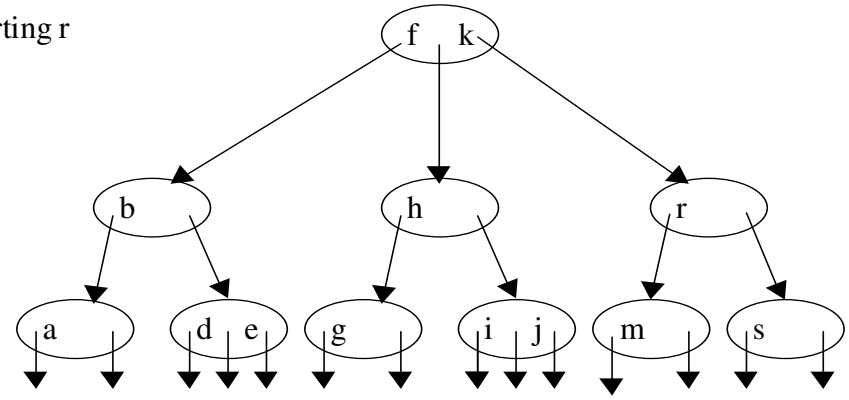
After inserting j



After inserting e, s, and i



After inserting r



Q.60 What are wait-for-graphs? Give the algorithm to construct a wait-for-graph from a given schedule of transactions? How can deadlocks be detected from wait-for-graphs? **(7)**

Ans: The wait-for-graph is a directed graph and contains nodes and directed arcs; the nodes of the graph are active transactions. An arc of the graph is inserted between two nodes if there is a data-item is required by the node at the tail of the arc, which is being held by the node at the head of the arc.

Algorithm to construct wait-for-graph is as follows:

1. For each transaction T_i active at the time of deadlock detection, create a node labeled T_i in the wait-for-graph.
2. For each case, if there is a transaction T_i, waiting for a data-item that is currently allocated and held by transaction T_j, then there is a directed arc from the node for transaction T_i, to the node for transaction T_j.
3. For each case, if there is a directed arc from the node for transaction T_i, to the node for transaction T_j and transaction Tj released the lock, then remove the arc between them.
4. There no deadlock if the wait-for-graph is acyclic.

The wait-for-graph is used for detection of deadlock in the system. If there is any cycle exits in the wait-for-graph, all the active transaction (represented as nodes) are in deadlock, which are in the cycle.

Q.61 What is the need of a log in a DBMS? Briefly describe the various types of records that are normally present in a log. **(7)**

Ans: The system log, which is usually written on stable storage, contains the redundant data required to recover from volatile storage failures and also from errors discovered by the transaction or the database system. System log is also called as log and

has sometimes been called he DBMS journal. It contains the following entries (also called as log records):

- [start_transaction, T]: Indicates that transaction T has started execution.
- [write_item, T, X, old_value, new_value]: Indicates that transaction T has changed the value of database item X from old_value to new_value.
- [read_item, T, X]: Indicates that transaction T has read the value of database item X.
- [commit, T]: Indicates that transaction T has completed successfully, and affirms that its effect can be committed (recorded permanently) to the database.
- [abort, T]: Indicates that transaction T has been aborted.

Q.62 Write short notes on (any **FOUR**) of the following:

(i) Deadlock recovery measures.
(ii) Query optimisation.
(iii) Hashing techniques.
(iv) Two phase locking protocol.
(v) ANSI SPARC architecture.
(vi) Domain calculus. **(3.5 x 4=14)**

Ans:

(i) To recover from deadlock, the cycle in the wait-for-graph must be broken. The common method of doing this is to rollback one or more transactions in the cycles until the system exhibits no further deadlock situation. The selection of the transaction to be rolled back is based on the following deadlock recovery measures:

- The progress of the transaction and the number of data-items it has used and modified. It is preferable to rollback a transaction that has just started or has not modified any data-item.
- The amount of computing remaining for the transaction and the number of data items that have yet to be accessed by the transaction. It is preferable not to rollback a transaction it has almost run to completion and/or it needs very few additional data-items before its termination.
- The relative cost of rolling back a transaction. It is preferable to roll back a less important or non-critical transaction.

(ii) The query parser typically generate a standard initial tree to correspond to an SQL query, without doing any optimization. Such a canonical query tree represents a relational algebra expression that is very inefficient if executed directly. A query typically has many possible execution strategies, and the process of choosing a suitable one for processing a query is known as query optimization. The main guidelines that are applied during query optimization are:

- Combine a cascade of selections
- Combine a cascade of projections
- Commute selection and projection
- Commuting selection with join or cartesian product Commuting projection with join or cartesian product
- Commute projection with set operations
- Commute selection with set operations

(iii) The hashing techniques can be classified as:

Static Hashing Techniques – In this technique, the data can be viewed a collection of buckets, with one primary page and possibly additional overflow pages per bucket. A file consists of buckets 0 through N-1, with one primary page per bucket initially and additional overflow pages chained with bucket, if required later. Buckets contain data entries (or data records). A major drawback of the static hashing is that the hash address space is fixed. Hence, it is difficult to expand or shrink the file dynamically.

Dynamic Hashing Techniques – There are two schemes in it: extendible hashing – stores an access structure in addition to the file , and hence is somewhat similar to indexing. The main difference is that the access structure is based on the values that the result after application of the hash function to the search field. The second one is, linear hashing – does not require additional access structures.

(iv) A transaction is said to follow the two-phase locking protocol if all locking operations precede the first unlock operation in the transaction. Such transaction can be divided into two phases: and expanding or growing (first) phase, during which new locks on items can be acquired but none can be released; and a shrinking (second) phase, during which existing locks can be released but no new locks can be acquired. This two-phase protocol can ensure the serializability because all the data items required by a transaction will be locked at the beginning of the transaction and if other transaction wants to use those data items then it has to wait until they become unlocked.

(v) The three-schema architecture is also known as ANSI SPARC architecture. The goal of the three-schema architecture is to separate the user applications and the physical database. The view at each of these levels is described by a schema. The processes of transforming requests and results between levels are called mappings. In this architecture, schemas can be defined at the following three levels:

- **External Level or Subschema –** It is the highest level of database abstraction where only those portions of the database of concern to a user or application program are included. Any number of user views may exist.
- **Conceptual Level or Conceptual Schema -** At this level of database abstraction all the database entities and the relationships among them are included. There is only one conceptual schema per database.
- **Internal Level or Physical Schema –** It is closest to the physical storage method used.

(**vi)** Domain calculus is one of the type of the relational calculus. The formal specification of the domain calculus was proposed after the development of the QBE system. Domain calculus has the variables range over single value from domains of attributes. To form a relation degree *n* for a query result, we must have *n* domain variables – one for each attribute. An expression of the domain calculus is of the form

$$\{X \mid F(X)\}$$

Where F is a formula on X and X represents a set of domain variables. The expression characterizes X such that F(X) is true.

Q.63 Describe five main functions of a database administrator. **(5)**

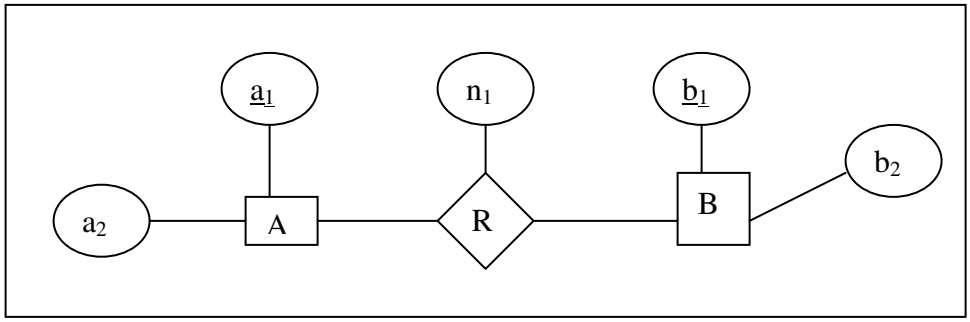
Ans: A **database administrator** (**DBA**) is a person who is responsible for the environmental aspects of a database. In general, these include:

- **Recoverability** - Creating and testing Backups
- **Integrity** - Verifying or helping to verify data integrity
- **Security** - Defining and/or implementing access controls to the data
- **Availability** - Ensuring maximum uptime

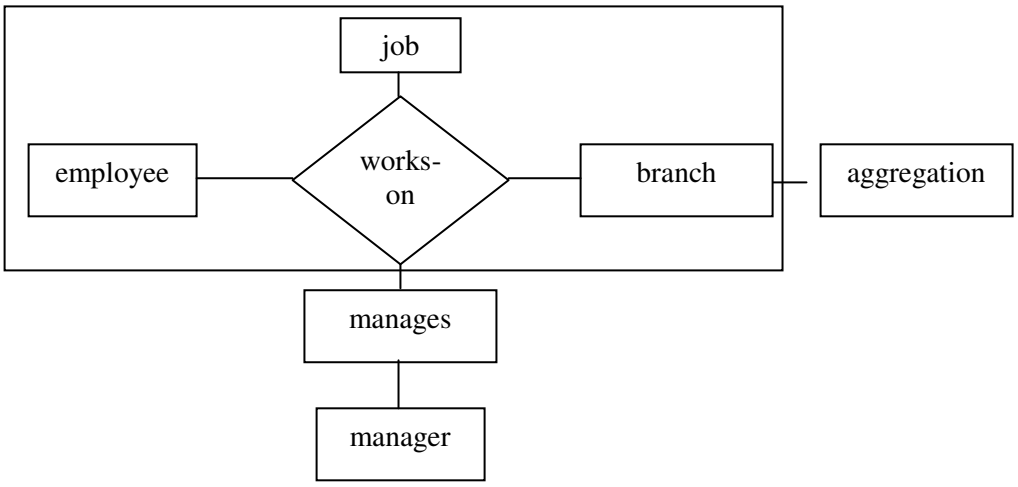
- **Performance** - Ensuring maximum performance
- **Development and testing support** - Helping programmers and engineers to efficiently utilize the database.

Q.64 Define the following with respect to an E-R diagram. Explain the manner in which each is mapped to a table. Illustrate with an example.
(i) Relationship set. **(ii)** Aggregation.
(iii) Multivalued attribute. **(9)**

Ans:
(i) Relationship set: It is a set of relationships of same type. Formally, it is a mathematical relation on n >=2 entity sets. If E_1, E_2-----,E_n are entity sets, then a relationship set R is a subset of { (e1, e2,-----, en) | e1, $e_1 \varepsilon E_1$, e2 εE_2 , ----, $e_n \varepsilon E_n$ } where (e_1,e_2,----,e_n) is a relationship
Mapping to a table: Let R be a relationship set, let a_1, a_2,-----, a_m be set of attributes formed by the union primary keys of each of entity sets participating in R, and let the descriptive attributes (if any) of R be b_1,b_2,------b_n. Then the table corresponding to R will have one column for each attribute of the set: { a_1, a_2, ----, a_m) U { b_1, b_2,----b_n}
Eg, given an ER diagram as



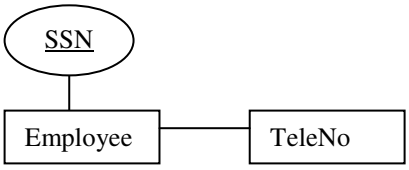
The table corresponding to R will be R(a_1, b_1, r_1)
ii) Aggregation: It is an abstraction through which relationships are treated as higher level entities.
Mapping to a table: This is explained through following example. Consider the E-R diagram given below:



The table for relationship set manages between aggregation onworks-on and entity set manager includes a column for each attribute in primary keys of entity set manages and relationship set works. It would also include a column for any descriptive attribute if they exist, of relationship manages.

iii) Multivalued attribute: An attribute that is having more than one values for a given attribute. Eg, Telephone no. of an employee

Mapping to a table: For a multivalued attribute M, a table is created with a column C that corresponds to M and columns corresponding to the primary key of the entity set or relationship set of which M is an attribute. Eg, Given ER diagram as below:



The corresponding table will be Telephone (SSN, TeleNo)

Q.65 Consider the following relations with primary keys underlined.
Salesperson (SNo, Sname, Designation)
Area (ANo, Aname, ManagerNo)
Product (PNo, Pname, Cost)
SAP (SNo, ANo, PNo)

(a) Define the schema in SQL specify the attributes, and keys assuming that ManagerNo is a foreign key. Specify the constraint that the cost of a product cannot be greater than Rs.10000/-. **(5)**

(b) Answer using SQL

(i) Get the names of all the products that are sold.

(ii) Get the product numbers which are marketed by alteast two sales persons.

(iii) Get the names of all salespersons who are not Managers. **(9)**

Ans:

(a) Create table Salesperson (SNo number(3) primary key, Sname char(30), Destination char(15));
Create table Area (Ao number(3) primary key, Aname char(30), ManagerNo number(3) references Salesperson(SNo));
Create table Product (PNo number(3) primary key, Pname char(30), Cost number(5) check cost < 100000);
Create table SAP (SNo number(3), ANo number(3), PNo number(3), primary key (SNo, ANo, PNo), SNo references Salesperson(SNo), PNo references Product(PNo), ANo references Area (ANo));

(b)

(i) Select PNo from Product, SAP where Product. PNo = SAP.PNo;

(ii) Select T1.PNo from SAP T1, SAP T2 where T1.PNo= T2.PNo and T1.SNo <> T2.SNo;

(iii) Select Sname from Salesperson where designation <> 'Managers'.

Q.66 What is the basic difference between relational algebra and relational calculus? Define the atoms in tuple relational calculus. Use these to define the formulae. **(5)**

Ans: An algebra consists of a set of *objects* and a set of *operators* that together satisfy certain *axioms* or *laws* (such as closure, commutativity, associativity, and so on). The word "algebra" itself ultimately derives from Arabic *al-jebr,* meaning a *resetting* (of something broken) or a *combination.*

Relational calculus is an applied form of a fundamental branch of logic called predicate calculus. In general, the term "calculus" signifies merely a system of computation (the Latin word calculus means a pebble, perhaps used in counting or some other form of reckoning). Thus, relational calculus can be thought of as a system for computing with relations

Atoms

For the construction of the formulas we will assume an infinite set V of tuple variables. The formulas are defined given a database schema S = (D, R, h) and a partial function *type* : $V \rightarrow 2^C$ that defines a *type assignment* that assigns headers to some tuple variables. We then define the *set of atomic formulas A*[*S*,*type*] with the following rules

1. if *v* and *w* in *V*, *a* in *type*(*v*) and *b* in *type*(*w*) then the formula " *v.a* = *w.b* " is in *A*[*S*,*type*],
2. if *v* in *V*, *a* in *type*(*v*) and *k* denotes a value in *D* then the formula " *v.a* = *k* " is in A[S,type], and
3. if *v* in *V, r* in *R* and *type(v) = h(r)* then the formula " *r(v)* " is in *A[S,type].*

   Examples of atoms are:
   - (*t*.age = *s*.age) — *t* has an age attribute and s has an age attribute with the same value
   - (*t*.name = "Codd") — tuple t has a name attribute and its value is "Codd"
   - Book*(t)* — tuple *t* is present in relation Book.

The formal semantics of such atoms is defined given a database *db* over *S* and a tuple variable binding *val* : $V \rightarrow T_D$ that maps tuple variables to tuples over the domain in *S*:

1. " *v.a* = *w.b* " is true if and only if *val*(*v*)(*a*) = *val*(*w*)(*b*)
2. " *v.a* = *k* " is true if and only if *val*(*v*)(*a*) = *k*
3. " *r*(*v*) " is true if and only if *val*(*v*) is in *db*(*r*)

Formulas

The atoms can be combined into formulas, as is usual in first-order logic, with the logical operators Λ (and), V (or) and ¬ (not), and we can use the existential quantifier (∃) and the universal quantifier (∀) to bind the variables. We define the *set of formulas F*[*S*,*type*] inductively with the following rules:

1. every atom in *A*[*S*,*type*] is also in *F*[*S*,*type*]
2. if f_1 and f_2 are in *F*[*S*,*type*] then the formula " f_1 V f_2 " is also in *F*[*S*,*type*]
3. if f_1 and f_2 are in *F*[*S*,*type*] then the formula " f_1 Λ f_2 " is also in *F*[*S*,*type*]
4. if *f* is in *F*[*S*,*type*] then the formula " ¬ *f* " is also in *F*[*S*,*type*]
5. if *v* in *V*, *H* a header and *f* a formula in $F[S,type_{[v->H]}]$ then the formula " ∃ *v* : *H* (*f*) " is also in *F*[*S*,*type*], where $type_{[v->H]}$ denotes the function that is equal to *type* except that it maps *v* to *H*,
6. if *v* in *V*, *H* a header and *f* a formula in $F[S,type_{[v->H]}]$ then the formula" ∀*v* : *H* (*f*) " is also in *F*[*S*,*type*]

Example of formula:

t.name = "C. J. Date" ∀ *t*.name = "H. Darwen"

Q.67 Consider the following relations

Person (name, street, city)

Owns (name, reg_no, model, year)
Accident (date, reg_no)
Answer the following using tuple relational calculus
(i) Find the names of persons who are not involved in any accident.
(ii) Find the names and street of persons who own a maruti car.
(iii) Find the registration numbers of the cars manufactured in the year 2004. **(9)**

Ans:
(i) $\{t[name] \mid t \in owns \wedge \neg \exists a \in Accident(t[reg\text{-}no] = a[reg\text{-}no])\}$
(ii) $\{t[name], t[street] \mid t \in person \wedge \exists O \in Owns(t[name] = O[name] \wedge O[model] = \text{'Maruti'})\}$
(iii) $\{t[reg\text{-}no] \mid t \in owns \wedge t(year] = 2004\}$

Q.68 Define functional and multivalued dependencies. **(2)**

Ans: A **functional dependency** is a property of the semantics of the attributes in a relation. The semantics indicate how attributes relate to one another, and specify the functional dependencies between attributes. When a functional dependency is present, the dependency is specified as a constraint between the attributes. Consider a relation with attributes A and B, where attribute B is functionally dependent on attribute A. If we know the value of A and we examine the relation that holds this dependency, we will find only one value of B in all of the tuples that have a given value of A, at any moment in time. Note however, that for a given value of B there may be several different values of A.
Multivalued Dependencies: A multivalued dependency is an outcome of two independent 1: N relationships A: B and A: C being mixed in the same relation R(A,B,C)

Q.69 Consider the relation Student (stid, name, course, year)
Given that
A student may take more than one course but has unique name and the year of joining.
(i) Identify the functional and multivalued dependencies for Student. **(4)**
(ii) Identify a candidate key using the functional and multivalued dependencies arrived at in step (b). **(4)**
(iii) Normalize the relation so that every decomposed relation is in 4NF. **(4)**

Ans:
(i) **F.D are:** stid → name, year
M.V.D. are: stid →→ Course
course →→ stid
(ii) **Candidate Keys:** (stid, course)
(iii) Since in both the MVDs, LHS is not a superkey . Therefore decomposition is as follows:

R1(stid, name, year)	Primary Key = stid
R2(stid, course)	Primary Key = (stid, course) Foreign Key = stid

Q.70 Explain the following
(i) ISA relationship.
(ii) NULL value.
(iii) Trigger.

(iv) EXEC statement in SQL. **(2 x 4)**

Ans:

(i) **ISA** notation is used in generalization/ specialization to show relationship between lower and higer level entity set.

(ii) **NULL value** NULL means something is unknown. It does NOT mean null (the digit 0). Null is also used as attribute value for a particular entity where attribute is not applicable eg, name of child for unmarried.

(iii) **Trigger:** A **database trigger** is procedural code that is automatically executed in response to certain events on a particular table in a database. Triggers can restrict access to specific data, perform logging, or audit data modifications.

(iv) **EXEC statement in SQL**

All statements that begin with EXEC SQL are embedded SQL database statements. High level languages like C can be used to write and SQL statements may be embedded in them using EXEC statement

Q.71 Define a view ProductArea in relational algebra and SQL, using the relations of question (65) above, which contains the area name and the names of products sold in that area.**(6)**

Ans: $\Pi_{aname,\ pname}$ (Area ⋈ SAP ⋈ Product)

Create view ProductArea as select Aname, Pname from Area, Product, SAP
where SAP.PNo = Product.PNo and SAP.ANo= Area. ANo

Q.72 Compare the two method for storing variable length records – byte string representation and fixed length representation. Discuss the merits and demerits of the two. **(6)**

Ans: **Variable-Length Records**

Variable-length records arise in a database in several ways:

- Storage of multiple items in a file
- Record types allowing variable field size
- Record types allowing repeating fields

Byte string representation uses the technique of attaching a special end-of record symbol(⊥)to the end of each record. Then we can store each record as a string of successive bytes.

Byte string representation has several disadvantages:

- It is not easy to re-use space left by a deleted record
- In general, there is no space for records to grow longer. (Must move to expand, and record may be pinned.)

So this method is not usually used.

Fixed-length representation uses one or more fixed-length records to represent one variable-length record. Two techniques:

- **Reserved space** - uses fixed-length records large enough to accommodate the largest variable-length record. (Unused space filled with end-of-record symbol.)
- **Pointers** - represent by a list of fixed-length records, chained together.

The reserved space method requires the selection of some maximum record length. If most records are of near-maximum length this method is useful.

- Otherwise, space is wasted.

- Then the pointer method may be used .
- Disadvantage is that space is wasted in successive records in a chain as non-repeating fields are still present.
- To overcome this last disadvantage we can split records into two blocks
  - **Anchor block** - contains first records of a chain
  - **Overflow block** - contains records other than first in the chain.
- Now all records in a block have the same length, and there is no wasted space.

Q.73 Describe the different RAID levels. Discuss the choices of the different RAID levels for different applications. **(8)**

Ans: There are various combinations of these approaches giving different trade offs of protection against data loss, capacity, and speed. RAID levels 0, 1, and 5 are the most commonly found, and cover most requirements.

- RAID 0 (striped disks) distributes data across several disks in a way that gives improved speed and full capacity, but all data on all disks will be lost if any one disk fails.
- RAID 1 (mirrored disks) could be described as a backup solution, using two (possibly more) disks that each store the same data so that data is not lost as long as one disk survives. Total capacity of the array is just the capacity of a single disk. The failure of one drive, in the event of a hardware or software malfunction, does not increase the chance of a failure nor decrease the reliability of the remaining drives (second, third, etc).
- RAID 2 It uses memory style redundancy by using Hamming codes, which contain parity bits for distinct overlapping subsets of components.
- RAID 3 It uses a single parity disk relying on the disk controller to figure out which disk has failed.
- RAID 4 It uses block-level data striping.
- RAID 5 (striped disks with parity) combines three or more disks in a way that protects data against loss of any one disk; the storage capacity of the array is reduced by one disk.
- RAID 6 (less common) can recover from the loss of two disks through P + Q redundancy scheme using Reed- Adoman codes

Q.74 Define two-phase locking protocol. **(2)**

Ans: Two-phase locking is important in the context of ensuring that schedules are serializable. The **two-phase locking protocol** specifies a procedure each transaction follows.

The two-phase locking protocol

1. Before operating on any row, a transaction must acquire a lock on that row.
2. After releasing a lock, a transaction must never acquire any more locks.

Q75 Differentiate between strict two-phase and rigorous two-phase with conversion protocol **(4)**

Ans: **Strict two-phase** locking holds all its exclusive (write) locks until commit time, once granted. While **Rigorous two-phase locking** is more restrictive than standard 2PL

in that it enforces the rule that no locks can be released by a transaction until after it commits or aborts. The former locks all its items before it starts so once transaction starts, it is in its shrinking phase whereas latter does not unlock any of its items until after it terminates to the transaction is in its expanding phase until it ends.

Q.76 Describe the nested-loop join and block-nested loop join. Compare them. **(8)**

Ans: The block nested- loop join algorithm is as given below:

for each block B_r of relation R do begin
    for each block B_s of relation S do begin
        for each tuple t_r in B_r do begin
            for each tuple t_s in B_s do begin
                test pair (t_r,t_s) to see if they satisfy the join condition
                if they do, add $t_r.t_s$ to the result
            end
        end
    end
end

The nested-loop from algorithm is as below:

for each tuple t_r in relation R do begin
  for each tuple t_s in relation S do begin
    test pair (t_r,t_s) to see if they satisfy the join condition Θ
      if they do, add $t_r.t_s$ to the result
        end
          end

Difference between them:

Nested loop join	Block-nested-loop join
It is expensive as worst case cost, no. of block accesses required is $n_r * b_s + b_r$ where bs and br represent no. of blocks containing n_s and n_r tuples of relations S and R, respective and in best case there will be $b_r + b_s$ block accesses	In worst case no. of block accesses is $b_r * b_s + b_r$. In best case, $b_r + b_s$ block accesses will be required.

Q.77 Two relations R with 60000 tuples and occupying of 300 blocks is to be joined with a relation S with 40000 tuples and occupying 400 blocks. What is the total cost using the two algorithms of (Q 76) above in terms of block transfers. Give both the best case and the worst case figures. **(6)**

Ans:
Nested-loop-join: Worst Case: If R as the outer relation 60000 X 400 + 300 = 24000300 Disk access, if S as the outer relation we need 40000 X 300 + 400 = 12000400 disk accesses
Best Case: 300 + 400 = 700 disk accesses will be required.
Block-nested loop join:
Worst Case: If R is the outer relation, we need 300 * 400 + 300 = 120300 disk access, if S is the outer relation we need 400 * 300 + 400 = 120400 disk access
Best Case: 300+ 400 = 700 disk accesses will be required.

Q.78 Explain the ACID properties of a transaction. **(6)**

Ans: ACID properties are an important concept for databases. The acronym stands for Atomicity, Consistency, Isolation, and Durability.
Atomicity: Either all operations of transaction are reflected properly in the database or none are.
Consistency: Execution of a transaction in isolation (i.e., with no other transaction executing concurrently) preserves the consistency of the database
Isolation: Even though multiple transactions may execute concurrently the system guarantees that for every pair of transaction Ti and Tj, it appears to Ti that either Tj that either Tj finished execution before Ti started or Tj started execution after Ti finished. Thus each transaction is unaware of other transactions executing concurrently in the system.
Durability: After a transaction completes successfully, the changes it has made to the database persist, even if there are system failures.

Q.79 Compare wait-die deadlock prevention scheme with wait-wound scheme. **(4)**

Ans: Wait-Die Scheme
- Based on a nonpreemptive technique.

- If Pi requests a resource currently held by Pj, Pi is allowed to wait only if it has a smaller timestamp than does Pj (Pi is older than Pj). Otherwise, Pi is rolled back (dies).
- Example: Suppose that processes P1, P2, and P3 have timestamps 5, 10, and 15, respectively.
  - If P1 requests a resource held by P2, then P1 will wait.
  - If P3 requests a resource held by P2, then P3 will be rolled back.

Wound-Wait Scheme

- Based on a preemptive technique; counterpart to the wait-die system.
- If Pi requests a resource currently held by Pj, Pi is allowed to wait only if it has a larger timestamp than does Pj (Pi is younger than Pj). Otherwise, Pj is rolled back (Pj is wounded by Pi).
- Example: Suppose that processes P1, P2, and P3 have timestamps 5, 10, and 15, respectively.
  - If P1 requests a resource held by P2, then the resource will be preempted from P2 and P2 will be rolled back.
  - If P3 requests a resource held by P2, then P3 will wait.

Q.80 What are the costs to be considered when a transaction has to be rolled back when recovering from deadlock? **(4)**

Ans: Some transaction will have to rolled back (made a victim) to break deadlock. Select that transaction as victim that will incur minimum cost.

- Rollback -- determine how far to roll back transaction.
  - **Total rollback**: Abort the transaction and then restart it.
  - More effective to roll back transaction only as far as necessary to break deadlock.
- Starvation happens if same transaction is always chosen as victim. Include the number of rollbacks in the cost factor to avoid starvation.

Q.81 Write short notes on

(i) Views in relational algebra.

(ii) Data dictionary.

(iii) Assertions in SQL.

(iv) B^+ tree. **(3.5 x 4=14)**

Ans:

(i) Views in relational algebra:

1. basic expression consists of either
   - A relation in the database.
   - A constant relation.
2. General expressions are formed out of smaller sub expressions using
   - $\sigma_p(E_1)$ select (p a predicate)
   - $\Pi s(E_1)$ project (s a list of attributes)
   - $P_x(E_1)$ rename (x a relation name)
   - $E_1 \cup E_2$ union
   - $E_1 - E_2$ set difference
   - $E_{1x} \times E_2$ cartesian product

(ii) Data dictionary

A data dictionary is a reserved space within a database which is used to store information about the database itself.

A data dictionary may contain information such as:

- Database design information
- Stored SQL procedures
- User permissions
- User statistics
- Database process information
- Database growth statistics
- Database performance statistics

(ii) Assertions in SQL

1. An **assertion** is a predicate expressing a condition we wish the database to always satisfy.
2. Domain constraints, functional dependency and referential integrity are special forms of assertion.
3. Where a constraint cannot be expressed in these forms, we use an assertion, e.g.
   - Ensuring the sum of loan amounts for each branch is less than the sum of all account balances at the branch.
   - Ensuring every loan customer keeps a minimum of $1000 in an account.

(iv) B+ tree

B+ tree is a type of tree which represents sorted data in a way that allows for efficient insertion, retrieval and removal of records, each of which is identified by a *key*. It is a dynamic, multilevel index, with maximum and minimum bounds on the number of keys in each index segment (usually called a 'block' or 'node'). In a B+ tree, in contrast to a B-tree, all records are stored at the lowest level of the tree; only keys are stored in interior blocks.

The primary value of a B+ tree is in storing data for efficient retrieval in a block-oriented storage context. Given a storage system with a block size of *b*, a B+ tree which stores a number of keys equal to a multiple of *b* will be very efficient when compared to a binary search tree (the corresponding data structure for non-block-oriented storage contexts).

Q.82 What is completeness constraint on generalization? Explain the difference between total and partial design constraint. Give an example each. **(6)**

Ans: Completeness Constraints

- total specialization constraint
  - every entity in the superclass must be a member of some subclass in the specialization

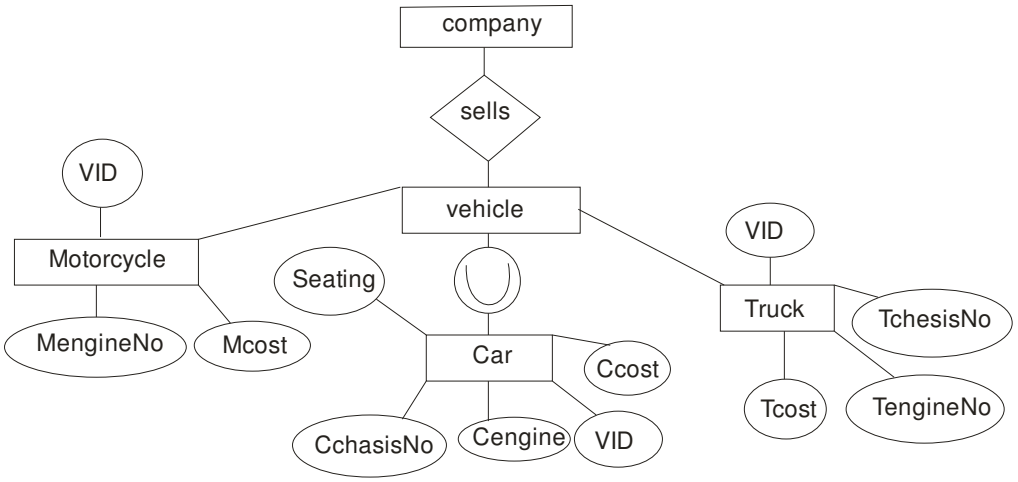
  e.g., {HOURLY_EMPLOYEE, SALARIED_EMPLOYEE}
  - notation: superclass
- partial specialization constraint

an entity may not belong to any of the subclasses

e.g., {SECRETARY, ENGINEER, TECHNICIAN}

Q.83 Design a generalization-specialization hierarchy for a motor-vehicle sales company. The company sells motorcycles which have an engine number and cost; cars which have a chassis number, an engine number, seating capacity and cost; trucks which have chassis number, an engine number and cost. **(10)**

Ans:



Q.84 Define the following operations of relational algebra and give an example each

(i) Division.

(ii) Cartesian product. **(7)**

Ans:

(i) Division:

Let R be a relation having attributes (A1,...Ap,Ap+1,...An) and S having attributes (Ap+1,...An)

DEF: Division

The division of R by S, denoted R ÷ S, gives a new relation Q, the quotient of the division, such that Q has attributes (A1,...Ap) and every tuple of Q must appear in R in combination with every tuple in S.

The division can be obtained from the difference, the Cartesian product and the projection as follows:

R ÷ S = T - Y, where T = $\pi_{A1,...Ap}$ (R) and Y = $\pi_{A1,...Ap}$ ((T X S) - R)

ii) Cartesian product:

The Cartesian product operation does not require relations to union-compatible. It means that the involved relations may have different schemas. Let R & S be relations that may have different schemas.

DEF: **Cartesian Product**

The Cartesian product of relations R and S, denoted R **X** S, is the set of all possible combinations of tuples of the two operation relations. Each resultant tuple consists of all the attributes of R and S.

The resultant relation has

cardinality = (cardinality of R) * (cardinality of S); and

arity = (arity of R) + (arity of S).

For example, in given table, STUDENT **X** MODULE is equal to

STUDENT #	STUDENT-NAME	COURSE	MODULE #	MODULE-NAME
A101	Astor	APPCOM	CS301	IS
A101	Astor	APPCOM	CS302	DBS
B334	Bernstein	INFTEC	CS301	IS
B334	Bernstein	INFTEC	CS302	DBS
J326	Jones	APPCOM	CS301	IS
J326	Jones	APPCOM	CS302	DBS

Q.85 Let R(A, B) and S(A, C) be two relations. Give relational algebra expressions for the following domain calculus expressions.

(i) $\{< a > | \exists b (< a, b > \in r \wedge b = 17)\}$

(ii) $\{< a, b, c > | < a, b > \in r \wedge < a, c > \in s\}$

(iii) $\{< a > | \exists b (\exists c \{< a, b > \in r) \wedge < a, c > \in s)\}\}$ **(9)**

Ans:

(i) $\Pi_A (\sigma_{B=17} (r))$

(ii) $r \bowtie s$

(iii) $\Pi_{r.A} ((r \bowtie s))$

Q.86 Consider the following relations with key underlined

lives (person_name, street, city)
works (person_name, company_name, salary)
located (company_name, city)
manages (person_name, manager_name)

Answer the following using SQL:

(i) Find the names and city of persons who work for manager John.

(ii) Find the names of persons who live in the same city as the company they work for.

(iii) John's manager has changed. The new manager is Anna.

(iv) Susan doesn't work anymore.

(v) Create a view BangWork (person_name, company_name, manager_name) of all people who work in Bangalore in ascending order of person name **(4x3x4)**

Ans:

(i) Select person_name, city from lives, manages
where manager_name = 'John' and lives.person_name = manages.person_name;

(ii) Select person_name from lives works, located where lives.person_name = works.person_name and works.company_name = located.company_name and lives.city = located. city;

(iii) Update manages
set manager_name = 'Anna'
where manager_name = 'John';

(iv) delete from works where person_name = 'Susan';

(v) create view BangWork as select person_name, company_name, manager_name from work where city = 'Bangalore' order by person_name;

Q.87 What are the problems if one were not to normalize? When do these problems surface? **(2)**

Ans: **Database normalization**, sometimes referred to as *canonical synthesis*, is a technique for designing relational database tables to minimize duplication of information and, in so doing, to safeguard the database against certain types of logical or structural problems, namely data anomalies. For example, when multiple instances of a given piece of information occur in a table, the possibility exists that these instances will not be kept consistent when the data within the table is updated, leading to a loss of data integrity. A table that is sufficiently normalized is less vulnerable to problems of this kind, because its structure reflects the basic assumptions for when multiple instances of the same information should be represented by a single instance only.

Q.88 Consider the relation
Book (accno, author, author_address, title, borrower_no, borrower_name, pubyear)
with the following functional dependencies
accno → title accno → pubyear
author → accno
accno → author author → author_address
accno → borrower_no borrower_no → borrower_name
(i) Normalize the relation. Clearly show the steps. **(6)**
(ii) For each decomposed relation identify the functional dependencies that apply and identify the candidate key. **(8)**

Ans:
(i) **Book1** (accno, title, author, borrow_no, pubyear, author_address)
Book2 (borrower_no, borrower_name)
Book3 (author, account)
ii) a) Functional Dependencies for "Book1" relation are:
accno → title
accno → author
accno → borrow_no
accno → pubyear
accno → author_address
b) F.Ds for "Book2" relation are:
borrow_no → borrower_name
c) F.Ds for "Book3" relation are:
author → account_no

Q.89 Describe sequential file organization. Explain the rules for
(i) inserting a new record.
(ii) Deleting an existing record. **(8)**

Ans: A sequential file contains records organized by the order in which they were entered. The order of the records is fixed.
Records in sequential files can be read or written only sequentially.

After you have placed a record into a sequential file, you cannot shorten, lengthen, or delete the record. However, you can update (`REWRITE`) a record if the length does not change. New records are added at the end of the file.
If the order in which you keep records in a file is not important, sequential organization is a good choice whether there are many records or only a few. Sequential output is also useful for printing reports.

(i) **Inserting a new record:** Insertion poses problems if no space where new record should go. If space, use it, else put new record in an **overflow block.**

(ii) **Deleting an existing record:** Deletion can be managed with the pointer chains

Q.90 Define and differentiate between ordered indexing and hashing. **(8)**

Ans: **Ordered indexing:** To gain fast random access to records in a file, we can use an index structure. Each index structure is associated with a particular search key. An ordered index stores the values of the search keys in sorted order and associates with each search key records that contain it. The records in the indexed file may themselves be stored in some sorted order. A file may have several indices on different search keys.
Hashing: It also provides a way of constructing indices. In hashing, the address of the disk lock containing a desired record directly is computed by a function on the search-key value of the record.
Differentiate between ordered indexing and hashing: Ordered indexing is stored in sorted order while in **Hashing** search keys are distributed uniformly across "buckets" using 'hash function'

Q.91 How do you compute the query cost for the following?

(i) Selection with linear search.

(ii) Negation. **(9)**

Ans:

(i) Scan each file block and test all records to see whether they satisfy the selection condition

- Cost estimate (number of disk blocks scanned) = b_r
  (b_r denotes number of blocks containing records from relation r)
- Selections on key attributes have an average cost $b_r/2$,
  but still have a worst-case cost of b_r
- Linear search algorithm can be applied to any file, regardless of
  - Ordering of records in the file
  - Availability of indices
  - Nature of the selection operation

(ii) Negation: In the absence of nulls the result of a selection $\sigma_{7\Theta}$(r) is simply the tuples of r that are not in σ_{Θ}(r). The number of tuples in $\sigma_{7\Theta}$(r) therefore estimated to be n (r) minus the estimated number of tuples in σ_{Θ} (r). To account for nulls, the number of tuples for which the condition $_{\Theta}$ would evaluate to unknown will be estimated and subtracting that number from above estimate ignoring nulls.

Q.92 Explain the statement 'Projection operation distributes over the union operation'. Give an example. **(4)**

Ans: $\Pi_L(E_1 \cup E_2) = (\Pi_L(E_1)) \cup (\Pi_L(E_2))$
This says that result obtained by projecting on an attribute set A, after taking union of two relations is same as union of two relations each of which is a projection of original relations on A.

Q.93 Explain pipelining. **(3)**

Ans: In order to explain pipelining in simple terms, think of it as breaking down processor functions into smaller and smaller parts. For example, the act of drinking milk can be broken down to opening the refrigerator, getting the milk carton, getting a glass from the cupboard, opening and pouring the milk, lifting the glass to your mouth, and pouring the milk inside. Each one of these subprocesses is much shorter than the entire process of drinking milk.A processor functions similarly. Instructions progress through, and as they do, they accomplish smaller parts of the overall execution. The more pipeline stages you have, the smaller each of these sub-processes are. Designers can design these sub-processes to be very fast, and efficient. The frequency of a processor advances each of these sub-processes to the next stage, so smaller sub-processes mean less time to complete them, and thus higher frequencies

Q.94 Explain the rules for creating a labelled precedence graph for testing view serializability. **(8)**

Ans: A schedule S is view serializable if it is view equivalent to a serial schedule

- Every conflict serializable schedule is also view serializable
- A schedule which is view serializable but not conflict serializable:

Testing for Serializability

- Consider some schedule of a set of transactions T1, T2, ..., Tn
- Precedence graph is a direct graph where the vertices represent the transactions
- We draw an arc from T_i to T_j if the two transaction conflict and T_i accessed the data item earlier on which the conflict arose
- We may label the arc by the item on which the conflict arose

A schedule is conflict serializable if and only if its precedence graph is acyclic

- If precedence graph is acyclic, the serializability order can be obtained by a topological sorting of the graph.
- The problem of checking if a schedule is view serializable falls in the class of *NP*-complete problems
- However practical algorithms that just check some sufficient conditions for view serializability can still be used

Q.95 Explain the difference between the three storage types – volatile, non volatile and stable. **(8)**

Ans: **Volatile storage:** if storage media loses data when power goes off, it is known as volatile storage media eg, RAM. It is the fastest between the three in terms of data access time.

Non-volatile storage: If storage media retains data even when power goes off, it is known as non-volatile storage media. Eg: hard disk. It is faster than stable storage but slower than volatile storage.

Stable Storage: Information residing in stable storage is never lost (theoretically). A natural catastrophe may result in a loss otherwise the probability of data loss is negligible. Eg, using multiple hard disks as in the case of RAID technology. This is the slowest of all storage media mentioned above.

Q.96 How does the system recover from a crash? **(8)**

Ans: Not all problems that cause computer crashes are severe. If you carefully analyze and try to find out probable causes, you usually have a good chance to quickly and completely recover your system.

Most of the time, it is quite simple to point out the main reason behind a problem. Suppose that your system starts freezing when you restart after installing a new hardware or software or if you have made some changes to system configuration, then you can safely say that the new device, driver, or software is the reason behind the problem.

Windows XP comes equipped with a number of options that you can use to troubleshoot and repair your PC problems. Here, we are going to discuss a few common options that can help you recover your Windows XP system after a crash.

Safe Mode

If your system frequently hangs during startup, then you must try to start your system in Safe Mode. To do this, restart your PC and press F8 as soon as your system starts booting up. Now, scroll down the displayed options using arrow keys and select one of the three Safe Mode options displayed. You can perform a number of activities to fix your computer in this mode. You can uninstall the suspect driver or software, and change configuration settings.

Last Known Good Configuration

The Last Known Good Configuration option also appears when you press the F8 key during system boot up. This option helps you in reversing any driver and hardware changes that you would have done since your last successful startup.

Recovery Console

In case your system is unable to start up even in the Safe Mode, then you may have to use the Recovery Console. You can start the Recovery Console by booting up from the Windows XP installation CD, and selecting the Repair option. This utility helps you delete corrupted software and driver files to stop services that might be causing frequent system crashes.

System Restore

System Restore is a boon for all computer users. The tool monitors all changes on your system, periodically takes snapshots of system files and settings, and stores original files in compressed form in a protected location on the hard drive. To perform system restore, log on as an Administrator, and open the Start menu. Next, select All Programs-Accessories-System Restore command to display the System Restore dialog box. Follow the simple instructions in the dialog box to rollback your computer to the most current system checkpoint.

Reinstallation

If none of the options above work, then you may have to reinstall the operating system. If you are careful with this step, you can easily reinstall the operating system without effecting your system preferences and settings.

Q.97 Describe the storage structure of indexed sequential files and their access method. **(7)**

Ans: Index provides a lookup capability to quickly reach the vicinity of the desired record.

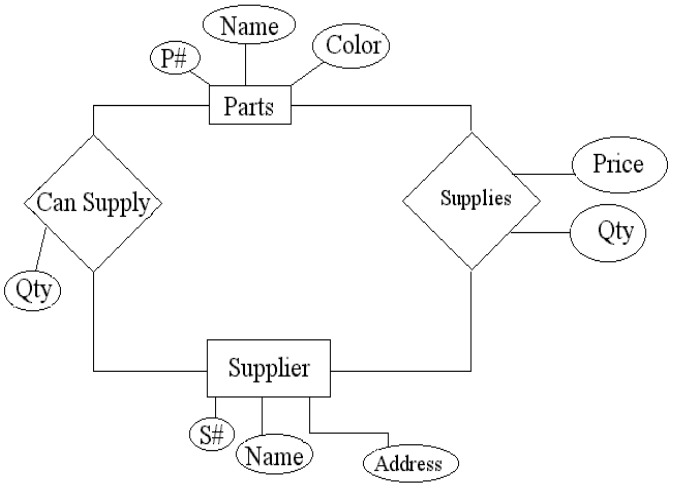
- Contains key field and a pointer to the main file
- Indexed is searched to find highest key value that is equal to or precedes the desired key value
- Search continues in the main file at the location indicated by the pointer.

If an index contains 1000 entries, it will take on average 500 accesses to find the key, followed by 500 accesses in the main file. Now on average it is 1000 accesses.

- New records are added to an overflow file
- Record in main file that precedes it is updated to contain a pointer to the new record
- The overflow is merged with the main file during a batch update
- Multiple indexes for the same key field can be set up to increase efficiency.

ISAM (Indexed Sequential Access Method) is a file management system developed at IBM that allows records to be accessed either sequentially (in the order they were entered) or randomly (with an index). Each index defines a different ordering of the records. An employee database may have several indexes, based on the information being sought. For example, a name index may order employees alphabetically by last name, while a department index may order employees by their department. A key is specified in each index. For an alphabetical index of employee names, the last name field would be the key.

Q.98 Map the following ER diagram to a relational database. Give the relation names and attributes in them. Also mention the primary key and foreign keys if any for each table. **(10)**



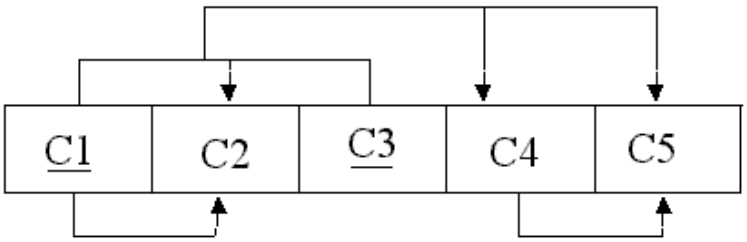
Ans:

Relation Name	**Attributes**	**Primary key**	**Foreign Key (s)**
Parts	P#, Name, color	P#	NIL
Supplier	S#, Name, Address	S#	NIL
Can_supply	P#, S#, QTY	P#,S#	P# references Parts.P# S# references Supplier. S#
Supplies	P#, S#, QTY, Price	P#, S#	P# references Parts. P# S# references Supplier.S#

Q.99 What are the three data anomalies that are likely to occur as a result of data redundancy? Can data redundancy be completely eliminated in database approach? Why or why not? **(5)**

Ans: The most common anomalies considered when data redundancy exists are: update anomalies, addition anomalies, and deletion anomalies. All these can easily be avoided through data normalization. Data redundancy produces data integrity problems, caused by the fact that data entry failed to conform to the rule that all copies of redundant data must be identical. The data redundancy cannot be totally removed from the database, but there should be controlled redundancy. Sometimes redundancy is allowed to overcome the problem of excessive join operations for computing queries.

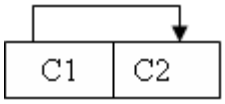
Q.100 Given the dependency diagram shown in the following figure, (the primary key attributes are underlined)



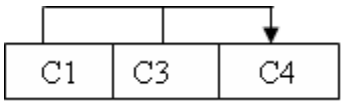
(i) Identify and discuss each of the indicated dependencies.
(ii) Create a database whose tables are atleast in 3NF, showing dependency diagram for each table **(4+5)**

Ans:
(i) C1 → C2 represents a *partial dependency*, because C2 depends only on C1, rather than on the entire primary key composed of C1 and C3.
C4 → C5 represents a *transitive dependency*, because C5 depends on an attribute (C4) that is not part of a primary key.
C1, C3 → C2, C4, C5 represents a functional dependency, because C2, C4, and C5 depend on the primary key composed of C1 and C3.
ii) After decomposition into 3NF, relations are

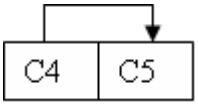
1) R1 (C1, C2) with FD {C1→ C2)} and functional dependency diagram:



2) R2 (C1,C3, C4) with FD {C1,C3 →C4) and functional dependency diagram:



3) R3 (C4, C5) with FD{C4 → C5} and functional dependency diagram:



Q.101 Consider the following relations with underlined primary keys

Product(P_code, Description, Stocking_date, QtyOnHand, MinQty, Price, Discount, V_code)

Vendor(V_code, Name, Address, Phone)

Here a vendor can supply more than one product but a product is supplied by only one vendor. Write SQL queries for the following :

(i) List the names of all the vendors who supply more than one product.

(ii) List the details of the products whose prices exceed the average product price.

(iii) List the Name, Address and Phone of the vendors who are currently not supplying any product. **(3 x 3)**

Ans: **(i)** Select name from Vendor where V_code in (Select V_code from Product group by V_code having count (V_code) >1)

(**ii**) Select * from Product where price > (select avg(Price) from product)

(iii) Select Name, Address, Phone from Vendor v where not exists (select * from Product p where p.V_code = v.V_code)

Q.102 Define the domain relational calculus. **(5)**

Ans: In DRC, *queries* have the form: { <x1,x2,--------xn> | P(x1,x2,--------xn)}

where each X_i is either a domain variable or constant, and p(<X1, X2, ..., Xn>) denotes a DRC *formula*. The result of the query is the set of tuples X_i to X_n which makes the DRC formula true.

This language uses the same operators as tuple calculus, the logical connectives (and), (or) and ¬ (not). The existential quantifier and the universal quantifier can be used to bind the variables.

Q.103 Consider the transactions t1, t2 and t3 and a schedule S given below.
$S: \mathbf{read}_1(A); \mathbf{read}_2(B); \mathbf{write}_1(C); \mathbf{read}_3(B); \mathbf{read}_3(C); \mathbf{write}_2(B); \mathbf{write}_3(A)$ Where the subscript denotes the transaction number. Assume that the time stamp of t1<t2<t3. Using time-stamp ordering scheme for concurrency control find out if the schedule will go through. If there is to be a rollback, which transaction(s) will be rolled back? **(8)**

Ans: **Schedule:**

t_1	t_2	t_3
read(A)		
	read(B)	
write (C)		
		read (B)
		read (C)
	*** write(B)**	
		write (A)

Since this write in t_2 is after a younger transaction (t_3) has read the value of B, therefore t_2 will be rolled back.

Q.104 Consider the following database with primary keys underlined
Project(P_No, P_Name, P_Incharge)
Employee(E_No, E_Name)
Assigned_To(P_No, E_No)
Write the relational algebra for the following :
(i) List details of the employees working on all the projects.
(ii) List E_No of employees who do not work on project number DB2003. **(6)**

Ans:

(i) $\Pi_{E\_No,\ E\_Name}$ (Project ⋈Employee ⋈Assigned) ÷ Π_{E_Name} (Employee)

(ii) Π_{E_No} (Assigned _To) - Π_{E_No} ($\sigma_{P\_No\ =\ \text{“DB2003”}}$ (Assigned_To)

Q.105 The following represents the sequence of events in a schedule involving transactions T1, T2, T3, T4 and T5. A,B, C, D, E, F are items in the database.

T2	:	Read	B
T4	:	Read	D
T2	:	Read	E
T2	:	Write	E
T3	:	Read	F
T2	:	Read	F
T1	:	Read	C
T5	:	Read	A
T5	:	Write	A
T1	:	Read	E
T5	:	Read	C
T3	:	Read	A
T5	:	Write	C
T2	:	Write	F
T4	:	Read	A

Draw a wait-for-graph for the data above and find whether the transactions are in a deadlock or not? **(8)**

1. T2 : Read B
The data item B is locked by T2 and no change in the wait-for-graph

2. T4 : Read D
The data item D is locked by T4 and no change in the wait-for-graph

3. T2 : Read E
The data item E is locked by T2 and no change in the wait-for-graph

4. T3 : Write E
The data item E is unlocked by T2 and no change in the wait-for-graph

5. T3 : Read F
The data item F is locked by T3 and no change in the wait-for-graph

6. T2 : Read F
Now T2 wants to lock F, which is already locked by T3 (in step 5) so insert an edge from T2 to T3 in the wait-for-graph.

7. T1 : Read C
The data item C is locked by T1 and no change in the wait-for-graph.

8. T5 : Read A
The data item A is locked by T5 and no change in the wait-for-graph.

9. T5 : Write A
The data item B is locked by T5 and no change in the wait-for-graph.

10. T1 : Read E
The data item E is locked by T1 and no change in the wait-for-graph.

11. T5 : Read C
Now T5 wants to lock C, which is already locked by T1 (in step 7) so insert an edge from T5 to T1 in the wait-for-graph.

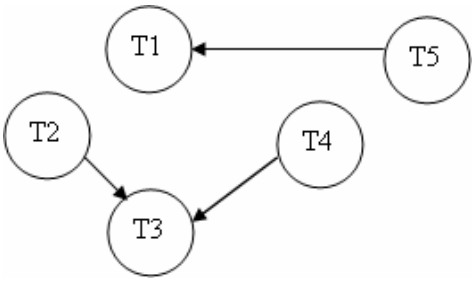
12 T3 : Read A
The data item A is locked by T3 and no change in the wait-for-graph.

13. T5 : Write C
The transaction cannot proceed further as it was not able to lock data item C (in step 11) so the transaction has to wait, hence there is no change in the wait-for-graph.

14. T2 : Write F

The transaction cannot proceed further as it was not able to lock data item F (in step 6) so the transaction has to wait, hence there is no change in the wait-for-graph.

15. T4 : Read A
Now T4 wants to lock A, which is already locked by T3 (in step 13) so insert an edge from T4 to T3 in the wait-for-graph.
Thus finally the wait-for graph will be as follows:



Since there is no cycle in the wait-for-graph, the system is not in a deadlock.

Q.106 Write short notes on any

(i) Disadvantages of file based systems.
(ii) Query-by-Example.
(iii) B-tree. **(3.5×3=14)**

Ans:

(i) Disadvantages of file based systems:

1. duplication of data
2. data integrity problem
3. limited data sharing
4. lengthy processing time

(ii) Query-by-Example:

Query by Example (QBE) is a method of query creation that allows the user to search for documents based on an example in the form of a selected text string or in the form of a document name or a list of documents. Because the QBE system formulates the actual query, QBE is easier to learn than formal query languages, such as the standard Structured Query Language (SQL), while still enabling powerful searches.

(iii) B-tree:

B-tree is a tree data structure that keeps data sorted and allows searches, insertions, and deletions in logarithmic amortized time. It is most commonly used in databases and filesystems.

In B-trees, internal (non-leaf) nodes can have a variable number of child nodes within some pre-defined range. When data is inserted or removed from a node, its number of child nodes changes. In order to maintain the pre-defined range, internal nodes may be joined or split. Because a range of child nodes is permitted, B-trees do not need re-balancing as frequently as other self-balancing search trees, but may waste some space, since nodes are not entirely full. The lower and upper bounds on the number of child nodes are typically fixed for a particular implementation. For example, in a 2-3 B-tree (often simply referred to as a **2-3 tree**), each internal node may have only 2 or 3 child nodes.

Q.107 Supreme Products manufactures products like pressure cookers, cookwares, water purifiers, food processors etc. The company markets its products to wholesalers all over the country and dealers sell them to customer. The company has five regional offices and many sales persons are attached to regional offices. Salespersons contact dealers and explain about products, incentives offered, training programs for wholesalers and demo for customers etc. Dealers place orders with the salespersons attached with the regional office of their location. After receiving goods they make payments, which may be in installments. Company would like to develop a system to monitor sales of different products, performance of salespersons and orders from wholesalers.

Do the following :

(i) Identify entities, attribute and relationships giving functionalities and draw E-R diagram for the system.

(ii) Convert this to relational tables explaining logic involved. **(5+5)**

Ans: The ER Diagram for the given problem is : The various entities for the given problem will be

WHOLESALER
REGIONALOFFICE
SALESPERSON
PAYMENT
DEALER
ORDER
ITEM

CUSTOMER

The various assumptions for the given system are :

- A salespersons is attached to only one regional office; however a regional office may have several salespersons under it.
- A dealer can place several orders but an order will be placed by a single dealer.
- A dealer will make payment to only one salesperson of his/her area but a salesperson can collect payments from several dealers of his/her area.
- An order may contain several items but an item can be placed only in one order.
- A wholesaler can appoint several dealers but a dealer will be only under one wholesaler.
- Many dealers can sales their products to several customers and several customers can purchase the products from many dealers.

The relational schema of the above ER Diagram is given here :

WHOLESALER:

WS_ID	WS_NAME	WS_ADD

REGIONAL OFFICE:

RO_ID	RO_NAME	RO_ADD

SALESPERSON:

SALES_ID	SALES_NAME	SALES_ADD	CATEGORY

PAYMENT

AMIT_PAYABLE	AMT_PAID	DATE

DEALER:

DEAL_ID	DEAL_NAME	DEAL_ADD

ORDER:

ORDER_ID	ITEM_ID	DATE	QTY

ITEM:

ITEM_ID	ITEM_NAME	STOCK	PRICE

CUSTOMER:

CUST_ID	CUST_NAME	CUST_ADD

For relationship, some of the relations will be as follows:

APPOINTS:

WS_ID	DEAL_ID

SALES:

CUST_ID	DEAL_ID
WS_ID	DEAL_ID
SALES:CUST_ID	DEAL_ID

PLACE:

DEAL_ID	ORDER_ID

ATTACHED:

SALES_ID	RO_ID

Some reports which can be proposed from the above system are :

- List the names of all salespersons with maximum sales from all regional offices.
- Name of the dealer who have maximum customers under him/her.
- All the dealers who have payments due against them.
- Lists the highest selling items.
- List of all regional offices along with their dealers and salespersons

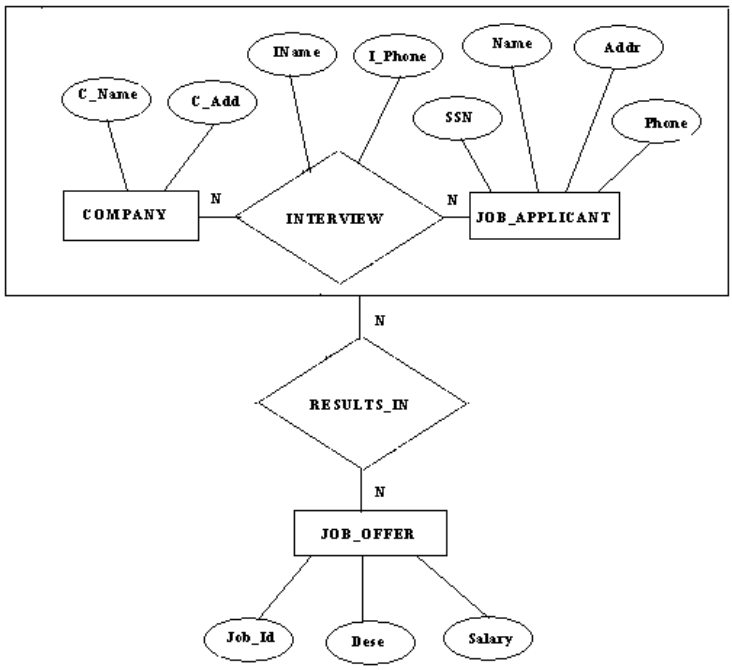
Q.108 Give examples of :

(i) A many – to – many relationship in which one of the participants is another relationship;

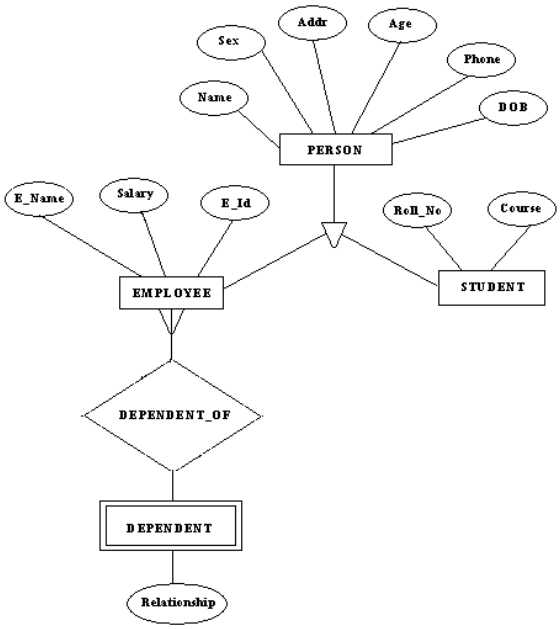
(ii) A subtype that has an associated weak entity that dose not apply to the super type

Ans:

(i) A many-to-many relationship in which one of the participants is another relationship;



(ii) A subtype that has an associated weak entity that does not apply to the supertype.



PERSON can b*e* EMPLOYEE or STUDENT. EMPLOYEE has DEPENDENT that works under him. DEPENDENT is a weak entity set because many dependents can have same Name, Age, Sex, etc.

Q.109 What is meant by heuristic optimisation? Discuss the main heuristics that are applied during query optimisation. **(6)**

Ans:

In heuristic opimization, heuristics are used to reduce the cost of optimization instead of analyzing the number of different plans to find out the optimal plan. For example. A analyzing optimizer would use the rule 'perform selection operation as early as possible' without finding out whether the cost is reduced by this transformation. Heuristics approach usually helps to reduce the cost but not always. The heuristics that are applied during query optimization are:

- Pushes the selection and projection operations down the query tree
- Left-deep join trees- convenient for pipelined evaluation
- Non- left-deep join trees

Q.110 Consider the following tables :
customer (c_id, c_name, c_address)
branch (br_name, br_city, assets)
account (c_id, act_no, br_name, balance)

(i) Customers who have accounts in all branches of Bhopal.

(ii) Customers who have accounts in branches with assets more than 50 crores.**(3X2)**

Ans:

The customers who have accounts in branches of Bhopal.

$$\prod_{c\_name}(\sigma_{br\_name='BHOPAL'}(\text{BRANCH } br\_name \bowtie \textit{ACCOUNT} \bowtie \textit{CUSTOMER } c\_id))$$

$$\prod_{c\_name}(\sigma_{assets>500000000}(\text{BRANCH } br\_name \bowtie \text{ACCOUNT} \bowtie \text{CUSTOMER } c\_id))$$

Q.111 Consider the following tables.

EMP (emp_no, name, salary, supervisor_no, sex_code, dept_code)

DEPT (dept_cd, dept_name)

Write down queries in SQL for getting following information:

(i) Employees getting more salary than their supervisor.

(ii) Department name and total number of employees in each department who earn more than average salary for their department.

(iii) Department(s) having maximum employees earning more than 25000.

(iv) Name of employee(s) who earn maximum salary in their organization. **(4x3)**

Ans:

(i) Employees getting more salary than their supervisor.

Again it will be solved by using self join concept.

```
SELECT E1.name, E2.name Supervisor
                    FROM EMPAS E1, EMP AS E2
                    WHERE E1.supervisor_no = E2.emp_no
                            AND E1.salary>E2.salary;
```

(ii) Department name and total number of employees in each department who earn more than average salary for their department.

This will use the correlated query concept.

```
SELECT dept_name, COUNT(*)
        FROM EMP E, DEPT
                WHERE E.dept_code=DEPT.dept_cd
                        AND salary>(SELECT AVG(salary)FROM EMP
                                WHERE EMP.dept-code=E.dept_code)
                                        GROUP BY dept_name;
```

(iii) Departments having maximum employees earning more than 25000.

```
SELECT dept_name, COUNT(*)
        FROM EMP E,DEPT
                WHERE E.dept_code=DEPT.dept_cd AND salary>25000
                        GROUP BY dept_name
                                HAVING COUNT(*)>=ALL(SELECT COUNT(*)
                FROM EMP GROUP BY dept_code);
```

(iv) Name of employee(s) who earn maximum salary in the organization.

```
SELECT name FROM EMP
        WHERE salary=(SELECT MAX (salary) FROM EMP);
```

Q.112 Let R = (A,B,C,D) and F be the set of functional dependencies for R given by {A→B, A→C, BC→D} **(4)**

Prove A→D.

Ans: Given set of functional dependencies for a relation R is {A→B, A→C, BC→D}

By using Armstrong Axioms, named projectivity, we can show that

A→BC (as A→B, A→C)

Since BC→D, so by transitivity rule,

A→BC and BC→D means A→D

Hence Proved.

Q113 **a.** Consider a relation TENANT (NAME, CITY, STREET#, APT#, APT_TYPE, RENT, LANDLORD, ADDRESS) where following functional dependencies hold

APT#STREET#CITY □ADDRESS

ADDRESS→APT_TYPE

NAME ADDRESS →RENT

LAND_LORD APT_TYPE→RENT

(i) Are the following relation schemes in 3NF?

APARTMENT (APT_TYPE, ADDRESS, RENT)

DWELLER (NAME, ADDRESS, RENT)

(ii) What updating and insertion anomalies do you foresee in TENANT, APARTMENT & DWELLER relations.

Do the APARTMENT & DWELLER relations have lossless join?

(iii) Find a key of this relation. How many keys does TENANT have? **(2x4)**

(iv) Find a key of this relation. How many keys does TENANT have? **(2x4)**

b. Find the decomposition of TENANT into 3NF having lossless join and preserving dependencies.

Ans:

(a) (i) Consider the relation APARTMENT (E,H,F)

Given H→E

The key in this relation will be (HF).

This relation will not be in 3NF as it is not in 2NF. The reason is that there exists a partial function dependency H→E, where (HF) is the key attribute. To make it in 2NF, decomposition will be R1(E,H) and R2(H,F) means E1(APT_TYPE, ADDRESS) and R2 (ADDRESS, RENT)

For another relation DWELLER (A,H,F), the key will be (AH) as AH → F. This relation does not have any partial as well as transitive functional dependency, therefore in 3NF.

(ii) TENANT relation is having redundant data, which needs updation and deletion in all corresponding records if an updation or deletion is made for one value. TENANT relation is not in 2NF. As APT_TYPE is functionally dependent on ADDRESS, therefore every time a apartment type is changed, its corresponding address will be changed every where, otherwise leading to inconsistent state. Moreover if an address is having only one record in the relation and it gets deleted then the information about the apartment type will also be lost. As the DWELLER relation is normalized, therefore there will not be any anomalies in this relation.

(iii) A relation scheme R can be decomposed into a collection of relation schemes to eliminate anomalies contained in the original scheme R. However any such decomposition required that the information contained in the original relation be maintained or in other words the joining of the decomposed relations must give the same set of tuples as the original relation. To check whether the decomposition is lossless or not, the following algorithm is used :

A decomposition of relation schema R<(X,Y,Z), F>into R1<(X,Z), F2> is lossless if the common attribute X of R1 and R2 form a super key of at least one of these i.e. X→Y or X → Z.

Applying the algorithm for the two relations, APARTMENT and DWELLER, the common attribute is (H,F), which is a key of APARTMENT RELATION. Therefore the two relations given have lossless join.

(iv) To find the key in the relation, following rule will be applied :

Given a relation scheme R {A1, A2, A3,…..An} and a set of functional dependencies F, a key of R is a subset of R such that K → A1, A2, …. An is in the closure of F and for any Y → K, Y → A1, A2, ….. An is not in closure of F.
For TENANT relation, we have to find a K such that al other attributes are functionally dependent on it but not on any its subset. The key in this case will be (ABCDG) as all other attributes are FD on it but not on any subset of it.

b. The key for the TENANT relation will be (A,B,C,D,G). Since H→E and H is FD on key i.e. (A,B,C,D,G) therefore a transitive FD exists here. Therefore the relation will be decomposed in R1(A,B,C,D,F,G,H) and R2(H,E). This decomposition is lossless decomposition as the common element in both the relation is H and H→E (as definition given in (iii).

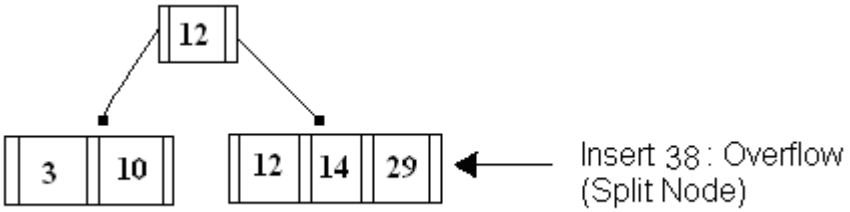
Q.114 Construct a B+ tree for the following set of key values under the assumption that the number of key values that fit in a node is 3. .Key values (3,10,12,14,29,38,45,55,60,68) Show the step involve in the following insertions (use your algorithm)
Insert 11, insert 30. **(10)**

Ans:

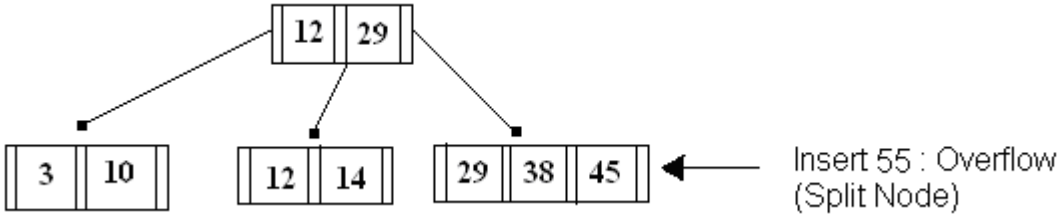
Construct a B+ tree for the following set of key values
n=4
K = n-1 =3
Max = 3 values allowed in leaf node
Min = n-1 | (n-1)/2 | = 2 values allowed in leaf node

Insert 3, 10 , 12

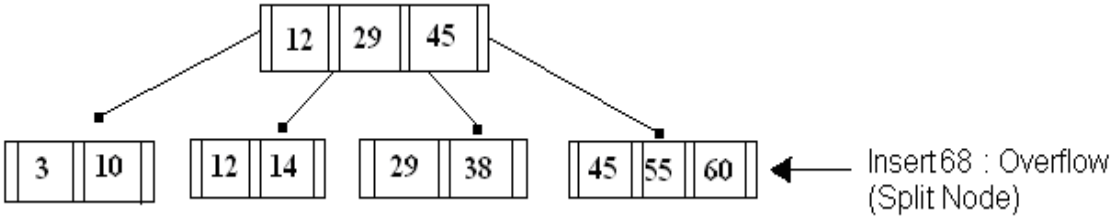
Insert 14,19



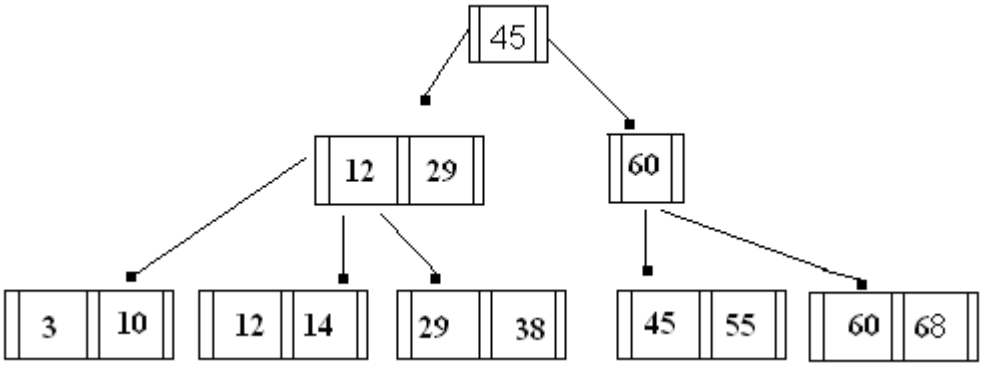
Insert 38,45



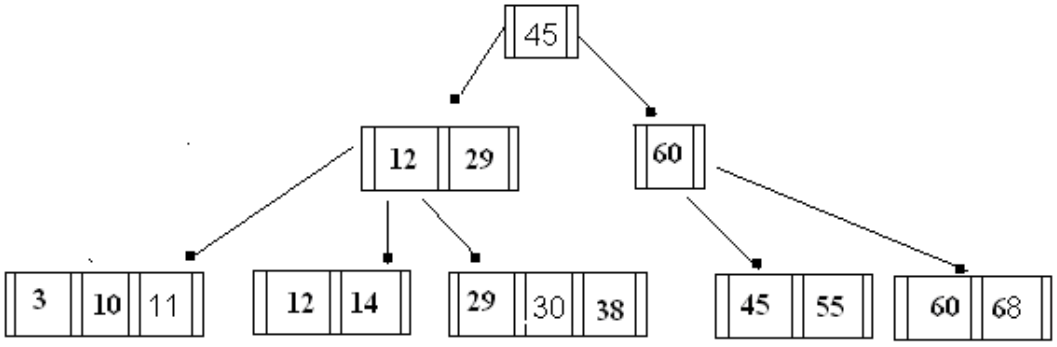
Insert 55,60



Insert 68



Insert 11 and 30



Q.115 Explain the concept of two phase locking and show that it guarantees serialisability. **(6)**

Ans: It concerns the positions of locking and unlocking operations in every transaction. In this protocol each transaction issue lock and unlock requests in two phases.

Growing Phase : In this phase a transaction may obtain locks but may not release any lock. Here the . of locks increases from zero to the maximum for the transaction.

Shrinking Phase : In this phase a transaction may release locks but may not obtain any new locks. Here the number. of locks decreases from maximum to zero.

The basic concept of this protocol is that, all the locks are first acquired before any of the locks released. No new locks can be granted after the releasing of any lock. There are several versions of this protocol

(i) **Static (or conservative) Two Phase Locking:** In this scheme, all the data items are locked before any operations on them and are released only after the last operation performed on any data item.

X(X)
X (Y)
Read (X)
Write (X)
Read (Y)
Y = Y+N
Write Y
Unlock (X)
Unlock (Y)

In static 2PL, requests from other transactions for data items locked by the previous transaction will be delayed unnecessarily, thus causing a serious impact on system performance.

(ii) **Dynamic Two-Phase Locking:** Here a transaction locks a data item immediately before any operation is applied on the data item. After finishing all the operations on all the data items, it releases all the locks. An example of dynamic 2PL is given below.

X(X)

Read (X)

X=X+M

Write (X)

X (Y)

Read (Y)

Y=Y+N

Write Y

Unlock (Y)

Unlock (X)

(iii) Strict Two-Phase locking : A transaction T does not release any of its exclusive (X) locks until that transaction commits or aborts. In this way no other transaction can access the item that is written by T unless the transaction T commits. This is a better technique than the two discussed.

In 2PL, a transaction may not be able to release its locks even after using it. The data item will remain locked until the transaction finishes all the actions on all the data items. If every transaction in a schedule follows 2PL then conflict serializability can be ensured. This is because if the schedule is not serializable, the precedence graph for schedule S consisting of transactions T1, T2 ….. Tn will have a cycle … Suppose that a cycle consists of T1, T2 ….. Tn ,T1. This means that a lock operation by T2 is followed by an unlock operation by T1; A Lockoperation T3 is followed by an unlock operation by T2 … and so on. However this is a contradiction of the assertion that T1 is using two phase locking protocol. Thus the assumption that graph is having a cycle is wrong and hence S is serializable.

Q.116 Explain recovery process after system failure using checkpoint. **(6)**

Ans:. Checkpoint scheme is an additional component of the log based recovery system. This scheme issued to limit the volume of log information that has to be handled and processed in the event of a system failure involving the loss of volatile data. The checkpoint operation is performed periodically copies log information onto the stable storage.

At each checkpoint, following information is kept.

- A start-of-checkpoint record, which gives the identification that it is a checkpoint.
- All log information from the buffers in the volatile storage is copied to the log on stable storage.
- All database updates from the buffers in the volatile storage are propagated to physical database.
- An end-of-checkpoint record is written and the address of the checkpoint record is saved on a file accessible to the recovery routine on start-up after a system crash.

When failure do occur, it is often possible to resume processing from the most recent checkpoint. All transactions that were saved before the checkpoint time need not be considered for the recovery operation. All transactions that were started before the checkpoint time but were not committed at that time are placed in an undo list, which is a list of transactions to be undone.

Now the recovery system scans the log in a backward direction from the time of the system crash. If it finds that a transaction in the undo list has committed, that transaction is removed from the undo list and placed in the REDO list, which contains all the transactions which have to be redone. Thus a system crash occurring during the checkpoint operation requires recovery to be done using the most recent previous checkpoint.

Q.117 What are the advantages of the tree locking protocol over the two-phase locking protocol? Explain the phantom phenomenon. Why this phenomenon may lead to an incorrect concurrency execution despite the use of a protocol that ensures serializability?

(12)

Ans: A tree locking protocol can be defined as follows :

- All locks are exclusive locks.
- Locking a node does not automatically lock any descendent of the locked node.
- The first item locked by a transaction can be any data item including the root node.
- Except for the first data item locked by a transaction, a node cannot be locked by a transaction unless the transaction has already successfully locked its parent.

- No items are locked twice by a transaction; thus releasing a lock on a data-item implies that the transaction will not attempt another lock on the data item.

This protocol is having an advantage over two-phase protocol in the sense that in tree locking protocol, a data item can be released earlier by a transaction if the data item and its unlocked descendants in the sub-tree are not required by the transaction. In this way a greater amount of concurrency is obtained.

Phantom Phenomenon : The phantom phenomenon can be explained by using the following example. Consider a transaction T1 that executes the following SQL statement on the employee relation.

```
SELECT SUM (sal) FROM employee
    WHERE deptno=10;
```

In this statement, transaction T1 requires to access all the tuples of relation employee pertaining to department number 10.

Problem will be encountered if there is another transaction, which is run to reflect the receive of SUM(sal). Let T2 be another transaction which issues an insert statement to the same relation, employee :

```
INSERT INTO employee
    VALUES
        (100, 'Ashok', 5000, 'A-34 Ashoka Road', 10):
```

Let S be the schedule involving T1 and T2. These two transactions can conflict from following two reasons:

- If T1 uses the tuple newly inserted by T2 in computing SUM(sal), then T1 read a value written by T2. Thus in a serial schedule equivalent to S, T2 must come before T1.
- If T1 does not use the newly created tuple by T2 in computing SUM(sal) then in a serial schedule equivalent to S, T1 must come before T2.

In second case, T1 and T2 do not access any tuple in common, yet they conflict with each other. Here locking of records did not prevent the creation of a new tuple, which was created after the existing records have been locked. If concurrency control is performed at the tuple granularity, this conflict would go undetected. This phenomenon is called as Phantom Phenomenon. The problem could be prevented if the locking of records also prevents the addition of such phantom records. The locking of a record belonging to a record type must guarantee that no new record occurrences of the record type can be added until the lock is released.

Q.118 Prove that a relation which is 4NF must be BCNF. **(4)**

Ans Let R be in 4NF. Assume it is not in BCNF. Therefore, there exists an FD $X \rightarrow Y$ in R such that x is not a super key. But by the rule M1 $X \rightarrow Y \models x \rightarrow\rightarrow Y$. Again x here is not a super key. This contradicts the fact that R is in 4NF. Therefore, our assumption is false. Every R in 4NF must be in BCNF.

Q.119 Explain ill effects of concurrency in terms of the 3 problems which occur. **(10)**

Ans: There are three ways in which a transaction through correct in itself can produce the wrong answer if interference occurs on the part of some other transaction.

The three problems are :

(i) The lost update problem

Example :

Transaction A	Time	Transaction B
Retrieve P	t1	
	t2	Retrieve P
Update P	t3	
	t4	Update P

Transaction A losses an update at time t4 .

(ii) The uncommitted dependency problem

Transaction A	Time	Transaction B
	t1	
	t2	Update P
Retrieve P	t3	
	t4	Rollback

Transaction A becomes dependent on an uncommitted change at time t3.

(iii)

Transaction A	Time	Transaction B
	t1	
	t2	Update P
Update P	t3	t4 Rollback

Transaction A updates change at time t3 and losses that update at time t4.

Q.120 Compare the two log-based recovery scheme in terms of ease of implementation and overhead cost. **(6)**

Ans: There are two log based recovery techniques : **deferred update** and **immediate update** scheme, which are also called as NO-UNDO/REDO and UNDO/NO-REDO techniques respectively.

In deferred update scheme, actual updates to the database are deferred or postponed until after a transaction completes its execution successfully and reaches its commit point. Before reaching commit, all transactions updates are recorded in the log and in the cache buffer. After the transaction reaches its commit point and the log is force written to the disk, the updates are recorded in the database. If a transaction fails before it reaches the commit point, it would not have modified the database and so no undo is required. But it may be required to redo some of the operations as their effects may not have reached the database. In case of failure, log files are used to perform recovery operations. We examine the log file starting the last entry and go back till the most recent checkpoint. The redo procedure performs all the writes to the database using the after-image log records for the transaction, in the order in which they were written to the log. Thus this method guarantees that we will update any data item that was not properly updated prior to the failure.

In immediate update technique, the database may be updated by some operations of transaction before the transaction reaches its commit point. In case of failure, we will have to redo the updates of committed transactions and undo the effects of uncommitted transactions. Similar to deferred update scheme, here also log files are used to perform the recovery scheme. Write-ahead protocol is used to record the update operations in the log (on disk) before it is written to the database. If a transaction aborts, the log can be used to undo it, since it contains all the old values for the update fields.

Q.121 Briefly describe the different kinds of users of a DBMS. **(6)**

Ans: Different types of DBMS users are:

(i) **Software Engineers:** These are the people responsible for developing application programs using DBMS.

(ii) **End users:** These are the people whose jobs require access to the database for querying, updating, and generating reports.

(iii) **Database Administrator**: They are responsible for authorizing access to the database, coordinating and monitoring its use.

(iv) **Database designers**: They are responsible for identifying data to be stored in the database.

Q.122 Define the concept of aggregation. Give two examples where this concept is useful. **(4)**

Ans: Aggregation transforms a relationship between objects into a higher-level object. A new data type, called aggregate, is developed which, under certain criteria of "well-definedness," specifies aggregation abstractions. Relational databases defined as collections of aggregates are structured as a hierarchy of n-ary relations. To maintain well-definedness, update operations on such databases must preserve two invariants. Well-defined relations are distinct from relations in third normal form. It is shown that these notions are complementary and both are important in database design. A top-down methodology for database design is described which separates decisions concerning aggregate structure from decisions concerning key identification. It is suggested that aggregate types, and other types which support real-world abstractions without introducing Key identification**.**

Q.123 Explain the following. Give an example

(i) Superkey

(ii) Weak entity set

(iii) Attribute inheritance **(6)**

Ans: (i) Superkey: A **superkey** is defined in the relational model of database organization as a set of attributes of a relation variable for which it holds that in all relations assigned to that variable there are no two distinct tuples (rows) that have the same values for the attributes in this set. Equivalently a superkey can also be defined as a set of attributes of a relvar upon which all attributes of the relvar are functionally dependent.

(ii) **Weak Entity Set:** An Entity set that does not have sufficient attribute to form a primay key is termed as weak entity set. For example, dependents of an employee in an organization do not have an identifying attribute.

(iii) **Attribute inheritance:** During the rendering of the objects in a view, attribute sets of objects higher in the view hierarchy are inherited by objects below them. For example, if the attribute set of a view specifies a particular diffuse color, then all objects in that view are rendered with that diffuse color, *unless* some other attribute set overrides the color specified in the view attributes. That is, if some face of some object has an attribute set containing a different diffuse color, the face's diffuse color overrides the diffuse color that otherwise would have been inherited from the view attribute set.

Q.124 Define the following

(i) A relation.

(ii) Atom of domain relational calculus. **(4)**

Ans: (i) A relation: A database relation is a predefined row/colum format for storing information in a relational database. Relations are equivalent to tables.
(ii) Atom of domain relational calculus: An atom has the form
$< x_1, x_2, \ldots, x_n > \rightarrow r$, where *r* is a relation on *n* attributes and $x_1, x_2, \ldots, x_n$ are the domain variables or domain constraints.
$x\Theta y$, where *x* and *y* are domain variables and Θ is the comparison operator ($<$,$>$, $\leq$, $\geq$, $=$, $\neq$). It is required that *x* and *y* have domains that can be compared by Θ.
$x\Theta c$, where *x* is a domain variable, Θ is a comparison operator, and *c* is a constraint in the domain of attributes for which x is a domain variable.

Q.125 Given the relations R(A, B, C) and S(C,D,E,F) give an expression in tuple relational calculus that is equivalent to each of the following

(i) $\Pi_{A,B,C}(R)$

(ii) $\sigma_{E=10}(S)$

(iii) $R \bowtie S$

(iv) $R \cup S$ **(12)**

Ans: (i) $\{ t \mid t \in R \}$
(ii) $\{ t \mid t \in S \wedge t[E] = 10 \}$
(iii) $\{ t,P \mid t \in R \wedge P \in S \wedge t[C] = P[C] \}$
(iv) R U S is not defined as these two relations are not union- compatible

Q.126 Given the relations Staff (staff No, position, salary) and Property (number, rent, staff No) given below. The staff looks after a given property.

Staff

Staff No	position	salary
SL21	Manager	50000.00
SL37	Assistant	15000.00
SG14	Supervisor	25000.00
SG5	Manager	45000.00

Property

Number	Rent	Staff No
PA14	5000.00	SL21
PG4	6000.00	SG5
PL94	10000.00	SL21

Give the result table for the following SQL queries

(i)
```
SELECT  position, COUNT(staff No) AS POS, my count
        FROM    Staff
```

(ii)
```
SELECT  staff No
        FROM    Staff
        WHERE   salary > (SELECT AVG(salary) FROM Staff)
```

(iii)
```
SELECT      staff No
FROM        Property
GROUP By    staff No
HAVING      COUNT(*) > 1
```
(iv)
```
INSERT INTO  Staff
VALUES       ('SG33', 'Assistant')
```
(16)

Ans:(i) This query will return an error message as it is not possible to use an aggregation function and an attribute without using the group by clause.

(ii)

Staff No
SL21
SG5

(iii)

Staff No
SL21

(iv)

Staff No	**Position**	**Salary**
SL21	Manager	50000.00
SL37	Assistant	15000.00
SL14	Supervisor	25000.00
SG5	Manager	45000.00
SG33	Assistant	

Q.127 Derive the union rule, decomposition rule and the pseudoTransitivity rule using the three Armstrong's axioms. **(6)**

Ans: **Union** $\{ X \rightarrow Y, X \rightarrow Z\} \models X \rightarrow YZ$
Proof: $X \rightarrow Y$
$X \rightarrow Z$
$X \rightarrow YZ$ (Augmentation)
$XY \rightarrow YZ$
$X \rightarrow YZ$
Decomposition $\{ X \rightarrow YZ\} \models X \rightarrow Y$
Proof: $X \rightarrow YZ$
$YZ \rightarrow Y$
$X \rightarrow Y$
Pseudo transitivity $\{ \mathbf{X} \rightarrow Y, WY \rightarrow Z\} \models WX \rightarrow Z$
Proof: $X \rightarrow Y$
$WY \rightarrow Z$
$WX \rightarrow WY$
$WX \rightarrow Z$

Q.128 Define multivalued dependency. What do understand by trivial multivalued dependency? **(4)**

Ans: A **multivalued dependency** *is a full constraint* between two sets of attributes in a relation. In contrast to the functional dependency, the **multivalued dependency** requires that certain tuples be present in a relation. Therefore, a multivalued dependency is a special case of *tuple-generating dependency*. The multivalued dependency plays a role in the 4NF database normalization.
An MVD $X \rightarrow\rightarrow Y$ in R is called a trivial MVD if (a) Y is a subset of X or

(b) X U Y = R

Q.129 Given R (A, B, C, D, E) and M the set of multivalued dependencies
(i) $A \rightarrow\rightarrow BC$
(ii) $B \rightarrow\rightarrow CD$
(iii) $E \rightarrow\rightarrow AD$
Is R in 4NF? Justify your answer. If it is not, decompose R into 4NF. **(6)**

Ans: A table is in 4NF if and only if, for every one of its non-trivial multivalued dependencies $X \rightarrow\rightarrow Y$, X is a superkey—that is, X is either a candidate key or a superset thereof.

Q.130 Describe the four main ways of optimising disk block access. **(8)**

Ans: 1. Disk
2. Non-volatile write buffers
3. File organization (***Clustering***)
4. Log-based file system

Q.131 Describe the algorithm for updating indices for a single level index when a record is
(i) Inserted **(ii)** deleted
What will be the modification if there are multilevel indices? **(8)**

Ans: **Inserted:** The time it takes to insert a new data item. This value includes the time it takes to find the correct place to insert the new data item, as well as the time it takes to update the index structure.
Deleted: The time it takes to delete a data item. This value includes the time it takes to find the item to be deleted, as well as the time it takes to update the index structure

Q.132 How do you estimate the query cost for natural join when
(i) $R \cap S = \phi$
(ii) $R \cap S$ is a foreign key **(6)**

Ans: (i) If $R \cap S = \Phi$, then $r \bowtie s$ is the same as $r \times s$.
(ii) If $R \cap S$ in S is a foreign key in S referencing R, Then the number of tuples in $r \bowtie s$ is exactly the same as the number of tuples in S.
The case for $R \cap S$ being a foreign key referencing S is symmetric

Q.133 Given two relations R (A,B) and S (B,C) with number of tuples in R and S equal to 500 and 1000 respectively and B is the foreign key in R, what is the number of tuples in $R \bowtie S$. **(4)**

Ans: The number of tuples in $R \bowtie S$ is 500000.

Q.134 Explain Thomas' Write rule. Show how it is different from timestamp ordering proto**(6)**

Ans: The Thomas Write rule is a rule in timestamp-based concurrency control.

Given a Timestamp on a transaction T, TS(T) and Write Timestamp on an object O, WTS(O):
It states if TS(T) < WTS(O), the current write action has been made obsolete by the most recent write of O, which follows the current write according to timestamp ordering.
Given a non-conflict serializable transaction schedule:
Text: T1:R(A), T2:W(A), T2 Commit, T1: W(A), T1 Commit.
The Thomas Write Rule relies on the fact that T1's write on object A is never seen by any transaction and postulates that the schedule above is equivalent to the schedule below where T2 occurs strictly after T1, and that hence the write of T1 can be ignored:
Text: T1:R(A), T1: W(A), T1 Commit, T2:W(A), T2 Commit.
This schedule has the same effect as the first and is conflict serializable

Q.135 Explain
(i) recoverable schedule. **(ii)** cascadeless schedule **(8)**

Ans: Recoverable Schedule: A recoverable schedule is one where for each pair of transactions Ti and Tj such that Tj reads a data item previously written by Ti, the commit operation of Ti appears before commit operations of Tj.
Cascadeless Schedule A cascadeless schedule is one where for each pair of transactions Ti and Tj such that Tj reads a data item previously written by Ti, the commit operation of Ti appears before the read operation of Tj.

Q.136 Define two-phase locking protocol. **(2)**

Ans: According to the *two phase locking* protocol, locks are handled by a transaction in two distinct, consecutive phases during the transaction's execution:
Phase 1: locks are acquired and no locks are released.
Phase 2: locks are released and no locks are acquired.
The serializability property is guaranteed for a schedule with transactions that obey the protocol. The 2PL *schedule class* is defined as the class of all the schedules comprising transactions with data access orders that could be generated by the 2PL protocol.

Q.137 Consider the following two transactions
T1 : read (A);
read (B);
B = A + B;
write (B)
T2 : write (A)
read (B)
Add lock and unlock instructions so that the transaction T1 and T2 observe two-phase locking protocol. Is it deadlock free? **(6)**

Ans:T1: lock- S(A)
Read (A)
Lock –X (B)
Read (B)
B = A +B
Write (B)
Unlock (A)

Unlock(B)
T2:lock-X(A)
Lock**-** S(B)
write (A)
Read (B)
Unlock(A)
Unlock(B)

Execution of these transactions can result in deadlock. For example, consider the following partial schedule:

T1	T2
Lock- S(A)	
Read(A)	
	Lock- X(A)
Lock – X(B)	
	Lock – S(B)

Q.138 Explain the recovery process of a checkpoint mechanism. How does the frequency of checkpoints affect
(i) system performance when no failure occurs.
(ii) the time it takes to recover from a system crash **(8)**

Ans: A checkpoint log record indicates that a log record and its modified data have been written to stable storage and that the transaction needs not to be redone in case of a system crash. Obviously, the more often checkpoint are performed, the less likely it is that redundant updates will have to be performed during the recovery process. System performance when no failure occurs—If no failures occur, the system must incur the cost of performing checkpoints that are essentially unnecessary. In this situation, performing checkpoints essential often will lead to better system performance.
(i) The time it takes to recover from a system crash—The existence of a checkpoint record means that an operation will not have to be redone during system recovery. In this situation, the more often checkpoints were performed, the faster the recovery time is from a system crash
(ii) The time it takes to recover from a disk crash—The existence of a checkpoint record means that an operation will not have to be redone during system recovery. In this situation, the more often checkpoints were performed, the faster the recovery time is from a disk crash.

Q.139 Write short notes on
(i) hash file organization.
(ii) physical and logical independence. **(8)**

Ans:i) Hash file organization
1. **Hashing** involves computing the address of a data item by computing a function on the search key value.
2. A **hash function h** is a function from the set of all search key values *K* to the set of all bucket addresses *B*.

We choose a number of buckets to correspond to the number of search key values we will have stored in the database.

To perform a lookup on a search key value K_i, we compute hk_i, and search the bucket with that address.
If two search keys *i* and *j* map to the same address, because $h(K_i)=h(K_j)$, then the bucket at the address obtained will contain records with both search key values.
In this case we will have to check the search key value of every record in the bucket to get the ones we want.
Insertion and deletion are simple

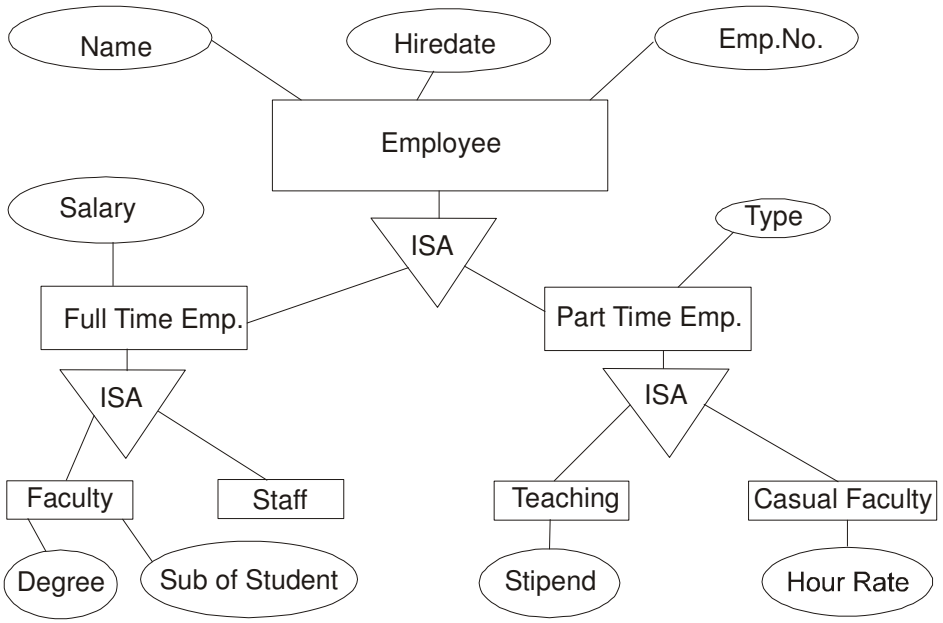
(ii) Physical and Logical independence

Physical data independence: The capability to modify physical level without causing application program to be rewritten.
Logical independence: The capability to modify logical level without causing application program to be rewritten.

Q.140 The entity set EMPLOYEE is a generalization of the entity sets FULL_TIME_EMPLOYEE and PART_TIME_EMPLOYEE. The former is a generalization of the entity sets FACULTY with attributes degree and subject of interest and STAFF with attribute classification; the latter, is a generalization of the entity sets TEACHING with attribute stipend and CASUAL_FACULTY with attribute hour rate. STAFF inherits the attribute Salary of the entity set FULL_TIME_EMPLOYEE and the latter, in turn, inherits the attributes of EMPLOYEE. FULL_TIME_EMPLOYEE is a specialization of the entity set EMPLOYEE and is differentiated by the additional attribute Salary. Similarly, PART_TIME_EMPLOYEE is specialization differentiated by the presence of the attribute Type. Each employee must have attributes empno, name and hire_date.

(i) Draw an E-R diagram for the system.
(ii) Convert this E-R diagram to relational tables. **(12)**

Ans. (i)



(ii) Emp(name, hiredate,empno)
Full Time Emp (name, hiredate, empno, salary)
Part Time Emp (name,hiredate,empno,type)
Faculty (empno,salary,degree,subofinterest)
Staff (empno,sname,sno,salary)
Teaching (empno, type, stipend)
Casual Faculty (empno,type,hourrate)

Q.141 Explain the difference between a one-to-many and a many-to-many relationship. Which logical data structures have one-to-many and which have many-to-many relationship?**(4)**

Ans: One-to-Many: means that at most one entity in set A is assigned to any number of entities in set B. For example Employee in set A has any number of Phone numbers in set B.
Many-to-Many: means that any numbers of entities in set A assigned to any number of entities in set B. For example several employees in set A may have several accounts in set B.

Q.142 What are the DBMS languages? Explain. **(6)**

Ans:1) **Data Definition Languages(DDL):** It is used to specify database schema e.g, CREATE DROP statements etc.
2) Data Manipulation language(DML): DML is used to express database queries and update. Eg, SELECT, UPDATE, DELETE statements, etc.

Q.143 What are Armstrong's inference rules?
Suppose we are given relation R with attributes A, B, C, D, E, F, and the FDs,
A→BC
B →E
CD →EF
Prove that FD AD→F also holds in R. **(10)**

Ans: Armstrong's axioms are a set of axioms (or, more precisely, inference rules) used to infer all the functional dependencies on a relational database. The axioms are sound in that they generate only functional dependencies in the closure of a set of functional dependencies (denoted as F^+) when applied to that set (denoted as F). They are also complete in that repeated application of these rules will generate all functional dependencies in the closure F^+.
Given: A → BC CD → EF
To Prove: AD → F
After applying decomposition rule
A→ B ----------1
A→C ----------2
B→E ----------3
CD→E----------4
CD→F----------5
Applying pseudotransitivity rule on 2 and 5
AD→F
Hence Proved

Q.144 Consider the relations:

PROJECT(proj#, proj_name)
EMPLOYEE(emp#, emp_name)
ASSIGNED(proj#, emp#)
Use relational algebra to express the following queries:

(i) Find the employee number of employees who work on at least all of the projects that employee 107 works on.

(ii) Get the employee number of employees who work on all projects. **(8)**

Ans: (i) ASSIGNED_TO ÷ $\Pi_{\text{Project \#}}$ ($\sigma_{\text{emp\# = 107}}$(ASSIGNED_TO))) -107

(ii) ASSIGNED_TO ÷ $\Pi_{\text{Project \#}}$ (PROJECT)

Q.145 Use tuple and domain calculus to express the following query Compile a list of employee number of employees who work on all projects **(8)**

Ans: Tuple calculus query:

t[Emp #] / t ∈ ASSIGNED_TO Λ∀
P(P ∈ PROJECT → ∃ u(u ∈ ASSIGNED_TOΛP[Project#]=u[Project #] Λ t[Emp#] =u[Emp #]))}

Domain Calculus query

{ e| ∃ p(<p,e> ∈ ASSIGNED_TO Λ ∀ P1(<P1,n1,c1> ∈ PROJECT → <p1,e> ∈ ASSIGNED_TO))}

Q.146 What is the difference between serial and sequential files? How searching is applied on both? **(8)**

Ans: A **serial file** is one in which the records have been stored in the order in which they have occurred. They have not been sorted into any particular order.
An **example** of a serial file is an **unsorted transaction file**.
A **shopping list** is an example of a non-computerised serial file. Items are appended to the list when that item runs low.
Serial files can be stored on tape, disc or in memory.
A **sequential file** is one in which the records are stored in **sorted** order of one or more key fields.
An **example** of a sequential file is a **sorted transaction file**.
A **class register** is an example of a non-computerised sequential file sorted on surname

Q.147 Discuss the problem of Spurious tuples and how we may prevent it. **(8)**

Ans: A spurious tuple is, basically, a record in a database that gets created when two tables are joined badly. In database-ese, spurious tuples are created when two tables are joined on attributes that are neither primary keys nor foreign keys. To avoid spurious tuples, avoid joining relations that contain matching attributes that are not (foreign key, primary key) combinations because joining on such attributes may produce spurious tuples.

Q.148 Explain the followings:

(i) Third normal form

(ii) Query Processing
(iii) Relational Completeness
(iv) Radix conversion method **(16)**

Ans: (i) Third Normal Form: The **third normal form** (**3NF**) is a normal form used in database normalization. 3NF was originally defined by E.F. Codd in 1971. Codd's definition states that a table is in 3NF if and only if both of the following conditions hold:The relation R (table) is in second normal form (**2NF**)
Every non-prime attribute of R is non-transitively dependent (i.e. directly dependent) on every key of R.

(ii) Query Processing

Data Query Processing allows a user to report or analyze structured and unstructured data pulled from multiple data sources.

(iii) Relational Completeness

Codd defined the term ***relational completeness*** to refer to a language that is complete with respect to first-order predicate calculus apart from the restrictions he proposed. In practice the restrictions have no adverse effect on the applicability of his relational algebra for database purposes.

(iv) Radix conversion method

One clever way to convert binary numbers to BCD notation (binary-coded decimal) is the "double dabble algorithm". It can be adapted to convert binary numbers directly to ASCII digits, and to convert binary numbers into other bases. The double dabble algorithm also works for mixed bases -- for example, for converting a binary number of seconds into the decimal digits for days, 10s of hours, hours, 10s of minutes, minutes, 10s of seconds and seconds

Q.149 Consider the relations
EMP(ENO,ENAME,AGE,BASIC)
WORK_ON(ENO,DNO)
DEPT(DNO,DNAME,CITY)
Express the following queries in SQL
(i) Find names of employees whose basic pay is greater than average basic pay.
(ii) Find the sum of the basic pay of all the employees, the maximum basic pay, the minimum basic pay and the average basic pay. **(8)**

Ans: (i) Select ENAME from EMP where BASIC > (select avg(BASIC) from EMP);
(ii) Select sum(BASIC), max(BASIC), min(BASIC), avg(BASIC) from EMP;

Q 150 What are the General Transformation Rules for Relational operations? **(8)**

Ans: Transformation rules transform one relational algebra expression to AN EQUIVALENT ONE

- Used by the query optimizer to optimize query tree
- Any rule, if applied, makes sure that the resulting tree is equivalent ➔ resulting execution plan is equivalent

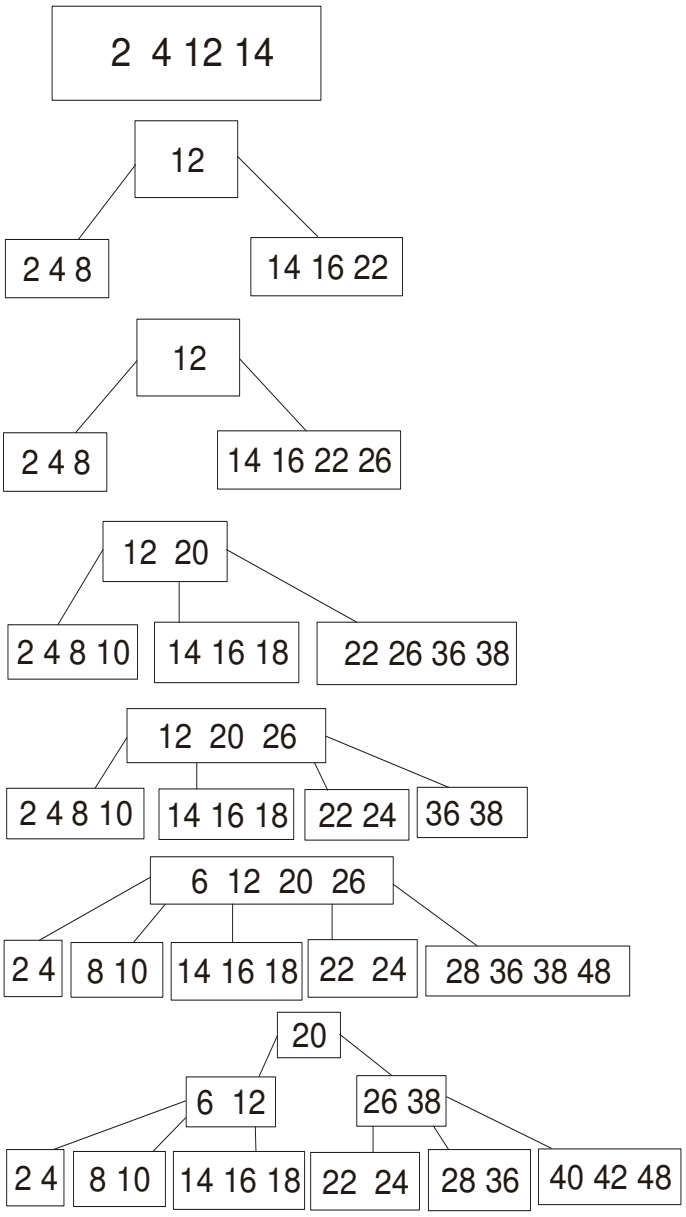
General Transformation Rules:

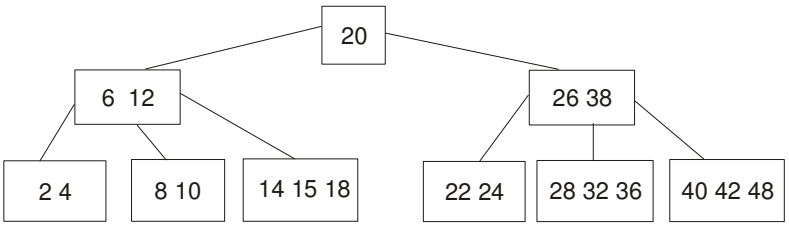
- *Commutativity of s*: The s operation is commutative:
  - $s_{c1}(s_{c2}(R)) = s_{c2}(s_{c1}(R))$
  - More selective selections first

Q.151 Draw a B-tree of order 5 by inserting the following data
(2 14 12 4 22 8 16 26 20 10 38 18 36 24 6 48 28 40 42 32)
After constructing the B-tree delete the following data:
(i) delete 36
(ii) delete 32

Ans.

(6+4=10)

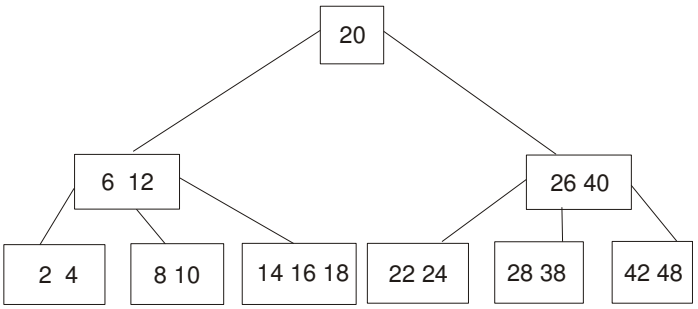




Final tree

While delete 36, remove it (no change in structure)

While delete 32, the B-tree looks as



Q.152 What is a view? Create a view of EMP table named DEPT 20, to show the employees in department 20 and their annual salary. **(6)**

Ans: View is a virtual table which does not contains it own data it fetch the data from the original table.

Create view DEPT20 as select * from dept;
Where dno=20 group by dno;

Q.153 Discuss the timestamp ordering protocol for concurrency control. How does strict timestamp ordering differs from basic timestamp ordering? **(8)**

Ans: The timestamp ordering protocol ensures that any conflicting read and write operations are executed in timestamp order. This protocol operates as follows:

1.Suppose that transaction T_i issues read (Q).

a. If *TS* (T_i) < W-timestamp (Q), then T_i needs to read a value of Q that was already overwritten. Hence, the read operation is rejected, and T_i is rolled back.

b. If *TS* (T_i) W-timestamp (Q), then the read operation is executed, and R timestamp (Q) is set to the maximum of R-timestamp (Q) and TS (T_i).

2. Suppose that transaction T_i issues write (Q).

a. If *TS* (T_i) < R-timestamp (Q), then the value of Q that T_i is producing was needed previously, and the system assumed that that value would never be produced. Hence, the system rejects the write operation and rolls T_i back.

b. If *TS* (T_i) < W-timestamp (Q), then T_i is attempting to write an obsolete value of Q. Hence, the system rejects this write operation and rolls T_1 back.

c. Otherwise, the system executes the write operation and sets W-time stamp (Q) to *TS* (T_i).

TS (T_i) = *R/W* - TS (Q).
If a transaction T_i is rolled back by the concurrency-control scheme as result of issuance of either a read or writes operation, the system assigns it a new timestamp and restarts it.

Q.154 Explain the shadow paging recovery technique. **(8)**

Ans: Shadow paging is a technique used to achieve atomic and durable transactions, and provides the ability to manipulate pages in a database. During a transaction, the pages affected by the transaction are copied from the database file into a workspace, such as volatile memory, and modified in that workspace. When a transaction is committed, all of the pages that were modified by the transaction are written from the workspace to unused pages in the database file. During execution of the transaction, the state of the database exposed to the user is that in which the database existed prior to the transaction, since the database file still contains the original versions of the modified pages, as they existed before being copied into the workspace. If a user accesses the database before the transaction is complete, or upon recovery of a failure, it will appear as though the transaction has not occurred

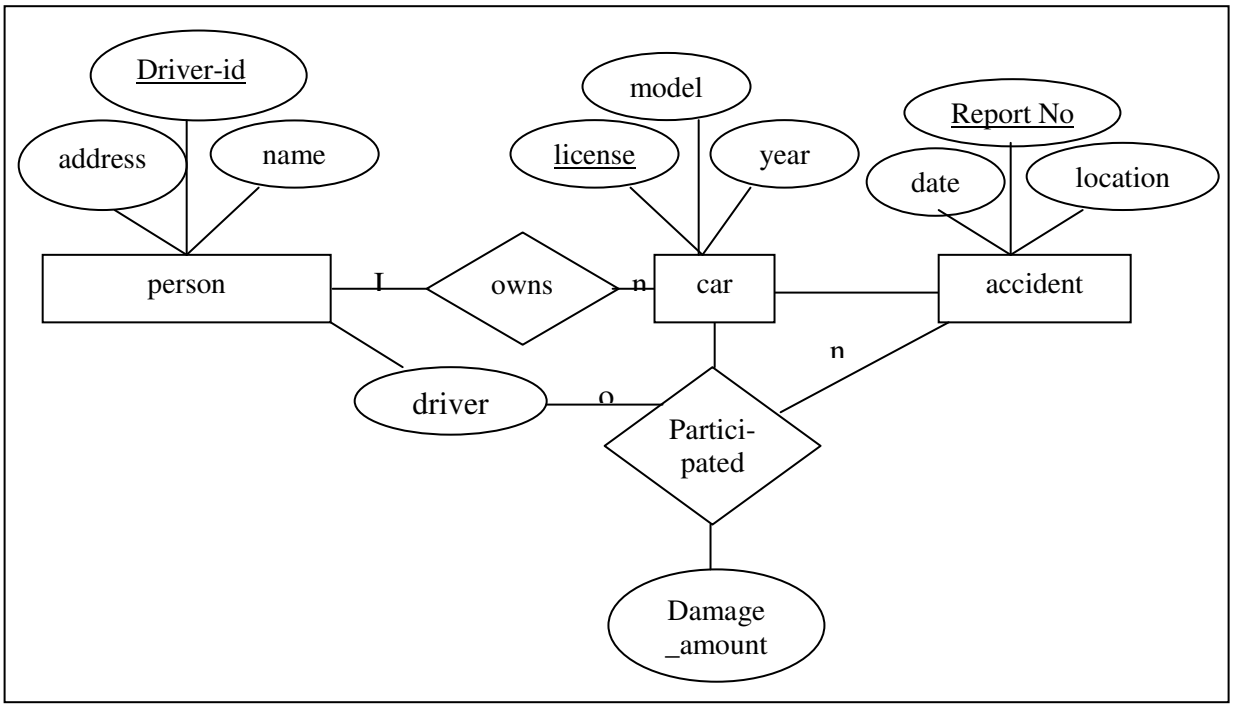
Q.155 Explain the steps for reduction of E-R model into relational model. **(8)**

Ans:(i) Entity set in E-R model will be treated as table name in relational Model.

(ii) Attributes of entity set in E-R diagram will be considered as fields or column name of table.
(iii) Underline attributes of Attribute in E-R model will be treated as Primary key of the table.

Q.156 Construct an E-R diagram for a car insurance company that has a set of customers,each of whom owns one or more cars. Each car has associated with it zero to any number of recorded accidents **(8)**

Ans:



Q.157 Consider the following relations **(8)**

SALESPERSON(SSN, Name, Start_Year, Dept_No)
TRIP(SSN,From_City,To_City,Departure_Date,Return_Date,Trip_ID)
EXPENSE(Trip_ID, Account#, Amount)
Write queries in relational algebra

(i) Give the details (all attributes to TRIP relation) for the trips that exceeded $2000 in expenses.
(ii) Print the SSN of salesman who took trips to 'Chandigarh'.
(iii) Print the total trip expenses incurred by the salesman with SSN ='234-56-7890'.

Ans:(i) $\Pi_{SSN, From\_City, To\_City, Departure\_Date, Return\_Date, Trip\_D}(\sigma_{AMT>2000}(TRIP \bowtie EXPENSE))$

(ii) $\Pi_{SSN}(\sigma_{To\_City = \text{"Chandigarh"}}(SALESPERSON \bowtie TRIP \bowtie EXPENSE))$

(iii) $\Pi_{AMOUNT}(\sigma_{SSN = '234\text{-}56\text{-}7890'}(TRIP) \bowtie EXPENSE)$

Q.158 How Relational Calculus is different from Relational Algebra? What do understand by TRC queries and DRC queries? **(8)**

Ans: Relational calculus consist of two calculi, the tuple relational calculus and the domain relational calculus, that are part of the relational model for databases and provide a declarative way to specify database queries. This in contrast to the relational algebra which is also part of the relational model but provides a more procedural way for specifying queries.
The relational algebra might suggest these steps to retrieve the phone numbers and names of book stores that supply *Some Sample Book*:

1. Join books and titles over the BookstoreID.
2. Restrict the result of that join to tuples for the book *Some Sample Book*.
3. Project the result of that restriction over StoreName and StorePhone.

The relational calculus would formulate a descriptive, declarative way:
Get StoreName and StorePhone for supplies such that there exists a title BK with the same BookstoreID value and with a BookTitle value of *Some Sample Book*.
The relational algebra and the relational calculus are essentially logically equivalent: for any algebraic expression, there is an equivalent expression in the calculus, and vice versa. This result is known as Codd's theorem.
TRC stands for tuple relational calculus and DRC stands for domain relational calculus. The TRC is based on specifying a query in terms of tuple variable and DRC is based on domain variables.

Q.159 What is DDL? Make a list of commands with short description used in DDL **(8)**

Ans: Data Definition Language (**DDL**) is a computer language for defining data structures. The term was first introduced in relation to the Codasyl database model, where the schema of the database was written in a Data Definition Language describing the records, fields, and "sets" making up the user Data Model.
List of DDL commands:

(i) To define a table/ relation
CREATE TABLE ------
(ii) To remove table/relation definition

DROP TABLE ----

(iii) To change the definition of an existing table
ALTER TABLE-------

Q.160 Consider the insurance database, where the primary keys are underlined.
person (ss#, name, address)
car (license, year, model)
accident (date, driver, damage-amount)
owns (ss#, license)
log (license, date, driver)
Construct the following SQL queries for this relational database.

(i) Find the total number of people whose cars were involved in accidents in 1989.

(ii) Find the number of accidents in which the cars belonging to "John Smith" were involved.

(iii) Add a new customer to the database.

(iv) Add a new accident record for the Toyota belonging to "Jones" **(8)**

Ans:(i) Select count(driver) from accident where date is between '01-Jan-89'and '31-DEC-89';

(ii) Select count(*) from accident,person,owns,log where name='John Smith' and person.ss# = owns.ss# and owns.License= log.License and log.driver = accident.driver

(iii) insert into person values(1,'Raj', 'A10 RajNagar Gzb');

(iv) insert into car values('L20', 1982, 'Toyata');

insert into person values('S1', 'Jones', 'A2 Kavi Nagar Gzb');
insert into owns values ('S1', 'L20');
insert into accident values (Sysdate,"Jones",10000)

Q.161 Explain Boyce-Codd Normal Form with example and also Compare BCNF and 3NF.**(8)**

Ans:
BCNF : For every functional dependency X->Y in a set *F* of functional dependencies over relation *R*, either:

- Y is a subset of X or,
- X is a *superkey* of R

whereas
3NF: For every functional dependency X->Y in a set *F* of functional dependencies over relation *R*, either:

- Y is a subset of X or,
- X is a *superkey* of R, or
- Y is a subset of K for some key K of R
  - no subset of a key is a key

Q.162 What are the reasons of bucket overflow? Explain any two methods for solving this problem. **(8)**

Ans: It is common for file structures to be divided into equal-length partitions, called buckets, into which records arrive for insertion and from which records are physically deleted. We give a simple algorithm which permits calculation of the average time until

overflow for a bucket of capacity n records, assuming that record insertions and deletions can be modelled as a stochastic process in the usual manner of queuing theory. We present some numerical examples, from which we make some general observations about the relationships among insertion and deletion rates, bucket capacity, initial fill, and average time until overflow. In particular, we observe that it makes sense to define the stable point as the product of the arrival rate and the average residence time of the records; then a bucket tends to fill up to its stable point quickly, in an amount of time almost independent of the stable point, but the average time until overflow increases rapidly with the difference between the bucket capacity and the stable point.

Q.163 What is irreducible set of dependencies? Relation R with attributes A, B, C, D, and FDs,
$A \rightarrow BC$
$B \rightarrow C$
$A \rightarrow B$
$AB \rightarrow C$
$AC \rightarrow D$
compute an irreducible set of FDs that is equivalent to this given set. **(8)**

Ans: Irreducible function depending set
A functional depending set S is irreducible if the set has three following properties:

1. Each right set of a functional dependency of S contains only one attribute.
2. Each left set of a functional dependency of S is irreducible. It means that reducing any one attribute from left set will change the content of S (S will lose some nformation).
3. Reducing any functional dependency will change the content of S.
   Sets of functional dependencies with these properties are also called *canonical* or *minimal*.

Step 1: After one attribute on LHS, FDs are:
A→ B---------- (1)
A→ C---------- (2)
B→ C---------- (3)
A → B---------- (4)
AB→C---------- (5)
AC→D---------- (6)
Since (1) and (4) are same, therefore removing (4)
Step 2: Checking if there is any FD implied by other FD for which closure of LHS of FD say **a**, is found by excluding **a** in set of FD. If closure has RHS of **a** then FD is redundant. Therefore

(1)	$\Rightarrow A^+ = ACD$	Since RHS of (1) is not in A^+ therefore (1)is redundant
(2)	$\Rightarrow A^+ = ABC$	Since RHS of (1) is in A^+ therefore (2) is redundant
(3)	$\Rightarrow B^+ = B$	implies non-redundant
(4)	$\Rightarrow AB^+ = ABC$	implies redundant
(5)	$\Rightarrow AC^+ = AC$	implies non-redundant

Therefore set of FDs left:
A→ B---------- (1)
B→ C---------- (3)
AC→ D---------- (6)

Step 3: These can be further reduced as
A→ B---------- (1)
B→ C---------- (3)
A→ D---------- (7)
Since adding A to both sides of (1) and (3) we get
AA→AB
AB→ AC
Implies A→ AC---------- (8)
(6) and (8) implies A→D which is (7).

Q.164 What is indexed sequential file organization? What are the applications of this organization? **(8)**

Ans: An index file can be used to effectively overcome the problem of storing and to speed up the key search as well. The simplest indexing structure is the single-level one: a file whose records are pairs key-pointer, where the pointer is the position in the data file of the record with the given key. Only a subset of data records, evenly spaced along the data file, are indexed, so to mark intervals of data records.
A key search then proceeds as follows: the search key is compared with the index ones to find the highest index key preceding the search one, and a linear search is performed from the record the index key points onward, until the search key is matched or until the record pointed by the next index entry is reached. In spite of the double file access (index + data) needed by this kind of search, the decrease in access time with respect to a sequential file is significant

Q.165 What are the General Transformation Rules for Relational Algebra Operations? **(5)**

Ans: General transformation rule for Relational Algebra are:

1. Cascade of s: A conjunctive selection condition can be broken up into a cascade (sequence) of individuals operations
   $s_{c1\ AND\ c2\ AND\ ...\ AND\ cn}(R) = s_{c1}(s_{c2}(...(s_{cn}(R))...))$
2. Commutativity of s: The s operation is commutative:
   $s_{c1}(s_{c2}(R)) = s_{c2}(s_{c1}(R))$
3. Cascade of p: In a cascade (sequence) of p operations all but the last one can be ignored:
   $p_{List1}(p_{List2}(...(p_{Listn}(R))...)) = p_{List1}(R)$
4. Commuting s with p: If the selection condition c involves only the attributes A1,….., An in the projection list, the two operations can be commuted:
   $p_{A1, A2, ..., An}(s_c(R)) = s_c(p_{A1, A2, ..., An}(R))$

Q.166 How does a query tree represent a relational algebra expression? **(5)**

Ans: This involves transforming an initial expression (tree) into an equivalent expression (tree) which is more efficient to execute. Two relational algebra expressions are said to be equivalent if the two expressions generate two relation of the same set of attributes and contain the same set of tuples although their attributes may be ordered differently.
The query tree is a data structure that represents the relational algebra expression in the query optimization process. The leaf nodes in the query tree corresponds to the input relations of the query. The internal nodes represent the operators in the query. When

executing the query, the system will execute an internal node operation whenever its operands available, then the internal node is replaced by the relation which is obtained from the preceding execution.

Q.167 Differentiate between static hashing and dynamic hashing. **(6)**

Ans: Static Hashing has the number of primary pages in the directory fixed. Thus, when a bucket is full, we need an overflow bucket to store any additional records that hash to the full bucket. This can be done with a link to an overflow page, or a linked list of overflow pages. The linked list can be separate for each bucket, or the same for all buckets that overflow. When searching for a record, the original bucket is accessed first, then the overflow buckets. Provided there are many keys that hash to the same bucket, locating a record may require accessing multiple pages on disk, which greatly degrades performance.
The problem of lengthy searching of overflow buckets is solved by Dynamic Hashing. In Dynamic Hashing the size of the directory grows with the number of collisions to accommodate new records and avoid long overflow page chains. Extendible and Linear Hashing are two dynamic hashing techniques.

Q.168 What is the two-phase locking protocol? How does it guarantee serializability? **(6)**

Ans: According to the *two phase locking* protocol, locks are handled by a transaction in two distinct, consecutive phases during the transaction's execution:
Phase 1: locks are acquired and no locks are released.
Phase 2: locks are released and no locks are acquired.
The serializability property is guaranteed for a schedule with transactions that obey the protocol. The 2PL *schedule class* is defined as the class of all the schedules comprising transactions with data access orders that could be generated by the 2PL protocol

Q.169 Explain the lost update problem and dirty read problem. **(6)**

Ans: Lost update problem: This problem occurs when two transactions that access the same database items have their operations interleaved in a way that makes the value of some database items incorrect. Eg: consider the following interleaved schedule:

T1	T2
read(X)	
X=X-N	
	read(X)
	X=X+N
write(X)	
	write(X)

In this case, X has incorrect value because its update by T1 is lost.
Dirty Read Problem: This problem occurs when one transaction updates a database item and then the transaction fails for some reason. The updated item is accessed by another transaction before it is changed lock to its original value. Consider the following interleaved schedule

T1	T2
	Sum=0
	read (A)

```
                    sum = sum + A
read(X)
X=X-N
Write(X)
                    read(X)
                    sum=sum+X
```

Suppose T1 fails this, then T2 has incorrect sum because change of T1 is not valid and should not have included when sum is computed.

Q.170 What are deadlocks? How can they be avoided? **(4)**

Ans: Deadlock refers to a specific condition when two or more processes are each waiting for another to release a resource, or more than two processes are waiting for resources in a circular chain. Deadlock is a common problem in multiprocessing where many processes share a specific type of mutually exclusive resource known as a *software,* or *soft,* lock. Computers intended for the *time-sharing* and/or *real-time* markets are often equipped with a *hardware lock* (or *hard lock*) which guarantees *exclusive access* to processes, forcing serialization. Deadlocks are particularly troubling because there is no *general* solution to avoid (soft) deadlocks.
One of the methods of deadlock avoidance related to timestamps there are two variations: wait-die and wound-wait.

Q.171 Explain the following:

(i) Normalization.
(ii) Joins in relational algebra.
(iii) Role of a database administrator.
(iv) B^+tree index files. **(4x4=16)**

Ans:(i) Normalization:
Database normalization, sometimes referred to as *canonical synthesis*, is a technique for designing relational database tables to minimize duplication of information and, in so doing, to safeguard the database against certain types of logical or structural problems, namely data anomalies. For example, when multiple instances of a given piece of information occur in a table, the possibility exists that these instances will not be kept consistent when the data within the table is updated, leading to a loss of data integrity. A table that is sufficiently normalized is less vulnerable to problems of this kind, because its structure reflects the basic assumptions for when multiple instances of the same information should be represented by a single instance only.

(ii) Joins in relational algebra:

a) Natural join

Natural join is a binary operator that is written as (R* S) where R and S are relations. The result of the natural join is the set of all combinations of tuples in R and S that are equal on their common attribute names. In this only one column along attributes having same name is retained

b) θ-join and equijoin

Consider tables *Car* and *Boat* which list models of cars and boats and their respective prices. Suppose a customer wants to buy a car and a boat, but she doesn't want to spend more money for the boat than for the car. The θ-join on the relation $CarPrice \geq BoatPrice$

produces a table with all the possible options. If the condition uses equality operator then it is also known as equijoin.

c) Outer join

They can be used when we want to keep all the tuples in R, or all those in S or all those in both relations in the result of the JOIN regardless of whether or not they have matching tuples in the other relation

(iii) Role of a database administrator

- **Defining the Schema**

The DBA defines the schema which contains the structure of the data in the application. The DBA determines what data needs to be present in the system ad how this data has to be represented and organized.

- **Liaising with Users**

The DBA needs to interact continuously with the users to understand the data in the system and its use.

- **Defining Security & Integrity Checks**

The DBA finds about the access restrictions to be defined and defines security checks accordingly. Data Integrity checks are also defined by the DBA.

- **Defining Backup / Recovery Procedures**

The DBA also defines procedures for backup and recovery. Defining backup procedures includes specifying what data is to backed up, the periodicity of taking backups and also the medium and storage place for the backup data.

- **Monitoring Performance**

The DBA has to continuously monitor the performance of the queries and take measures to optimize all the queries in the application.

(iv) B^+ tree index files

Indexing mechanisms used to speed up access to desired data.

E.g., author catalog in library

Search Key - attribute to set of attributes used to look up records in a file.

An index **file** consists of records (called index **entries**) of the form

Index files are typically much smaller than the original file

Ordered indices: search keys are stored in sorted order.

- All paths from root to leaf are of the same length
- Each node that is not a root or a leaf has between $[n/2]$ and n children.
- A leaf node has between $[(n–1)/2]$ and $n–1$ values

Special cases:

If the root is not a leaf, it has at least 2 children.

If the root is a leaf (that is, there are no other nodes in the tree), it can have between 0 and $(n–1)$ values

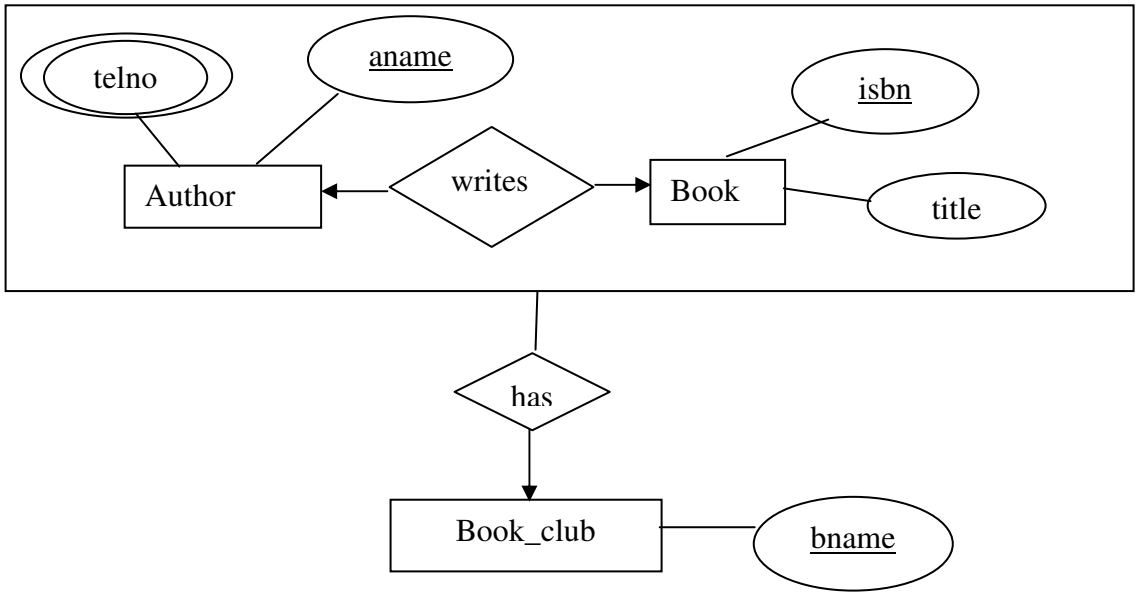
Q.172 Define cardinality and participation constraints on a relationship type, completeness constraint on generalization. **6**

Ans: Cardinality expresses the number of entities to which another entity can be associated via a relationship set. It may be 1:1, 1:N or M:N.

Participation of a entity set in a relationship set can be total or partial. It is total if every entity participates in the relationship. If only some entities participate in a relationship then it is partial.

Completeness constraint can be partial or total. In the case of total each higher level entity must belong to a lower level entity and in the case of partial some higher level entities may not belong to lower level entities.

Q.173. Consider the following ER diagram



Map the diagram to tables. Specify the table names and their attributes. **10**

Ans. The tables are

Author, with attributes aname

Book, with attributes isbn, title

Writes, with attributes isbn, aname

Has, with attributes isbn, aname, bname

Book_club, with attributes bname

Telno, with attributes phoneno, aname

Q.174 Describe the structure of a well formed formulae in relational calculus. **6**

Ans:

- An atom is a formula
- If P1 is a formula, then so are ¬P1 and (P1)
- If P1 and P2 are formulae, then so ar P1 V P2, P1 ^ P2, P1 → P2
- If P1(s) is a formula containing a free variable s and r is a relation then

$\exists s \varepsilon r$ (P1(s)) and $\forall s \varepsilon r$ (P1(s)) are also formulae

Q.175. Consider the relations given below with keys underlined
Branch(branchno, street, city)
Staff(staffNo, name, salary, branchNo, position, DOB)
propertyForRent(propertyNo, staffNo, rent)
Answer the following queries in relational algebra
1. Find the names of Staff who work in Delhi
2. List the staffNo of Staff who have not rent any property **6**

Ans.
1. $\prod_{name} (\sigma_{city = 'Delhi'}$ (Staff join Branch))
2. $\prod$ staffNo (Staff) - $\prod$ staffNo (propertyForRent)

Q.176 What is the result relation if we perform the relational algebra operator as shown below
OFFICE $\div \prod_{HQ} (\sigma_{NAME = 'Signet' \text{ or } NAME = 'Dell'} (PUBLISHER))$
for the relations given below **4**

PUBLISHER

PID	NAME	HQ
01	Acer	Mumbai
02	Signet	New York
03	Dell	London
04	HP	Sydney

OFFICE

PID	CITY
01	New York
01	London
02	London
02	Sydney
03	London
03	New York
04	New York
04	London

Ans.

PID
01
03
04

Q.177 In SQL, differentiate between
1. a subquery and a join
2. WHERE and HAVING clauses
3. DROP and DELETE
4. LEFT OUTER JOIN and RIGHT OUTER JOIN **8**

Ans:
1. If we need to obtain information from one or more tables then either subquery or join can be used. If the columns that are to appear in the result table are from different tables then a join must be used.
2. Predicates in the WHERE clause applies to each tuple whereas the predicate in the HAVING clause applies to groups.
3. DELETE deletes one or more tuples of a given relation from the database. DROP is used to remove a table from the schema. Whereas DROP removes the definition of

the relation as well as the data in the given relation, delete only deletes the tuples but maintains the table definition.

4. In left outer join tuples from the left-hand-side relation that do not match any tuple in the right-hand-side relation are padded with nulls and are added to the result of the left outer join. Similarly, in right outer join tuples from the right-hand-side relation that do not match any tuple in the left-hand-side relation are padded with nulls and are added to the result of the left outer join.

Q.178 For the relations given in Q 175, answer the following queries in SQL

1. How many properties cost more than RS.5000/- per month
2. Give all Managers a 5% pay increase
3. Find the number of staff working in each branch and the sum of their salaries

(2 + 3 + 3)

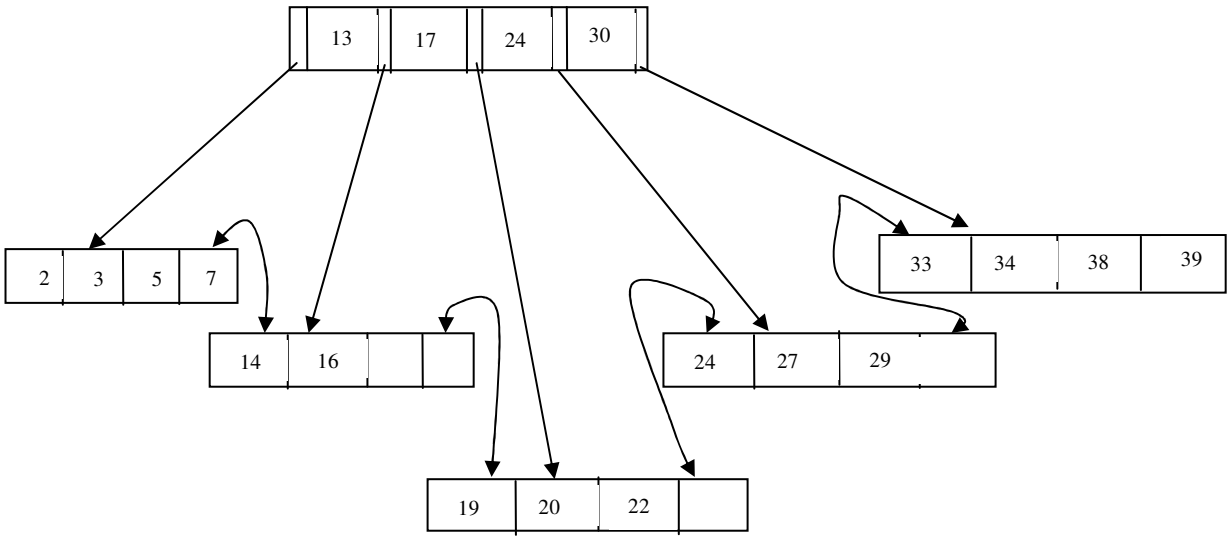
Ans:

1. SELECT COUNT(*)
   FROM PropertyForRent
   WHERE rent > 5000
2. UPDATE Staff
   SET salary = salary*1.05
   WHERE position = 'Manager'
3. SELECT branchno, COUNT(staffNo), SUM(salary)
   FROM Staff
   GROUP BY branchNo

Q.179 Define a B+ tree. **(2)**

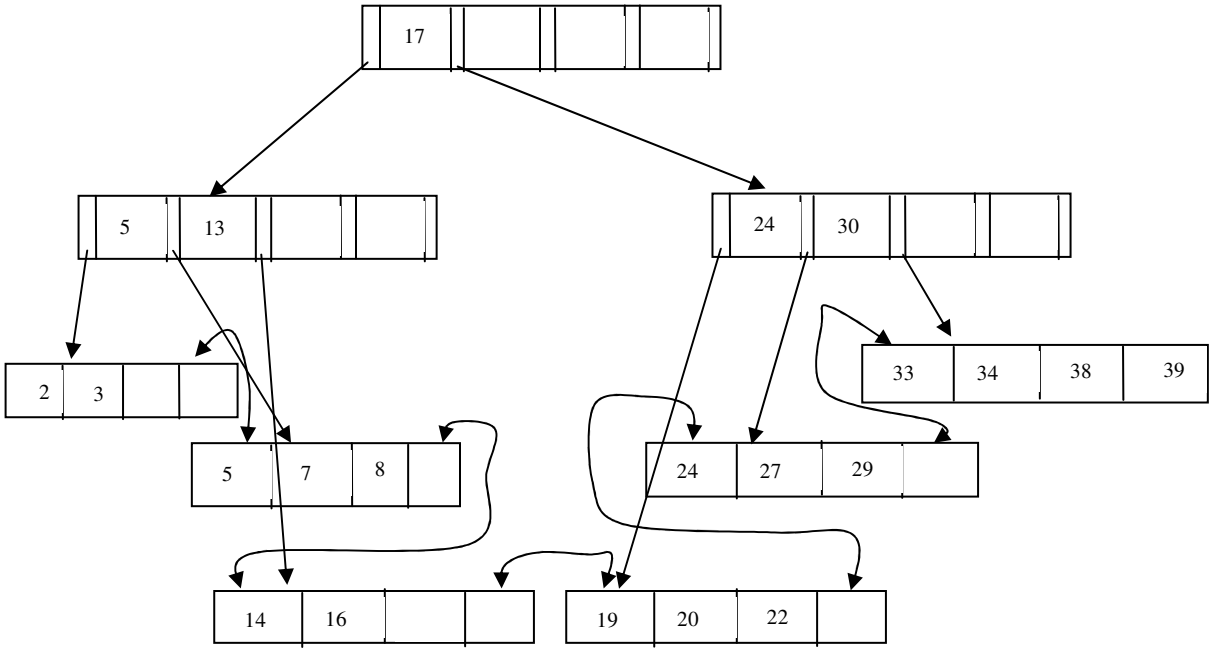
Ans: A B+ tree is a dynamic, height balanced index structure . Each node except the root has between d and 2d entries. The number d is called the order of the tree.

Q.180 Consider a B+ tree with order 2 with the following elements

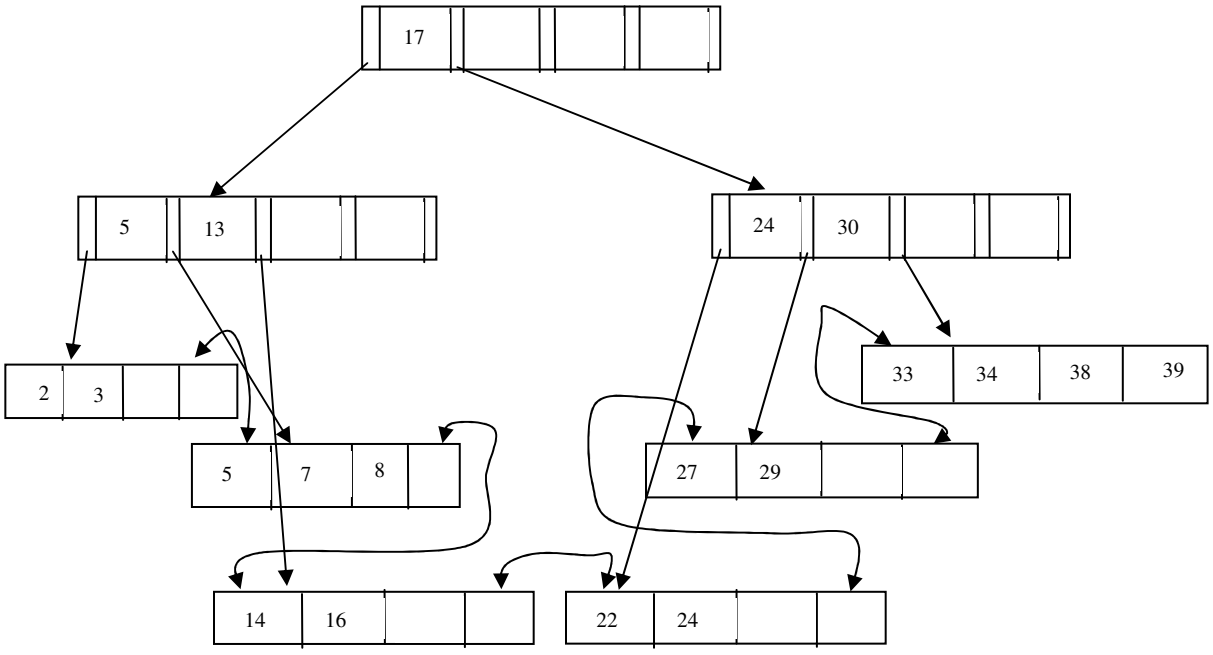


(i) Insert entry 8
(ii) Delete entry 19 and 20 5 * 2

Ans**. (i)**



(ii)



Q.181 Describe the definition of a transaction in SQL

Ans. The SQL transaction begins implicitly. Transactions are ended by either Commit work which commits the transaction and begins a new one Rollback work which causes the transaction to abort.
If a program terminates without either of these statements then updates are either committed or rolled back and the choice is implementation dependent.

Q.182 Define dependency preserving and lossless join decomposition **4**

Ans: A decomposition of R into $R_1$and R_2 is a lossless decomposition, if at least one of the following dependencies is in F^+ :
$R_1 \cap R_2 \rightarrow R_1$
$R_1 \cap R_2 \rightarrow R_2$
A decomposition of R into R_1, R_2 , … R_n is a dependency preserving decomposition, if either F' = F or $F'^+ = F^+$ where $F' = F_1 \cup F_2 \cup \ldots \cup F_n$ and F_i is the restriction of F to R_i.

Q.183 Given R = ABCD with the FD set F = { A → B, B → C, C → D}
Determine all 3NF violations. Decompose the relations into relations which are in 3NF. **8**

Ans: The only candidate key is A. The FD B → C violates 3NF as B is not a superkey and C is not a prime attribute. The FD C → D violates 3NF as C is not a superkey and D is not a prime attribute. The decomposition is R1 (AB) with key A and FD = A → B
R2 (BC) with key B and FD = B → C and R3 (CD) with key C and FD = C → D

Q.184 Determine a candidate key for R = ABCDEG with the FD set
F = { AB → C, AC → B, AD → E, B → D, BC → A, E → G } **4**

Ans: $BC^+ = R$. $B^+ = BD$ and $C^+ = C$. Therefore, BC is the candidate key.

Q.185 What are the objectives of query processing? **3**

Ans: The aim of query processing is to transform a query written in a high level language into a correct and efficient execution strategy expressed in a low level language and execute the strategy to retrieve the data.

Q.186 Explain the transformation rules that apply to
(a) Selection
(b) Projection
(c) Theta join operations **9**

Ans:
(i) Conjunctive selection operations can cascade into individual selection operations
$\sigma_{p \wedge q \wedge q \wedge r}(R) = \sigma_p(\sigma_q(\sigma_r(R)))$
(ii) Commutativity of selection operations
$\sigma_p(\sigma_q(R)) = \sigma_q(\sigma_p(R))$
(iii) In a sequence of projection operations only the last one is required
(iv) Commutativity of projection and selection
(v) Commutativity of theta join
(vi) Commutativity of selection and theta join
(vii) Commutativity of projection and theta join

(viii) Associativity of theta join

Q.187 When are two schedules said to be view equivalent? 4

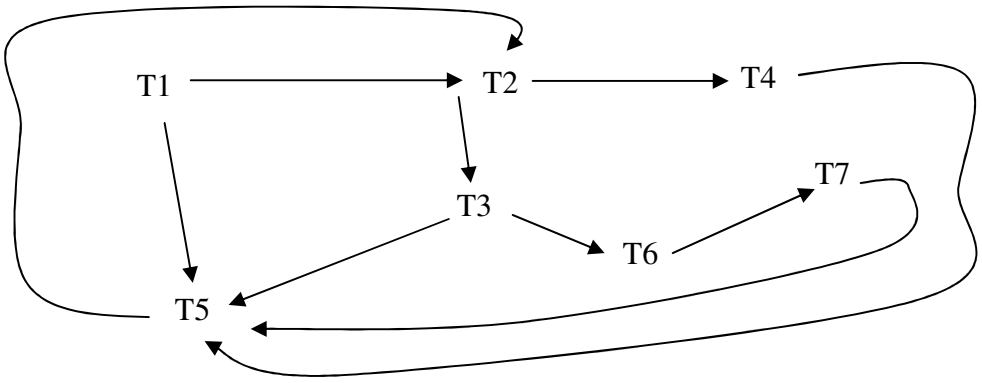
Ans:
Two schedules S and S' are said to be view equivalent if the following three conditions are met

a) For each data item Q, if a transaction Ti reads the initial value of Q in schedule S, then transaction Ti must in schedule S' also read the initial value of Q.
b) For each data item Q, if a transaction Ti executes read(Q) in schedule S and that value was produced by Tj (if any) , then transaction Ti must in schedule S' also read the value of Q produced by Tj..
c) For each data item Q, if a transaction if any that performs the final write(Q) in schedule S must perform the final write(Q) in schedule S'.

Q.188 Define deadlock. Produce a wait-for graph for the following transaction scenario and determine whether a deadlock exists

Transaction	Data items locked	Data items waiting for	
T1	x2	x1,x3	
T2	x3, x10	x7, x8	
T3	x8	x4, x5	
T4	x7	x1	
T5	x1, x5	x3	
T6	x4, x9	x6	
T7	x6	x5	**2 + 8**

Ans: An impasse that may result when two or more transactions are each waiting for locks to be released that are held by the other



Since there is cycle in the wait- for graph ($T_2 \rightarrow T_4 \rightarrow T_5 \rightarrow T_2$) in the graph, therefore, system is in a deadlock.

Q.189 Define granularity, hierarchy of granularity of locks and multiple granularity locking. Describe the modified two phase locking with multiple granularity locking. **6**

Ans: The size of the data item chosen as a unit of protection by a concurrency control protocol is granularity.
When the granularity is represented in a hierarchical structure where each node represents data items of different sizes we get the hierarchy of granularity.
In multiple granularity locking there are three types of locks – shared, exclusive and intention lock.
Modified two-phase locking strategy

a) No lock can be granted once any node has been unlocked
b) No node may be locked until its parent has an intention lock
c) No node may be unlocked until all its descendants are unlocked.

Q.190. What are the advantages of using a DBMS

8

Ans: The main advantages of using a DBMS are

Controlled redundancy

Every user group maintains its own files in traditional systems. For example, in the University system two different programmers will maintain different files for Courses, Teachers and Student information. Much of the information about the teachers teaching different courses to different students will be repeated in these files. This redundancy leads to several problems. Firstly, there is no single place to update data. Secondly, there is a waste of storage. Thirdly, the files may become inconsistent. The information about student registration may be different in different files.

Restricting Unauthorized access

When multiple users use the data base it is necessary to control access to sensitive data. Not all people may have access to the salary of employees in an organization. This is ensured by the DBMS.

Persistent Storage

The data that is stored survives beyond the program that created it

Deductive rules

Deductive DBMS permit the specification of inference rules to derive new information from the stored facts.

Multiple user interfaces

Some DMBS provide different user interfaces so that the user can choose the on that best suits his needs. Query languages, Data manipulation languages, graphical user interfaces are some of the interfaces that are provided.

Complex relationships

A DBMS is capable of representing a variety of complex relationships among data.

Integrity Constraints

It is possible to express constraints which are enforced. For example, if there is a constraint that no employee can be aged more than 62 years of age that this can be specified and enforced by a DBMS.

Backup and Recovery

In a DBMS environment, the system automatically recovers the data base to a consistent state when there are program and system failures.

Q.191 Define the following

1. Attribute Inheritance
2. Cascading rollback
3. Exec statement in SQL

4. Project-Join normal form **8**

Ans:

1. **Attribute Inheritance**
   The attributes of a higher level entity set are inherited by a lower level entity set created by specialization-generalization hierarchy. That is, all the attributes of the higher level entity set are also the attributes of the lower level entity set.
2. **Cascading rollback**
   The phenomenon in which a single transaction failure leads to a series of transaction rollbacks is called Cascading rollback.
3. **Exec statement in SQL**
   Exec statement has the form
   EXEC SQL <embedded SQL statement> END-EXEC
   Embeded SQl statements are of a form similar to non embedded SQL statements.
4. **Project-Join normal form**
   A relation R is in PJNF with resoect to a et of functional ependencies D if for all join dependencies of the form $*(R_1, R_2, \ldots, R_n)$ and $R = R_1 \cup R_2 \cup \ldots, \cup R_n$ at least one of the following holds:

$*(R_1, R_2, \ldots, R_n)$ is trivial
Every R_i is a superkey of R

Q.192 What are the disadvantages of file-processing system? **(4)**

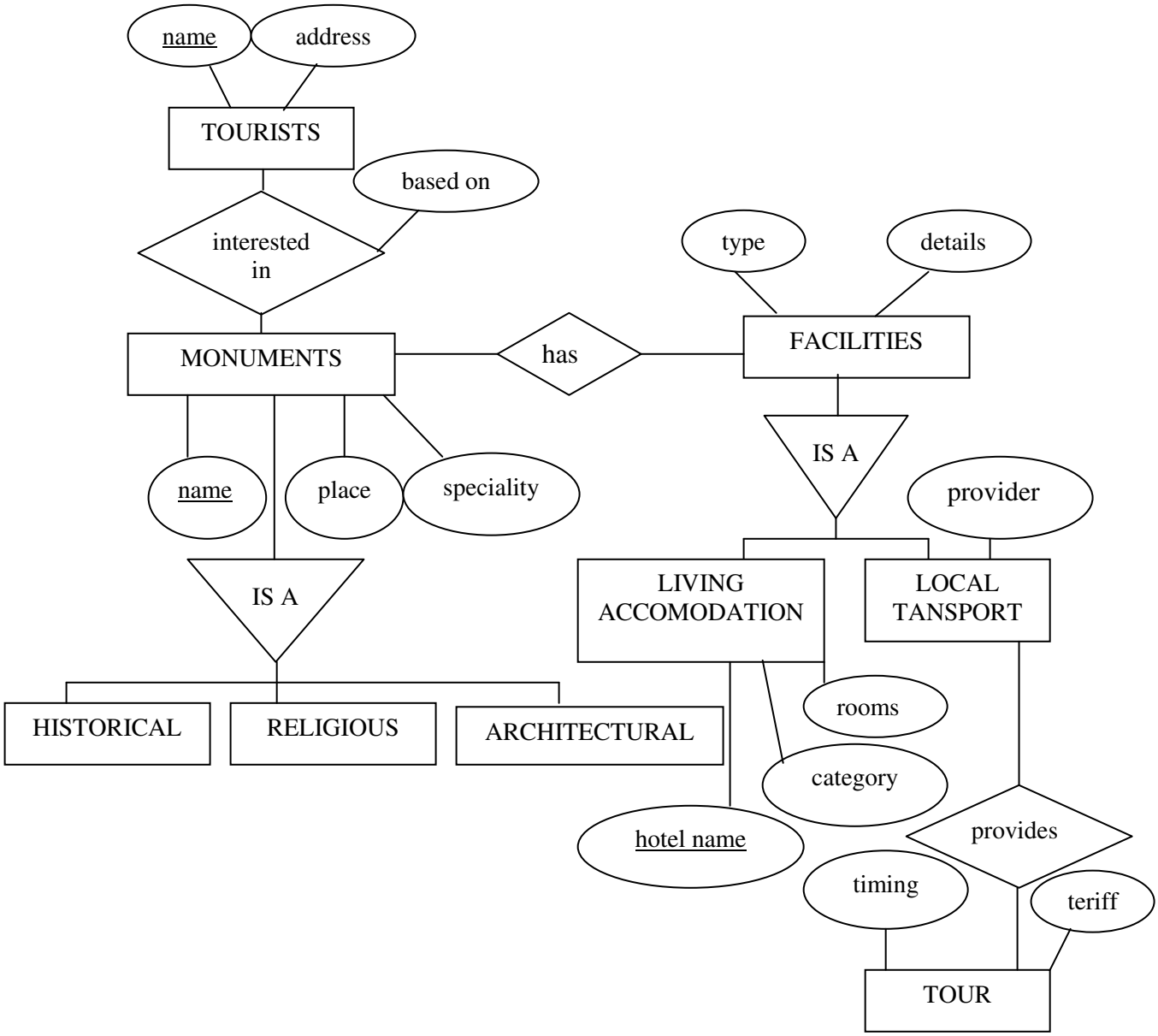
Ans:

The Disadvantages Of File-Processing System

1. **Data redundancy and inconsistency** – Data redundancy means unnecessary duplication of data. In file processing systems, the information may be duplicated in several places (files) due to the different structure. This redundancy leads to higher storage access cost. In addition, it may lead to data inconsistency, means the various copies of the same data may no longer agree.
2. **Difficulty in accessing data –** The conventional file-processing environments do not allow needed data to be retrieved in convenient and efficient manner
3. **Data isolation –** Because data are scattered in various files and may be in different formats, writing new program to retrieve the appropriate data is difficult.
4. **Security problem –** In file processing system. Enforcing security constraints is difficult.

Q.193 The tourism department wishes to computerize its data. The information consists of monuments of tourist interest, their location and history. Monuments are classified according to historical, religious and architecture importance. The list of facilities available at each sport is also available. These give (i) living accommodation in terms of hotels, their names, category and the number of rooms available and (ii) local transport facilities in terms of service provider name, tours with their tariff and timing. Arrive at an E-R diagram by identifying the entities, relationships, attributes, primary keys and cardinality. **(10)**

Ans:



Q.194 Describe the division and the join operation of the relational algebra. Give an example for each. Express each of them in terms of the basic operations. **(5)**

Ans: Join (⋈)- Allows the combining of two relations to form a single new relation. The combined operations of Cartesian product followed by selection operation is the join operation. There are different types of join operations: Theta, Equi, Natural. and outer Join. Theta join can be

$$R=P_{A\theta B} \bowtie Q$$

Join can be defined in terms of basic operations as: $\sigma_{A\theta B}$ (PxQ)
For example

Employee

E#	Name	D#
101	Jones	D1
103	Smith	D1
104	Lalonde	D2

Department

D#	Dname
D1	Sales
D2	Accounts

The natural join of employee and department

E#	Name	D#	Dname
101	Jones	D1	Sales
103	Smith	D1	Sales
104	Lalonde	D2	Accounts

Division (÷) Produces a relation R(X) that includes all the tuples t[X] in P(Z) that appear in relation P in combination with every tuple from Q(Y), where Z=X∪Y. The Cartesian product of relation Q and R is a subset of P (P⊇Q x R). Useful when a query involves the phrase " for all objects having all the specified properties."

For example : R= P÷Q

P

A	B
a_1	b_1
a_1	b_2
a_2	b_1
a_3	b_1
a_4	b_2
a_5	b_1
a_5	b_2

Q

B
b_1
b_2

R(result)

A
a_1
a_5

Q

B
b_1

Then R is

A
a_1
a_2
a_3
a_2

(a)

Q

B
b_1
b_2
b_3

Then R is

A

(b)

Q

B

Then R is

A
a_1
a_2
a_3
a_4
a_5

Division can be expressed as a sequence of projection, cross product, and difference operations as follows:

1. $R_1 \leftarrow \Pi_A(P)$
2. $R_2 \leftarrow \Pi_A(Q \times R_1) - P)$
3. $R \leftarrow R_1 - R_2$

Q.195 Consider the schema
Airport (code, name, city, country)
Flight (number, airline, from_airport_code, to_airport_code)
Reservation(flight_number, seat_number, date, passenger_name)
Answer the following using relational algebra

(i) List the flight numbers of flights that take off from India
(ii) List the passenger who are on flight number 'SA 747'.
(iii) List all the flight information for Indian Airlines and Jet Airways.

Ans: (i) Π_{number} ($\sigma_{country=India}$ (FLIGHT $\bowtie$ from _airport_code=codeAIRPORT))
(ii) $\Pi_{passenger\_name}$ ($\sigma_{flight\_number='Sa\ 747'}$(RESERVATION))
(iii) $\sigma_{airline('IndianAirlines'\ \wedge\ 'Jet\ Airways')}$ (FLIGHT)

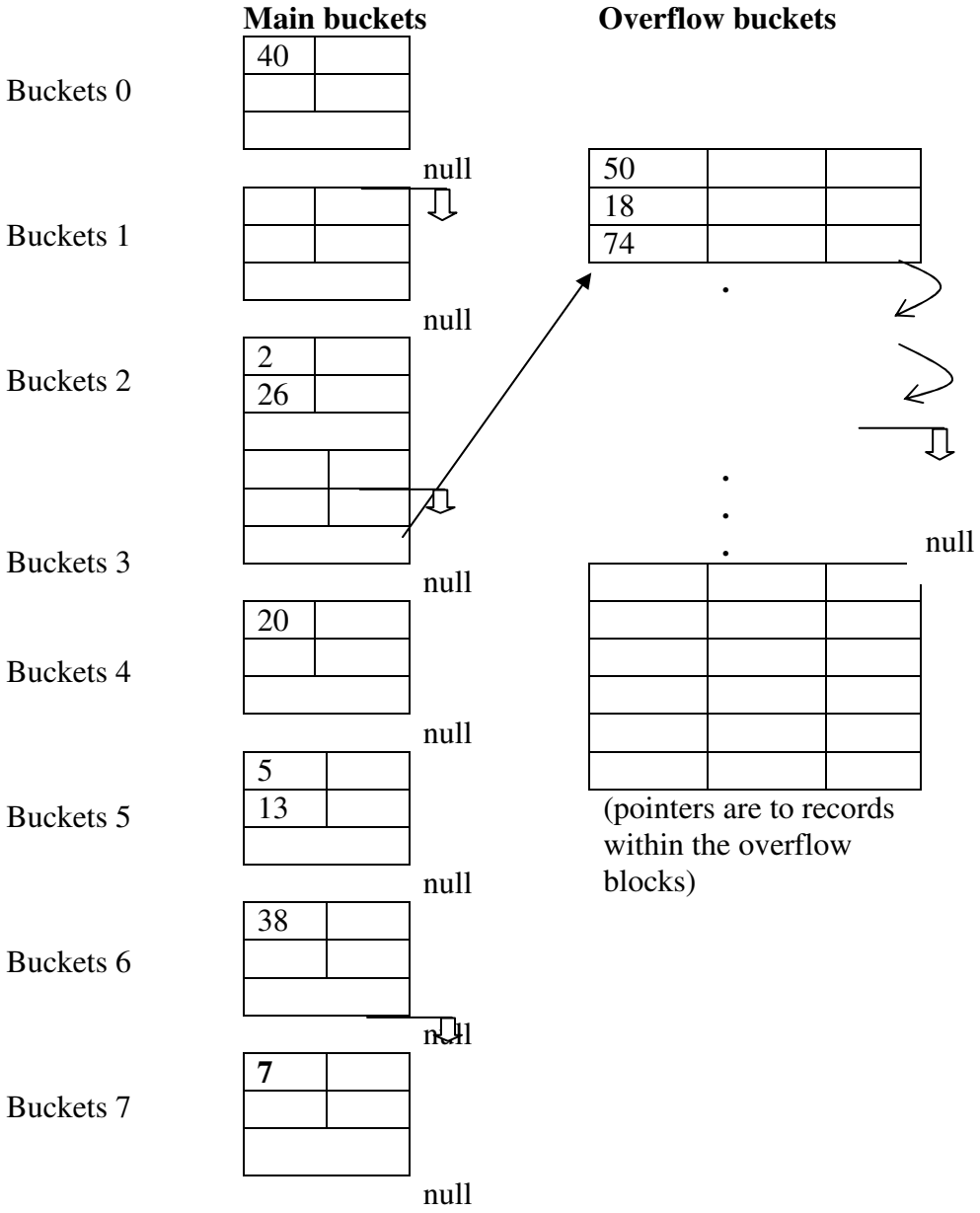
Q. 196 What is the difference between primary index and secondary index? **(4)**

Ans:

Primary Index	Secondary index
1. It is an ordered file whose records are of fixed length with two fields.	1. It provides a secondary means of accessing a file for which some primary access already exists.
2. Only based on the primary key.	2. May be based on candidate key or secondary key.
3. The total number of entries in the index is the same as the number of disk blocks in the ordered data file.	3. It has a large number entries due to duplication.
4. Primary index is a king of nondense (sparse) index.	4. Secondary index is a kind of dense index.
5. There may be at most one primary index for a file	5. There may be more than one secondary indexes for the same file.
6. Needs less storage space.	Needs more storage space and longer search time.

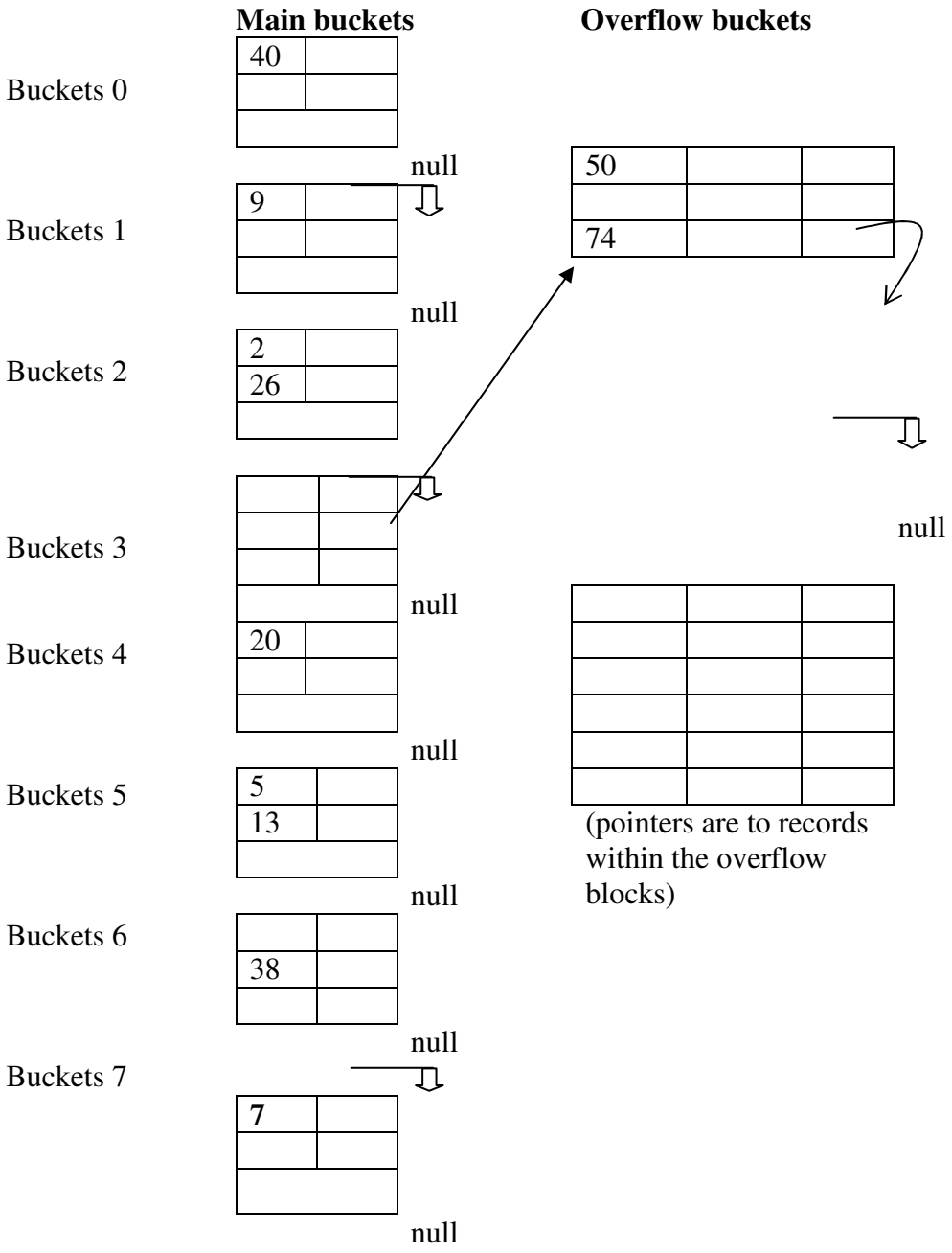
Q.197 Suppose there is a hash function h(x)=x mod 8. Show the hash structure for a file with the following key values 2,5,7,26,13,20,38,40,50,18,74. Assume a bucket size of 2 records. **(6)**

Ans:



Q.198 Explain how you would delete the record with key 18 and then add a record with key 9? What is the hash structure after the deletion and after the insertion?

Ans: To delete the record with key 18 the record pointer of the previous record i.e. 50 will be pointing to the next record of the 18, i.e. 74, and the space of the record with key 18 will be added to the pool of free space. The record with key 9 is added to the main bucket 1 because there is a space available in it. The hash structure after the deletion and after the insertion will be:



Q. 199 Consider the data base with two areas A1 and A2 with files F1, F2 and A2 with files F3. F1 has records r_{11} to r_{19}, and F2 has records r_{21} to r_{29}, F3 has records r_{31} to r_{39}.

(i) What is a granularity hierarchy? Draw the granularity hierarchy for the above database **(4)**

(ii) Suppose transaction T1 wants to read record r_{35} and T2 wants to read file F3 what are the locks to be placed? Can T1 and T2 execute concurrently? Justify. **(4)**

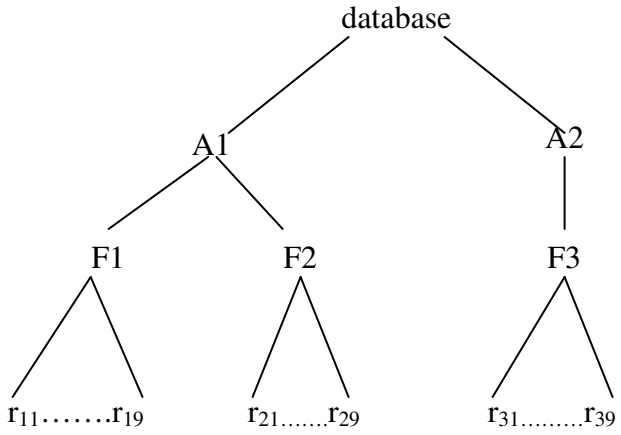
(iii) Is it possible for T1 to unlock file F3. Justify. **(3)**

(iv) If a transaction T3 now wants to write r_{11} will it be allowed to proceed? **(3)**

Ans: (i) All concurrency control techniques assumed that the database was formed of a number of named data items. A database item could be chosen to be one of the following:

- A field value of a database record
- A database record
- A disk block
- A whole file
- The whole database

It seems appropriate that a database system support multiple levels of granularity, where the granularity level can be different for various mixes of transactions, called as granularity hierarchy.



Granularity Hierarchy

(ii) In granularity hierarchy, a lock can be requested at any level. However, additional types of locks will be needed to efficiently support such protocol. When transaction T1 want to read the record r_{35}, the record r_{35} will be locked with shared lock and the higher level items are locked with intention shared mode locks. When transaction T2 want to read the file F3, the file F3 will be locked with shared lock and the higher level items are locked with intention shared mode locks. T1 and T2 can execute concurrently because both the transactions performing the only the read – read operations. Read operations will not effect the consistency of the database.

(iii) Yes, The transaction T1 can unlock the file F3 because if any transaction acquired a lock then it can release that lock at any time.

(iv) Yes, transaction T3 will be allowed to write record r_{11} because neither its lower level items nor higher level items are already locked. Hence, it will not cause any inconsistency.

Q.200 When shadow paging is used, show the steps that are executed when a transaction executed write (X) operation and X resides on the 1^{th} page.

Ans:

In the shadow page scheme, the database is considered to be made up of logical units of storage called pages, the shadow page scheme handles the write operation, e.g., write(x),to a given page(1^{th} page) as follows;

- A free block of non-volatile storage is located from the pool of free blocks accessible by the database system.
- The page to be modified (1^{th} page) is copied onto this block.

- The original entry in the current page table is changed to point to this new block.
- The updates, i.e. write(X), are propagated to the block pointed to by the current page table, which in this case would be the newly created block.

Q.201 Define entity integrity and referential integrity. How does SQL allow specification of these?

Ans:
Entity Integrity Rule- If the attribute A of relation R is a prime attribute of R then A cannot accept null values.
Referential Integrity Rule- In referential integrity. It is ensured that a value that appears in one relation for a given set of attributes also appears for a certain set of attributes in another relation.
In SQL, entity integrity and referential integrity rules are implemented as constraints on the relation called as primary Key constraint and reference Key constraint respectively.
These constraints can be specified with relation at the time of creation of the relations or after the creation of the relations by altering the definition of the relations. For example:

```
Create Table Dept
      (Deptno      Number primary key.
       DName       Varchar2(15));
Create Table Emp
      (Empno       Number primary Key.
       EName       Varchar2(15),
       Job         Varchar2(10),
       Deptno      Number  References Dept(Deptno));
```

Q.202 Let R (A,B,C) and S(B,C,D) be the relations

R

A	**B**	**C**
a1	b1	c2
a2	b4	c1
a3	b3	c3
a2	b3	c5
a1	b1	c2

S

B	C	D
b1	c1	c2
b2	c4	d1
b3	c3	d4

Computer the following SQL statements for the above relations
(i) Select*From R,S
(ii) Select A from R,S Where R,B=S,B
(iii) Select A,C From R Group by B
(iv) Select R,A From R where R,B=NOT ALL
(Select S,B From S Where R.C= S.C) **(10)**

Ans: (i) It will returns the Cartesian product of relations R and S.

A	R.B	R.C	S.B	S.C	D
a1	b1	c2	b1	c1	c2
a1	b1	c2	b2	c4	d1
a1	b1	c2	b3	c3	d4
a2	b4	c1	b1	c1	c2
a2	b4	c1	b2	c4	d1
a2	b4	c1	b3	c3	d4
a3	b3	c3	b1	c1	c2
a3	b3	c3	b2	c4	d1
a3	b3	c3	b3	c3	d4
a2	b3	c5	b1	c1	c2
a2	b3	c5	b2	c4	d1
a2	b3	c5	b3	c3	d4
a1	b1	c2	b1	c1	c2
a1	b1	c2	b2	c4	d1
a1	b1	c2	b3	c3	d4

(ii)

A
a1
a1
a3
a2

(iii) This is an invalid SQL command because in SELECT clause we can use only group expressions (e.g.. column name(s) on which the grouping is done and/ or aggregate functions) when we are using GROUP BY clause. In the given command A and C are not group expressions.

(iv) **<>ALL**

A
a1
a2
a2
a1

Q.203 Consider the relation R(A, B, C, D, E) with the set of function dependencies F={A, B → C, D → E, A → D}

(i) Is AB a candidate Key? Justify. **(2)**

(ii) Giving reasons find out whether R is in 3NF or BCNF. **(4)**

(iii) Consider the decomposition of R into R1 (A,B,C) and R2 (A,D,E). Is the decomposition lossless and dependency preserving? Justify. **(6)**

(iv) Define fifth normal form. **(2)**

Ans:

(i) Yes, AB qualifies as a candidate of the given relation R. Because, AB can derive all other attributes CDE as illustrated below:

- A can derives the value of D based on the FD $A \rightarrow D$.
- D can derives the value of E based on the FD $D \rightarrow E$ or transitively A derives the value of E.
- AB can derive the value of C based on the FD $AB \rightarrow C$.

(ii) Above relation neither in 3NF nor in BCNF. Even it is not in 2NF because the Key of the relation is AB and there is partial functional dependency $A \rightarrow D$. which is not permissible in 2NF relation. Therefore, it is justified the while the given relation not in 2NF then it is neither in 3NF nor in BCNF.

(iii) Yes, the given decompositions R1 and R2 of the relation R is dependency preserving because $(F1 \cup F2)^+ \equiv F^+$. The given decompositions are also lossless because the Key AB of the relation R is preserved in one of the decomposition R1.

(iv) A relation schema R is in fifth normal form (5NF) (or project join normal form [PJNF]) with respect to a set of F of functional. Multivalued, and join dependencies if, for every nontrivial join dependency JD $(R_1, R_{2,\ldots} R_n)$ in F^+ (that is, implied by F), every R_1 is a superkey of R,

Q.204 Discuss with respect to SQL

(i) How nulls are treated in comparison operator.
(ii) The count function.
(iii) The check clause.
(iv) The difference between drop table R and delete from R. **(8)**

Ans:

(i) With comparison (relational) operators, the null values are ignored because we cannot derive the relation with the given operand. Nulls cannot be equate. Nulls can be checked by containment using IS NULL or IS NOT NULL operators.

(ii) The COUNT function returns the number of tuples or values specified in a query. The count function has two types of syntax: (1) COUNT (*) – returns the number of rows in the result query, (2) COUNT ([DISTINCT] <column>) – in this way, the NULL values are discarded and if used then the duplicates are also discarded in the count.

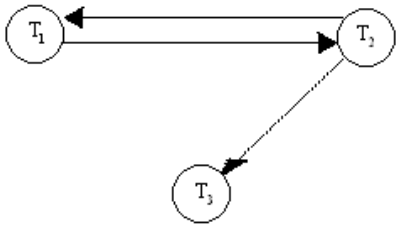
(iii) The CHECK clause is used to at the end of a CREATE TABLE statement specify table constraints. This is called table-based constraint because it applies to each tuple individually and are checked whenever a tuple is inserted or modified. The CHECK clause can also be used to specified more general constraints using the CREATE ASSERTION statement of SQL.

(iv) DROP TABLE command deletes all the records along with the table definition. This command will automatically committed and cannot be rolled back. But . DELETE FROM <table> command deletes only the records; the table definition remains existed. The effect of this command may be rolled back, if required.

Q.205 Consider the schedule of three transactions T1,T2 and T3.

$S: r_2(A); r22_1(B); w_2(A); r_2(B); r_3(A); w_1(B); w_3(A); w_2(B)$

Where r stands for READ, w for WRITE and determine if the schedule is seializable. If so, give the schedule.



The given schedule is not serializable because the precedence graph is cyclic.

Q.206 Under what condition does the selection operation distribute over theta join operation? **(2)**

Ans:

The conditions are:

- If all the attributes in the selection condition c involve only in the attributes of one of the relations being joined-say, R – the two operations can be commuted a follows:

$$\sigma_C(R \bowtie S) \equiv (\sigma_C(R)) \bowtie R$$

- If the selection condition c can be written as c1 and c2, where condition c1 involves only the attributes of R and/or condition c2 involves only the attributes of S, the operations can commute as follows:

$$\sigma_C(R \bowtie S) \equiv (\sigma_{C1}(R)) \bowtie (\sigma_{C2}(S))$$

OR

$$\sigma_C(R \bowtie S) \equiv \sigma_{C2}((\sigma_{C1}(R)) \bowtie S)$$

OR

$$\sigma_C(R \bowtie S) \equiv \sigma_{C1}((R \bowtie (\sigma_{C2}(S)))$$

Q.207 For the relations

Emp(name, fname, address, dno)
Dept(deptno, dname, address)

And the query

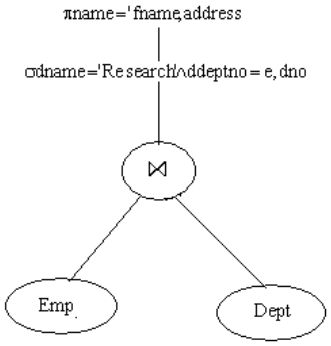
Π fname, ename, address (σ dname= 'Research' AND d,deptno=e. dno(EMP$\bowtie$Dept))

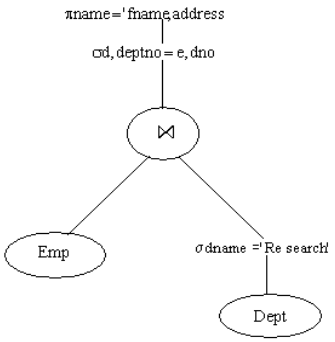
(i) Draw the initial query tree. **(2)**

(ii) Optimize the query write(X) operation and X resides on the 1th page. **(4)**

Ans:

$\pi_{\text{name, fname, address}}$ ($\sigma_{\text{dname='Research'^d.deptno=e.dno}}$(Emp$\bowtie$Dept))





Distribution of selection peration

The initial query tree **The optimized query tree**

Q.208 Compare shadow paging with log based recovery methods. **(4)**

Ans:

Shadow paging – In the shadow page scheme, the database is considered to be made up of logical units of storage called pages. The shadow paging scheme uses two page tables for transaction that is going to modify the database. The original page table is called the shadow page table and the transaction addressed the database using another page table known as the current page table. In case of a system crash, before the transaction commits, the shadow page table and the corresponding blocks containing the old database, which was assumed to be in a consistent state, will continue to be accessible. The recovery from system crash is relatively inexpensive, is an advantage of this scheme. The main disadvantages of the shadow scheme are: (1) the problem of scattering, and (2) the original version of the changed blocks pointed to by the shadow page table have to be returned to the pool of free blocks.

Log Based Recovery Method – The system log, which is usually written on stable storage, contains the redundant data required to recover from volatile storage failures and also from errors discovered by the transaction or the database system. System log is also called as log and has sometimes been called he DBMS journal. In case of system crash the log is processed from the beginning to the point where system crashes. All the transaction, those have been recorded their-of-transaction marker, will be committed otherwise rolled back. The simplicity and easier to use the main advantages of this method. The disadvantage are: (1) inefficient for the application, which have large volume of information, (2) in case of system crash with loss of volatile information, the log information collected in buffers will also be lost and transaction that had been completed for some period prior to the system crash may be missing their respective end-of-transaction markers in the log; if such transactions are rolled back then likely to be partially undone.

Q.209 What is a checkpoint? List the operations to be performed by the system when a checkpoint is to be taken. What does the recovery system do if there is a crash. **(6)**

Ans:

In a large on-line database system there could be hundreds of transactions handled per minute. The log for this type of database contains a very large volume of information. A scheme called checkpoint is used to limit the volume of log information that has to be handled and processed in the event of a system failure involving the loss of volatile information. The checkpoint scheme is an additional component of the logging scheme.

A checkpoint operation, performed periodically, copies log information onto stable storage and put the checkpoint marker at that time in the log. For all transaction active at checkpoint, their identifiers and their database modification actions. Which at that time are reflected only in the database buffers, will be propagated to the appropriate storage.

If there is a crash, it scans the log in a backward direction from the time of system crash. Initially, all the transaction are added to the undo list. After that, if it finds that a transaction in the undo list has committed, that transaction is removed from the undo list and placed in the redo list. The redo list contains all the transactions that have to be redone. All transactions that were committed before the checkpoint time need not be considered for the recovery operation. Therefore, the overhead of the recovery from a system crash is reduced. However, in a system that uses the transaction paradigm. Checkpoint is a strategy to minimize the search of the log and the amount of undo and redo required to recover from a system failure with loss of volatile storage.

Q.210 Write short notes on

(i) Clustering file organization.
(ii) Deadlock detection and recovery.
(iii) Outer join
(iv) Mapping of ISA relationship of E-R diagram to tables. **(3.5x4=14)**

Ans:

(i) If a record of file are physically ordered on a nonkey field – which does not have a distinct value for each record – that field is called the clustering field. The type of index that is created using clustering field is called as clustering index. Clustering index is used to speed up retrieval of records that have the same value for the clustering field. The file that is organized based on the clustering field and clustering index is called clustering file organization.

(ii) Deadlock Detection – A deadlock is said to occur when there is a circular chain of transaction, each waiting for the release of a data – item held by the next transaction in the chain, the algorithm to detect a deadlock is based on the detection of such a circular chain in the current system in wait – for – graph. The algorithm generates the wait – for – graph at regular intervals and examines it for a chain. The algorithm starts with the assumption that there is no deadlock.

Deadlock Recovery – To recover from deadlock. The cycles in the wait – for – graph must be broken. The common method of doing this is to roll back one or more transactions in the cycles until the system exhibits no further deadlock situation. The process of deadlock recovery must ensure that a given transaction is not continuously the one selected for rollback. The selection of the transaction to be rolled back is based on the following considerations:

- The progress of the transaction and the number of data – items it has used and modified.
- The amount of computing remaining for the transaction and the number of data – items that have yet to be accessed by the transaction.

- The relative cost of rolling back a transaction.

(iii) Outer join – If there are any values in the one table that do not have corresponding value(s) in the other, in an equi-join that will not be selected. Such rows can be forcefully selected by using the outer join, The corresponding columns for that row will have NULLs. There are actually three forms of the outer-join operation: left outer join (_X), right outer join (X_) and full outer join (_X_).

(iv) There are several option for mapping of IS_A relationship to relation. For the mapping. Convert each specialization with m subclasses and superclass C into relation schemas using one of the four following options:

(a) Multiple relations – Superclass subclasses: In this option, one relation schema is created for the subperclass and for the each subclasses, the separate relation schemas are with the respective attributes and the key of the superclass. This option works for any specialization (total or partial, disjoint or overlapping).

(b) Multiple relation – Subclass relations only: In this option. Only the relation schemas are created for each subclasses separately by including all the attributes of the superclass with each subclass. (This option only works for a specialization whose subclasses are total.

(c) Single relation with one type attribute: In this option, only one relation is created by combining all the attributes of the superclass and all the subclasses with an extra attributes may named as 'type' to distinguish each entity. This option works only for a specialization whose subclasses are disjoint, and has the potential for generating many null values if many specific attributes exist in the subclasses.

(d) Single relation with multiple type attributes: In this option, only one relation is created by combining all the attributes of the superclass and all the subclasses with some extra attributes to distinguish each entity. This option works for a specialization whose subclasses are overlapping (also works for disjoint specialization).

TYPICAL QUESTIONS & ANSWERS

OBJECTIVE TYPE QUESTIONS

Each question carries 2 marks.

Choose the correct or best alternative in the following:

Q.1 In the relational modes, cardinality is termed as:

(A) Number of tuples. **(B)** Number of attributes.
(C) Number of tables. **(D)** Number of constraints.

Ans: A

Q.2 Relational calculus is a

(A) Procedural language. **(B)** Non- Procedural language.
(C) Data definition language. **(D)** High level language.

Ans: B

Q.3 The view of total database content is

(A) Conceptual view. **(B)** Internal view.
(C) External view. **(D)** Physical View.

Ans: A

Q.4 Cartesian product in relational algebra is

(A) a Unary operator. **(B)** a Binary operator.
(C) a Ternary operator. **(D)** not defined.

Ans: B Cartesian product in relational algebra is a binary operator.
(It requires two operands. e.g., P X Q)

Q.5 DML is provided for

(A) Description of logical structure of database.
(B) Addition of new structures in the database system.
(C) Manipulation & processing of database.
(D) Definition of physical structure of database system.

Ans: C DML is provided for manipulation & processing of database.
(Data stored in the database is processed or manipulated using data manipulation language commands as its name)

Q.6 'AS' clause is used in SQL for

(A) Selection operation. **(B)** Rename operation.
(C) Join operation. **(D)** Projection operation.

Ans: B 'AS' clause is used in SQL for rename operation.
(e.g., SELECT ENO AS EMPLOYEE_NO FROM EMP)

Q.7 ODBC stands for

(A) Object Database Connectivity.
(B) Oral Database Connectivity.
(C) Oracle Database Connectivity.
(D) Open Database Connectivity.

Ans: D

Q.8 Architecture of the database can be viewed as

(A) two levels. **(B)** four levels.
(C) three levels. **(D)** one level.

Ans: C

Q.9 In a relational model, relations are termed as

(A) Tuples. **(B)** Attributes
(C) Tables. **(D)** Rows.

Ans:

Q.10 The database schema is written in

(A) HLL **(B)** DML
(C) DDL **(D)** DCL

Ans: C

Q.11 In the architecture of a database system external level is the

(A) physical level. **(B)** logical level.
(C) conceptual level **(D)** view level.

Ans: D

Q.12 An entity set that does not have sufficient attributes to form a primary key is a

(A) strong entity set. **(B)** weak entity set.
(C) simple entity set. **(D)** primary entity set.

Ans: B

Q.13 In a Hierarchical model records are organized as

(A) Graph. **(B)** List.
(C) Links. **(D)** Tree.

Ans: D

Q.14 In an E-R diagram attributes are represented by
(A) rectangle. **(B)** square.
(C) ellipse. **(D)** triangle.

Ans: C

Q.15 In case of entity integrity, the primary key may be
(A) not Null **(B)** Null
(C) both Null & not Null. **(D)** any value.

Ans: A

Q.16 In tuple relational calculus $P_1 \rightarrow P_2$ is equivalent to
(A) $\neg P_1 \vee P_2$ **(B)** $P_1 \vee P_2$
(C) $P_1 \wedge P_2$ **(D)** $P_1 \wedge \neg P_2$

Ans: A In tuple relational calculus $P_1 \rightarrow P_2$ is equivalent to $\neg P_1 \vee P_2$.
(The logical implication expression A → B, meaning if A then B,is equivalent to $\neg A \vee B$)

Q.17 The language used in application programs to request data from the DBMS is referred to as the
(A) DML **(B)** DDL
(C) VDL **(D)** SDL

Ans: A

Q.18 A logical schema
(A) is the entire database.
(B) is a standard way of organizing information into accessible parts.
(C) describes how data is actually stored on disk.
(D) both (A) and (C)

Ans: A

Q.19 Related fields in a database are grouped to form a
(A) data file. **(B)** data record.
(C) menu. **(D)** bank.

Ans: B Related data fields in a database are grouped to form a data record.
(A record is a collection of related fields)

Q.20 The database environment has all of the following components except:
(A) users. **(B)** separate files.
(C) database. **(D)** database administrator.

Ans: A

Q.21 The language which has recently become the defacto standard for interfacing application programs with relational database system is

(A) Oracle. **(B)** SQL.
(C) DBase. **(D)** 4GL.

Ans: B

Q.22 The way a particular application views the data from the database that the application uses is a

(A) module. **(B)** relational model.
(C) schema. **(D)** sub schema.

Ans: D

Q.23 In an E-R diagram an entity set is represent by a

(A) rectangle. **(B)** ellipse.
(C) diamond box. **(D)** circle.

Ans: A

Q.24 A report generator is used to

(A) update files. **(B)** print files on paper.
(C) data entry. **(D)** delete files.

Ans: B

Q.25 The property / properties of a database is / are :

(A) It is an integrated collection of logically related records.
(B) It consolidates separate files into a common pool of data records.
(C) Data stored in a database is independent of the application programs using it.
(D) All of the above.

Ans: D

Q.26 The DBMS language component which can be embedded in a program is

(A) The data definition language (DDL).
(B) The data manipulation language (DML).
(C) The database administrator (DBA).
(D) A query language.

Ans: B

Q.27 A relational database developer refers to a record as

(A) a criteria. **(B)** a relation.
(C) a tuple. **(D)** an attribute.

Ans: C

Q.28 The relational model feature is that there

(A) is no need for primary key data.
(B) is much more data independence than some other database models.
(C) are explicit relationships among records.
(D) are tables with many dimensions.

Ans: B

Q.29 Conceptual design
(A) is a documentation technique.
(B) needs data volume and processing frequencies to determine the size of the database.
(C) involves modelling independent of the DBMS.
(D) is designing the relational model.

Ans:C

Q.30 The method in which records are physically stored in a specified order according to a key field in each record is
(A) hash. **(B)** direct.
(C) sequential. **(D)** all of the above.

Ans: A A method in which records are physically stored in a specified order according to a key field in each record is hash.
(In hash method, a hash function is performed on the key value to determine the unique physical address of the record to store or retrieve)

Q.31 A subschema expresses
(A) the logical view. **(B)** the physical view.
(C) the external view. **(D)** all of the above.

Ans: C A subschema expresses the external view.
(External schemas are called also called as subschemas)

Q.32 Count function in SQL returns the number of
(A) values. **(B)** distinct values.
(C) groups. **(D)** columns.

Ans: A Count function in SQL returns the number of values.
(Count function counts all the not null values in the specific column. If we want to count only distinct values than the DISTINCT keyword is also to be used)

Q.33 Which one of the following statements is false?
(A) The data dictionary is normally maintained by the database administrator.
(B) Data elements in the database can be modified by changing the data dictionary.
(C) The data dictionary contains the name and description of each data element.
(D) The data dictionary is a tool used exclusively by the database administrator.

Ans: B

Q.34 An advantage of the database management approach is
(A) data is dependent on programs.
(B) data redundancy increases.
(C) data is integrated and can be accessed by multiple programs.
(D) none of the above.

Ans: C

Q.35 A DBMS query language is designed to
(A) support end users who use English-like commands.
(B) support in the development of complex applications software.
(C) specify the structure of a database.
(D) all of the above.

Ans: D

Q.36 Transaction processing is associated with everything below except
(A) producing detail, summary, or exception reports.
(B) recording a business activity.
(C) confirming an action or triggering a response.
(D) maintaining data.

Ans: C

Q.37 It is possible to define a schema completely using
(A) VDL and DDL. **(B)** DDL and DML.
(C) SDL and DDL. **(D)** VDL and DML.

Ans: B

Q.38 The method of access which uses key transformation is known as
(A) direct. **(B)** hash.
(C) random. **(D)** sequential.

Ans: B

Q.39 Data independence means
(A) data is defined separately and not included in programs.
(B) programs are not dependent on the physical attributes of data.
(C) programs are not dependent on the logical attributes of data.
(D) both (B) and (C).

Ans: D both (B) and (C)

Q.40 The statement in SQL which allows to change the definition of a table is
(A) Alter. **(B)** Update.
(C) Create. **(D)** select.

Ans: **A**

Q.41 E-R model uses this symbol to represent weak entity set ?
(A) Dotted rectangle.
(B) Diamond
(C) Doubly outlined rectangle
(D) None of these

Ans: C

Q.42 SET concept is used in :
(A) Network Model
(B) Hierarchical Model
(C) Relational Model
(D) None of these

Ans: A

Q.43 Relational Algebra is
(A) Data Definition Language **.**
(B) Meta Language
(C) Procedural query Language
(D) None of the above

Ans: C

Q.44 Key to represent relationship between tables is called
(A) Primary key **(B)** Secondary Key
(C) Foreign Key **(D)** None of these

Ans: C

Q.45 _____ produces the relation that has attributes of R1 and R2
(A) Cartesian product **(B)** Difference
(C) Intersection **(D)** Product

Ans: A

Q.46 The file organization that provides very fast access to any arbitrary record of a file is
(A) Ordered file **(B)** Unordered file
(C) Hashed file **(D)** B-tree

Ans: C

Q.47 DBMS helps achieve
(A) Data independence **(B)** Centralized control of data
(C) Neither (A) nor (B) **(D)** both (A) and (B)

Ans: D

Q.48 Which of the following are the properties of entities?
(A) Groups **(B)** Table

(C) Attributes **(D)** Switchboards

Ans: C

Q.49 In a relation
(A) Ordering of rows is immaterial
(B) No two rows are identical
(C) (A) and (B) both are true
(D) None of these**.**

Ans: C

Q.50 Which of the following is correct:
(A) a SQL query automatically eliminates duplicates.
(B) SQL permits attribute names to be repeated in the same relation.
(C) a SQL query will not work if there are no indexes on the relations
(D) None of these

Ans: D

Q.51 It is better to use files than a DBMS when there are
(A) Stringent real-time requirements.
(B) Multiple users wish to access the data.
(C) Complex relationships among data.
(D) All of the above.

Ans: B

Q.52 The conceptual model is
(A) dependent on hardware.
(B) dependent on software.
(C) dependent on both hardware and software .
(D) independent of both hardware and software.

Ans: D

Q.53 What is a relationship called when it is maintained between two entities?
(A) Unary **(B)** Binary
(C) Ternary **(D)** Quaternary

Ans: B

Q.54 Which of the following operation is used if we are interested in only certain columns of a table?
(A) PROJECTION **(B)** SELECTION
(C) UNION **(D)** JOIN

Ans: A

Q.55 Which of the following is a valid SQL type?
(A) CHARACTER **(B)** NUMERIC
(C) FLOAT **(D)** All of the above

Ans: D

Q.56 The RDBMS terminology for a row is
(A) tuple. **(B)** relation.
(C) attribute. **(D)** degree.

Ans: A

Q.57 Which of the following operations need the participating relations to be union compatible?
(A) UNION **(B)** INTERSECTION
(C) DIFFERENCE **(D)** All of the above

Ans: D

Q.58 The full form of DDL is
(A Dynamic Data Language **(B)** Detailed Data Language
(C) Data Definition Language **(D)** Data Derivation Language

Ans: C

Q.59 Which of the following is an advantage of view?
(A) Data security **(B)** Derived columns
(C) Hiding of complex queries **(D)** All of the above

Ans: D

Q.60 Which of the following is a legal expression in SQL?
(A) SELECT NULL FROM EMPLOYEE;
(B) SELECT NAME FROM EMPLOYEE;
(C) SELECT NAME FROM EMPLOYEE WHERE SALARY = NULL;
(D) None of the above

Ans: B

Q.61 The users who use easy-to-use menu are called
(A) Sophisticated end users. **(B)** Naïve users.
(C) Stand-alone users. **(D)** Casual end users.

Ans: B

Q.62 Which database level is closest to the users?
(A) External **(B)** Internal
(C) Physical **(D)** Conceptual

Ans: A

Q.63 Which are the two ways in which entities can participate in a relationship?
(A) Passive and active **(B)** Total and partial
(C) Simple and Complex **(D)** All of the above

Ans: B

Q.64 The result of the UNION operation between R1 and R2 is a relation that includes
(A) all the tuples of R1
(B) all the tuples of R2
(C) all the tuples of R1 and R2
(D) all the tuples of R1 and R2 which have common columns

Ans: D

Q.65 Which of the following is a comparison operator in SQL?
(A) = **(B)** LIKE
(C) BETWEEN **(D)** All of the above

Ans: D

Q.66 A set of possible data values is called
(A) attribute. **(B)** degree.
(C) tuple. **(D)** domain.

Ans: D

Q.67 Which of the operations constitute a basic set of operations for manipulating relational data?
(A) Predicate calculus **(B)** Relational calculus
(C) Relational algebra **(D)** None of the above

Ans:C

Q.68 Which of the following is another name for weak entity?
(A) Child **(B)** Owner
(C) Dominant **(D)** All of the above

Ans: A

Q.69 Which of the following database object does not physically exist?
(A) base table **(B)** index
(C) view **(D)** none of the above

Ans: C

Q.70 NULL is
(A) the same as 0 for integer
(B) the same as blank for character
(C) the same as 0 for integer and blank for character

(D) not a value

Ans: D

Q.71 Which of the following is record based logical model?

(A) Network Model **(B)** Object oriented model
(C) E-R Model **(D)** None of these

Ans: A

Q.72 A data dictionary is a special file that contains:

(A) The name of all fields in all files.
(B) The width of all fields in all files.
(C) The data type of all fields in all files.
(D) All of the above.

Ans: D

Q.73 A file manipulation command that extracts some of the records from a file is called

(A) SELECT **(B)** PROJECT
(C) JOIN **(D)** PRODUCT

Ans: A

Q.74 The physical location of a record is determined by a mathematical formula that transforms a file key into a record location is :

(A) B-Tree File **(B)** Hashed File
(C) Indexed File **(D)** Sequential file.

Ans: B

Q.75 Using Relational Algebra the query that finds customers, who have a balance of over 1000 is

(A) $\Pi_{Customer\_name}(\sigma_{balance>1000}(Deposit))$
(B) $\sigma_{Customer\_name}(\Pi_{balance>1000}(Deposit))$
(C) $\Pi_{Customer\_name}(\sigma_{balance>1000}(Borrow))$
(D) $\sigma_{Customer\_name}(\Pi_{balance>1000}(Borrow))$

Ans: **A**

Q.76 A primary key is combined with a foreign key creates

(A) Parent-Child relation ship between the tables that connect them.
(B) Many to many relationship between the tables that connect them.
(C) Network model between the tables that connect them.
(D) None of the above.

Ans: A

Q.77 In E-R Diagram derived attribute are represented by

(A) Ellipse
(B) Dashed ellipse
(C) Rectangle
(D) Triangle

Ans B

Q.78 Cross Product is a:
(A) Unary Operator
(B) Ternary Operator
(C) Binary Operator
(D) Not an operator

Ans: C

Q.79 An instance of relational schema R (A, B, C) has distinct values of A including NULL values. Which one of the following is true?
(A) A is a candidate key
(B) A is not a candidate key
(C) A is a primary Key
(D) Both (A) and (C)

Ans: B

Q.80 Consider the join of a relation R with relation S. If R has m tuples and S has n tuples, then the maximum size of join is:
(A) mn
(B) m+n
(C) (m+n)/2
(D) 2(m+n)

Ans: A

Q.81 The natural join is equal to :
(A) Cartesian Product
(B) Combination of Union and Cartesian product
(C) Combination of selection and Cartesian product
(D) Combination of projection and Cartesian product

Ans: D

Q.82 Which one of the following is not true for a view:
(A) View is derived from other tables.
(B) View is a virtual table.
(C) A view definition is permanently stored as part of the database.
(D) View never contains derived columns.

Ans: C

Q.83 A primary key if combined with a foreign key creates
(A) Parent-Child relationship between the tables that connect them.
(B) Many to many relationship between the tables that connect them.
(C) Network model between the tables that connect them.
(D) None of the above.

Ans: A

Q.84 In E-R Diagram relationship type is represented by
(A) Ellipse **(B)** Dashed ellipse
(C) Rectangle **(D)** Diamond

Ans: D

Q.85 Hierarchical model is also called
(A) Tree structure **(B)** Plex Structure
(C) Normalize Structure **(D)** Table Structure

Ans: A

Q.86 To delete a particular column in a relation the command used is:
(A) UPDATE **(B)** DROP
(C) ALTER **(D)** DELETE

Ans: C

Q.87 The _____ operator is used to compare a value to a list of literals values that have been specified.
(A) BETWEEN **(B)** ANY
(C) IN **(D)** ALL

Ans: A

Q.88 A logical schema
A) is the entire database
B) is a standard way of organizing information into a accessible part
C) describe how data is actually stored on disk
D) none of these

Ans: D

Q.89 A B-tree of order m has maximum of _____ children
(A) m **(B)** m+1
(C) m-1 **(D)** m/2

Ans: A

Q.90 _____ function divides one numeric expression by another and returns the remainder.
(A) POWER **(B)** MOD
(C) ROUND **(D)** REMAINDER

Ans: B

Q.91 A data manipulation command the combines the records from one or more tables is called
(A) SELECT **(B)** PROJECT
(C) JOIN **(D)** PRODUCT

Ans: C

Q.92 In E-R diagram generalization is represented by

(A) Ellipse **(B)** Dashed ellipse
(C) Rectangle **(D)** Triangle

Ans: D

Q.93 _____ is a virtual table that draws its data from the result of an SQL SELECT statement.

(A) View **(B)** Synonym
(C) Sequence **(D)** Transaction

Ans: A

Q.94 The method of access which uses key transformation is known as

(A) Direct **(B)** Hash
(C) Random **(D)** Sequential

Ans: B

Q.95 A table joined with itself is called

(A) Join **(B)** Self Join
(C) Outer Join **(D)** Equi Join

Ans: B

Q.96 _____ data type can store unstructured data

(A) RAW **(B)** CHAR
(C) NUMERIC **(D)** VARCHAR

Ans: A

Q.97 Which two files are used during operation of the DBMS

(A) Query languages and utilities
(B) DML and query language
(C) Data dictionary and transaction log
(D) Data dictionary and query language

Ans: C

PART-II

DESCRIPTIVES

Q.1 What is a database? Describe the advantages and disadvantages of using of DBMS. **(7)**

Ans: Database – A database is a collection of related data and/or information stored so that it is available to many users for different purposes.

Advantages Of DBMS

1. **Centralized Management and Control** - One of the main advantages of using a database system is that the organization can exert, via the DBA, centralized management and control over the data.
2. **Reduction of Redundancies and Inconsistencies -** Centralized control avoids unnecessary duplication of data and effectively reduces the total amount of data storage required. Removing redundancy eliminates inconsistencies.
3. **Data Sharing** - A database allows the sharing of data under its control by any number of application programs or users.
4. **Data Integrity** - Data integrity means that the data contained in the database is both accurate and consistent. Centralized control can also ensure that adequate checks are incorporated in the DBMS to provide data integrity.
5. **Data Security** - Data is of vital importance to an organization and may be confidential. Such confidential data must not be accessed by unauthorized persons. The DBA who has the ultimate responsibility for the data in the DBMS can ensure that proper access procedures are followed. Different levels of security could be implemented for various types of data and operations.
6. **Data Independence** - Data independence is the capacity to change the schema at one level of a database system without having to change the schema at the next level. It is usually considered from two points of view: physical data independence and logical data independence. Physical data independence is the capacity to change the internal schema without having to change conceptual schema. Logical data independence is the capacity to change the conceptual schema without having to change external schemas or application programs.
7. **Providing Storage Structures for Efficient Query Processing -** Database systems provide capabilities for efficiently executing queries and updates. Auxiliary files called *indexes* are used for this purpose.
8. **Backup and Recovery -** These facilities are provided to recover databases from hardware and/or software failures.

Some other advantages are:

- Reduced Application Development Time
- Flexibility
- Availability of up-to-date Information

Disadvantages Of DBMS

1. **Cost of Software/Hardware and Migration** - A significant disadvantage of the DBMS system is cost.

2. **Reduced Response and Throughput** - The processing overhead introduced by the DBMS to implement security, integrity, and sharing of the data causes a degradation of the response and throughput times.
3. **Problem with Centralization** - Centralization also means that the data is accessible from a single source namely the database. This increases the potential of security breaches and disruption of the operation of the organization because of downtimes and failures.

Q.2 Explain five duties of Database Administrator. **(7)**

Ans:

1. DBA administers the three levels of the database and, in consultation with the overall user community, sets up the definition of the global view or conceptual level of the database.
2. Mappings between the internal and the conceptual levels, as well as between the conceptual and external levels, are also defined by the DBA.
3. DBA ensures that appropriate measures are in place to maintain the integrity of the database and that the database is not accessible to unauthorized users.
4. DBA is responsible for granting permission to the users of the database and stores the profile of each user in the database.
5. DBA is responsible for defining procedures to recover the database from failures with minimal loss of data.

Q.3 Explain the terms primary key, candidate key and foreign key. Give an example for each. **(7)**

Ans: **Primary Key** – Primary key is one of the candidate keys that uniquely identifies each row in the relation.

Candidate Key – A candidate key of an entity set is a minimal superkey, that uniquely identifies each row in the relation.

Foreign Key – Let there are two relations (tables) *R* and *S*. Any candidate key of the relation *R* which is referred in the relation *S* is called the *foreign key* in the relation *S* and *referenced key* in the relation ***R*.** The relation *R* is also called as *parent table* and relation *S* is also called as *child table***.**

For example:

STUDENT

Enrl No	**Roll No**	**Name**	**City**	***Mobile***
11	17	Ankit Vats	Delhi	9891663808
15	16	Vivek Rajput	Meerut	9891468487
6	6	Vanita	Punjab	
33	75	Bhavya	Delhi	9810618396

GRADE

Roll No	Course	*Grade*
6	C	A
17	VB	C
75	VB	A
6	DBMS	B
16	C	B

- Roll No is the primary key in the relation STUDENT and Roll No + Course is the primary key of the relation GRADE.
- Enrl No and Roll No are the candidate keys of the relation STUDENT.
- Roll No in the relation GRADE is a foreign key whose values must be one of those of the relation STUDENT.

Q.4 Differentiate between logical database design and physical database design. Show how this separation leads to data independence. **(7)**

Ans:

Basis	**Logical Database Design**	**Physical Database Design**
Task	Maps or transforms the conceptual schema (or an ER schema) from the high-level data model into a relational database schema.	The specifications for the stored database in terms of physical storage structures, record placement, and indexes are designed.
Choice of criteria	The mapping can proceed in two stages: ▪ System-independent mapping but data model-dependent ▪ Tailoring the schemas to a specific DBMS	The following criteria are often used to guide the choice of physical database design options: ▪ Response Time ▪ Space Utilization ▪ Transaction Throughput
Result	DDL statements in the language of the chosen DBMS that specify the conceptual and external level schemas of the database system. But if the DDL statements include some physical design parameters, a complete DDL specification must wait until after the physical database design phase is completed.	An initial determination of storage structures and the access paths for the database files. This corresponds to defining the internal schema in terms of Data Storage Definition Language.

The database design is divided into several phases. The logical database design and physical database design are two of them. This separation is generally based on the concept of three-level architecture of DBMS, which provides the data independence. Therefore, we can say that this separation leads to data independence because the output of the logical database design is the conceptual and external level schemas of the database

system which is independent from the output of the physical database design that is internal schema.

Q.5 Consider the following relation schemes: **(2×7=14)**

Project (Project#, Project_name, chief_architect)
Employee (Emp#, Empname)
Assigned_To (Project#, Emp#)
Give expression in Tuple calculus and Domain calculus for each of the queries below:

(i) Get the employee numbers of employees who work on all projects.

(ii) Get the employee numbers of employees who do not work on the COMP123 project.

Ans:

(i) Tuple Calculus:

$\{t[Emp\#] \mid t \in ASSIGNED\_TO \wedge \forall p\ (p \in PROJECT \rightarrow \exists u\ (u \in ASSIGNED\_TO \wedge p[Project\#] = u[Project\#] \wedge t[Emp\#] = u[Emp\#]))\}$

Domain Calculus:

$\{e \mid \exists p\ (<p, e> \in ASSIGNED\_TO \wedge \forall p_1\ (<p_1, n_1, c_1> \in PROJECT \rightarrow <p_1, e> \in ASSIGNED\_TO))\}$

(ii) Tuple Calculus:

$\{t[Emp\#] \mid t \in ASSIGNED\_TO \wedge \neg\exists u\ (u \in ASSIGNED\_TO \wedge u[Project\#] = \text{'COMP123'} \wedge t[Emp\#] = u[Emp\#])\}$

Domain Calculus:

$\{e \mid \exists p\ (<p, e> \in ASSIGNED\_TO \wedge \forall p_1, e_1\ (<p_1, e_1> \notin ASSIGNED\_TO \vee p_1 \neq \text{'COMP123'} \vee e_1 \neq e))\}$

Q.6 What is ODBC? How does Oracle act as ODBC and give examples of front end uses with ODBC. **(7)**

Ans:

ODBC – Open DataBase Connectivity (ODBC) enable the integration of SQL with a general-purpose programming language. ODBC expose database capabilities in a standardized way to the application programmer through an application programming interface (API). Using ODBC, an application can access not just one DBMS but several different ones simultaneously.

ODBC achieve portability at the level of the executable by introducing an extra level of indirection. All direct interaction with a specific DBMS happens through a DBMS-specific driver. A driver is a software program that translates the ODBC calls into DBMS-specific calls. Drivers are loaded dynamically on demand since the DBMSs the application is going to access are known only at run-time. Available drivers are registered with a driver manager. The Oracle database driver translates the SQL commands from the application into equivalent commands that the Oracle DBMS understands and takes the result from the DBMS and translate into equivalent form for the application.

Example: Let there be a DSN named EMPLOYEE through, which we want to access the Oracle database in Visual Basic.

```
Dim CN As New ADODB.Connection
Dim RS As New ADODB.Recordset
CN.Open "DSN=employee", "scott", "tiger"
```

RS.Open "Select * From Emp", CN

Q.7 Define the five basic operators of relational algebra with an example each. **(7)**

Ans: Five basic operators of relational algebra are:

1. Union (∪) - Selects tuples that are in either P or Q or in both of them. *The duplicate tuples are eliminated.*

$R = P \cup Q$

2. Minus (–) - Removes common tuples from the first relation.

$R = P - Q$

3. Cartesian Product or Cross Product (×) - The cartesian product of two relations is the concatenation of tuples belonging to the two relations and consisting of all possible combination of the tuples**.**

$R = P \times Q$

For Example:

P:

ID	Name
101	Jones
103	Smith
104	Lalonde

Q:

ID	Name
100	John
104	Lalonde

R = P ∪ Q

ID	Name
100	John
101	Jones
103	Smith
104	Lalonde

R = P – Q

ID	Name
101	Jones
103	Smith

R = P × Q

P.ID	P.Name	Q.ID	Q.Name
101	Jones	100	John
101	Jones	104	Lalonde
103	Smith	100	John
103	Smith	104	Lalonde
104	Lalonde	100	John
104	Lalonde	104	Lalonde

4. **Projection (π) -** The projection of a relation is defined as a projection of all its tuples over some set of attributes, i.e., it yields a *vertical subset* of the relation. It is used to either *reduce* the number of attributes (degree) in the resultant relation or to *reorder* attributes. The projection of a relation T on the attribute A is denoted by $\pi_A(T)$.
5. **Selection (σ) -** Selects only some of the tuples, those satisfy given criteria, from the relation. It yields a *horizontal subset* of a given relation, i.e., the action is defined over a complete set of attribute names but only a subset of the tuples are included in the result.
   $R = \sigma_B(P)$

   For Example:

EMPLOYEE:

Id	Name
101	Jones
103	Smith
104	Lalonde
106	Byron

→

Name
Jones
Smith
Lalonde
Byron

Projection of relation EMPLOYEE over attribute Name

EMPLOYEE:

Id	Name
101	Jones
103	Smith
104	Lalonde
106	Byron

→ **Result of Selection**

Id	Name
104	Lalonde
106	Byron

Result of Selection over **EMPLOYEE** for **ID > 103**

Q.8 Explain entity integrity and referential integrity rules in relational model. Show how these are realized in SQL. **(7)**

Ans:

Entity Integrity Rule – No primary key value can be null.

Referential Integrity Rule – In referential integrity, it is ensured that a value that appears in one relation for a given set of attributes also appears for a certain set of attributes in another relation.

In SQL, entity integrity and referential integrity rules are implemented as constraints on the relation called as primary key constraint and reference key constraint respectively. These constraints can be specified with relation at the time of creation of the relations or after the creation of the relations by altering the definition of the relations. For example:

```
CREATE TABLE DEPT
(DEPTNO     NUMBER PRIMARY KEY,
 DNAME      VARCHAR2(15));
CREATE TABLE EMP
(EMPNO      NUMBER PRIMARY KEY,
```

```
ENAME      VARCHAR2(15),
JOB        VARCHAR2(10),
DEPTNO     NUMBER REFERENCES DEPT(DEPTNO));
```

Q.9 What are the advantages of embedded query language? Give an example of a embedded SQL query. **(7)**

Ans:

Embedded query language – SQL can be implemented in two ways. It can be used interactively or embedded in a host language or by using API. The use of SQL commands within a host language (e.g., C, Java, etc.) program is called embedded query language or Embedded SQL. Although similar capabilities are supported for a variety of host languages, the syntax sometimes varies. Some of the advantages of embedded SQL are:

- SQL statements can be used wherever a statement in the host language is allowed.
- It combines the strengths of two programming environments, the procedural features of host languages and non-procedural features of SQL.
- SQL statements can refer to variables (must be prefixed by a colon in SQL statements) defined in the host program.
- Special program variables (called null indicators) are used to assign and retrieve the NULL values to and from the database.
- The facilities available through the interactive query language are also automatically available to the host programs.
- Embedded SQL along with host languages can be used to accomplish very complex and complicated data access and manipulation tasks.

Example: The following Embedded SQL statement in C inserts a row, whose column values are based on the values of the host language variables contained in it**.**

```
EXEC SQL
INSERT INTO Sailors VALUES (:c_sname, :c_sid, :c_rating, :c_age);
```

Q.10 Consider the following relations: **(3.5 x 2=7)**

S (S#, SNAME, STATUS, CITY)
SP (S#, P#, QTY)
P (P#, PNAME, COLOR, WEIGHT, CITY)

Give an expression in SQL for each of queries below:

(i) Get supplier names for supplier who supply at least one red part
(ii) Get supplier names for supplier who do not supply part P2.

Ans:(i)

```
SELECT SNAME FROM S
        WHERE S# IN (SELECT S# FROM SP
                WHERE P# IN (SELECT P# FROM P
                        WHERE COLOR = RED'))
```

(ii)

```
SELECT SNAME FROM S
WHERE S# NOT IN (SELECT S# FROM SP WHERE P# = 'P2')
```

Q.11 Define a view and a trigger. Construct a view for the above relations which has the information about suppliers and the parts they supply. The view contains the S#, SNAME, P# , PNAME renamed as SNO, NAME, PNO, PNAME. **(7)**

Ans:

View – A view is a virtual table which is based on the one or more physical tables and/or views. In other words, a view is a named table that is represented, not by its own physically separate stored data, but by its definition in terms of other named tables (base tables or views).

Trigger – A trigger is a procedure that is automatically invoked by the DBMS in the response to specified changes to the database. Triggers may be used to supplement declarative referential integrity, to enforce complex business rules or to audit changes to data.

Command:

```
CREATE VIEW SUP_PART (SNO, NAME, PNO, PNAME) AS
        SELECT S.S#, SNAME, P.P#, PNAME
            FROM S, SP, P
            WHERE S.S# = SP.S# AND P.P# = SP.P#
```

Q.12 Differentiate between the following: **(10)**

(i) Theta Join. (ii) Equi Join. (iii) Natural Join
(iv) Outer Join.

Ans:(i) Theta Join – The theta join operation is an extension to the natural-join operation that allows us to combine selection and a Cartesian product into a single operation. Consider relations *r(R)* and *s(S)*, and let θ be a predicate on attributes in the schema $R \cup S$. The theta join operation $r \rhd\lhd_\theta s$ is defined as follows:

$$r \rhd\lhd_\theta s = \sigma_\theta (r \times s)$$

(ii) Equi Join – It produces all the combinations of tuples from two relations that satisfy a join condition with only equality comparison (=).

(iii) Natural Join - Same as equi-join except that the join attributes (having same names) are not included in the resulting relation. Only one sets of domain compatible attributes involved in the natural join are present.

(iv) Outer Join - If there are any values in one table that do not have corresponding value(s) in the other, in an equi-join that will not be selected. Such rows can be forcefully selected by using the outer join. The corresponding columns for that row will have NULLs. There are actually three forms of the outer-join operation: left outer join ($\overline{\times}$), right outer join ($\times\!\!\!\!-$) and full outer join ($\overline{\times}\!\!\!\!-$).

Q.13 What are temporary tables? When are they useful? Justify with an example. **(4)**

Ans:

Temporary tables exists solely for a particular session, or whose data persists for the duration of the transaction. The temporary tables are generally used to support specialized rollups or specific application processing requirements. Unlike a permanent table, a space is not allocated to a temporary table when it is created. Space will be dynamically allocated for the table as rows are inserted. The CREATE GLOBAL TEMPORARY TABLE command is used to create a temporary table in Oracle.

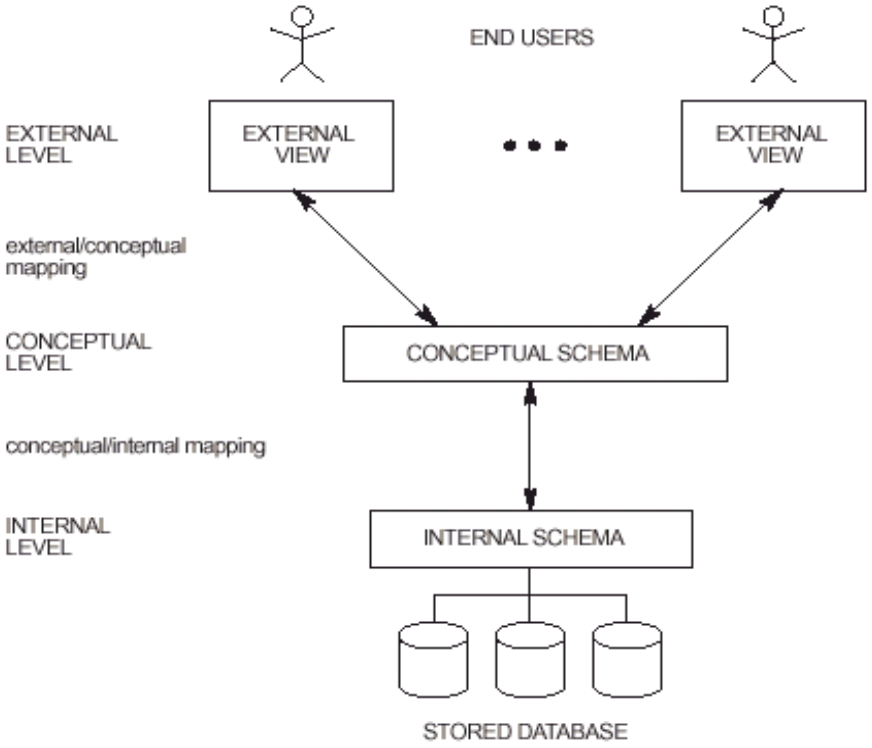
```
CREATE GLOBAL TEMPORARY TABLE <table_name> (
```

<columns details>
) ON COMMIT {PRESERVE|DELETE} ROWS;

Q.14 Draw and explain the three level architecture of the database system. **(7)**

Ans:

A DBMS provides three levels of data is said to follow three-level architecture. The goal of the three-schema architecture is to separate the user applications and the physical database. The view at each of these levels is described by a schema. The processes of transforming requests and results between levels are called *mappings*. In this architecture, schemas can be defined at the following three levels:



- **External Level or Subschema –** It is the highest level of database abstraction where only those portions of the database of concern to a user or application program are included. Any number of user views (some of which may be identical) may exist for a given global or conceptual view. Each external view is described by means of a schema called an *external schema* or *subschema*.
- **Conceptual Level or Conceptual Schema -** At this level of database abstraction all the database entities and the relationships among them are included. One conceptual view represents the entire database. This conceptual view is defined by the *conceptual schema*. There is only one conceptual schema per database. The description of data at this level is in a format independent of its physical representation. It also includes features that specify the checks to retain data consistency and integrity.
- **Internal Level or Physical Schema –** It is closest to the physical storage method used. It indicates how the data will be stored and describes the data structures and access methods to be used by the database. The internal view is expressed by the *internal schema*.

Q.15 Explain (a) Heap file (b) Sorted file. Also discuss their advantages and disadvantages.

Ans: Heap File is an unordered set of records, stored on a set of pages. This class provides basic support for inserting, selecting, updating, and deleting records. Temporary heap files are used for external sorting and in other relational operators. A sequential scan of a heap file (via the Scan class) is the most basic access method.

Sorted file The *sort* utility shall perform one of the following functions:

1.Sort lines of all the named files together and write the result to the specified output.

2.Merge lines of all the named (presorted) files together and write the result to the specified output.

3.Check that a single input file is correctly presorted.

Comparisons shall be based on one or more sort keys extracted from each line of input (or, if no sort keys are specified, the entire line up to, but not including, the terminating <newline>), and shall be performed using the collating sequence of the current locale**.**

Q.16 Describe a method for direct search? Explain how data is stored in a file so that direct searching can be performed.

Ans: For a file of unordered fixed length records using unspanned blocks and contiguous allocation, it is straight forward to access any record by its position in the file. If the file records are numbered 0,1,2,---,r-1 and the records in each block are numbered 0,1,---bfr-1; where bfr is the blocking factor, then ith record of the file is located in block [(i/bfr)] and is the (I mod bfr)th record in that block. Such a file is often called a relative or direct file because records can easily be accessed directly by their relative positions. Accessing a record based on a search condition; however, it facilitates the construction of access paths on the file, such as the indexes.

Q17 Explain the integrity constraints: Not Null, Unique, Primary Key with an example each. Is the combination 'Not Null, Primary Key' a valid combination. Justify. **(7)**

Ans: Not Null – Should contain valid values and cannot be NULL.
Unique – An attribute or a combination of two or more attributes must have a unique value in each row. The unique key can have NULL values.

Primary Key – It is same as unique key but cannot have NULL values. A table can have at most one primary key in it.

For example:

STUDENT

Roll No	**Name**	**City**	***Mobile***
17	Ankit Vats	Delhi	9891663808
16	Vivek Rajput	Meerut	9891468487
6	Vanita	Punjab	NULL
75	Bhavya	Delhi	9810618396

- Roll No is a primary key.
- Name is defined with NOT NULL, means each student must have a name.
- Mobile is unique**.**

'*Not Null, Primary Key*' is a valid combination. Primary key constraint already includes 'Not Null' constraint in it but we can also add 'Not Null' constraint with it. The use of 'Not Null' with 'Primary Key' will not have any effect. It is same as if we are using just 'Primary Key'.

Q.18 Explain the followings :
(i) Nested Queries.
(ii) Cursors in SQL.
(iii) RDBMS.
(iv) View
(v) Application Programming Interface **(14)**

Ans: (i) Nested Queries – A SELECT query can have subquery(s) in it. When a SELECT query having another SELECT query in it, is called as nested query. Some operations cannot be performed with single SELECT command or with join operation. There are some operations which can be performed with the help of nested queries (also referred to as subqueries). For example, we want to compute the second highest salary:
SELECT MAX(SAL) FROM EMP WHERE SAL < (SELECT MAX(SAL) FROM EMP)
Some operations can be performed both by Join and subqueries. The Join operation is costlier in terms of time and space. Therefore, the solution based on subqueries is preferred.
(ii) Cursors in SQL – An object used to store the output of a query for row-by-row processing by the application programs. Cursors are constructs that enable the user to name a private memory area to hold a specific statement for access at a later time. Cursors are used to process multi-row result sets one row at a time. Additionally, cursors keep track of which row is currently being accessed, which allows for interactive processing of the active set.
(iii) RDBMS – RDBMS is a database management system (DBMS) that stores data in the form of relations. Relational databases are powerful because they require few assumptions about how data is related or how it will be extracted from the database. As a result, the same database can be viewed in many different ways. An important feature of

relational system is that a single database can be spread across several tables. This differs from flat-file databases, in which each database is self-contained in a single table.

(iv) View – A view is a relation (virtual rather than base) and can be used in query expressions, that is, queries can be written using the view as a relation. In other words, a view is a named table that is represented, not by its own physically separate stored data, but by its definition in terms of other named tables (base tables or views). The base relations on which a view is based are sometimes called the existing relations. The definition of a view in a create view statement is stored in the system catalog. The syntax to create a view is:CREATE [OR REPLACE] VIEW <view_name> [(<aliases>)] AS <query> WITH {READ ONLY|CHECK OPTION [CONSTRAINT <constraint_name>]};

(v) Application Programming Interface – Commercial SQL implementations take one of the two basic techniques for including SQL in a programming language – embedded SQL and application program interface (API). In the application program interface approach, the program communicates with the RDBMS using a set of functions called the Application Program Interface (API). The program passes the SQL statements to the RDBMS using API calls and uses API calls to retrieve the results. In this method, the precompiler is not required.

Q.19 Consider the following relational schema: **(7)**

PERSON (SS#, NAME, ADDRESS)
CAR (REGISTRATION_NUMBER, YEAR, MODEL)
ACCIDENT (DATE, DRIVER, CAR_REG_NO)
OWNS (SS#, LICENSE)

Construct the following relational algebra queries:

(i) Find the names of persons who are involved in an accident.

(ii) Find the registration number of cars which were not involved in any accident.

Ans: **(i)** $\pi_{NAME}(PERSON) \cap \pi_{DRIVER}(ACCIDENT)$

(ii) $\pi_{REGISTRATION\_NUMBER}(CAR) - \pi_{CAR\_REG\_NO}(ACCIDENT)$

Q.20 What is a key? Explain Candidate Key, Alternate Key and Foreign Key. **(7)**

Ans:

Key – A single attribute or a combination of two or more attributes of an entity set that is used to identify one or more instances (rows) of the set (table) is called as key.

Candidate Key – A candidate key is a minimal superkey, which can be used to uniquely identify a tuple in the relation.

Alternate Key – All the *candidate keys except primary key* are called as alternate keys.

Foreign Key – Let there are two relations (tables) *R* and *S*. Any candidate key of the relation *R* which is referred in the relation *S* is called the *foreign key* in the relation *S* and *referenced key* in the relation *R*. The relation *R* is also called as *parent table* and relation *S* is also called as *child table*.

Q.21 What is data independence? Explain the difference between physical and logical data independence. **(7)**

Ans: Data independence is the capacity to change the schema at one level of a database system without having to change the schema at the next level. The three-schema architecture allows the feature of data independence. Data independence occurs

because when the schema is changed at some level, the schema at the next level remains unchanged; only the *mapping* between the two levels is changed. Types of data independence are:

- **Physical Data Independence –** It is capacity to change the internal schema without having to change conceptual schema. Hence, the external schemas need not be changed as well. Changes to the internal schema may be needed because some physical files had to be reorganized to improve the performance of retrieval or update. If the same data as before remains in the database, the conceptual schema needs not be changed.
- **Logical Data Independence -** It is the capacity to change the conceptual schema without having to change external schemas or application programs. The conceptual schema may be changed to expand the database (by adding a record type or data item), to change constraints, or to reduce the database (by removing a record type or data item). Only the view definition and the mappings need be changed in a DBMS that supports logical data independence. Changes to constraints can be applied to the conceptual schema without affecting the external schemas or application programs.

Q.22 Write short notes on:

(i) Weak and strong entity sets.
(ii) Types of attributes.
(iii) Oracle Instance.
(iv) Mid square method of hashing. **(4 x 4 = 16)**

Ans: (i) Weak and Strong entity sets: A strong entity set has a primary key. All tuples in the set are distinguishable by that key. A weak entity set has no primary key unless attributes of the strong entity set on which it depends are included. Tuples in a weak entity set are partitioned according to their relationship with tuples in a strong entity set. Tuples within each partition are distinguishable by a discriminator, which is a set of attributes. A strong entity set has a primary key. All tuples in the set are distinguishable by that key. A weak entity set has no primary key unless attributes of the strong entity set on which it depends are included. Tuples in a weak entity set are partitioned according to their relationship with tuples in a strong entity set. Tuples within each partition are distinguishable by a discriminator, which is a set of attributes.

(ii) Types of attributes: An attribute's type determines the kind of values that are allowed in the attribute. For example, the value *version 1* is not valid for an attribute defined as an integer, but the value *1* is valid. Numeric types (such as integer or real) can also be limited to a predefined range by their attribute definition.

Choice : An attribute with a list of predefined values.

ID Reference: An attribute with a value that is a Unique ID value from another element. It is typically used for element-based cross-references.

ID References: An attribute with a value of one or more Unique ID values from another element.

Integer: An attribute with a whole number value (no decimal parts). Examples of valid integers are 22, -22, and +322. An integer can be defined to fall within a range.

Integers: An attribute with a value of one or more integers. Enter each number on a separate line in the Attribute Value text box.

Real An attribute with a real number value, with or without a decimal part (the value can also be expressed in scientific notation). Examples of valid real numbers are 2, 22.4, -0.22, and $2.3e^{-1}$. A real number can be defined to fall within a range.

Reals: An attribute with a value of one or more real numbers. Enter each number on a separate line in the Attribute Value text box.

String: An attribute with a value of a series of characters (text).

Strings: An attribute with a value of one or more strings. Enter each string on a separate line in the Attribute Value text box.

Unique ID: An attribute with a value of a unique text string. An element can have only one ID attribute (which can be of type Unique ID or Unique IDs). All ID values must be unique in the document or book. An element with a Unique ID attribute can be the source for an element-based cross-reference.

Unique IDs: An attribute with a value of one or more unique text strings. Enter each string on a separate line in the Attribute Value text box.

(iii) Oracle Instances: An instance is the (executed) Oracle software and the memory they use. It is the instance that manipulates the data stored in the database. It can be started independent of any database. It consists of:

1) A shared memory area that provides the communication between various processes.

2) Upto five background processes which handled various tasks.

Whenever an oracle instance starts, the file 'INIT.ORA' is executed.

(iv) Mid square method of hashing: In midsquare hashing, the key is squared and the address selected from the middle of the squared number.

Mid square method

* Square K.
* Strip predetermined digits from front and rear.
* e.g., use thousands and ten thousands places.

Q.23 Consider the following relational schemas:

EMPLOYEE (EMPLOYEE_NAME, STREET, CITY)
WORKS (EMPLOYEE_NAME, COMPANYNAME, SALARY)
COMPANY (COMPANY_NAME, CITY)

Specify the table definitions in SQL. **(5)**

Ans:

```
CREATE TABLE EMPLOYEE
( EMPLOYEE_NAME        VARCHAR2(20) PRIMARY KEY,
  STREET               VARCHAR2(20),
  CITY          VARCHAR2(15));
CREATE TABLE COMPANY
( COMPANY_NAME         VARCHAR2(50) PRIMARY KEY,
  CITY          VARCHAR2(15));
CREATE TABLE WORKS
( EMPLOYEE_NAME        VARCHAR2(20)
        REFERENCES EMPLOYEE(EMPLOYEE_NAME,
  COMPANYNAME          VARCHAR2(50)
        REFERENCES COMPANY(COMPANY_NAME,
  SALARY               NUMBER(6),
  CONSTRAINT WORKS_PK PRIMARY KEY(EMPLOYEE_NAME,
        COMPANY_NAME));
```

Q.24 Give an expression in SQL for each of queries below: **(9)**

(i) Find the names of all employees who work for first Bank Corporation.
(ii) Find the names and company names of all employees sorted in ascending order of company name and descending order of employee names of that company.
(iii) Change the city of First Bank Corporation to 'New Delhi'

Ans:

(i)
```
SELECT EMPLOYEE_NAME
    FROM WORKS
    WHERE COMPANYNAME = 'First Bank Corporation';
```
(ii)
```
SELECT EMPLOYEE_NAME, COMPANYNAME
    FROM WORKS
    ORDER BY COMPANYNAME, EMPLOYEE_NAME DESC;
```
(iii)
```
UPDATE COMPANY
    SET CITY = 'New Delhi'
    WHERE COMPANY_NAME = 'First Bank Corporation';
```

Q.25 Discuss the correspondence between the E-R model construct and the relation model construct. Show how each E-R model construct can be mapped to the relational model using the suitable example?

Ans: **An entity-relationship model (ERM):** An entity-relationship model (ERM) is an abstract conceptual representation of structured data. Entity-relationship modeling is a relational schema database modeling method, used in software engineering to produce a type of conceptual data model (or semantic data model) of a system, often a relational database, and its requirements in a top-down fashion. Diagrams created using this process are called *entity-relationship diagrams*, or *ER diagrams* or *ERDs* for short.

ER-to-Relational Mapping Algorithm:

1) Step 1: Mapping of regular entity types: For each strong entity type E, create a relation T that includes all the simple attributes of a composite attribute.

2) Step2: Mapping of weak entity types: For each weak entity type W with owner entity type E, create relation R and include all simple attributes (or simple components of composite attributes) of W as attributed of R. In addition, include as foreign key attributes of R, the primary key attribute (s) of relation(s) that correspond to the owner(s) and the partial key of the weak entity type W, if any.

3) Mapping of relationship types: form a relation R, for relationship with primary keys of participating relations A and B as foreign keys in R. In addition to this, any attributes of relationship become an attribute of R also.

4) Mapping of multivalued attributes: For each multilvalued attribute A, create a new relation R. This relation R will include an attribute corresponding to A, plus primary key attribute K-as a foreign key in R-of the relation that represents the entity type or relationship type that has A as an attribute.

Q.26 Explain the concepts of relational data model. Also discuss its advantages and disadvantages. **(7)**

Ans:

Relational Data Model – The relational model was first introduced by Prof. E.F. Codd of the IBM Research in 1970 and attracted immediate attention due to its simplicity and

mathematical foundation. The model uses the concept of a mathematical relation (like a table of values) as its basic building block, and has its theoretical basis in set theory and first-order predicate logic. The relational model represents the database as a collection of *relations*. The relational model like all other models consists of three basic components:

- a set of domains and a set of relations
- operation on relations
- integrity rules

Advantages

- **Ease of use –** The revision of any information as tables consisting of rows and columns is quite natural and therefore even first time users find it attractive.
- **Flexibility –** Different tables from which information has to be linked and extracted can be easily manipulated by operators such as project and join to give information in the form in which it is desired.
- **Security –** Security control and authorization can also be implemented more easily by moving sensitive attributes in a given table into a separate relation with its own authorization controls. If authorization requirement permits, a particular attribute could be joined back with others to enable full information retrieval.
- **Data Independence –** Data independence is achieved more easily with normalization structure used in a relational database than in the more complicated tree or network structure. It also frees the users from details of storage structure and access methods.
- **Data Manipulation Language –** The possibility of responding to ad-hoc query by means of a language based on relational algebra and relational calculus is easy in the relational database approach. Provides simplicity in the data organization and the availability of reasonably simple to very powerful query languages.

Disadvantages

- **Performance** – If the number of tables between which relationships to be established are large and the tables themselves are voluminous, the performance in responding to queries is definitely degraded.
- **Unsuitable for Hierarchies** – While the relational database approach is a logically attractive, commercially feasible approach, but if the data is for example naturally organized in a hierarchical manner and stored as such, the hierarchical approach may give better results.

Q.27 Consider the following relational schema: **(14)**

Doctor(DName,Reg_no)
Patient(Pname, Disease)
Assigned_To (Pname,Dname)

Give expression in both Tuple calculus and Domain calculus for each of the queries:

(i) Get the names of patients who are assigned to more than one doctor.

(ii) Get the names of doctors who are treating patients with 'Polio'.

Ans:

(i)Tuple Calculus:

$\{p[PName] \mid p \in PATIENT \wedge \exists a_1, a_2 (a_1 \in ASSIGNED\_TO \wedge a_2 \in ASSIGNED\_TO \wedge p[PName] = a_1[PName] \wedge a_1[PName] = a_2[PName] \wedge a_1[DName] \neq a_2[DName])\}$

Domain Calculus:

$\{p \mid \exists p_1, d_1, p_2, d_2 (<p, s> \in PATIENT \wedge <p_1, d_1> \in ASSIGNED\_TO \wedge <p_2, d_2> \in ASSIGNED\_TO \wedge p_1 = p_2 \wedge d_1 \neq d_2)\}$

(ii)Tuple Calculus:

$\{u[Dname] \mid u \in ASSIGNED\_TO \wedge \exists t (t \in PATIENT \wedge t[Disease] = \text{'Polio'} \wedge t[PName] = u[PName])\}$

Domain Calculus:

$\{d \mid \exists p_1, p_2, s_2 (<p_1, d> \in ASSIGNED\_TO \wedge <p_2, s_2> \in PATIENT \wedge p_1 = p_2 \wedge s_2 = \text{'Polio'})\}$

Q.28 What are the features of embedded SQL? Explain. **(7)**

Ans: Embedded SQL – SQL can be implemented in two ways. It can be used interactively or embedded in a host language or by using API. The use of SQL commands within a host language (e.g., C, Java, etc.) program is called embedded query language or Embedded SQL. Although similar capabilities are supported for a variety of host languages, the syntax sometimes varies. Some of the features of embedded SQL are:

- SQL statements can be used wherever a statement in the host language is allowed.
- It combines the strengths of two programming environments, the procedural features of host languages and non-procedural features of SQL.
- SQL statements can refer to variables (must be prefixed by a colon in SQL statements) defined in the host program.
- Special program variables (called null indicators) are used to assign and retrieve the NULL values to and from the database.
- The facilities available through the interactive query language are also automatically available to the host programs.
- Embedded SQL along with host languages can be used to accomplish very complex and complicated data access and manipulation tasks.

Q.29 What is the purpose of tables, private synonyms and public synonyms? If there are multiple objects of same name on an Oracle database, which order are they accessed in?

Ans: The purpose of table is to store data. If we use the PUBLIC keyword (or no keyword at all), anyone who has access to the database can use our synonym. If the database is not ANSI-compliant, a user does not need to know the name of the owner of a public synonym. Any synonym in a database that is not ANSI-compliant **and** was created in an Informix database server is a public synonym. In an ANSI-compliant database, all synonyms are private. If you use the PUBLIC or PRIVATE keywords, the database server issues a syntax error. If you use the PRIVATE keyword to declare a synonym in a database that is not ANSI-compliant, the unqualified synonym can be used by its owner. Other users must qualify the synonym with the name of the owner.

Q.30 Explain the followings: **(14)**

(i) Temporary Tables

(ii) Integrity Constraints.

Ans:

(i) Temporary Tables – Temporary tables exists solely for a particular session, or whose data persists for the duration of the transaction. The temporary tables are generally used to support specialized rollups or specific application processing requirements. Unlike a permanent table, a temporary table does not automatically allocate space when it is created. Space will be dynamically allocated for the table as rows are inserted.
The CREATE GLOBAL TEMPORARY TABLE command is used to create a temporary table in Oracle.

```
CREATE GLOBAL TEMPORARY TABLE <table_name>
  (
        <columns details>
  )
  ON COMMIT {PRESERVE|DELETE} ROWS;
```

(ii)Integrity Constraints – A database is only as good as the information stored in it, and a DBMS must therefore help prevent the entry of incorrect information. An *integrity constraint* is a condition specified on a database schema and restricts the data that can be stored in an instance of the database. If a database instance satisfies all the integrity constraints specified on the database schema, it is a legal instance. A DBMS enforces integrity constraints, in that it permits only legal instances to be stored in the database. Integrity constraints are specified and enforced at different times:

- When the DBA or end user defines a database schema, he or she specifies the integrity constraints that must hold on any instance of this database.
- When a database application is run, the DBMS checks for violations and disallows changes to the data that violate the specified integrity constraints.

Many kinds of integrity constraints can be specified in the relational model, such as, Not Null, Check, Unique, Primary Key, etc.

Q.31 Explain different types of failures that occur in Oracle database. **(7)**

Ans: **Types of Failures –** In Oracle database following types of failures can occurred:

- Statement Failure
  - Bad data type
  - Insufficient space
- Insufficient Privileges (e.g., object privileges to a role)
- User Process Failure
  - User performed an abnormal disconnect
  - User's session was abnormally terminated
  - User's program raised an address exception
- User Error
  - User drops a table
  - User damages data by modification
- Instance Failure
- Media Failure
  - User drops a table
  - User damages data by modification
- Alert Logs
  - Records informational and error messages
  - All Instance startups and shutdowns are recorded in the log

- Every Create, Alter, or Drop operation on a rollback segment, tablespace, or database is record in the log
- Recovery Views
- DB Verify
  - Used to insure that a datafile is valid before a restore

Q.32 What is ODBC? What are the uses of ODBC? Under what circumstances we use this technology? **(7)**

Ans:
ODBC – Open DataBase Connectivity (ODBC) enable the integration of SQL with a general-purpose programming language. ODBC expose database capabilities in a standardized way to the application programmer through an application programming interface (API). In contrast to Embedded SQL, ODBC allows a single executable to access different DBMSs without recompilation. Thus, while Embedded SQL is DBMS-independent only at the source code level, applications using ODBC are DBMS-independent at the source code level and at the level of the executable.
All direct interaction with a specific DBMS happens through a DBMS-specific driver. A driver is a software program that translates the ODBC calls into DBMS-specific calls. Drivers are loaded dynamically on demand since the DBMSs the application is going to access are known only at run-time. Available drivers are registered with a driver manager. The driver translates the SQL commands from the application into equivalent commands that the DBMS understands. An application that interacts with a data source through ODBC selects a data source, dynamically loads the corresponding driver, and establishes a connection with the data source. ODBC achieves portability at the level of the executable by introducing an extra level of indirection. In addition, using ODBC, an application can access not just one DBMS but several different ones simultaneously.

Q.33 List any two significant differences between a file processing system and a DBMS. **(4)**

Ans:
File Processing System vs. DBMS
Data Independence - Data independence is the capacity to change the schema at one level of a database system without having to change the schema at the next level. In file processing systems the data and applications are generally interdependent, but DBMS provides the feature of data independence.
Data Redundancy – Data redundancy means unnecessary duplication of data. In file processing systems there is redundancy of data, but in DBMS we can reduce data redundancy by means of normalization process without affecting the original data. If we do so in file processing system, it becomes too complex.

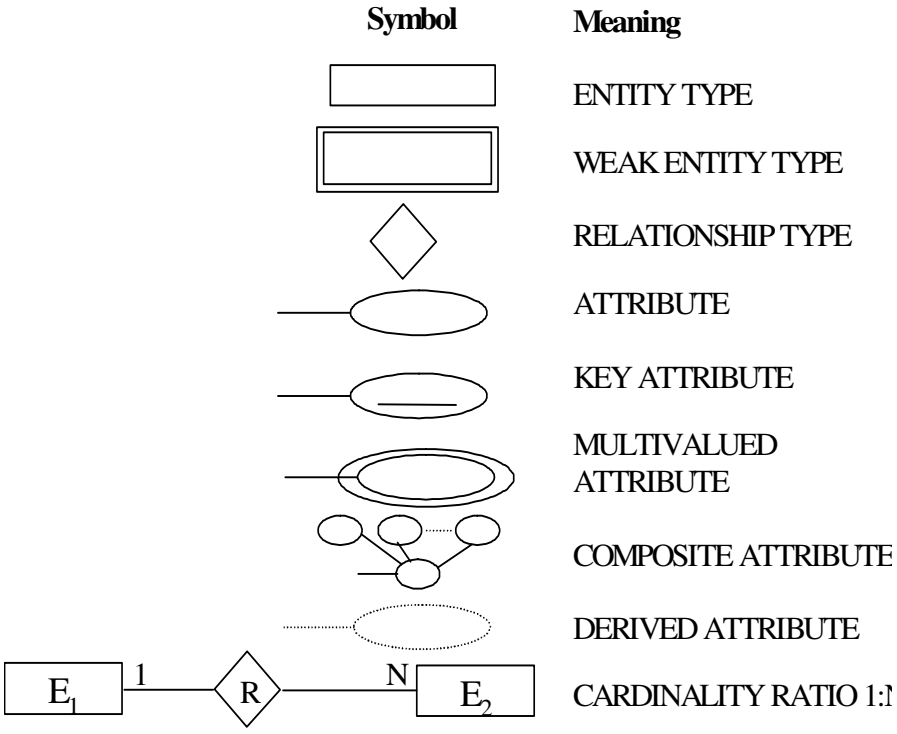
Q.34 Differentiate between various levels of data abstraction. **(5)**

Ans: Data Abstraction – Abstraction is the process to hide the irrelevant things from the users and represent the relevant things to the user. Database systems are often used by non-computer professionals so that the complexity must be hidden from database system users. This is done by defining levels of abstract as which the database may be viewed, there are logical view or external view, conceptual view and internal view or physical view.

- **External View** – This is the highest level of abstraction as seen by a user. It describes only the part of entire database, which is relevant to a particular user.
- **Conceptual View –** This is the next higher level of abstraction which is the sum total of Database Management System user's views. It describes what data are actually stored in the database. It contains information about entire database in terms of a small number of relatively simple structure.
- **Internal View –** This is the lowest level of abstraction. It describes how the data are physically stored

Q.35 What are the various symbols used to draw an E-R diagram? Explain with the help of an example how weak entity sets are represented in an E-R diagram. **(6)**

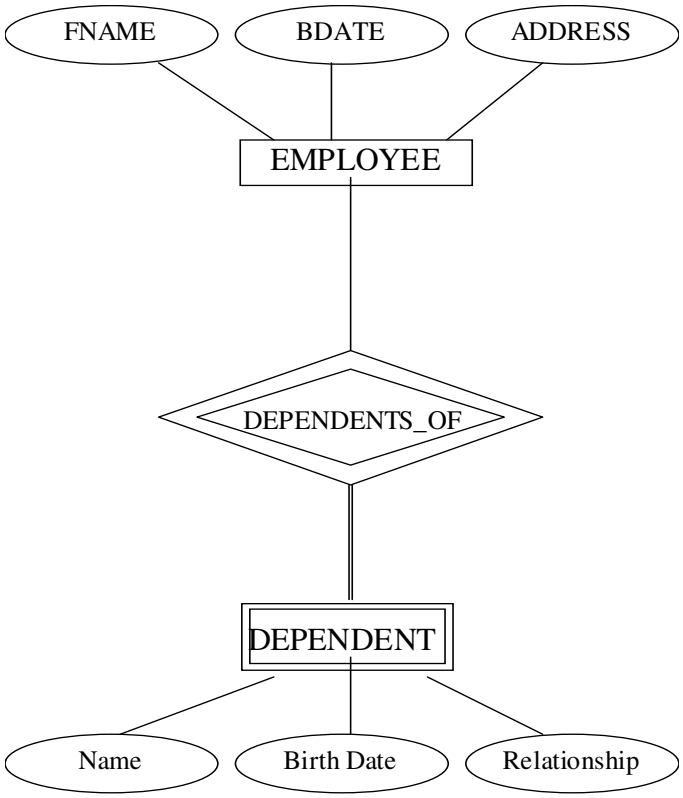
Ans: Various symbols used to draw an E-R diagram



Weak Entity Sets - An entity set that does not have a key attribute is called weak entity set. A weak entity must participate in an identifying relationship type with an owner or identifying entity type Entities are identified by the combination of:

- A partial key of the weak entity type
- The primary key of the identifying entity type

Example:
Suppose that a DEPENDENT entity is identified by the dependent's first name and birhtdate, *and* the specific EMPLOYEE that the dependent is related to. DEPENDENT is a weak entity type with EMPLOYEE as its identifying entity type via the identifying relationship type DEPENDENT_OF



Q.36 Define the following terms: **(8)**

a) Primary key. **b)** DML **c)** Multivalued attribute **d)** Relationship instance

Ans: Primary Key – Primary key is one of the candidate keys. It should be chosen such that its attribute values are never, or very rarely, changed.

b) Data Manipulation Language (DML) – A data manipulation language is a language that enables users to access or manipulate data as organized by the appropriate data model.

c) Multivalued Attribute – Multivalued attribute may have more than one value for an entity. For example, PreviousDegrees of a STUDENT.

d) Relationship Instance – A relationship is an association among two or more entities. An instance of relationship set is a set of relationships.

Q.37 Define a table in SQL called Client, which is used to store information about the clients. Define CLIENT_NO as the primary key whose first letter must start with 'C'. Also ensure that the column 'NAME' should not allow NULL values.

Column name	**Data type**	**Size**
CLIENT_NO	Varchar2	6
NAME	Varchar2	20
ADDRESS1	Varchar2	30
ADDRESS2	Varchar2	30
CITY	Varchar2	15
STATE	Varchar2	15
PINCODE	Number	6
BAL_DUE	Number	10, 2

(7)

Ans:

```
CREATE TABLE CLIENT
( CLIENT_NO      VARCHAR2(6) PRIMARY KEY CHECK (CLIENT_NO LIKE 'C%'),
        NAME          VARCHAR2(20) NOT NULL,
        ADDRESS1      VARCHAR2(30),
        ADDRESS2      VARCHAR2(30),
        CITY          VARCHAR2(15),
        STATE         VARCHAR2(15),
        PINCODE  NUMBER(6),
        BAL_DUE NUMBER(10,2))
```

Q.38 An orchestra database consists of the following relations: **(3.5 x 2=7)**

CONDUCTS (conductor, composition)
REQUIRES (composition, Instrument)
PLAYS (Player, Instrument)
Give the relational calculus queries for the following:

(i) List the compositions and the players.
(ii) List the compositions which require the 'violin' and the 'congo'

Ans: (i) Tuple Calculus:

$\{r[Composition] \| p[Player] \mid r \in REQUIRES \wedge p \in PLAYS \wedge r[Instrument] = p[Instrument]\}$

Domain Calculus:

$\{c \| p \mid \exists i_1, i_2 (<c, i_1> \in REQUIRES \wedge <p, i_2> \in PLAYS \wedge i_1 = i_2)\}$

(ii) Tuple Calculus:

$\{r[Composition] \mid r \in REQUIRES \wedge \exists u (u \in REQUIRES \wedge r[Composition] = u[Composition] \wedge r[Instrument] = \text{'violin'} \wedge u[Instrument] = \text{'congo'})\}$

Domain Calculus:

$\{c \mid \exists i_1, c_2, i_2 (<c, i_1> \in REQUIRES \wedge <c_2, i_2> \in REQUIRES \wedge c_1 = c \wedge i_1 = \text{'violin'} \wedge i_2 = \text{'congo'})\}$

Q.39 Perform the following with syntax and a suitable example
(i) Create a table from existing table.
(ii) Insert data in your table from another table.

Ans: (i) Create table <table-name> as <select-statement>
e.g, To create a new table 'N_emp' with employee names and their identification numbers only from employee table, statement is to create table N_emp as select empname, empid from employee
(ii) insert into <table-name > <select-statement>
e.g, To insert tuples from employee into N_emp created above, use following statement

```
Insert into N_emp select empname, empid from employee
```

Q.40 What is an INDEX as defined in ORACLE? Write the syntax of creating an INDEX. Create an index for the table Client, field CLIENT_NO of Q. **(2+2+3)**

Ans: Indexes in Oracle – Index is typically a listing of keywords accompanied by the location of information on a subject. In other words, An index can be viewed as an auxiliary table which contains two fields: the key and the location of the record of that key. Indexes are used to improve the performance of the search operation. Indexes are not strictly necessary to running Oracle, they do speed the process.

Syntax of Creating an Index:

CREATE [BITMAP] [UNIQUE] INDEX <index_name> ON
<table_name>(<column_name1> [, <column_name2>] . . .);

Command:

CREATE INDEX client_client_no ON client(client_no);

Q.41 Consider the following relational database:
STUDENT (name, student#, class, major)
COURSE (course name, course#, credit hours, department)
SECTION (section identifier, course#, semester, year, instructor)
GRADE_REPORT (student#, section identifier, grade)
PREREQUISITE (course#, presequisite#)
Specify the following queries in SQL on the above database schema. **(3.5 x 4=14)**

(i) Retrieve the names of all students majoring in 'CS' (Computer Science).
(ii) Retrieve the names of all courses taught by Professor King in 1998
(iii) Delete the record for the student whose name is 'Smith' and whose student number is 17.
(iv) Insert a new course <'Knowledge Engineering', 'CS4390', 3, 'CS'>

Ans: (i) SELECT NAME FROM STUDENT WHERE MAJOR = 'CS'

(ii)
```
SELECT COURSE_NAME FROM COURSE C, SECTION S
        WHERE C.COURSE# = S.COURSE#
            AND INSTRUCTOR = 'KING' AND YEAR = 1998
                        OR
SELECT COURSE_NAME FROM COURSE
    WHERE COURSE# IN (SELECT COURSE# FROM SECTION
    WHERE INSTRUCTOR = 'KING' AND YEAR = 1998)
```

(iii) DELETE FROM STUDENT WHERE NAME = 'Smith' AND STUDENT# = 17

(iv)
```
INSERT INTO COURSE
        VALUES('Knowledge Engineering', 'CS4390', 3, 'CS')
```

Q.42 Explain the concept of a data model. What data models are used in database management systems? **(7)**

Ans:

Data Model – Model is an abstraction process that hides irrelevant details while highlighting details relevant to the applications at hand. Similarly, a data model is a collection of concepts that can be used to describe structure of a database and provides the necessary means to achieve this abstraction. Structure of database means the data types, relationships, and constraints that should hold for the data. In general a data model consists of two elements:

- A mathematical notation for expressing data and relationships.
- Operations on the data that serve to express queries and other manipulations of the data.

Data Models used in DBMSs:

- **Hierarchical Model -** It was developed to model many types of hierarchical organizations that exist in the real world. It uses tree structures to represent relationship among records. In hierarchical model, no dependent record can occur without its parent record occurrence and no dependent record occurrence may be connected to more than one parent record occurrence.
- **Network Model -** It was formalised in the late 1960s by the Database Task Group of the Conference on Data System Language (DBTG/CODASYL). It uses two different data structures to represent the database entities and relationships between the entities, namely *record type* and *set type*. In the network model, the relationships as well as the navigation through the database are predefined at database creation time.
- **Relational Model -** The relational model was first introduced by E.F. Codd of the IBM Research in 1970. The model uses the concept of a mathematical relation (like a table of values) as its basic building block, and has its theoretical basis in set theory and first-order predicate logic. The relational model represents the database as a collection of *relations*.
- **Object Oriented Model –** This model is based on the object-oriented programming language paradigm. It includes the features of OOP like inheritance, object-identity, encapsulation, etc. It also supports a rich type system, including structured and collection types.
- **Object Relational Model –** This model combines the features of both relational model and object oriented model. It extends the traditional relational model with a variety of features such as structured and collection types.

Q.43 Briefly explain the differences between a stand alone query language, embedded query language and a data manipulation language. **(7)**

Ans: Stand alone Query Language – The query language which can be used interactively is called stand alone query language. It does not need the support of a host language.

Embedded Query Language – A query language (e.g., SQL) can be implemented in two ways. It can be used interactively or embedded in a host language. The use of query language commands within a host language (e.g., C, Java, etc.) program is called embedded query language. Although similar capabilities are supported for a variety of host languages, the syntax sometimes varies.

Data Manipulation Language (DML) – A data manipulation language is a language that enables users to access or manipulate data as organized by the appropriate data model.

Q.44 Consider the following relations for a database that keeps track of business trips of salespersons in a sales office:

SALESPERSON (SSN, Name, start_year, Dept_no)
TRIP (SSN, From_city, To_city, Departure_Date, Return_Date, Trip_ID)
EXPENSE(TripID, Account#, Amount)

Specify the following queries in relational algebra: **(4x3 =12)**

(i) Give the details (all attributes of TRIP) for trips that exceeded $2000 in expenses.

(ii) Print the SSN of salesman who took trips to 'Honolulu'

(iii) Print the trip expenses incurred by the salesman with SSN= '234-56-7890'.Note that the salesman may have gone on more than one trip. List them individually

Ans: **(i)** $\pi_{TRIP.*}$ ($\sigma_{amount > 2000}$ (TRIP ▷◁ EXPENSE))
(ii) π_{SSN} ($\sigma_{to\_city = 'Honolulu'}$ (TRIP))
(iii) $\pi_{EXPENSE.tripid, amount}$ ($\sigma_{SSN = '234-56-7890'}$ (TRIP ▷◁ EXPENSE))

Q.45 What is the difference between a key and a superkey? **(2)**

Ans: Key – A key a single attribute or a combination of two or more attributes of an entity set that is used to identify one or more instances (rows) of the set (table). It is a minimal combination of attributes.
Super Key – A super key is a set of one or more attributes that, taken collectively, allows us to identify uniquely a tuple in the relation.

Q.46 Why are cursors necessary in embedded SQL? **(2)**

Ans: A cursor is an object used to store the output of a query for row-by-row processing by the application programs. SQL statements operate on a set of data and return a set of data. On other hand, host language programs operate on a row at a time. The cursors are used to navigate through a set of rows returned by an embedded SQL SELECT statement. A cursor can be compared to a pointer.

Q.47 Write a program in embedded SQL to retrieve the total trip expenses of the salesman named 'John' for the relations of Q. 44 **(6)**

Ans:

```
EXEC SQL BEGIN DECLARE SECTION;
long total_expenses;
EXEC SQL END DECLARE SECTION;
EXEC SQL
SELECT SUM(AMOUNT) INTO :total_expenses FROM EXPENSE WHERE TRIPID
IN (SELECT TRIP_ID FROM TRIP
WHERE SSN = (SELECT SSN FROM SALEPERSON WHERE NAME = 'John'));
printf("\nThe total trip expenses of the salesman John is: %ld", total_expenses);
```

Q.48 What are views? Explain how views are different from tables. **(6)**

Ans:
A view in SQL terminology is a single table that is derived from other tables. These other tables could be base tables or previously defined views. A view does not necessarily exist in physical form; it is considered a virtual table, in contrast to base tables, whose tuples are actually stored in the database. This limits the possible update operations that can be applied to views, but it does not provide any limitations on querying a view. A view represents a different perspective of a base relation(s). The definition of a view in a create view statement is stored in the system catalog. Any attribute in the view can be updated as long as the attribute is simple and not derived from a computation involving two or more

base relation attribute. View that involve a join may or may not be updatable. Such views are not updatable if they do not include the primary keys of the base relations.

Q.49 What do you mean by integrity constraints? Explain the two constraints, check and foreign key in SQL with an example for each. Give the syntax. **(8)**

Ans:Integrity Constraints –An *integrity constraint* is a condition specified on a database schema and restricts the data that can be stored in an instance of the database. If a database instance satisfies all the integrity constraints specified on the database schema, it is a legal instance. A DBMS enforces integrity constraints, in that it permits only legal instances to be stored in the database.

CHECK constraint – CHECK constraint specifies an expression that must always be true for every row in the table. It can't refer to values in other rows.

Syntax:

ALTER TABLE <table_name>
ADD CONSTRAINT <constraint_name> CHECK(<expression>);

FOREIGN KEY constraint – A foreign key is a combination of columns with values based on the primary key values from another table. A foreign key constraint, also known as referential integrity constraint, specifies that the values of the foreign key correspond to actual values of the primary or unique key in other table. One can refer to a primary or unique key in the same table also.

Syntax:

ALTER TABLE <table_name>
ADD CONSTRAINT <constraint_name> FOREIGN KEY(<column_name(s)>)
REFERENCES <base_table>(<column_name>) ON {DELETE | UPDATE}
CASCADE;

Q.50 Define the following constraints for the table client of Q.37 **(6)**

(i) BAL_DUE must be at least 1000.
(ii) NAME is a unique key.

Ans: (i)ALTER TABLE CLIENT
ADD CONSTRAINT CLIENT_BAL_DUE_C1 CHECK(BAL_DUE < 1000);

(ii) ALTER TABLE CLIENT
ADD CONSTRAINT CLIENT_NAME_U UNIQUE(NAME);

Q.51 What are the different types of database end users? Discuss the main activities of each. **(7)**

Ans:

End-Users – End-users are the people whose jobs require access to the database for querying, updating, and generating reports; the database primarily exists for their use. The different types of end-users are:

- **Casual end-users –** occasionally access the database, need different information each time
- **Naive or Parametric end-users –** includes tellers, clerks, etc., make up a sizable portion of database end-users, main job function revolves around constantly querying and updating the database

- **Sophisticated end-users –** includes engineers, scientists, business analyst, etc., use for their complex requirements
- **Stand-alone users –** maintain personal databases by using ready-made program packages, provide easy-to-use menu-based or graphics-based interfaces

Q.52 Discuss the typical user friendly interfaces and the types of users who use each. **(7)**

Ans:

User-friendly interfaces provided by a DBMS may include the following:

- **Menu-Based Interfaces for Web Clients or Browsing –** These interfaces present the user with lists of options, called menus, that lead the user through the formulation of a request. Pull-down menus are a very popular techniques in Web-based user interfaces. They are also used in browsing interfaces, which allow a user to look through the contents of a database in an exploratory and unstructured manner.
- **Forms-Based Interfaces –** A forms-based interface displays a form to each user. Forms are usually designed and programmed for naive users and interfaces to canned transactions. Many DBMSs have forms specification languages.
- **Graphical User Interfaces (GUIs) –** A GUI typically displays a schema to the user in diagrammatic form. The user can then specify a query by manipulating the diagram. In many cases, GUIs utilizes both menus and forms. Most GUIs use a pointing device to pick certain parts of the displayed schema diagram.
- **Natural Language Interfaces –** These interfaces accept requests written in English or some other language and attempt to "understand" them. A natural language interface usually has its own "schema," which is similar to the database conceptual schema, as well as a dictionary of important words.
- **Interfaces for Parametric Users –** Parametric users, such as bank tellers, often have a small set of operations that they must perform repeatedly. The interfaces for these users usually have a small set of abbreviated commands with the goal of minimizing the number of keystrokes required for each request.
- **Interfaces for the DBA –** Most database systems contain privileged commands that can be used only by the DBA's staff. These include commands for creating accounts, setting system parameters, granting account authorization, changing a schema, and a reorganizing the storage structures of a database.

Q.53 With the help of an example show how records can be deleted and updated in QBE. **(5)**

(i) Increase Pay_Rate of employees with the skill of 'cook' by 10%.

(ii) Delete employee record for EMP# 123459

Ans: (i)

EMPLOYEE	Emp#	Name	Skill	Pay_Rate
	EX		Cook	PX
U.	EX			PX * 1.1

(ii)

EMPLOYEE	Emp#	Name	Skill	Pay_Rate
D.	123459			

Q.54 Describe cardinality ratios and participation constraints for relationship types. **(4)**

Ans:

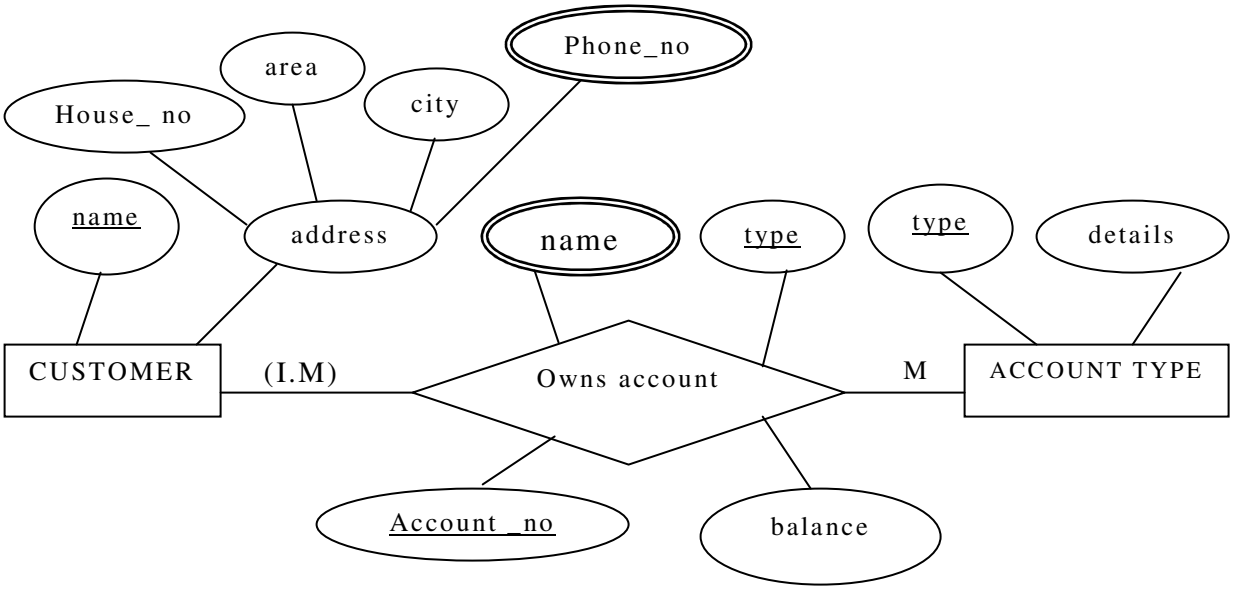
Cardinality Ratios – The cardinality ratios for a relationship type specifies the *maximum* number of relationship instances that an entity can participate in. The possible cardinality ratios for relationship types are one-to-one (1:1), one-to-many or many-to-one (1:M or M:1), and many-to-many (M:N).

Participation Constraints – The participation constraint specifies whether the existence of an entity depends on its being related to another entity via the relationship type. This constraint specifies the *minimum* number of relationship instances that each entity can participate in. It is sometimes called the minimum cardinality constraint. There are two types of participation constraints – total and partial.

Q.55 Information about a bank is about customers and their account. Customer has a name, address which consists of house number, area and city, and one or more phone numbers. Account has number, type and balance. We need to record customers who own an account. Account can be held individually or jointly. An account cannot exist without a customer.

Arrive at an E-R diagram. Clearly indicate attributes, keys, the cardinality ratios and participation constraints. **(10)**

Ans:

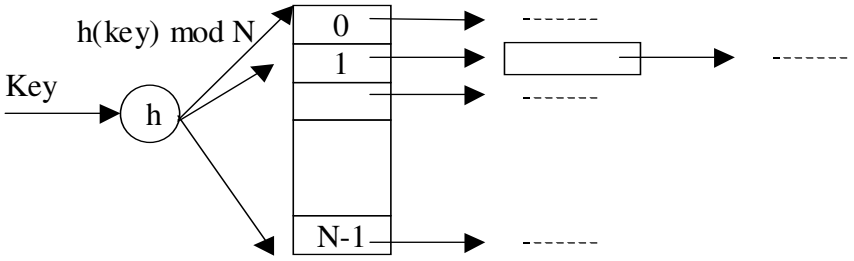


Q.56 Describe the static hash file with buckets and chaining and show how insertion, deletion and modification of a record can be performed. **(9)**

Ans:

In static hash file organization, the term bucket is used to denote a unit storage that can store one or more records.A file consists of buckets 0 through N-1, with one primary page per bucket initially and additional overflow pages chained with bucket, if required later. Buckets contain data entries (or data records). In hashing scheme, a hash function, h, is performed on the key of the record to identify the bucket to which data record belongs to. The hash function is an important component of the hashing approach. The main problem with static hash file is that the number of buckets is fixed.

Insertion of a record – To insert a data entry, the hash function is used to identify the



Primary Bucket Pages Overflow Pages

Static Hash File

correct bucket and then put the data entry there. If there is no space for this data entry, a

new overflow page will be allocated, put the data entry on this page, and the page to the overflow chain of the bucket.

Deletion of a record – To delete a data entry, the hash function is used to identify the correct bucket, locate the data entry by searching the bucket, and then remove it. If the data entry is the last in an overflow page, the overflow page is removed from the overflow chain of the bucket and added to a list of free pages.

Modification of a record – To modify a data entry, the hash function is used to identify the correct bucket, locate the data entry by searching the bucket and get it, modify the data entry, and then rewrite the modified data entry on it.

Q.57 What are the reasons for having variable length records? What types of separator characters are needed for each? **(5)**

Ans:

Variable-Length Records – Variable-length records are those records, which are of different sizes. A file may contain variable-length records in any of the following situations:

- **Records having variable length fields –** In this case, the end-of-field symbol along with end-of-record symbol can be used as the separator characters.
- **Records having repeating fields** – In this case, the number of repetitions of repeating fields can be used with end-of-record symbol as the separator characters.
- **Records having optional fields –** In this case, the end-of-field symbol along with end-of-record symbol can be used as the separator characters.
- **File containing records of different record types -** In this case, a special field (or control field) can be prefixed with each record and end-of-record symbol can be used as the separator characters.

In each of the above situations either we can use end-of-record symbol or the length of each record can be prefixed at the beginning of the record for the separation of the variable-length records.

Q.58 Define the following terms **(7 x 2 = 14)**

(i) Derived and stored attribute.

(ii) Distributed system.
(iii) Interblock gap
(iv) Degree of a relation.
(v) Catalog
(vi) Conceptual schema
(vii) DDL and SDL.

Ans: (i) Derived and Stored Attribute - In some cases, two or more attribute values are related, for example, Age and BirthDate attributes of a person. For particular person entity, the value of Age can be determined from the current date and the value of that person's BirthDate. Hence, the attribute Age is called as derived attribute and the attribute BirthDate is called as stored attribute.
(ii) Distributed System – A distributed system consists of a number of processing elements that are interconnected by a computer network and that cooperate in performing certain assigned tasks.
(iii) Interblock Gap – A track of a disk is divided into equal-sized disk blocks. Blocks are separated by fixed-size gaps, called as interblock gaps, which include specially coded control information written during disk initialization.
(iv) Degree of a Relation – The degree or arity of a relation is the number of attributes *n* of its relation schema.
(v) Catalog – A relational DBMS maintains information about every table and index that it contains. A catalog is a collection of special tables, which stores the descriptive information of every table and index.
(vi) Conceptual Schema – Conceptual schema describes the structure of the whole database for a community of users. It hides the details of physical storage structures and concentrates on describing entities, data types, relationships, and constraints.
(vii) DDL and SDL – The data definition language (DDL) is used by DBA and database designers to define conceptual schema, internal schema, and mappings between these two. In some DBMSs, a clear separation is maintained between conceptual schema and internal schema. In that case, DDL is used to specify the conceptual schema only. Another language, storage definition language (SDL) is used to specify the internal schema. The mappings between the two schemas may be specified in either one of these languages.

Q.59 Define a relation. **(2)**

Ans: Relation – A relations is a named two-dimensional table of data. Mathematically, a relation can be defined as a subset of the cartesian product of a list of domains. Each relation consists of a set of named columns and an arbitrary number of rows. The columns correspond to the fields describing each tuple in the table or relation. The rows correspond to each instance of the entity described by the table or relation.

Q.60 Describe entity integrity and referential integrity. Give an example of each. **(6)**

Ans:
Entity Integrity Rule – If the attribute A of relation R is a prime attribute of R then A cannot accept null values.

Referential Integrity Rule – In referential integrity, it is ensured that a value that appears in one relation for a given set of attributes also appears for a certain set of attributes in another relation.
For example:

STUDENT

Enrl No	**Roll No**	**Name**	**City**	***Mobile***
11	17	Ankit Vats	Delhi	9891663808
15	16	Vivek Rajput	Meerut	9891468487
6	6	Vanita	Punjab	
33	75	Bhavya	Delhi	9810618396

GRADE

Roll No	**Course**	***Grade***
6	C	A
17	VB	C
75	VB	A
6	DBMS	B
16	C	B

- **Roll No** is the primary key in the relation STUDENT and Roll No + Course is the primary key of the relation GRADE. (Entity Integrity)
- **Roll No** in the relation GRADE (child table) is a foreign key, which is referenced from the relation STUDENT (parent table). (Referential Integrity).

Q.61 Consider the two relations given below

R

A	B	C
A1	b1	c1
Null	b2	null
a1	b1	c1

S

D	A	F
d1	a1	f1
d1	a2	null

Given that A is the primary key of R, D is the primary key of S and there is a referential integrity between S.A and R.A, discuss all integrity constraints that are violated. **(6)**

Ans: (i) Primary key of R contains the 'null' value and the value 'a1' is duplicated, hence it violates the entity integrity constraint in the relation R.

(ii) In primary key of S, the value 'd1' is duplicated, hence it violates the entity integrity constraint in the relation S.

(iii) The foreign key S.A contains the value 'a2', which is not available in the parent key R.A, hence it violates the referential integrity constraint in the relation S.

Q.62 Given the following relations
TRAIN (NAME, START, DEST)
TICKET (PNRNO., START, DEST, FARE)
PASSENGER (NAME, ADDRESS, PNRNO.)
Write SQL expressions for the following queries: **(3.5 x 4 = 14)**
Note: Assume NAME of Train is a column of Ticket.

(i) List the names of passengers who are travelling from the start to the destination station of the train.
(ii) List the names of passengers who have a return journey ticket.
(iii) Insert a new Shatabti train from Delhi to Bangalore.
(iv) Cancel the ticket of Tintin.

Ans

(i)
```
SELECT P.NAME FROM TRAIN T, TICKET I, PASSENGER P
WHERE P.PNRNO = I.PNRNO AND T.NAME = I.NAME
AND T.START = I.START AND T.DEST = I.DEST
```
(ii)
```
SELECT NAME FROM PASSENGER
    WHERE PNRNO IN (SELECT DISTINCT A.PNRNO
    FROM TICKET A, TICKET B WHERE A.PNRNO = B.PNRNO
    AND A.START = B.DEST AND A.DEST = B.START)
```
(iii)
```
INSERT INTO TRAIN
    VALUES('Shatabdi', 'Delhi', 'Banglore'
```
(iv)
```
DELETE FROM TICKET
    WHERE PNRNO = (SELECT PNRNO FROM PASSENGER
    WHERE NAME = 'Tintin')
```

Q.63 Define outer union operation of the relational algebra. Compute the outer union for the relations R and S given below. **(2 x 3)**

R

A	B	C
a1	b1	c1
a3	b2	c2

S

D	A	F
d1	a1	f1
d1	a2	null

Ans:
Outer Join - If there are any values in one table that do not have corresponding value(s) in the other, in an equi-join that will not be selected. Such rows can be forcefully selected by using the *outer join. The corresponding columns for that row will have NULLs.* There are actually three forms of the outer-join operation: left outer join (⟕), right outer join (⟖) and full outer join (⟗).

R.A	**B**	**C**	**D**	**S.A**	**F**
a1	b1	c1	d1	a1	f1
a3	b2	c2	Null	Null	Null
Null	Null	Null	d1	a2	Null

Q.64 Given the following relations **(3 x 3)**
Vehicle (Reg_no, make, colour)
Person(eno, name, address)
Owner(eno, reg_no)
Write expressions in the relational algebra to answer the following queries:-
(i) List the reg_no of vehicles owned by John.

(ii) List the names of persons who own maruti cars.
(iii) List all the red coloured vehicle.

Ans: (i) $\pi_{reg\_no}(\sigma_{name='John'}(PERSON \triangleright\triangleleft OWNER))$
(ii) $\pi_{name}(\sigma_{make='maruti'}(PERSON \triangleright\triangleleft OWNER \triangleright\triangleleft VEHICAL))$
(iii) $\sigma_{colour='red'}(VEHICAL)$

Q.65 Describe DROP TABLE command of SQL with both the options – CASCADE and RESTRICT. **(5)**

Ans:
DROP TABLE command – This command drops the specified table. Dropping a table also drops indexes and grants associated with it. Objects built on dropped tables are marked invalid and cease to work.
CASCADE and RESTRICT options – The DROP TABLE with RESTRICT option destroy the table unless some view or integrity constraint refer to the given table; if so, the command fails. But with the CASCADE option, any referencing views or integrity constraints are (recursively) dropped as well.

Q.66 With respect to Oracle describe the following: **(3 x 3)**

(i) Data Block.

(ii) Data dictionary.

(iii) Segments.

Ans:Data Block – In oracle the data blocks are referred to as tablespaces. A tablespace is an area of disk consisting of one or more disk files. A tablespace can contain many tables, indexes, or clusters.
(ii) Data Dictionary – Data dictionaries are the system tables that contain descriptions of the database objects and how they are structured.
(iii) Segments – Each table has single area of disk space, called segment, set aside for it in the tablespace. Segments consists of contiguous sections called extents.

Q.67 Describe the responsibilities of the DBA and the database designer. **(7)**

Ans: The responsibilities of DBA and database designer are:

1. Planning for the database's future storage requirements
2. Defining database availability and fault management architecture
3. Defining and creating environments for development and new release installation
4. Creating physical database storage structures after developers have designed an application
5. Constructing the database
6. Determining and setting the size and physical locations of data files
7. Evaluating new hardware and software purchase
8. Researching, testing, and recommending tools for Oracle development, modeling, database administration, and backup and recovery implementation, as well as planning for the future
9. Providing database design and implementation

10. Understanding and employing the optimal flexible architecture to ease administration, allow flexibility in managing I/O, and to increase the capability to scale the system

Q.68 What are the four main characteristics of the database approach? **(7)**

Ans: The four main characteristics of the database approach are:

1. Self-describing nature of a database system.
2. Insulation between programs and data, and data abstraction.
3. Support of multiple views of the data.
4. Sharing of data and multi-user transaction processing.

Q.69 Differentiate between DDL and DML. **(4)**

Ans: **DDL - Data Definition Language:** statements used to define the database structure or schema. Some examples:

- CREATE - to create objects in the database
- ALTER - alters the structure of the database
- DROP - delete objects from the database
- TRUNCATE - remove all records from a table, including all spaces allocated for the records are removed
- COMMENT - add comments to the data dictionary
- RENAME - rename an object

DML - Data Manipulation Language: statements used for managing data within schema objects. Some examples:

- SELECT - retrieve data from the a database
- INSERT - insert data into a table
- UPDATE - updates existing data within a table
- DELETE - deletes all records from a table, the space for the records remain
- MERGE - UPSERT operation (insert or update)
- CALL - call a PL/SQL or Java subprogram
- EXPLAIN PLAN - explain access path to data

Q.70 List any two disadvantages of a database system. **(3)**

Ans: **The disadvantages of database system are:**

- Database systems are complex, difficult, and time-consuming to design.
- Substantial hardware and software start-up costs.
- Damage to database affects virtually all applications programs.
- Extensive conversion costs in moving form a file-based system to a database system.
- Initial training required for all programmers and users.

Q.71 **Explain the utilities that help the DBA to manage the database.** **(7)**

Ans: Every DBA uses database utilities to manage and control their databases. But there is a lot of confusion in the field as to what, exactly, is a database utility. There are a

lot of definitions floating around out there. DBAs constantly refer to utilities, tools, solutions, and suites.

So, first of all, let's be clear on what a utility is and what is a "tool" or "solution." A utility is generally a single purpose program for moving and/or verifying database pages; examples include LOAD, UNLOAD, REORG, CHECK, COPY, and RECOVER. A database tool is a multi-functioned program designed to simplify database monitoring, management, and/or administrative tasks. A solution is a synergistic group of tools and utilities designed to work together to address a customer's business issue. A suite is a group of tools that are sold together, but are not necessarily integrated to work with each other in any way. Of course, these are just my definitions. But there are useful definitions that make it easier to discuss DBA products and programs

Q.72 Differentiate between

(i) WHERE and HAVING clause in SQL.
(ii) Strong entity set and weak entity set.
(iii) Spanned and unspanned organisation. **(3.5 x 3)**

Ans: (i) WHERE and HAVING clause in SQL

The WHERE clause is basically used for implementing conditions on every tuple of the relation.

The HAVING clause is used in combination with the GROUP BY clause. It can be used in a SELECT statement to filter groups of the records that a GROUP BY returns.

The syntax for the HAVING clause is:

SELECT column1, column2, ... column_n, aggregate_function (expression)
FROM tables
WHERE predicates
GROUP BY column1, column2, ... column_n
HAVING condition1 ... condition_n;

Aggregate_function can be a function such as SUM, COUNT, MIN, or MAX.

(ii) Strong entity set and weak entity set: A strong entity set has a primary key. All tuples in the set are distinguishable by that key. A weak entity set has no primary key unless attributes of the strong entity set on which it depends are included. Tuples in a weak entity set are partitioned according to their relationship with tuples in a strong entity set. Tuples within each partition are distinguishable by a discriminator, which is a set of attributes. A strong entity set has a primary key. All tuples in the set are distinguishable by that key. A weak entity set has no primary key unless attributes of the strong entity set on which it depends are included. Tuples in a weak entity set are partitioned according to their relationship with tuples in a strong entity set. Tuples within each partition are distinguishable by a discriminator, which is a set of attributes.

(iii) Spanned and unspanned organization: If records are not allowed to cross block boundaries then the organisation **is** called unspanned *record organisation* otherwise it is called spanned *record organisation*

Q.73 Discuss with examples about various types of attributes present in the ER model. **(6)**

Ans: Types of Attributes are:

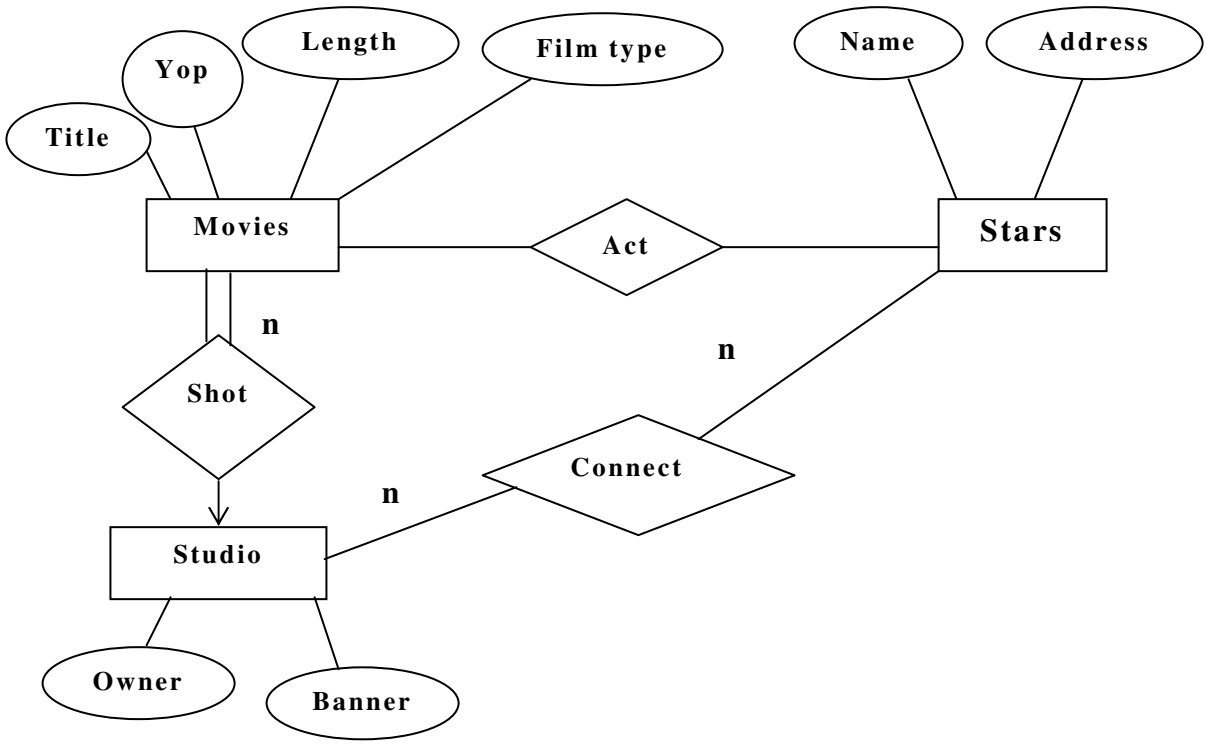
- SIMPLE attributes are attributes that are drawn from the atomic value domains

E.g. Name = {John} ; Age = {23}

- COMPOSITE attributes: Attributes that consist of a hierarchy of attributes
  E.g. Address may consists of "Number", "Street" and "Suburb" → Address = {59 + 'Meek Street' + 'Kingsford'}
- SINGLE VALUED attributes: Attributes that have only one value for each entity
  E.g. Name, Age for EMPLOYEE
- MULTIVALUED attributes: Attributes that have a set of values for each entity
  E.g. Degrees of a person: ' BSc' , 'MIT', 'PhD'
- DERIVED attributes: Attributes Contain values that are calculated from other attributes
  Eg. Age can be derived from attribute DateOfBirth. In this situation, DateOfBirth might be called Stored Attribute

Q.74 Information about films contains information about movies, stars and studios. Movies have a title, year of production, length and the film type. Stars have a name and address. Studios have a owner and a banner. Movies are shot in studios which own them. A movie is shot in only one studio. Stars are connected to one or more studios but can act in any film which may or may not be owned by the studio. Arrive at an E-R diagram. Clearly indicate attributes, keys, the cardinality ratios and participation constraints. **(8)**

Ans:



Q.75 What is the main goal of RAID technology? Describe the levels 1 through 5. (6)

Ans: **RAID** stands for **R**edundant **A**rray of **I**nexpensive (or sometimes "Independent")**D**isks.
RAID is a method of combining several hard disk drives into one logical unit (two or

more disks grouped together to appear as a single device to the host system). RAID technology was developed to address the fault-tolerance and performance limitations of conventional disk storage. It can offer fault tolerance and higher throughput levels than a single hard drive or group of independent hard drives. While arrays were once considered complex and relatively specialized storage solutions, today they are easy to use and essential for a broad spectrum of client/server applications.

RAID 1 : Mirrored set without parity. Provides fault tolerance from disk errors and failure of all but one of the drives. Increased read performance occurs when using a multi-threaded operating system that supports split seeks, very small performance reduction when writing. Array continues to operate so long as at least one drive is functioning. Using RAID 1 with a separate controller for each disk is sometimes called *duplexing*. SNIA definition.

RAID 2: Redundancy through Hamming code. Disks are synchronised and striped in very small stripes, often in single bytes/words. Hamming codes error correction is calculated across corresponding bits on disks, and is stored on multiple parity disks. SNIA definition.

RAID 3: Striped set with dedicated parity/Bit interleaved parity. This mechanism provides an improved performance and fault tolerance similar to RAID 5, but with a dedicated parity disk rather than rotated parity stripes. The single parity disk is a bottle-neck for writing since every write requires updating the parity data. One minor benefit is the dedicated parity disk allows the parity drive to fail and operation will continue without parity or performance penalty.

RAID 4: Block level parity. Identical to RAID 3, but does block-level striping instead of byte-level striping. In this setup, files can be distributed between multiple disks. Each disk operates independently which allows I/O requests to be performed in parallel, though data transfer speeds can suffer due to the type of parity. The error detection is achieved through dedicated parity and is stored in a separate, single disk unit.

RAID 5: It distributes data and parity information across all disks.

Q.76 An employee record has the following structure

```
struct employee {
    int     eno;
    char    name[22];
    float   salary;
    char    dept[10];};
```

(i) Calculate the record size R in bytes.

(ii) If the file has 500 records, calculate the blocking factor bfr and the number of blocks b, assuming an unspanned organization with block size B = 512 bytes.

(iii) What is the unused space in each block and in the last block? **(8)**

Ans: (i) 2 + 22+ 4 + 10 =38 bytes

(ii) bfr = floor(B/R) = [512/38] =13

Number of blocks b = ceiling (500/13) =39

(iii) Unused space in each block

= 512 – 38 *13 = 512 – 494= 18 bytes

Unused space in the last block

= (39 *13 – 500) * 38 bytes = (507 -500) * 38 bytes

= 7 * 38 =266 bytes

Q.77 Define the following terms

(i) Hashing **(ii)** Specialization
(iii) Value set. **(iv)** DBMS.
(v) Host language. **(vi)** Database state.
(vii) Trigger. **(2 x 7)**

Ans: (i) Hashing: Hashing is a method to store data in an array so that storing, searching, inserting and deleting data is fast (in theory it's O(1)). For this every record needs an unique key.

The basic idea is not to search for the correct position of a record with comparisons but to compute the position within the array. The function that returns the position is called the 'hash function' and the array is called a 'hash table'.

(ii) Specialization: Specialization allows you to define new kinds of information (new structural types or new domains of information), while reusing as much of existing design and code as possible, and minimizing or eliminating the costs of interchange, migration, and maintenance.

(iii) Value set: It is a set of values. You do not always want a user to enter junk free text into all the fields. Hence, Oracle Apps uses value set to validate that correct data is being entered in the fields in screen.

(iv) DBMS: database management system (DBMS) is computer software designed for the purpose of managing databases based on a variety of data models.

(v) Host Language: You can write applications with SQL statements embedded within a host language. The SQL statements provide the database interface, while the host language provides the remaining support needed for the application to execute.

(vi) Database state: The actual data in the database at a particular moment in time is called a database state.

(vii) Trigger: A database trigger is procedural code that is automatically executed in response to certain events on a particular table in a database. Triggers can restrict access to specific data, perform logging, or audit data modifications.

Q.78 Differentiate between natural join and outer join. **(2)**

Ans: Natural join is a binary operator that is written as (*R* * *S*) where *R* and *S* are relations. The result of the natural join is the set of all combinations of tuples in *R* and *S* that are equal on their common attribute names. In this only one column out of columns having same name attributes is retained.

An **Outer join** contains those tuples and additionally some tuples formed by extending an unmatched tuple in one of the operands by "fill" values for each of the attributes of the other operand.

Q.79 For the relations R and S given below:

R

A	B	C
1	2	3
4	5	6
7	8	9

S

B	C	D
2	3	10
2	3	11
6	7	12

Compute

(i) $\Pi_{A,C}(R)$ **(ii)** $\sigma_{B=2}(S)$

(iii) natural join **(iv)** outer join **(12)**

Ans: (i)

A	C
1	3
4	6
7	9

(ii)

B	C	D
2	3	10
2	3	11

(iii)

A	B	C	D
1	2	3	10
1	2	3	11

(iv) Assuming left outer join

A	B	C	D
1	2	3	10
1	2	3	11
4	5	6	NULL
7	8	9	NULL

Q.80 Define union compatibility? Explain why INTERSECTION of two relations can not be performed if they are not union compatible? **(4)**

Ans: Two relations R(A1, A2,-------, A_n) and S(B1, B2,-----, B_n) are said t union compatible if they have the same degree n and if dom (A_i) = dom(B_i) for all i=1---n
Intersection requires two relations to be union compatible since otherwise it is not possible to formulate the criterion as to on what basis will attributes be considered as common.

Q.81 What is a view in SQL? When can views be updated? **(4)**

Ans: A view is a virtual table that consists of columns from one or more tables. Though it is similar to a table, it is stored in the database. It is a query stored as an object. Hence, a view is an object that derives its data from one or more tables. These tables are referred to as base or underlying tables.
Views can be updated if they are defined on a single table without any aggregate functions can be mapped to an update on the underlying base table under certain conditions**.**

Q.82 Using SQL create a view RS for the relations R and S of Q79. The view consists of the columns A and D renamed as X and Y respectively. Insert a tuple <10, 15> into it. Show the contents of the view. **(6)**

Ans: create view RS as select A as X , D as Y from Q79;
insert into RS values (10, 15);

Q.83 Consider the relations defined below:
PHYSICIAN (regno, name, telno, city)

PATIENT (pname, street, city)
VISIT (pname, regno, date_of_visit, fee)
Where the regno and pname identify the physician and the patient uniquely respectively. Express queries (i) to (iii) in SQL.

(i) Get the name and regno of physicians who are in Delhi.

(ii) Find the name and city of patient(s) who visited a physician on 31 August 2004.

(iii) Get the name of the physician and the total number of patients who have visited her.

(iv) What does the following SQL query answer

```
SELECT DISTINCT name
    FROM PHYSICIAN P
    WHERE NOT EXISTS
        ( SELECT *
          FROM VISIT
          WHERE regno = p.regno )
```

(3.5 x 4)

Ans: **(i)** Select name, regno from PHYSICIAN where city = 'Delhi';

(ii) Select pname, city from PATIENT,VISIT where PATIENT.pname=VISIT.pname and date_of_visit = '31-Aug-04';

(iii) select name, count(*) from PHYSICIAN, VISIT where PHYSICIAN.regno = VISIT.regno group by Physician . regno;

(iv) This will give the name of physicians who have not visited any patient.

Q.84 Write short notes on

(i) Data models.

(ii) Oracle database structure.

(iii) Group By clause in SQL.

(iv) Retrieval in QBE. **(3.5 x 4)**

Ans: (i) Data models: A data model is an abstract model that describes how data is represented and accessed.

The term data model has two generally accepted meanings:

A data model *theory*, i.e. a formal description of how data may be structured and accessed.

A data model *instance*, i.e. applying a data model *theory* to create a practical data model *instance* for some particular application.

(ii) Oracle database structure:

The relational model has three major aspects:

Structures: Structures are well-defined objects that store the data of a database. Structures and the data contained within them can be manipulated by operations.

Operations: Operations are clearly defined actions that allow users to manipulate the data and structures of a database. The operations on a database must adhere to a pre-defined set of integrity rules.

Integrity Rule: Integrity rules are the laws that govern which operations are allowed on the data and structures of a database. Integrity rules protect the data and the structures of a database.

An ORACLE database has both a physical and a logical structure. By separating physical and logical database structure, the physical storage of data can be managed without affecting the access to logical storage structures.

(iii) Group By clause in SQL: The GROUP BY clause can be used in a SELECT statement to collect data across multiple records and group the results by one or more columns.

The syntax for the GROUP BY clause is:

SELECT column1, column2, ... column_n, aggregate_function (expression)
FROM tables
WHERE predicates
GROUP BY column1, column2, ... column_n;

aggregate_function can be a function such as SUM, COUNT, MIN, or MAX.

(iv) Retrieval in QBE : A "GUI" for expressing queries.

Based on the Domain Relational Calculus (DRC)
Actually invented before GUIs.
Very convenient for simple queries.
Awkward for complex queries.
QBE and IBM trademark.
But has influenced many projects
Especially PC Databases: Paradox, Access, etc.

Q.85 Explain the disadvantages of file oriented approach. **(6)**

Ans: Applications are designed in isolation. Design of application is optimized for one application. Independently developed applications leads to data redundancy. Wasted storage space due to redundancy.

a. There is loss of data integrity because integrity checking is not automated.
b. Difficulty in accessing data.
c. Information is available only in reports.
d. Data Isolation.
e. Inadequate security
f. Limited Flexibility
g. High Maintenance cost.
h. Each data file of an application is a separate entity.

Q.86 Explain the three data models namely relational, network and hierarchical and compare their relative advantages and disadvantages. **(10)**

Ans: Hierarchical Model: In hierarchical model, data elements are connected to one another through links. Records are arranged in a top-down structure that resembles a tree or genealogy chart. The top node is called the root, the bottom nodes are called leaves, and intermediate nodes have one parent node and several child nodes. The root can have any number of child nodes but a child node can have only one parent node. Data are related in a nested, one-to-many set of relationships, while many-to-many relationship cannot be directly expressed.

A child record occurrence must have a parent record occurrence; deleting a parent record occurrence requires deleting all its child record occurrences.

A network data model can be regarded as an extended form of the hierarchical model; the principle distinction between the two being that in a hierarchical model, a child record

has exactly one parent whereas in network model, a child record can have any number of parents. It may have zero also.

Data in the network model is represented by collection of records and relationship among data is represented by links, which can be viewed, as pointers. The records in the database are organized as collection of arbitrary graphs, which allows to have one-to-many as well as many-to-many relationship is a collection of data items which can be retrieved from a database, or which can be stored in a database as an undivided object. Thus, A DBMS may STORE, DELETE or MODIFY records within a database. In this way, a number of records within a network database are dynamically changed. The network model can be graphically represented as follows:

A labeled rectangle represents the corresponding entity or record type. An arrow represents the set type, which denotes the relationship between the owner record type and member record. The arrow direction is from the owner record type to the member record type.

A labeled rectangle represents the corresponding entity or record type. An arrow represents the set type, which denotes the relationship between the owner record type and member record. The arrow direction is from the owner record type to the member record type.

Each many to many relationship is handled by introducing a new record type to represent the relationship wherein the attributes, if any, of the relationship are stored. We when create two symmetrical 1:M sets with the member in each of the sets being the newly introduced record type. In this model, the relationships as well as the navigation through the database are predefined at database creation time.

In relational model the data and the relations among them are represented by a collection of tables. A tables is a collection of records and each record in a table contains the same fields. The attractiveness of the relational approach arouses from the simplicity in the data organization and the availability of ably simple to very powerful query languages. The relational model is based on a technique called “Normalization” proposed by E.F. Codd. This model reduces the complexity of the Network and Hierarchical Models. This model uses the certain mathematical operations from relational algebra and relational calculus on the relation such as projection, union and joins etc. where fields in two different tables take values from the same set, a join operation can be performed to select related records in the two tables by matching values in those fields. A description of data in terms of a data model is called a schema. In relation model, the schema for a relation specifies its name, the name of each field and the type of each field.

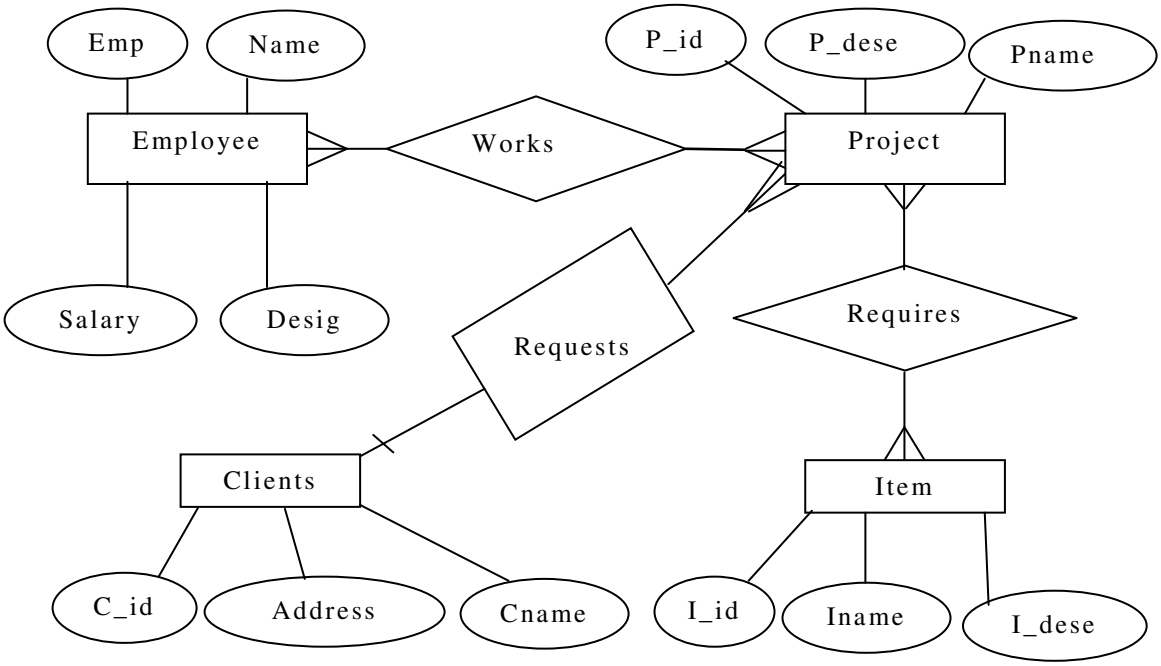
Navigation through relations the represent an M:N relationship is just as simple as through a 1:M relationship. This leads us to conclude that it is easier to specify how to manipulate a relational database than a network or hierarchical one. This in turn leads to a query language for the relational model that is correct, clear, and effective in specifying the required operations. Unfortunately, the join operation is inherently inefficient and demands a considerable amount of processing and retrieval of unnecessary data. The structure for the network and hierarchical model can be implemented efficiently. Such an implementation would mean that navigating through these databases, though awkward, requires the retrieval of relatively little unnecessary data.

Q.87 In an organization several projects are undertaken. Each projects can employ one or more employees. Each employee can work on one or more projects. Each project is undertaken on the required of client. A client can request for several projects. Each

project have only one client. A project can use a number of items and a item may be used by several projects. Draw an E-R diagram and convert it to a relational schema. **(10)**

Ans: The ER Diagram can be converted into the following relational schema.
Entity Sets

- **(i)** Employee (Empno, Name, Salary, Desig)
- **(ii)** Project (P_id, Pname, P-desc)
- **(iii)** Item (I_id, Iname, I-desc)
- **(iv)** Clients (C_id,Cname, Address)
- **(v)** Works_on (Empno, P_id)
- **(vi)** Requests (P_id, C_id)
- **(vii)** Requires (P_id, I_id)



Q.88 Define : (i) Identifying relationship (ii) Specialisation / generalization (iii) Aggregation. **(6)**

Ans: (i) Identifying relationship : It is relationship between strong entity and a weak entity it is represented by a doubly outlined diamond in Chen notation.

(ii) Specialisation /generalization : This is a top down / bottom up approach to the design of database. It shows the IS_A relationship between a super class (parent) and associated sub-classes. It is represented by inverted triangle in the Chen notation.

(iii) Aggregation : This is used whenever we intend to show a relationship between an entity and the relationship. There is no provision for this in the ER Model. This is shown by placing the relationship between two entities alongwith the entities in a box and setting up a relationship between this box and another entity.

Q.89 A well-maintained relational DBMS has a high level of data integrity. What features of a relational DBMS contribute towards this level of integrity? **(6)**

Ans: Relational DBMS provides high level of data integrity, by using the following rules:

Entity Integrity: Entity integrity says that a prime attribute in a relation can not accept null values.

Referential Integrity: The referential integrity rule is related with the foreign key concept. Let R1 and R2 are two relations where R1 is having an attribute(s) with primary key. Let R2 be having a foreign key, which refers to relation R1 via the same set of attributes. Then the value of the foreign key in a tuple in R2 relation must either be equal to primary key of a tuple in a relation R1 relation or be entirely NULL.

In a more general way, the referential integrity rule states that every foreign key value must match a primary key value in an associated table. Referential integrity ensures that we can correctly navigate between related entities> Referential Integrity is a state where each foreign key value has matching primary key value. In other words, it guarantees that data in a dependent table has matching records in the table(s) on which it depends. With modern databases, referential integrity is enforced with the use of primary and foreign keys, which will not allow insertions, deletions, or changes to data that violate the rules of integrity configured by the database administrator.

Student_id	X (6)
Subject_code	X(6)
Semester_code	X(6)
Subject_name	X(20)
Lecturer_id	X(8)
Lecturer-name	X(20)
Grade	X(6)

PRIMARY KEY (student_id, subject_code)
FOREIGN KEY (subject_code) REFERENCES TABLE subjects.
The student_grade table stores the information of the grade obtained by all the students in their respective subjects taught by a lecturer in a semester. The design of this table is not normalized and it would lead to various anomalies.

Every time a grade for a student is inserted for some subject taught by a lecturer, subject_name and lecturer_name have also to be inserted. So these field will be repeated for all the students for a particular subject taught by a teacher. For example if there are five students who studied Mathematics subject taught by Mr. Ashok then these two values along with their Ids will be repeated in the relation. This will lead to redundancy. If a change is required in subject_name then it will be reqired to modify the data in both the tables student_grade and subjects, otherwise leading to a **inconsistent** state. For example if we wish to change the subject name of Maths to Mathematics in subjects table then this change has to be made in student_grade also.

If we declare a record in student_grade table, which have only one entry for a particular lecturer, then it will loose the information of that lecturer. Moreover, if a new lecturer joins, his/her information can not be inserted in the relation unless he/she teaches a subject to the students of any semester.

Q.90 Explain the relevance of Data Dictionary in a Database System. **(10)**

Ans: Data dictionary is a database in its own right residing on the disk which consist of Meta data which is = Data about all entity sets + attributes + relationships among entity sets + constraints. It consist of compiled form of definitions, structure and usage information on data stored, design decisions, usage standards, application programme descriptions, user information. It is consulted by DBMS before DML operation and by user to learn what each piece of data and various synonymous of data fields mean. Data dictionary can be integrated system where it is part of DBMS or add ons to DBMS. In integrated system data dictionary contains information concerning external, conceptual and internal level of data base. Both in source and object form. It contains source code of each data field value, frequency of its use, audit trail concerning updates and cross reference information. Present system are all add ons standards do not exist for iintegrity data dictionary with DBMS. Data dictionary should be integrated in database it defines and thus include its own definition so that it can be queried with the same language use for queering database.

Q.91 Differentiate between **(3 x 3)**

(i) Procedural and non procedural languages.
(ii) Key and superkey
(iii) Primary and secondary storage

Ans: (i) Procedural and non procedural languages - A procedural language specifies the operations to be performed on the existing data to derive the results. It also specifies the sequence of operations in which they will be performed. But, a non procedural language specifies only the result or information required not how it is obtained.

(ii) Key and superkey - A key a single attribute or a combination of two or more attributes of an entity set that is used to identify one or more instances (rows) of the set (table). If we add some additional attributes to a primary key then that *augmented key* is called as super key. Therefore, *the primary key is the minimum super key*.

(iii) Primary and secondary storage – Primary storage device stores the data temporarily. Primary storage is generally used by the processing unit to temporary store the data, intermediate results, and the final results before storing to the secondary storage because the secondary storage devices are not directly accessible by the CPU. But, if we want to store data permanently then the secondary storage devices are required. Secondary storage devices are slower than the primary storage devices.

Q.92 Consider the following relations:
BRANCH(bno, street, area, city, pcode, Tel_no, Fax_no)
STAFF(Sno, Fname, Lname, address, position , salary, bno)
Express the following queries in SQL:
(i) List the staff who work in the branch at '163 main street'
(ii) Find staff whose salary is larger than the salary of every member of staff at branch B3.

Ans: (i) Select Fname, Lname from STAFF, BRANCH where STAFF.bno = BRANCH.bno and street = '163 main street'

(ii) Select Fname, Lname from STAFF where salary > (select max (salary) from Staff where bno='B3'

Q.93 **(i)** Consider employee (e_no, e_name, e_salary, d_code), dept (d_code, d_name) and dependent (depndt_name, e_no, relation).
Show the names of employees in purchase and accounts departments with at least one dependent.

(ii) Consider student (std_id, std_name, date_of_birth, phone, dept_name). Put the data for a student with student id200, name arun, birth date 1st February, 1985, phone number (01110 32818 and dept name English in the student table.

(iii) A constraint named less_than_20 was defined on the field date_of_birth of table student. Delete this constraint.

(iv) Consider the table student and list names of students in the departments other than maths and computer**.**

(v) Consider employee table of (i) and list names of department(s) for which average salary for department is more than 10,000.

(vi) Create role named role_table that allows a user to create tables. Using role_table allow users kripa and reena to create tables.

(vii) Create a view emp_dep containing e_name and number of dependents from the tables employee and dependent of (i) **(2x7)**

Ans:(i)
```
SELECT e_name FROM employee, dependent, dept
    WHERE employee.d_code=dept.d_code
        AND employee.e_no=dependant.e_no
            AND d_name IN ('ACCOUNTS', 'PURCHASE')
                GROUP BY e_name
                HAVING COUNT (e_no)>=1
```

(ii)
```
INSERT INTO student VALUES (200,'arun,''01-FEB-85',
'(0111)32818','English');
```

(iii)
```
ALTER TABLE student DROP CONSTRAINT less_than_20;
```

(iv)
```
SELECT std_name FRM student WHERE dept_name NOT IN
('Maths','Computer);
```

(v)
```
(SELECT d_name FROM dept
        WHERE d_code IN (SELECT d_code
                FROM emp GROUP BY d_code
            HAVING AVG(sal) ? 10000);
```

(vi)
```
CREATE ROLE role_table;
GRANT CREATE ANY TABLE TO role_table;
                GRANT role_table TO kripa, reena;
```

(vii)
```
CREATE VIEW emp_dept AS SELECT ename,
COUNT(*) FROM employee, dependent
WHERE employee.e_no = dependant.e_no GROUP BY e_name;
```

Q.94 Mention the kind of constraints we can specify in the CREATE command of DDL. **(2)**

Ans: The kind of constraints which can be specified are as follows:-
PRIMARY KEY, NOT NULL, UNIQUE, FOREIGN KEY, CHECK**.**

Q.95 What is the difference between a primary index and a secondary index? What are the advantages of using an index and what are its disadvantages. **(10)**

Ans: Primary Index: A primary index is an ordered file whose records are of fixed length with two fields. The first field is the ordering key field-called primary key-of the data file, and the second field is a pointer to a disk block. There is one index entry in the index file for each block in the data file. Each index entry has the value of the primary key field for the first record in a block and a pointer to that block as its two field values. A major problem with a primary index is insertion and deletion of records. If we attempt to insert a record in it's correct positioning the data file, we have to not only move records to make space for the new record but also change some index entries.

Secondary Index: A secondary index is also an ordered file with two fields. The first field is non-ordering field of the data file that is an indexing field. The second field is either a block pointer or a record pointer. A secondary index on a candidate key looks just like a dense primary index, except that the records pointed to by successive values in the index are not stored sequentially.

In contrast, if the search key of a secondary index is not a candidate key, it is not enough to point to just the first record with each search-key value. The remaining records with the same search – key value could be anywhere in the file, since the records are ordered by the search key of the primary index, rather than by the search key if the secondary index. Therefore, a secondary index must contain pointers to all the records. Secondary indices improve the performance of queries that use keys other than the search key of the primary index. However, they impose a significant overhead on modification of the database. The designer of a database decides which secondary indices are desirable on an estimate of the relative frequency of query's and modifications.

Some of the advantages of using an index are:

(i) Indexes speed up search on the indexed attributes(s). Without an index either a sequential search or some sort of binary search would be needed.

(ii) Indexes can also speed up sequential processing of the file when the file is not stored as a sequential file.

Some of the disadvantages of using an index are :

(i) An index requires additional storage. This additional storage can be significant when a number of indexes are being used on a file.

(ii) Insertion, deletion and updates on a file with indexes takes more time than on a file without any indexes.

Q.96 What are the causes of bucket overflow in a hash file organization? What can be done to reduce the occurrence of bucket overflow? **(6)**

Ans: When a record is inserted, the bucket to which it is mapped has space to store the record. If the bucket does not have enough space, a **bucket** overflow is said to occur. Bucket overflow can occur for several reasons:

Insufficient buckets : The number of buckets, which we denote n_b, must be chosen such than $n_b > n_r/f_r$, where n_r, denotes the total number of records that will be stored, and f_r, denotes the number of records that will fit in a bucket. This designation, of course, assumes that the total number of records is known when the hash function is chosen.

Skew : Some buckets are assigned more records than are others, so a bucket may overflow even when other buckets still have space. This situation is called bucket skew.

Skew can occur for two reasons:

1. Multiple records may have the same search key.
2. The chosen hash function may result in non-uniform distribution **of** search keys.

So, that the probability of bucket overflow is reduced, the number of buckets is chosen to be $(n_r/f_r)*(1+d)$, where d is a fudge factor typically around 0.2. Some space is wasted: About 20 percent o the space in the buckets will be empty. But the benefit is that the probability of overflow is reduced.

Despite allocation of a few more buckets than required, bucket overflow can still occur. We handle bucket overflow by using overflow buckets. If a record must be inserted into a bucket b, and b is already full, the system provides an overflow bucket for b, and inserts the record into the overflow bucket, and so on.

All the overflow buckets of a given bucket are chained together in a linked list. Overflow handling using such linked list is called overflow chaining.

Q.97 Explain the concept of QBE **(8)**

Ans: Query-by-example represents a visual/graphical approach for accessing information in a database through the use of query templates called as skeleton tables. It is used by entering example values directly into a query template to represent what is to be achieved. QBE was developed at IBMs. T.J. Watson Research Centre and it is used by many database systems for personal computers. QBE is a very powerful facility that gives the user the capability to access the information a user wants without the knowledge of any programming language.

Queries in QBE are expressed by skeleton tables. QBE has two distinct features :

- QBE has two-dimensional syntax : Queries look like tables.
- QBE queries are expressed "by example". Instead of giving a procedure for obtaining the desired answer, the user gives an example of what is desired.

An example of QBE is given as

EMPNO	NAME	SALARY	DESIG	DEPTNO
	P..x		"Manager"	

This query tells the system to look for tuples in EMP relation that have designation Manager. For each such tuple, the system assigns the value of the Ename attribute to the variable x and prints it.

Q.98 Describe the function of each of the following types of keys:
Primary, alternative, secondary and foreign. **(8)**

Ans: **Primary Key :** The primary key is an attribute or a set of attributes that uniquely identify a specific instance of an entity. Every entity in the data model must have a primary key whose values uniquely identify instances of the entity.

To qualify as a primary key for an entity, an attribute must have the following properties :

* It must have a non-null value for each instance of the entity.
* The value must be unique for each instance of an entity
* The values must not change or become null during the life of each entity instance.

Candidate Key and Alternate Key : In some instances, an entity will have more than one attribute that can serve as a primary key. Any key or minimum set of keys that could be a primary key is called a candidate key. Ones candidate keys are identified, one of

them is chosen as primary key. The choice of Primary key is based on guaranteed uniqueness and minimalism.

Candidate keys which are not chosen as the primary key are known as alternate keys.

Foreign Key : The primary key of one file or table which is implanted in another file or table to implement the relationships between them. Foreign keys are used to implement some types of relationships. Foreign keys do not exist in information models.

Q.99 Explain the structural components of a DBMS. **(10)**

Ans: DML compiler, Embedded DML compiler, DDL Interpreter, Query Evaluation Engine, Authorization and integrity Manager, Transaction Manager, File Manager, Buffer Manager.

DML Compiler : Translates DML statements in a query language into low level instructions understandable by the query evaluation engine. Attempts to transforms users request into an equivalent and more efficient from for executing the query understandable by Data Manager, Interprets DDL statements and records them in a set of tables containing Meta data in a form that can be used by other components of a DBMS.

Query Evaluation engine : Vehicle executes low-level instructions generated by the DML compiler.

The Storage Manager components provide interface between the low level data stored in the database and application programs and queries submitted to the system. It is central software component of DBMS and called database control system also.

Authorization and Integrity Manager : Tests for the satisfaction of integrity constraints and checks the authority of users to access data.

Transaction Manager : Ensure the database remains in a constant (correct) state despite system failure, and that concurrent transaction executions proceed without conflicting.

File Manager : Manager allocation of space on disk storage and the data structures used to represent into stored on disk. It is also responsible for –

- Locating block of target record
- Requesting block from disk Manager
- Transmitting required record to data Manager

This file Manager can be implemented using an interface to the existing file subsystem provided by the O.S. of the host computer or it can include a file subsystem written especially for DBMS.

Disk Manager : Responsible for fetching data regulated by file Manager from disk storage into Main memory and deciding what data to cache in memory.

It is a part of the O.S. of host computer and all physical input/output operations are performed by it. It provides disk abstraction to file Manager s.t. file Manager need not be concerned with the physical characteristics of the underlying storage media.

Functions of Data Manager :

- Convert operations in users queries coming directly via query processor or indirectly in an application program from users logical view to a physical file system.
- It interfaces with file system enforces constraints to maintain consistency. And integrity of the data as well as its security. (Authorization and Integrity Manager).
- Synchronizing simultaneous operations performed by concurrent users. (Transaction Manager)
- Back up and recovery operations (Transaction Manager)

Q.100 Describe the GRANT function and explain, how it relates to security. What types of privileges may be granted? How are they revoked? **(6)**

Ans: Since more than one user can access one database, there exists the need to restrict user from using entire database in any way. Setting up access privilege can do this. Any privilege can be granted to or revoked from user with taken of the GRANT and REVOKE statements. Each privilege can be granted or revoked individually or in a group. Such group of rights is called ROLE and rights can be granted or revoked into in the same way as to the user. Access privileges include SELECT, INSERT, UPDATE and DELETE.
GRANT {statement|role_name} ON {object_name} TO {user|role_name_2} [WITH GRANT OPTION]

statement	A statement to be granted or revoked.
role_name	The name of a role to be granted or revoked.
object_name	The name of an object (table, view or role) on which the rights will be changed.

user	The name of a user, the rights will be granted to or revoked from. All users are represented by the keyword PUBLIC.
role_name_2	The name of a role, the rights will be granted to or revoked from.
WITH GRANT	Gives the administration right on object.
OPTION	right can grant rights, revoke rights, create role, change role and delete role.

An example of granting permission to user B for reading EMP table is:
GRANT SELECT ON emp to B;
The REVOKE statement is used to revoke the permissions from the user.
The syntax for REVOKE statement is given as:

REVOKE {statement|role_name} FROM {user|role_name};

Q.101 Define the following terms

(i) Catalog and meta data
(ii) Parametric end users
(iii) DBA
(iv) Controlled redundancy
(v) Snapshot
(vi) Data sublanguage
(vii) High level DML
(viii) Data abstraction **(16)**

Ans: i) Catalog and meta data
A catalog is a compilation of records describing the contents of a particular collection or group of collections.Metadata (meta data, or sometimes metainformation) is "data about data", of any sort in any media. An item of metadata may describe an individual datum, or content item, or a collection of data including multiple content items and hierarchical levels, for example a database schema.
(ii) **Parametric end users** may be given update access, but are generally not allowed to change the structure of data.
(iii) **DBA:** A database administrator (DBA) is a person who is responsible for the environmental aspects of a database. In general, these include:

- Recoverability - Creating and testing Backups
- Integrity - Verifying or helping to verify data integrity

- Security - Defining and/or implementing access controls to the data
- Availability - Ensuring maximum uptime
- Performance - Ensuring maximum performance
- Development and testing support - Helping programmers and engineers to efficiently utilize the database

(iv) Controlled redundancy The replication of data within the data warehouse for the purposes of improved data access or understand ability.

(v) Snapshot A record of the state of an entry fetched from a persistent store at the time is it fetched

(vi) Data sublanguage In relational database theory, the term sublanguage, first used for this purpose by E. F. Codd in 1970, refers to a computer language used to define or manipulate the structure and contents of a relational database management system (RDBMS). Typical sublanguages associated with modern RDBMS's are QBE (Query by Example) and SQL (Structured Query Language).

(vii) High level DML A high-level or non-procedural DML allows the user to specify what data is required without specifying how it is to be obtained. Many DBMSs allow high-level DML statements either to be entered interactively from a terminal or to be embedded in a general-purpose programming language.

(viii) Data abstraction Data abstraction is a methodology that enables us to isolate how a compound data object is used from the details of how it is constructed from more primitive data objects.

Q.102 Differentiate between

(i) Procedural and non procedural DML
(ii) Forms based and graphical interface
(iii) Internal and external schema **(4 x 3 = 12)**

Ans (i) Procedural and non procedural DML Procedural DML specify what data is needed and how to get those data. Non procedural DML specify what data is needed without specifying how to get those data.

(ii) Forms based and graphical interface: graphical user interface (GUI) **is a** type of user interface which allows people to interact with electronic devices like computers, hand-held devices (MP3 Players, Portable Media Players, and Gaming devices), household appliances and office equipment. A *GUI* offers graphical icons, and visual indicators as opposed to text-based interfaces, typed command labels or text navigation to fully represent the information and actions available to a user. The actions are usually performed through direct manipulation of the graphical elements.

Forms-based applications are inherently less usable and less functional than traditional GUI-based applications.

(iii) Internal and external schema: Internal Schema

- Describes the physical storage structure
- Uses a physical data model

External Schema

- Includes a number user views
- Uses a conceptual or an implementation data model

Q.103 Explain the conventions for displaying an ER schema as an ER diagram. **(8)**

Ans: An E-R Diagram scheme may define certain constraints to which the contents of a database must conform.

Mapping Cardinalities: express the number of entities to which another entity can be associated via a relationship. For binary relationship sets between entity sets A and B, the mapping cardinality must be one of:

1. **One-to-one:** An entity in A is associated with at most one entity in B, and an entity in B is associated with at most one entity in A. (Figure 2.3)
2. **One-to-many:** An entity in A is associated with any number in B. An entity in B is associated with at most one entity in A. (Figure 2.4)
3. **Many-to-one:** An entity in A is associated with at most one entity in B. An entity in B is associated with any number in A. (Figure 2.5)
4. **Many-to-many:** Entities in A and B are associated with any number from each other.

The appropriate mapping cardinality for a particular relationship set depends on the real world being modeled. Cardinality ratios for binary relationships are represented on ER diagram by displaying 1,M, N etc. on the diamonds.

Existence Dependencies: if the existence of entity X depends on the existence of entity Y, then X is said to be existence dependent on Y

Q.104 Describe the two alternatives for specifying structural constraints on relationship types. **(8)**

Ans: Relationship types have certain constraints that limit the possible combination of entities that may participate in relationship.

An example of a constraint is that if we have the entities Doctor and Patient, the organization may have a rule that a patient cannot be seen by more than one doctor. This constraint needs to be described in the schema.There are two main types of structural relationship constraints on: cardinality ratio, and participation

Q.105 Discuss the techniques for a hash file to expand and shrink dynamically. What are the advantages and disadvantages of each? **(8)**

Ans:

The hashing techniques that allow dyanamic file expansion are:

(i) Extendible hashing

(ii) Linear hashing

The main advantage of extendible hashing that makes it attractive is that performance of the file does not degrade as the file grows. Also, no space is allocated in extendible hashing for future growth, but additional buckets can be allocated dynamically as needed. A disadvantage is that the directory must be searched before accessing the buckets themselves, resulting in two blocks accesses instead of one in static hashing.

Q.106 What are the reasons for having variable length records? What are the various ways to store variable length records? **(4)**

Ans: Variable-length records arise in a database in several ways:

(i)Storage of multiple items in a file.

(ii)Record types allowing variable field size

(iii)Record types allowing repeating fields

Various ways of storing variable length records:

(i) In fixed form by allocating every record, maximum amount of space that can be occupied a record and padding smaller sized records with nulls.

(ii) Special separator characters between the records to delineate variable length records.

Q107 Discuss the mechanism to read data from and write to a disk. **(4)**

Ans: Disk read/write heads are mechanisms that read data from or write data to disk drives. The heads have gone through a number of changes over the years.

In a hard drive, the heads 'fly' above the disk surface with clearance of as little as 3 nanometres. The "flying height" is constantly decreasing to enable higher area density. The flying height of the head is controlled by the design of an air-bearing etched onto the disk-facing surface of the *slider*. The role of the air bearing is to maintain the flying height constant as the head moves over the surface of the disk. If the head hits the disk's surface, a catastrophic head crash can result.

The heads themselves started out similar to the heads in tape recorders—simple devices made out of a tiny C-shaped piece of highly magnetizable material called ferrite wrapped in a fine wire coil. When writing, the coil is energized, a strong magnetic field forms in the gap of the C, and the recording surface adjacent to the gap is magnetized. When reading, the magnetized material rotates past the heads, the ferrite core concentrates the field, and a current is generated in the coil. The gap where the field is very strong is quite narrow. That gap is roughly equal to the thickness of the magnetic media on the recording surface. The gap determines the minimum size of a recorded area on the disk. Ferrite heads are large, and write fairly large features.

Q.108 Consider the following relations
RENTER(rno, fname, lname, address, tel_no, pref_type, max_rent)
VIEWING(rno, pno, date, comment)
PROPERTY_FOR_RENT(pno, street, area ,city, pcode, type, rooms, rent)
Express the following queries in relational algebra.
(i) List the name and comments of all renters who have viewed a property.
(ii) Identify all renters who have viewed all properties with three rooms.

Ans: (i) $\Pi_{fname,lname,comment}$(RENTER $\bowtie$VIEWING)

(ii) $\Pi_{fname,lname}$ ($\sigma_{rooms=3}$ (PROPERTY_FOR_RENT)$\bowtie$RENTER $\bowtie$VIEWING)

Q.109 Consider the relations
City (city_name, state)
Hotel (name, address)
City_hotel (hotel_name, city_name, owner)
Answer the following queries in relational algebra
(i) Find the names and address of hotels in Agra.
(ii) List the names of cities which have no hotel.
(iii) List the names of the hotels owned by 'Taj Group'. **(9)**

Ans: (i) $\Pi_{name, address}$ ($\sigma_{city\_name= 'Agra'}$ ($Hotel_{name=city\_name}$ $\bowtie$City_hotel))
(ii) Π_{city_name} (City) – Π_{city_name}(City_hotel)

(iii) $\Pi_{\text{hotel\_name}} (\sigma_{\text{owner='Taj Group'}} (\text{City\_hotel}))$

Q.110 Explain the difference between using functions with and without grouping attributes in relational algebra. Give examples. **(6)**

Ans: Group functions are used to group data of similar type.
GAMMA = grouping and aggregation
Applying $GAMMA_L(R)$
- Group *R* according to all the grouping attributes on list *L*.
  - That is: form one group for each distinct list of values for those attributes in *R*.
- Within each group, compute AGG(*A*) for each aggregation on list *L*.
- Result has one tuple for each group:
  - The grouping attributes and
  - Their group's aggregations.

Example: Grouping /Aggregation
R = (A B C)
1 2 3
4 5 6
1 2 5
$GAMMA_{A,B,AVG(C)}$ (R) = ?
First, group *R* by *A* and *B* :
A B C
1 2 3
1 2 5
4 5 6
Then, average *C* within groups:
A B AVG(C)
1 2 4
4 5 6

Q.111 Define the following with respect to SQL
(i) Specifying alias (ii) UNIQUE function
(iii) ORDER BY clause (iv) LIKE predicate
(v) Asterisk (*) **(10)**

Ans:(i) Specifying alias: is used to rename column or attribute in a table.
(ii) UNIQUE function: If **UNIQUE** is specified then only **unique** values are used to calculate the mean.
(iii) ORDER BY clause: The **ORDER BY clause** allows you to sort the records in your result set. The **ORDER BY clause** can only be used in select statements.
(iv) LIKE predicate: The **LIKE predicate** searches for strings that have a certain pattern.
(v) Asterisk (*): **In SQL**, the columns for all the tables and views in the FROM clause will be displayed.

Q.112 Consider the relations given below
Borrower (id_no, name)

Book (accno., title, author, borrower_idno)

(a) Define the above relations as tables in SQL making real world assumptions about the type of the fields. Define the primary keys and the foreign keys. **(7)**

(b) For the above relations answer the following queries in SQL

What are the titles of the books borrowed by the borrower whose id-no in 365.

(i) Find the numbers and names of borrowers who have borrowed books on DBMS in ascending order in id_no.

(ii) List the names of borrowers who have borrowed at least two books. **(9)**

Ans: **a)** Create table Book
(Accno int Primary Key,
title char(30),
author char(30),
borrow-idno int references Borrower_id.no);
Create table borrower
(id-no int Primary Key,
name char(30));

b) **(i)** Select title from Book, Borrower where Borrower.id_no =Book.borrower-idno and borrower.id_no = 365

(ii) Select id_no, name from Borrower, Book where Borrower.id_no= Book.borrower_idno and title= 'DBMS' order by id_no asc;

(iii) Select name from Borrower, Book where Borrower.id_no = book.borrower_id_no having count (*) > 2

Q.113 Describe the structure of Oracle data dictionary. **(4)**

Ans: Generally, the data dictionary consists of base tables and user-accessible views. The base tables contain all database information that is dynamically updated by Oracle RDBMS. Oracle strictly discourages using those tables even for selects; the database users normally have no access to them, and even DBAs do not typically query these tables directly. The information stored in the base tables is cryptic and difficult to understand. The user-accessible views summarize and display the information stored in the base tables; they display the information from the base tables in readable and/or simplified form using joins, column aliases, and so on. Different Oracle users can have SELECT privileges on different database views.

Q.114 In Oracle what is system global area and how is it organized? **(6)**

Ans: The SGA is a collection of shared memory areas that along with the Oracle processes constitute an Oracle instance. All SGA components allocate and deallocate space in units of granules. Granule size is determined by total SGA size. On most platforms, the size of a granule is 4 MB if the total SGA size is less than 1 GB, and granule size is 16MB for larger SGAs. Some platform dependencies arise. For example, on 32-bit Windows, the granule size is 8M for SGAs larger than 1GB

Q.115 Write a short note on QBE**.** **(6)**

Ans: Stands for "Query By Example." QBE is a feature included with various database applications that provides a user-friendly method of running database queries.

Typically without QBE, a user must write input commands using correct SQL (Structured Query Language) syntax. This is a standard language that nearly all database programs support. However, if the syntax is slightly incorrect the query may return the wrong results or may not run at all. The Query By Example feature provides a simple interface for a user to enter queries. Instead of writing an entire SQL command, the user can just fill in blanks or select items to define the query she wants to perform. For example, a user may want to select an entry from a table called "Table1" with an ID of 123. Using SQL, the user would need to input the command, "SELECT * FROM Table1 WHERE ID = 123". The QBE interface may allow the user to just click on Table1, type in "123" in the ID field and click "Search." QBE is offered with most database programs, though the interface is often different between applications. For example, Microsoft Access has a QBE interface known as "Query Design View" that is completely graphical. The phpMyAdmin application used with MySQL, offers a Web-based interface where users can select a query operator and fill in blanks with search terms. Whatever QBE implementation is provided with a program, the purpose is the same – to make it easier to run database queries and to avoid the frustrations of SQL errors

Q.116 What three main types of actions involve databases? Briefly discuss these. **(6)**

Ans: The three main types of actions involving databases are:
(i) Defining: It involves specifying the data types, structures and constraints of data types, structures and constraints the data to be stored in the database.
(ii) Manipulating: It includes functions such as querying the database to retrieve specific data updating the database to reflect changes in the miniworld and generating reports form data.
(iii) Sharing: It allows multiple users and programs to access the database simultaneously.

Q.117 What are the situations when DBMS should not be used? **(6)**

Ans: DBMS should not be used in situations like:
(i) No need of security.
(ii) Not difficulty to access the data
(iii) No redundancy and inconsistency etc.

Q.118 What is the difference between a database schema and a database state? **(4)**

Ans: The collection of information stored in database at particular moment in time is called database state while the overall design of database is called database schema.

Q.119 With the help of a diagram describe the typical component modules of a DBMS. **(8)**

Ans: The functional components of a database system can be broadly divided into:
1) Storage manager and **2)** query processor

1) Storage manager components includes:

(i) Authorization and integrity manager to test integrity constraints and authorized access to data.

(ii) Transaction manager to ensure that database remains in a consistent state.

(iii) File manager to manage disk storage space.

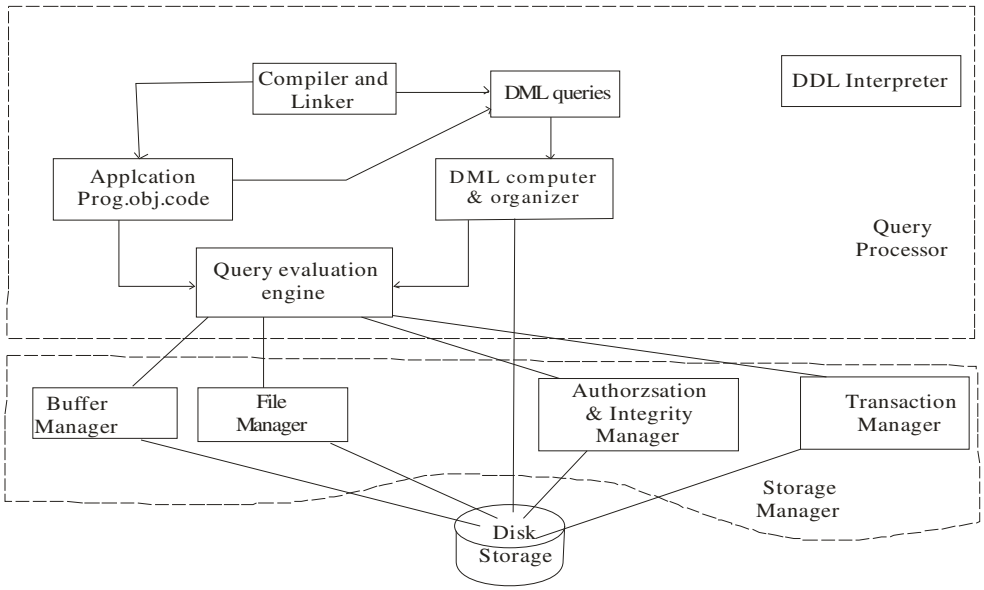
(iv) Buffer manager is responsible for buffering data.

2) The query processor component includes:

(i) DDL interpreter to interpret DDL statements

(ii) DML compiler which translation DML queries into query evaluation plan for query evaluation engine.

(iii) Query evaluation engine execute low level instructions generated by DML compiler.



Q.120 Define the following:

(i) Participation role **(ii)** Recursive relationship type

(iii) Composite attributes **(iv)** Entity

Ans: (i) Participation role: roles are indicating in E-R diagram by labelling the lines that connect diamonds to rectangles.

(ii) Recursive relationship type: A recursive relationship is one in which the same entity participates more than once in the relationship

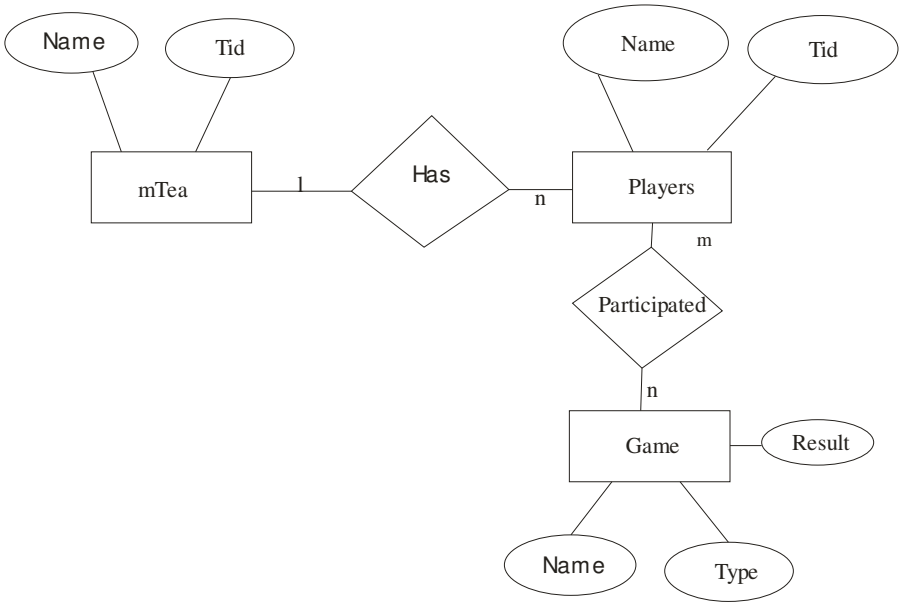
(iii) Composite attribute: A composite attribute has multiple components, each of which is atomic or composite.

(iv) Entity: A DBMS entity is *either* a thing in the modeled world *or* a drawing element in an ERD.

Q.121 A database is to be constructed to keep track of the teams and games of a sport league. A team has a number of players, not all of whom participate in each game. It is desired to keep track of the players participating in each game of each team and the result of the game.

Create an ER diagram, completely with attributes, keys and constraints, for the above description. State any assumptions that you make. **(8)**

Ans:



Q.122 Discuss the types of integrity constraints that must be checked for the update operations – Insert and Delete. Give examples. **(8)**

Ans: Insert operation can violet any of the following four constraints:

1) Domain constraints can be violated if given attribute value does not appear in corresponding domain.
2) Key constraints can be violated if given attribute value does not appear in corresponding domain.
3) Entity integrity can be violated if the primary key of the new tuple t is NULL.
4) Referential integrity can be violated if value of any foreign key in t refers to a tuple that does not exist in referenced relation.

Delete operation can violate only referential integrity constraints, if the tuple being deleted is referenced by the foreign keys from other tuples in the database.

Q.123 Differentiate between the following giving advantages and disadvantages of each.

(i) Primary and secondary storage.
(ii) Open addressing and chaining for collision resolution.
(iii) Unordered and ordered file. **(12)**

Ans: i) Primary and secondary storage

*Computer storage is classified into primary(main) memory and secondary(peripheral) storage.

a) Primary storage is usually RAM; ~10ns access time
b) Secondary storage is usually hard disk drives; ~10ms access time
c) Secondary storage is a lot cheaper than primary, 3¢/Mb vs.$1/Mb.
   - Secondary storage is persistent

a) Databases are stored in secondary memory, and large databases are manipulated in secondary storage.
b) We need to minimise disk accesses when accessing and manipulating databases.

(ii) Open addressing and chaining for collision resolution

In open addressing, proceeding from the occupied position specified by hash address the program checks the subsequent positions in order until an unused (empty) position is found.

Chaining: In this method various overflow locations are kept, usually by extending the way with a number of overflow positions. Additionally, a pointer field is added to each record location. A collision is resolved by placing the new record in an unused overflow location and setting the pointer of occupied hash address location to the address of that overflow location. A linked list of overflow records for each hash address is thus maintained.

(iii) Unordered and ordered file

Unordered file do no has any sequence while ordered file has arranged in some sequence and data are assigned in ordered format.

Q.124 What is recursive closure? Why is it not possible to define this operation in relational algebra? **(4)**

Ans: Recursive closure is applied to recursive relationship. An example of recursive operation is to retrieve all SUPERVISEES of an Employee e at all levels- that is , all EMPLOYEE e directly supervised by e; all employees e" directly supervised by each employee e" and so on

Q.125 Consider the two tables R1 (A,B,C) and R2 (P,Q,R) shown below

R1

A	B	C
a	15	10
a	25	8
b	15	5

R2

P	Q	R
b	15	5
c	10	8
b	5	6

Show the results of the following operations:

(i) $\Pi_{A,B}(R1)$

(ii) $F_{COUNT\ P,AVERAGE\ Q}(R2)$ (where F is the aggregate function)

(iii) $R_1 X R_{1.B} > R_{2.Q} R_2$

(iv) R1 X R2 **(12)**

Ans: (i)

A	B
a	15
a	25
b	15

(ii)

Count p	Average Q
3	10

(iii) $R_1 \times R_{1.B} > R_{2.Q} R_2$

A	B	C	P	Q	R
a	15	10	c	10	8
A	15	10	b	5	6
a	25	8	b	15	5
a	25	8	c	10	8
a	25	8	b	5	6
b	15	5	c	10	8
b	15	5	b	5	6

(iv) $R_1 \times R_2$

A	B	C	P	Q	R
a	15	10	b	15	10
a	15	10	c	10	8
a	15	10	b	5	6
a	25	8	b	15	5
a	25	8	c	10	8
a	25	8	b	5	6
b	15	5	b	15	5
b	15	5	c	10	8
b	15	5	b	5	6

Q.126 Explain the EXISTS and UNIQUE functions of SQL. Give an example for each. **(6)**

Ans: EXISTS: The `EXISTS` function takes one parameter which is a SQL statement. If any records exist that match the criteria it returns true, otherwise it returns false. This gives you a clean, efficient way to write a stored procedure that does either an insert or update.

UNIQUE: If **UNIQUE** is specified then only **unique** values are used to calculate the mean.

Q.127 What is NULL? Give an example to illustrate testing for NULL in SQL. **(4)**

Ans: The NULL SQL keyword is used to represent either a missing value or a value that is not applicable in a relational table.

Consider there is a relation:

Person(id, name, address, phone)

Now to find ids and names of person who do not have a phone is:

```
Select id, name
from Person
where phone is null
```

Q.128 Describe substring comparison in SQL. For the relation Person(name, address), write a SQL query which retrieves the names of people whose name begins with 'A' and address contains 'Bangalore'. **(6)**

Ans: SUBSTR is used to extract a set of characters from a string by specifying the character starting position and end position and length of characters to be fetched.
example substr('hello',2,3) will return 'ell'

```
Select name
from Person
where name like 'A%' and address = 'Bangalore'
```

Q.129 Consider the following relations with keys underlined
Street (name, location, city)
House (number, street_name)
Lives (name, house_number)
Define the above relations as tables in SQL making real world assumptions about the type of the fields. Define the primary keys and the foreign keys. **(7)**

Ans: Create table street (name character(30) primary key, location character(30), city character(30));
Create table house(number integer primary key, street_name character(30) references street(name));
Create table lives (name character(30) primary key, house number integer references house(number));

Q.130 For the relations given in Q129 answer the following queries in SQL

(i) Get the names of persons who live in the street named 'Mahatma Gandhi'.
(ii) Get the house numbers street wise.
(iii) Get the numbers of houses which are not occupied. **(9)**

Ans: **(i)** Select lives.name from Street, House, Lives where Street.name = House.street_name and House. Number = Lives. House_number and street_name = 'Mahatma Gandhi';

(ii)
```
Select number from House
order by street_name;
```

(iii)
```
Select number from house
        minus
Select house_number from lives;
```

Q.131 How is the database organised in Oracle? **(8)**

Ans: Database organized in Oracle in terms of table and table contains attributes and values.

A database consists of one or more logical units called table spaces. Each table space in turn, consists of one or more physical structures called data files. These may be either, files managed by the operating system or raw devices.
Oracle database consists of following table spaces
(i) System **(ii)** Table spaces to store user data **(iii)** Temporary table spaces.
The space in table space is divided into units, called segments, that each contain data for a specific data structure. These are four types of segments:
(i) Data segments
(ii) Index segments
(iii) Temporary segments
(iv) Rollback segments

Q.132 Explain the following with respect to QBE
(i). CNT
(ii) I
(iii) P
(iv) D **(8)**

Ans: (i) CNT: It is one of the aggregation commands provide in QBE. It is used to identify number of tuples
(ii) I : It is the command in QBE which is used to insert tuple (s) into the database
(iii) P: It is the command that is used to print (logically display) the value of the attribute in whose cell it is written. To display the entire relation, P has to be written in every field or by placing a single P in the column headed by the relation name also displays entire relation.
(iv) D: It is the command in QBE to delete tuple(s) from the relation.

Q.133 Discuss the differences between the candidate keys and the primary key of a relation. Give example to illustrate your answer.

Ans: A candidate key is one which can be used as primary key that is not null and unique constraint both holding true. In short all primary keys are definitely candidate keys. That is one of the candidate keys is chosen as primary key.

Q.134 What are the DBMS languages? Briefly explain. **(6)**

Ans: Data Definition language (DDL):A database schema is specified by a set of definitions expressed by a special language called data definition language. For example, the following statement in the SQL defines the account table
Create table account (account_number char(10), balance integer)
Execution of the above DDL statement creates the account table. DDL allows a user to specify storage structure and access methods used by the database system as well as specify certain consistency constraints like balance of an account should on fall below Rs1000.
Data Manipulation language (DML): A data-manipulation language (DML) is a language that enables users to access or manipulate data as organized by the appropriate data model.

There are basically two types:

1) Procedural DMLs require a user to specify what data are needed and how to get those data.

2) Declarative DMLs (also referred to as nonprocedural DMLs) require a user to specify what data are needed without specifying how to get those data.

Q.135 Explain the terms primary key, candidate key, alternate key and secondary key. In the given table identify each key.
STUDENT(SID,Regno, Name, City) **(8)**

Ans: **Primary Key:** The primary key of a relational table uniquely identifies each record in the table. It can either be a normal attribute (or set of attributes) that is guaranteed to be unique (such as Social Security Number in a table with no more than one record per person) or it can be generated by the DBMS (such as a globally unique identifier, or GUID, in Microsoft SQL Server).

Candidate Key: A candidate key is a combination of attributes that can be uniquely used to identify a database record without any extraneous data. Each table may have one or more candidate keys**.** One of these candidate keys is selected as the table primary key.

Alternate Key: An alternate key (**or secondary key**) is any candidate key which is not selected to be the primary key (PK).

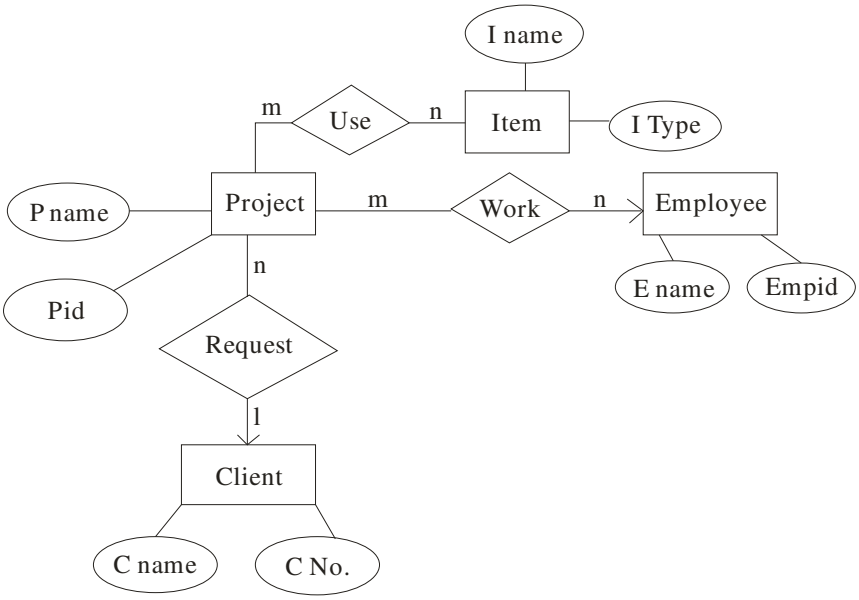
For example, a relational database with a table "employee" could have attributes like "employee_id", "bank_acct_no", and so on. In this case, both "employee_id" and "bank_acct_no" serve as unique identifiers for a given employee, and could thus arguably be used for a primary key. Hence, both of them are called "candidate keys". If, **for** example, "bank_acct_no" was chosen as the primary key, "employee_id" would become the alternate key

Secondary Key: Same as above

In above table SID is a primary key, SID and Regno are candidate keys and SIDor Regno are alternate or secondary key.

Q.136 In an organisation several projects are undertaken. Each projects can employ one or more employees. Each employee can work on one or more projects.Each project is undertaken on the request of client. A client can request for several projects. Each project has only one client. A project can use a number of items and a item may be used by several projects. Draw an E-R diagram and convert it to a relational schema. **(10)**

Ans:



The relational schema has following tables

1) Project (Pid, Pname)
2) Employee (Empid, Ename)
3) Client (Cno, Cname)
4) Item (I type, I name)
5) Use (Pid, I type)
6) Work (Pid, Empid)
7) Request (Pid, Cno)

Q.137 Define

(i) Identifying relationship.
(ii) Specialisation / generalization.
(iii) Aggregation. **(6)**

Ans: (i) Identifying relationship: An identifying relationship means that the child table cannot be uniquely identified without the parent. For example, you have this situation in the intersection table used to resolve a many-to-many relationship where the intersecting table's Primary Key is a composite of the left and right (parents) table's Primary Keys

(ii) Specialisation / generalization: Generalization/Specialization represents the is a relationship set, an essential element of the object oriented paradigm. The main idea in Generalization/Specialization is that one object class (the specialization) is a *subset* of another (the generalization).

(iii) Aggregation: Aggregation refers to an abstraction in which a relationship between objects is regarded as a higher-level object.

Q.138 What is hash file organization? What are the causes of bucket overflow in a hash file organization? What can be done to reduce the occurrence of bucket overflow? **(8)**

Ans: Hashing involves computing the address of a data item by computing a function on the search key value.

A hash function h is a function from the set of all search key values *K* to the set of all bucket addresses *B*.

We choose a number of buckets to correspond to the number of search key values we will have stored in the database.

To perform a lookup on a search key value K_i, we compute $h(K_i)$, and search the bucket with that address.

If two search keys *i* and *j* map to the same address, because, $h(K_i)=h(K_j)$ then the bucket at the address obtained will contain records with both search key values.

The hash functions selected should be such that it should result in uniform distribution of search keys.

Q.139 What do you understand by RAID? Explain RAID Level 4 and Level5. **(8)**

Ans: RAID — which stands for Redundant Array of Inexpensive Disks (as named by the inventor), or alternatively Redundant Array of Independent Disks (a less relative name, and thus now the generally accepted one) — is a technology that employs the simultaneous use of two or more hard disk drives to achieve greater levels of performance, reliability, and/or larger data volume sizes.

RAID Level 4: In this setup, files can be distributed between multiple disks. Each disk operates independently which allows I/O requests to be performed in parallel, though data transfer speeds can suffer due to the type of parity. The error detection is achieved through dedicated parity and is stored in a separate, single disk unit.

RAID Level 5: Distributed parity requires all drives but one to be present to operate; drive failure requires replacement, but the array is not destroyed by a single drive failure. Upon drive failure, any subsequent reads can be calculated from the distributed parity such that the drive failure is masked from the end user. The array will have data loss in the event of a second drive failure and is vulnerable until the data that was on the failed drive is rebuilt onto a replacement drive

Q.140 Consider the relations:

PROJECT(proj#,proj_name,chief_architect)

EMPLOYEE(emp#,emp_name)

ASSIGNED(proj#,emp#)

Use relational algebra to express the following queries:

- **(i)** Get details of employees working on project COMP33.
- **(ii)** Get the employee number of employees who work on all projects.
- **(iii)** Get details of project on which employee with name 'RAM' is working. **(9)**

Ans:(i) $EMPLOYEE \bowtie \Pi_{Emp\#}(\sigma_{project\#='comp33'}(ASSIGNED))$

(ii) $\Pi_{Emp\#}(EMPLOYEE) - (ASSIGNED \div \Pi_{project\#}(PROJECT))$

(iii) $PROJECT \bowtie (\Pi_{project\#}(\sigma_{Emp\_name='RAM'}(EMPLOYEE) \bowtie ASSIGNED))$

Q.141 Differentiate between join and outer join. **(4)**

Ans:Outer joins return all rows from at least one of the tables or views mentioned in the FROM clause, as long as those rows meet any WHERE or HAVING search conditions.

All rows are retrieved from the left table referenced with a left outer join, and all rows from the right table referenced in a right outer join. All rows from both tables are returned in a full outer join.
A table can be joined to itself in a **self-join.**

Q.142 Consider a table student (std_id, std_name, date_of_birth, percent_marks, dept_name). Write a QBE query to display names of Computer Science department students who have scored more than 80%. **(3)**

Ans:

Stud_id	Stud_name	date_of_birth	Percent_marks	Dept_name
	P._S		>80	Computer Science Department

Q.143 Consider the relations
EMP(ENO,ENAME,AGE,BASIC_SALARY)
WORK_IN(ENO,DNO)
DEPT(DNO,DNAME,CITY)
Express the following queries in SQL

(i) Find names of employees who work in a deptt. in Delhi.
(ii) Get the deptt. number in which more than one employee is working.
(iii) Find name of employee who earns highest salary in 'HR' department.

(9)

Ans:(i) select ENAME from EMP, WORK_IN, DEPT where EMP.ENo= WORK_IN.ENO and WORK_IN.DNO= DEPT.DNO and CITY= 'Delhi';
(ii) select DNO from WORK_IN group by DNO having count(*) >1
(iii) select ENAME from EMP e where BASIC_SALARY >= (select max(BASIC_SALARY) from DEPT,WORK_IN where DNAME = 'HR' and e.ENO = WORK_IN.ENO and WORK_IN. DNO= DEPT.DNO)

Q.144 Explain various kinds of constraints that can be specified using CREATE TABLE command. Explain CASCADE and RESTRICT clauses of DROP SCHEMA command. **(7)**

Ans:

Constraint	Description
PRIMARY KEY	Determines which column(s) uniquely identifies each record. The primary key cannot be NULL, and the data value(s) must be unique.
FOREIGN KEY	In a one-to-many relationship, the constraint is added to the "many" table. The constraint ensures that if a value is entered into a specified column, it must already exist in the "one" table, or the record is not added.
UNIQUE	Ensures that all data values stored in a specified column are unique. The UNIQUE constraint differs from the PRIMARY KEY constraint in that it allows NULL values.
CHECK	Ensures that a specified condition is true before the data value is added to a table. For example, an order's ship date cannot be earlier than its order date.
NOT NULL	Ensures that a specified column cannon contain a NULL value. The NOT NULL constraint can only be created with the column-level approach to table creation.

Prevents the DROP from taking place if any dependent objects exists (RESTRICT) or causes all dependent objects do also be dropped (CASCADE).

Q.145 Write short notes on following:

(i) Extension and Intension.
(ii) Weak and strong entity type.
(iii) Views in SQL.
(iv) Built in function in QBE.

($4 \times 4 = 16$)

Ans: (i) In any data model, it is important to distinguish between the description of the database and the database itself the description of a database is called database schema, which is specified during database design and is not expected to change frequently. The actual data in a database is database itself. May change quite frequently.

The schema is sometimes called the intension, and a database state is called an extension of the schema.

(ii) Weak and Strong entity type: An entity set that does not have a primary key is referred to as a weak entity set. The existence of a weak entity set depends on the existence of a strong entity set; it must relate to the strong set via a on-to-many relationship set. A discriminator of a weak entity set is the set of attributes that distinguishes among all the entities of a weak entity set. The primary key of a weak entity set is formed by the primary key of the strong entity set on which the weak entity set is existence depends, plus the weak entity set's discriminator.

(iii) We define a view in SQL by using the create view command. To define a view, we must give the view a name and must state the query that computers the view. The form of the create view command is

Create view z as <query expression> where z is the view name

< query expression > is any legal query expression.

(iv) Built in function in QBE: QBE includes several built in functions like AVG, MAX,MIN,SUM and CNT. We must prefix these operators with ALL to create a multiset on which the aggregate operation is evaluated. The ALL operator ensures that duplicates

are not eliminated. We can use UNQ to specify that we want duplicates eliminated. QBE also offers the ability to compute functions on groups of tuples using the G operator, which is analogous to SQL's group by construct.

Q.146 What is Oracle Process? Explain any four processes started by Oracle. **(8)**

Ans: Oracle Process is as follows:
SMON: The System Monitor carries out a crash recovery when a crashed instance is started up again. It also cleans temporary segments.
PMON: The Process Monitor checks if a user process fails and if so, does all cleaning up of resources that the user process has aquired.
DBWR : The Database Writer writes *dirty blocks* from the *database* buffer to the datafiles. How many DBWn Processes are started is determined by the initialization parameter `DB_WRITER_PROCESSES`. DBWR also writes the *actual SCN* with the Block.
LGWR : The Log Writer writes the redo log buffer from the SGA to the *online redo log file*.

Q.147 What is DBMS and what are functions of DBMS **(10)**
Ans: DBMS consist of collection of integrated data and set of program to access those data.
The functions performed by a typical DBMS are the following:

- **Data Definition**: The DBMS provides functions to define the structure of the data in the application. These include defining and modifying the record structure, the type and size of fields and the various constraints/conditions to be satisfied by the data in each field.
- **Data Manipulation:** Once the data structure is defined, data needs to be inserted, modified or deleted. The functions which perform these operations are also part of the DBMS. These functions can handle planned and unplanned data manipulation needs. Planned queries are those which form part of the application. Unplanned queries are ad-hoc queries which are performed on a need basis.
- **Data Security & Integrity:** The DBMS contains functions which handle the security and integrity of data in the application. These can be Thus the DBMS provides an environment that is both convenient and efficient to use when there is a large volume of data and many transactions to be proved

Q.148 How many types of users works on database? **(6)**

Ans: Users are differentiated by the way they expect to interact with the system
(i) Application programmers–interact with system through DML calls
(ii) Sophisticated users–form requests in a database query language
(iii) Specialized users–write specialized database applications that do not fit into the traditional data processing framework
(iv) Naive users–invoke one of the permanent application programs that have been written previously Examples, people accessing database over the web, bank tellers, clerical staff

Q.149 Describes the various relationship constraints by giving suitable example. **(8)**

Ans: Constraints on relationships: There are two types of constraints on relationships.

1) Mapping cardinalities or cardinality ratios express the number of entities to which another entity can be associated via a relationship set. For a binary relationship set between entities A and B. The mapping cardinality may be any one of the following:

a) **One to One:** An entity in A is associated with at most one entity in B and an entity in B is associated with at most one entity in A.

b) **One to Many:** An entity in A can be associated with any number in B but an entity in B is associated with at most one in A.

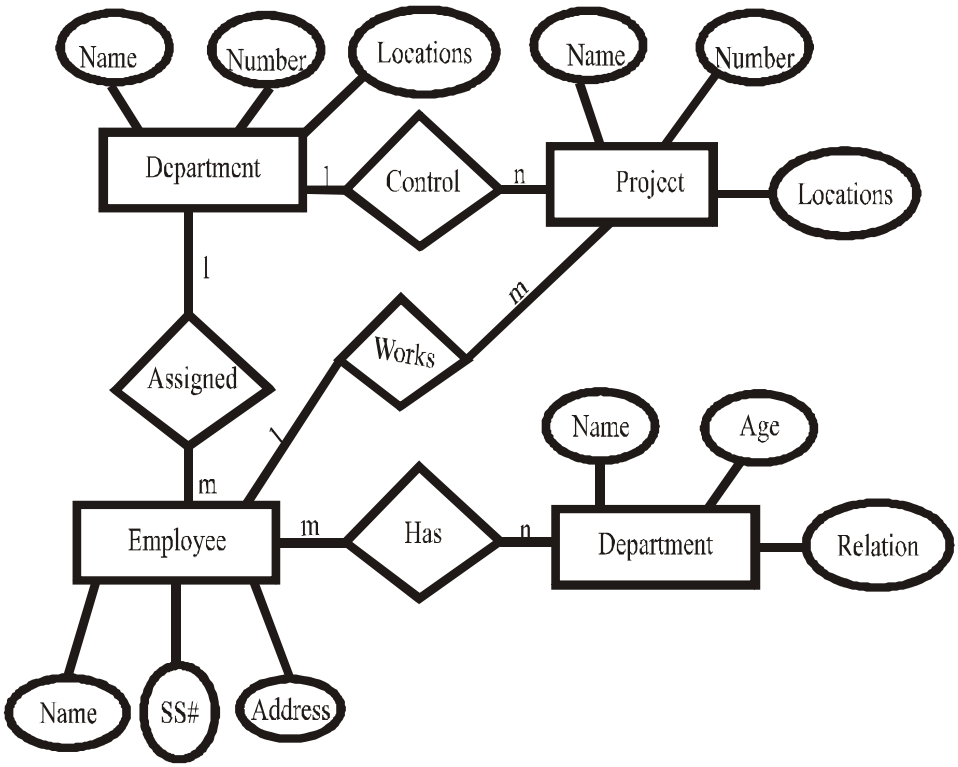
c) **Many to one:** An entity in A is associated with at most one in B but an entity in B is associated with any number in A.

d) **Many to Many:** An entity in A and B can be associated with any number of entities in the other entity set.

2) **Participation constraints:** The participation of an entity set E in a relationship set R is said to be total if every entity in E participates in at least one relationship in R. If only some entities in E participation of entity set E in relationship R is said to be partial.

Q.150 What is an E-R model? Draw an E-R Diagram for the company database with following Descriptions:
The company is organized into departments. Each department has a unique name and a unique number with several locations.
A department controls a number of projects, each of which has a unique name, unique number and a single location.
We store each employees name, social security number, address, and salary. An employee is assigned to one department but may work on several projects, which are not necessarily controlled by the same departments.
We want to keep track of the dependents of each employee for insurance purposes. We keep each dependent's name, age and relationship to the employee. **(8)**

Ans: An entity-relationship model (ERM) is an abstract conceptual representation of structured data. Entity-relationship modeling is a relational schema database modeling method, used in software engineering to produce a type of conceptual data model (or semantic data model) of a system, often a relational database, and its requirements in a top-down fashion. Diagrams created using this process are called *entity-relationship diagrams*, or *ER diagrams* or *ERDs* for short.



Q.151 Define the following

(i) Record-Based Logical Models
(ii) Data Independence **(8)**

Ans:(i) Record-Based logical Models: Also describe data at the conceptual and view levels. Unlike object-oriented models, are used to specify overall logical structure of the database, and provide a higher-level description of the implementation. Named so because the database is structured in fixed-format records of several types. Each record type defines a fixed number of fields, or attributes. Each field is usually of a fixed length (this simplifies the implementation). Record-based models do not include a mechanism for direct representation of code in the database. Separate languages associated with the model are used to express database queries and updates. The three most widely-accepted models are the relational, network, and hierarchical.

(ii) Data Independence: Techniques that allow data to be changed without affecting the applications that process it. There are two kinds of data independence. The first type is data independence for data, which is accomplished in a database management system (DBMS). It allows the database to be structurally changed without affecting most existing programs. Programs access data in a DBMS by field and are concerned with only the data fields they use, not the format of the complete record. Thus, when the record layout is updated (fields added, deleted or changed in size), the only programs that must be changed are those that use those new fields.

Q.152 What is bucket overflow and how bucket overflow is handled by Over Flow Chaining or Closed Hashing? **(8)**

Ans: **Bucket:** unit of storage containing records. Bucket is a disk block or contiguous block .Bucket contains multiple records .

Hash Function: maps records to bucket numbers Function h from set of all search key values K to the set of all bucket addresses(numbers) B .Records with different search key values may be mapped same bucket. Entire bucket has to be searched to locate record.If bucket is full need *overflow buckets* and pointers

How to handle bucket overflow:

- Buckets overflow is handled by using overflow buckets
  - *Closed Hashing (Chaining):* The overflow buckets of a given bucket are chained together in a linked list
  - *Open Hashing:* place in next available bucket (not suitable for database applications)
  - *Multiple Hashing :* use a second hash function

Q.153 What are the Constituents of File? Also explain all the possible file operations. **(8)**

Ans: A file is organised logically as sequence of record. These records are mapped onto disk blocks. Although blocks are of fixed size determined by the physical properties of the disk and by the operating system, record sizes vary. In a relational database, tuples of distinct relations are generally of different sizes.

Possible operations on a file:

(i) Create: a file can be created
(ii) Delete: a file can be deleted
(iii) Changing attributes of a file: The attributes of a file say read only, author's name etc. can be changed.
(iv) Contents of the file can be altered.

Q.154 Express the following queries in SQL assumes that the data is stored in EMPLOYEE table with relevant fields.

(i) Display name, job, salary, and hire date of employee who are hired between May 10, 1975 and December 20, 1980. Order the query in ascending order of hire date.

(ii) Display name and hire date of employee who are employed after employee 'RAGHAV'. **(6)**

Ans: (i) Select name, job, salary, hire date
from employee where hire date between
'May-10-1975' and 'Dec-20-1980'
order by hire date;

(ii) Select name, hire date
from employee
Where hire date> (select hire date from employee where name= 'Raghav');

Q.155 What are the various types of the update operations on relations? Also explain the constraints on these update operation. Give examples in support of your answer. **(10)**

Ans: There are three basic update separation on relations:

(i) **Insert :** It is used to insert a new tuple or tuples in a relation. Insert can violate any of the four tuples of constraints : Domain constraints, key constraints , entity integrity and referential integrity

(ii) **Delete :** It is used to delete tuples. The delete operation can violate only referential integrity, if the tuple being deleted is referenced by the foreign keys from other tuples in the database

(iii) **Modify:** It is used to change value of some attributes in existing tuples. Modify can also violate any of the four constraints as specified in insert operation.

Examples:

(1) Inserting a tuple having null value for primary key violates entity integrity

(2) Deleting a tuple in a table for which tuples exist other tables that are dependent on those tuples will violate referential key integrity.

(3) Modifying a primary key attribute to any other such that is aheady exists in the table will violate entity integrity.

Q.156 Consider the relations

EMPLOYEE(emp#, name)
ASSIGNED_TO(project#, emp#)
PROJECT(project#, project_name, chief)
Express the following queries in Relational Algebra
(i) Get details of employee working on both comp354 and comp345 project numbers.
(ii) Find the employee number of employee who do not work on project comp678 **(8)**

Ans:(i) $\text{EMPLOYEE} \bowtie (\text{ASSIGNED TO} \div \Pi_{\text{project \#}} (\sigma_{(\text{project\#='comp351'} \vee \text{project\#='comp 345}} \text{ ASSIGNED \_TO})))$

(ii) $\Pi_{\text{Emp\#}}(\text{EMPLOYEE}) - \Pi_{\text{Emp\#}} (_{\text{project\#='c 678'}} \text{ ASSIGNED\_TO}))$

Q.157 Explain the SQL operators BETWEEN-AND, IN, LIKE and IS_NULL by taking suitable examples. **(8)**

Ans:

Comparison Operators	Description
LIKE	column value is similar to specified character(s). % and – are two special characters to match any substring and character respectively in a string. Eg, 'Perry%' matches any sting with Perry.
IN	column value is equal to any one of a specified set of values. Eg, SQL query to find all customers who are borrowers also from a bank as well as account holders in the bank is : Select cust# from borrower where cust# in(Select cust# in depositor)
BETWEEN...AND	column value is between two values, including the end values specified in the range. Eg, SQL query to find all employees hiredate between 1-JAN-2008 to 31-DEC-2008: Select * from employee where hiredate between 1-JAN-2008' and '31-DEC-2008'
IS NULL	column value does not exist. Eg, SQL statement to find all salesperson who are not earning any commission: Select * from employee where job= 'Sales' and commission is null.

Q.158 Write short note on followings:

(i) Relational Constraints
(ii) Disadvantages of Relational Approach
(iii) Instances and Schemas

(4 × 4 = 16)

Ans: (i) Relational Constraints are:

- NOT NULL
- Unique
- Primary key
- Foreign key
- Table check

(ii) Disadvantages of relational approach:

- Substantial hardware and system software overhead
- May not fit all business models
- Can facilitate poor design and implementation

- May promote "islands of information" problems

(iii) Instances and schemas: Databases changes over time as the information is inserted and deleted. The collection of information stored in database at a particular moment in time is called Instances and the overall design of database is called schemas.

Q.159 What are the key features of Oracle? **(8)**

Ans: Key features of Oracle are as follows:

- Read Consistency
- Concurrency
- Locking Mechanisms
- Quiesce Database
- RAC
- Portability

Q.160 Explain the following functions of Oracle with suitable examples:
(i) To_Char( ) (ii) Count( )
(iii) Trim( ) (iv) Length( )
(8)

Ans: (i) To_char( ): The TO_CHAR function converts a DATETIME, number, or NTEXT expression to a TEXT expression in a specified format. This function is typically used to format output data.

(ii) Count( ): The COUNT function returns the number of rows in a query.
The syntax for the COUNT function is:

```
SELECT COUNT (expression)
FROM tables
WHERE predicates;
```

(iii) Trim( ): In Oracle/PLSQL, the trim function removes all specified characters either from the beginning or the ending of a string.
The syntax for the trim function is:

```
trim( [ leading | trailing | both  [ trim_character ] ]  string1 )
```

leading - remove *trim_string* from the front of *string1*.
trailing - remove *trim_string* from the end of *string1*.
both - remove *trim_string* from the front and end of *string1*.

(iv) Length( ): In Oracle/PLSQL, the length function returns the length of the specified string.
The syntax for the length function is:

```
length(string1)
```

string1 is the string to return the length for. If *string1* is NULL, then the function returns NULL.

Q.161 Explain the disadvantages of a file processing system.

Ans: Disadvantages of File Processing Systems include:
1) Data Redundancy
2) Data Inconsistency
3) Difficult to access data
4) Data isolation
5) Atomicity problem

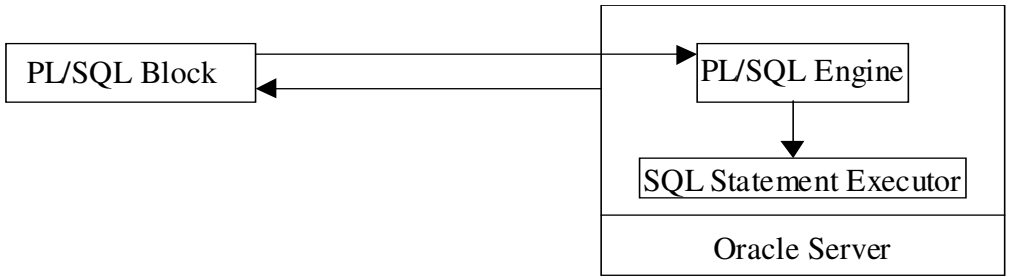
6) Concurrent access anomalies
7) Security problem

Q.162 What is data model? Explain object based and record based data models.

Ans: A data model is an abstract model that describes how data is represented and accessed.
(i) Object based data models: Similar to a relational database model, but objects, classes and inheritance are directly supported in database schemas and in the query language
(ii) Record based data models: is a database model based on first-order predicate logic. Its core idea is to describe a database as a collection of predicates over a finite set of predicate variables, describing constraints on the possible values and combinations of values.

Q.163 How is Oracle used in PL/SQL? Define the features of procedures and how they are defined. **(7)**

Ans:
PL/SQL – PL/SQL is Oracle's procedural language (PL) superset of the Structured Query Language (SQL). PL/SQL is block-structured language that enables developers to code procedures, functions and unnamed blocks that combine SQL with procedural statements. PL/SQL has its own processing engine that executes PL/SQL blocks and subprograms. If the host program does not have the PL/SQL engine, the blocks of code are sent to the Oracle server for processing. The following figure shows the PL/SQL engine as an integrated component of Oracle server. The engine executes procedure statements and passes SQL statements to the SQL statement executor in the Oracle server.



PL/SQL Engine

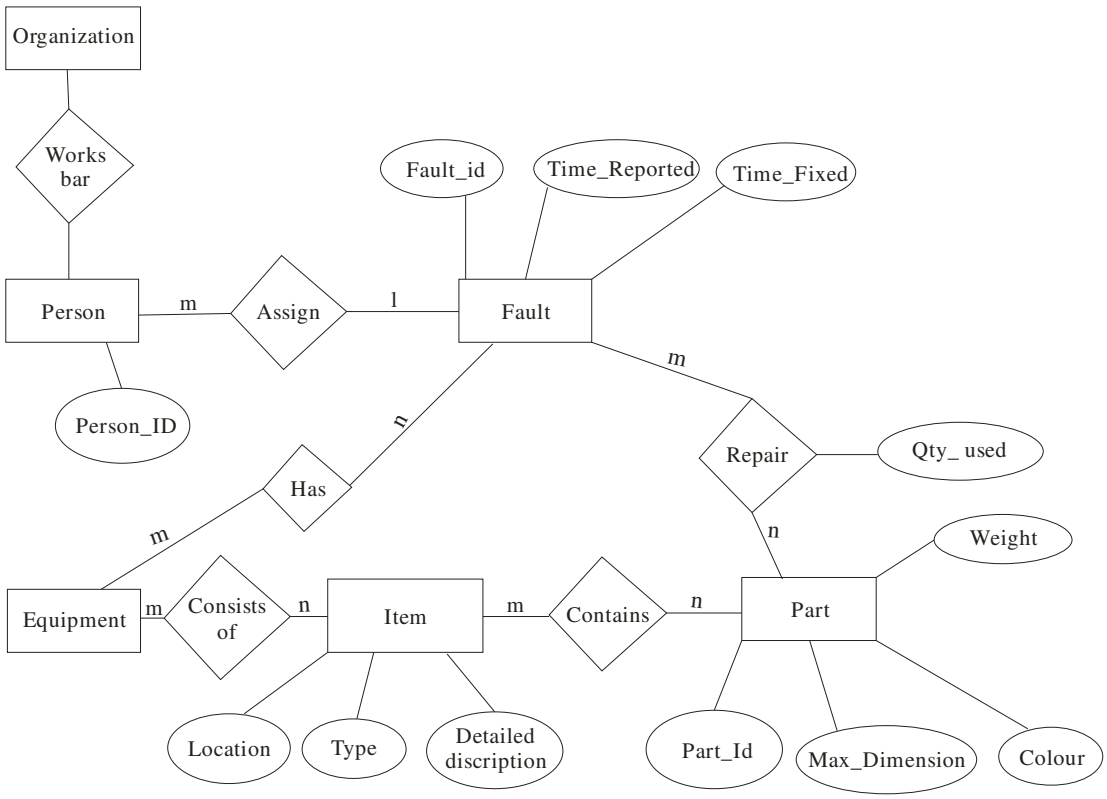
Procedures – Sophisticated business rules and application logic can be stored as procedures within Oracle. Stored procedures – groups of SQL and PL/SQL statements – allow you to move code that enforces business rules from the application to the database. As a result the code will be stored once for all applications to use. Procedures can be defined or created with the CREATE PROCEDURE command. The syntax in Oracle to define a procedure is:

```
CREATE [OR REPLACE] PROCEDURE [<user>.]<procedure_name>
   [(<arg1> [IN|OUT|IN OUT] <data type>
         [,<arg2> [IN|OUT|IN OUT] <data type>] ...)] {IS|AS}
         [<declaration>]
   BEGIN
```

```
<body>
[EXCEPTION
        WHEN <exception_name> THEN
                <error handling statements>
        [WHEN <exception_name> THEN
                <error handling statements>
        ...
        ...]
        [WHEN OTHERS THEN
                <error handling statements>]]
END;
```

Q.164 Draw an E_R Model for the following:
An organization uses number of items of a equipment to produce goods. Each item is at one LOCATION, of one TYPE and has a DETAILED_DISCRIPTION. Faults on the equipment are identified by a unique FAULT_ID and are reported at a TIME_REPORTED. Any number of persons may be assigned to a fault and work on the fault until it is fixed. The TIME_FIXED is recorded as is the TIME_SPENT by each person on a fault. Any number of parts may be used to repair a fault. The QTY_USED of each parts is recorded against the fault. Each part is identified by a PART_ID and has a given weight and MAX_DIMENSION and can have any number of colors.

Ans:



Q.165 Explain the concept of generalization and aggregation in E_R diagrams. Give one example for each one of them.

Ans: Generalization: Consider extending the entity set account by classifying accounts as being either savings-account or chequing-account.
Each of these is described by the attributes of account **plus additional attributes.**
(savings has interest-rate and chequing has overdraft-amount.)
We can express the similarities between the entity sets by **generalization.** This is the process of forming containment relationships between a **higher-level** entity set and one or more **lower-level** entity sets.
Aggregation: The E-R model cannot express relationships among relationships.
When would we need such a thing?
Consider a DB with information about employees who work on a particular project and use a number of machines doing that work. We get the E-R diagram shown in Figure 1.

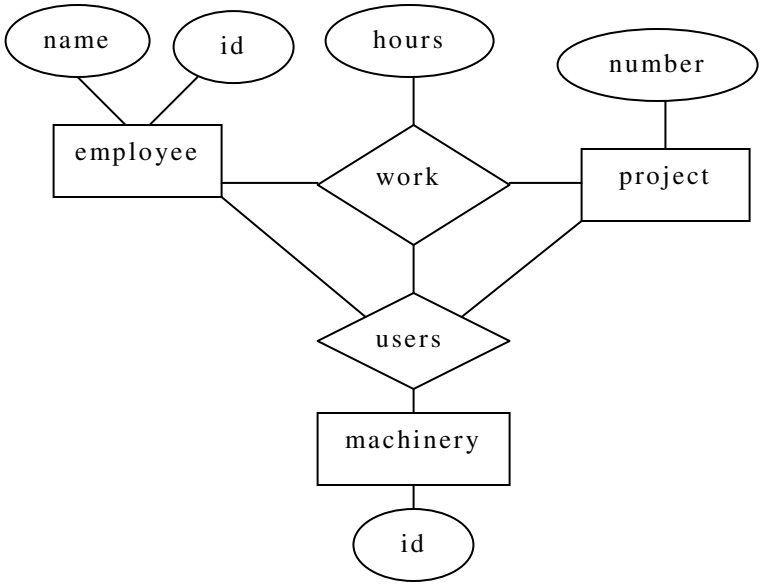


Figure 1: E-R diagram with redundant relationships

Relationship sets **work** and **uses** could be combined into a single set. However, **they** shouldn't be, as this would obscure the logical structure of this scheme.
The solution is to use aggregation.

- An abstraction through which relationships are treated as higher-level entities.
- For our example, we treat the relationship set **work** and the entity sets **employee** and **project** as a higher-level entity set called **work**.
- Figure 2 shows the E-R diagram with aggregation.

work

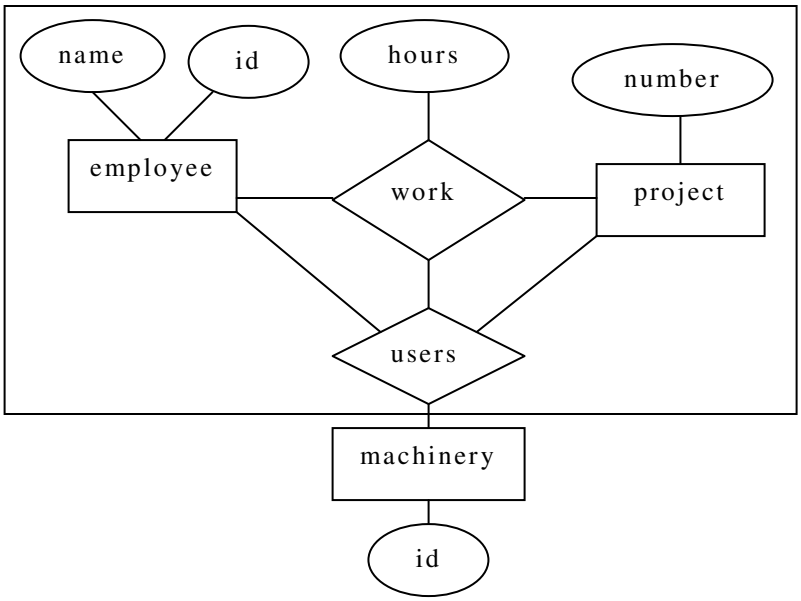


Figure 2: E-R diagram with aggregation

TYPICAL QUESTIONS & ANSWERS

PART - I

OBJECTIVE TYPE QUESTIONS

Each Question carries 2 marks.

Choose correct or the best alternative in the following:

Q.1 The NAND gate output will be low if the two inputs are

(A) 00 **(B)** 01

(C) 10 **(D)** 11

Ans: D

The NAND gate output will be low if the two inputs are 11
(The Truth Table of NAND gate is shown in Table.1.1)

X(Input)	Y(Input)	F(Output)
0	0	1
0	1	1
1	0	1
1	1	0

Table 1.1 Truth Table for NAND Gate

Q.2 What is the binary equivalent of the decimal number 368

(A) 101110000 **(B)** 110110000

(C) 111010000 **(D)** 111100000

Ans: A

The Binary equivalent of the Decimal number 368 is 101110000
(Conversion from Decimal number to Binary number is given in Table 1.2)

2	368
2	184 --- 0
2	92 --- 0
2	46 --- 0
2	23 --- 0
2	11 --- 1
2	5 --- 1
2	2 --- 1
2	1 --- 0
	0 --- 1

Table 1.2 Conversion from Decimal number to Binary number

Q.3 The decimal equivalent of hex number 1A53 is

(A) 6793 **(B)** 6739
(C) 6973 **(D)** 6379

Ans: B

The decimal equivalent of Hex Number 1A53 is 6739
(Conversion from Hex Number to Decimal Number is given below)

1	A	5	3	Hexadecimal
16^3	16^2	16^1	16^0	Weights

$(1A53)_{16} = (1X16^3) + (10 X 16^2) + (5 X 16^1) + (3 X 16^0)$
$= 4096 + 2560 + 80 + 3$
$= 6739$

Q.4 $(734)_8 = (\ )_{16}$

(A) C 1 D **(B)** D C 1
(C) 1 C D **(D)** 1 D C

Ans: D

$(734)_8 = (1\ D\ C)_{16}$
0001 | 1101 | 1100
1 D C

Q.5 The simplification of the Boolean expression $\left(\overline{\overline{A}B\overline{C}}\right) + \left(\overline{A\overline{B}C}\right)$ is

(A) 0 **(B)** 1
(C) A **(D)** BC

Ans: B

The Boolean expression is $\left(\overline{\overline{A}B\overline{C}}\right) + \left(\overline{A\overline{B}C}\right)$ is equivalent to 1

$\left(\overline{\overline{A}B\overline{C}}\right) + \left(\overline{A\overline{B}C}\right) = \overline{\overline{A}} + \overline{B} + \overline{\overline{C}} + \overline{A} + \overline{\overline{B}} + \overline{C} = A + \overline{B} + C + \overline{A} + B + \overline{C}$

$= (A + \overline{A})(B + \overline{B})(C + \overline{C}) = 1X1X1 = 1$

Q.6 The number of control lines for a 8 – to – 1 multiplexer is

(A) 2 **(B)** 3
(C) 4 **(D)** 5

Ans: B

The number of control lines for an 8 to 1 Multiplexer is 3
(The control signals are used to steer any one of the 8 inputs to the output)

Q.7 How many Flip-Flops are required for mod–16 counter?

(A) 5 **(B)** 6
(C) 3 **(D)** 4

Ans: D

The number of flip-flops is required for Mod-16 Counter is 4.

(For Mod-m Counter, we need N flip-flops where N is chosen to be the smallest number for which 2N is greater than or equal to m. In this case 24 greater than or equal to 1)

Q.8 EPROM contents can be erased by exposing it to

(A) Ultraviolet rays. **(B)** Infrared rays.
(C) Burst of microwaves. **(D)** Intense heat radiations.

Ans: A

EPROM contents can be erased by exposing it to Ultraviolet rays
(The Ultraviolet light passes through a window in the IC package to the EPROM chip where it releases stored charges. Thus the stored contents are erased).

Q.9 The hexadecimal number 'A0' has the decimal value equivalent to

(A) 80 **(B)** 256
(C) 100 **(D)** 160

Ans: D

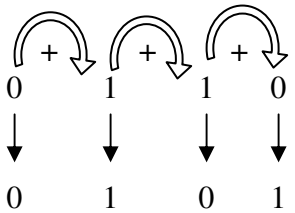
The hexadecimal number 'A0' has the decimal value equivalent to 160
(A 0
16^1 16^0 $= 10X16^1 + 0X16^0 = 160$)

Q.10 The Gray code for decimal number 6 is equivalent to

(A) 1100 **(B)** 1001
(C) 0101 **(D)** 0110

Ans: C

The Gray code for decimal number 6 is equivalent to 0101
(Decimal number 6 is equivalent to binary number 0110)



Q.11 The Boolean expression $\overline{A}.B + A.\overline{B} + A.B$ is equivalent to

(A) A + B **(B)** $\overline{A}.B$
(C) $\overline{A+B}$ **(D)** A.B

Ans: A

The Boolean expression $\overline{A}.B + A.\overline{B} + A.B$ is equivalent to A + B
$(\overline{A}.B + A.\overline{B} + A.B = B(\overline{A} + A) + A.\overline{B}$
$= B + A.\overline{B}$ $\{\because (\overline{A} + A) = 1\}$
$= A + B$ $\{\because (B + A.\overline{B}) = B + A\}$

Q.12 The digital logic family which has minimum power dissipation is

(A) TTL **(B)** RTL
(C) DTL **(D)** CMOS

Ans: D

The digital logic family which has minimum power dissipation is CMOS.
(CMOS being an unipolar logic family, occupy a very small fraction of silicon Chip area)

Q.13 The output of a logic gate is 1 when all its inputs are at logic 0. the gate is either
(A) a NAND or an EX-OR **(B)** an OR or an EX-NOR
(C) an AND or an EX-OR **(D)** a NOR or an EX-NOR

Ans: D

The output of a logic gate is 1 when all inputs are at logic 0. The gate is either a NOR or an EX-NOR .

(The truth tables for NOR and EX-NOR Gates are shown in fig.1(a) & 1(b).)

Input		Output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Fig.1(a) Truth Table for NOR Gate

Input		Output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

Fig.1(b) Truth Table for EX-NOR Gate

Q.14 Data can be changed from special code to temporal code by using
(A) Shift registers **(B)** counters
(C) Combinational circuits **(D)** A/D converters.

Ans: A

Data can be changed from special code to temporal code by using Shift Registers.
(A Register in which data gets shifted towards left or right when clock pulses are applied is known as a Shift Register.)

Q.15 A ring counter consisting of five Flip-Flops will have
(A) 5 states **(B)** 10 states
(C) 32 states **(D)** Infinite states.

Ans: A

A ring counter consisting of Five Flip-Flops will have 5 states.

Q.16 The speed of conversion is maximum in
(A) Successive-approximation A/D converter.
(B) Parallel-comparative A/D converter.
(C) Counter ramp A/D converter.
(D) Dual-slope A/D converter.

Ans: B

The speed of conversion is maximum in Parallel-comparator A/D converter
(Speed of conversion is maximum because the comparisons of the input voltage are carried out simultaneously.)

Q.17 The 2's complement of the number 1101101 is

(A) 0101110 **(B)** 0111110
(C) 0110010 **(D)** 0010011

Ans: D

The 2's complement of the number 1101101 is 0010011
(1's complement of the number 1101101 is 0010010
2's complement of the number 1101101is 0010010 + 1 =0010011)

Q.18 The correction to be applied in decimal adder to the generated sum is

(A) 00101 **(B)** 00110
(C) 01101 **(D)** 01010

Ans: B

The correction to be applied in decimal adder to the generated sum is 00110.
When the four bit sum is more than 9 then the sum is invalid. In such cases, add +6(i.e. 0110) to the four bit sum to skip the six invalid states. If a carry is generated when adding 6, add the carry to the next four bit group .

Q.19 When simplified with Boolean Algebra (x + y)(x + z) simplifies to

(A) x **(B)** x + x(y + z)
(C) x(1 + yz) **(D)** x + yz

Ans: D

When simplified with Boolean Algebra (x + y)(x + z) simplifies to x + yz
$[(x + y)(x + z)] = xx + xz + xy + yz = x + xz + xy + yz$ ($\because xx = x$)
$= x(1+z) + xy + yz = x + xy + yz$ { $\because (1+z) = 1$}
$= x(1 + y) + yz = x + yz$ { $\because (1+y) = 1$}]

Q.20 The gates required to build a half adder are

(A) EX-OR gate and NOR gate **(B)** EX-OR gate and OR gate
(C) EX-OR gate and AND gate **(D)** Four NAND gates.

Ans: C

The gates required to build a half adder are EX-OR gate and AND gate
Fig.1(d) shows the logic diagram of half adder.

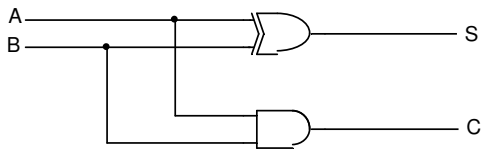


Fig.1(d) Logic diagram of Half Adder

Q.21 The code where all successive numbers differ from their preceding number by single bit is

(A) Binary code. **(B)** BCD.
(C) Excess – 3. **(D)** Gray.

Ans: D

The code where all successive numbers differ from their preceding number by single bit is Gray Code.

(It is an unweighted code. The most important characteristic of this code is that only a single bit change occurs when going from one code number to next.)

Q.22 Which of the following is the fastest logic

(A) TTL **(B)** ECL
(C) CMOS **(D)** LSI

Ans: B

ECL is the fastest logic family of all logic families.

(High speeds are possible in ECL because the transistors are used in difference amplifier configuration, in which they are never driven into saturation and thereby the storage time is eliminated.

Q.23 If the input to T-flipflop is 100 Hz signal, the final output of the three T-flipflops in cascade is

(A) 1000 Hz **(B)** 500 Hz
(C) 333 Hz **(D)** 12.5 Hz.

Ans: D

If the input to T-flip-flop is 100 Hz signal, the final output of the three T-flip-flops in cascade is 12.5 Hz

{The final output of the three T-flip-flops in cascade is

$$(T) = \frac{Frequency}{2^N} = \frac{100}{2^3} = 12.5\text{Hz}\}$$

Q.24 Which of the memory is volatile memory

(A) ROM **(B)** RAM
(C) PROM **(D)** EEPROM

Ans: B

RAM is a volatile memory

(Volatile memory means the contents of the RAM get erased as soon as the power goes off.)

Q.25 -8 is equal to signed binary number

(A) 10001000 **(B)** 00001000
(C) 10000000 **(D)** 11000000

Ans: A

- 8 is equal to signed binary number 10001000

(To represent negative numbers in the binary system, Digit 0 is used for the positive sign and 1 for the negative sign. The MSB is the sign bit followed by the magnitude bits. i.e.,

- 8 = 1000 1000

Sign Magnitude

Q.26 DeMorgan's first theorem shows the equivalence of
(A) OR gate and Exclusive OR gate.
(B) NOR gate and Bubbled AND gate.
(C) NOR gate and NAND gate.
(D) NAND gate and NOT gate

Ans: B
DeMorgan's first theorem shows the equivalence of NOR gate and Bubbled AND gate
(Logic diagrams for De Morgan's First Theorem is shown in fig.1(a)

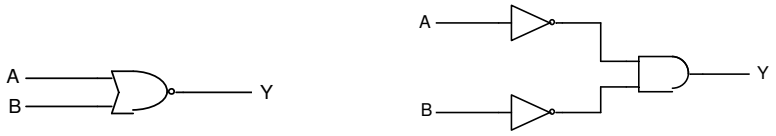


Fig.1(a) Logic Diagrams for De Morgan's First Theorem

Q.27 The digital logic family which has the lowest propagation delay time is
(A) ECL **(B)** TTL
(C) CMOS **(D)** PMOS

Ans: A
The digital logic family which has the lowest propagation delay time is ECL
(Lowest propagation delay time is possible in ECL because the transistors are used in difference amplifier configuration, in which they are never driven into saturation and thereby the storage time is eliminated).

Q.28 The device which changes from serial data to parallel data is
(A) COUNTER **(B)** MULTIPLEXER
(C) DEMULTIPLEXER **(D)** FLIP-FLOP

Ans: C
The device which changes from serial data to parallel data is demultiplexer.
(A demultiplexer takes in data from one line and directs it to any of its N outputs depending on the status of the select inputs.)

Q.29 A device which converts BCD to Seven Segment is called
(A) Encoder **(B)** Decoder
(C) Multiplexer **(D)** Demultiplexer

Ans: B

A device which converts BCD to Seven Segment is called DECODER.
(A decoder coverts binary words into alphanumeric characters.)

Q.30 In a JK Flip-Flop, toggle means

(A) Set Q = 1 and $\overline{Q}$ = 0.

(B) Set Q = 0 and $\overline{Q}$ = 1.

(C) Change the output to the opposite state.

(D) No change in output.

Ans: C

In a JK Flip-Flop, toggle means Change the output to the opposite state.

Q.31 The access time of ROM using bipolar transistors is about

(A) 1 sec **(B)** 1 msec

(C) 1 μsec **(D)** 1 nsec.

Ans: C

The access time of ROM using bipolar transistors is about 1 μ sec.

Q.32 The A/D converter whose conversion time is independent of the number of bits is

(A) Dual slope **(B)** Counter type

(C) Parallel conversion **(D)** Successive approximation.

Ans: C

The A/D converter whose conversion time is independent of the Number of bits is Parallel conversion.
(This type uses an array of comparators connected in parallel and comparators compare the input voltage at a particular ratio of the reference voltage).

Q.33 When signed numbers are used in binary arithmetic, then which one of the following notations would have unique representation for zero.

(A) Sign-magnitude. **(B)** 1's complement.

(C) 2's complement. **(D)** 9's complement.

Ans: A

Q.34 The logic circuit given below (Fig.1) converts a binary code $y_1y_2y_3$ into

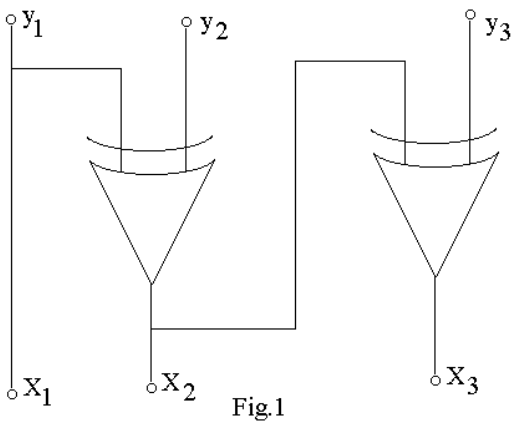


Fig.1

(A) Excess-3 code. **(B)** Gray code.
(C) BCD code. **(D)** Hamming code.

Ans: B

Gray code as

X1=Y1, X2=Y1 XOR Y2 , X3=Y1 XOR Y2 XOR Y3

For	Y1	Y2	Y3	X1	X2	X3
	0	0	0	0	0	0
	0	0	1	0	0	1
	0	1	0	0	1	1
	0	1	1	0	1	0

Q.35 The logic circuit shown in the given fig.2 can be minimised to

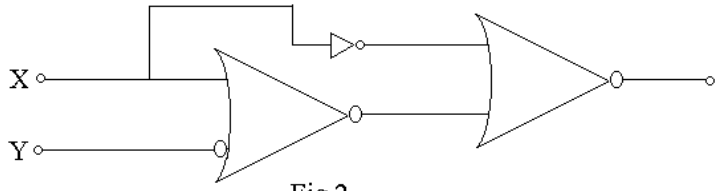
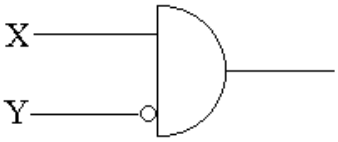
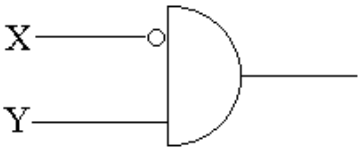


Fig.2

(A) 

(B) 

(C) 

(D) 

Ans: D

As output of the logic circuit is
Y=(X+Y')'+(X'+(X+Y')')'
(X+Y')'=X'Y Using DE Morgan's
Now this is one of input of 2nd gate.
F=(A+X')'=A'X=[(X'Y)'.X]
=[(X+Y')X]=X+XY'=X(Y')
=X

Q.36 In digital ICs, Schottky transistors are preferred over normal transistors because of their
(A) Lower Propagation delay. **(B)** Higher Propagation delay.
(C) Lower Power dissipation. **(D)** Higher Power dissipation.

Ans: A

Lower propagation delay as shottky transistors reduce the storage time delay by preventing the transistor from going deep into saturation.

Q.37 The following switching functions are to be implemented using a Decoder:

$f_1 = \sum m(1,2,4,8,10,14)$ $f_2 = \sum m(2,5,9,11)$ $f_3 = \sum m(2,4,5,6,7)$

The minimum configuration of the decoder should be

(A) 2 – to – 4 line. **(B)** 3 – to – 8 line.
(C) 4 – to – 16 line. **(D)** 5 – to – 32 line.

Ans: C

4 to 16 line decoder as the minterms are ranging from 1 to 14.

Q.38 A 4-bit synchronous counter uses flip-flops with propagation delay times of 15 ns each. The maximum possible time required for change of state will be

(A) 15 ns. **(B)** 30 ns.
(C) 45 ns. **(D)** 60 ns.

Ans: A

15 ns because in synchronous counter all the flip-flops change state at the same time.

Q.39 Words having 8-bits are to be stored into computer memory. The number of lines required for writing into memory are

(A) 1. **(B)** 2.
(C) 4. **(D)** 8.

Ans: D

Because 8-bit words required 8 bit data lines.

Q.40 In successive-approximation A/D converter, offset voltage equal to $\frac{1}{2}$ LSB is added to the D/A converter's output. This is done to

(A) Improve the speed of operation.
(B) Reduce the maximum quantization error.
(C) Increase the number of bits at the output.
(D) Increase the range of input voltage that can be converted.

Ans: B

Q.41 The decimal equivalent of Binary number 11010 is

(A) 26. **(B)** 36.
(C) 16. **(D)** 23.

Ans: A

$$11010 = 1 \text{ X } 2^4 + 1 \text{ X } 2^3 + 0 \text{ X } 2^2 + 1 \text{ X } 2^1 = 26$$

Q.42 1's complement representation of decimal number of -17 by using 8 bit representation is

(A) 1110 1110 **(B)** 1101 1101
(C) 1100 1100 **(D)** 0001 0001

Ans: A

$(17)_{10} = (10001)_2$

In 8 bit = 00010001

1's Complement = 11101110

Q.43 The excess 3 code of decimal number 26 is

(A) 0100 1001 **(B)** 01011001
(C) 1000 1001 **(D)** 01001101

Ans: B

$(26)_{10}$ in BCD is (00100110) BCD
Add 011 to each BCD 01011001 for excess – 3

Q.44 How many AND gates are required to realize Y = CD+EF+G

(A) 4 **(B)** 5
(C) 3 **(D)** 2

Ans: D

To realize Y = CD + EF + G
Two AND gates are required (for CD & EF).

Q.45 How many select lines will a 16 to 1 multiplexer will have

(A) 4 **(B)** 3
(C) 5 **(D)** 1

Ans: A

In 16 to 1 MUX four select lines will be required to select 16 (2^4) inputs.

Q.46 How many flip flops are required to construct a decade counter

(A) 10 **(B)** 3
(C) 4 **(D)** 2

Ans: C

Decade counter counts 10 states from 0 to 9 (i.e. from 0000 to 1001)
Thus four FlipFlop's are required.

Q.47 Which TTL logic gate is used for wired ANDing

(A) Open collector output **(B)** Totem Pole
(C) Tri state output **(D)** ECL gates

Ans: A

Open collector output.

Q.48 CMOS circuits consume power

(A) Equal to TTL **(B)** Less than TTL
(C) Twice of TTL **(D)** Thrice of TTL

Ans: B

As in CMOS one device is ON & one is Always OFF so power consumption is low.

Q.49 In a RAM, information can be stored

(A) By the user, number of times.

(B) By the user, only once.
(C) By the manufacturer, a number of times.
(D) By the manufacturer only once.

Ans: A

RAM is used by the user, number of times.

Q.50 The hexadecimal number for $(95.5)_{10}$ is

(A) $(5F.8)_{16}$ **(B)** $(9A.B)_{16}$
(C) $(2E.F)_{16}$ **(D)** $(5A.4)_{16}$

Ans: A

$(95.5)_{10} = (5F.8)_{16}$

Integer part

16	95	
16	5	15 ↑
	0	5

Fractional part
0.5x16=8.0

Q.51 The octal equivalent of $(247)_{10}$ is

(A) $(252)_8$ **(B)** $(350)_8$
(C) $(367)_8$ **(D)** $(400)_8$

Ans: C

$(247)_{10} = (367)_8$

8	247	
8	30	↑ 7
8	3	6
	0	3

Q.52 The chief reason why digital computers use complemented subtraction is that it

(A) Simplifies the circuitry.
(B) Is a very simple process.
(C) Can handle negative numbers easily.
(D) Avoids direct subtraction.

Ans: C

Using complement method negative numbers can also be subtracted.

Q.53 In a positive logic system, logic state 1 corresponds to

(A) positive voltage **(B)** higher voltage level
(C) zero voltage level **(D)** lower voltage level

Ans: B

We decide two voltages levels for positive digital logic. Higher voltage represents logic 1 & a lower voltage represents logic 0.

Q.54 The commercially available 8-input multiplexer integrated circuit in the TTL family is
(A) 7495. **(B)** 74153.
(C) 74154. **(D)** 74151.

Ans: B
MUX integrated circuit in TTL is 74153.

Q.55 CMOS circuits are extensively used for ON-chip computers mainly because of their extremely
(A) low power dissipation. **(B)** high noise immunity.
(C) large packing density. **(D)** low cost.

Ans: C
Because CMOS circuits have large packing density.

Q.56 The MSI chip 7474 is
(A) Dual edge triggered JK flip-flop (TTL).
(B) Dual edge triggered D flip-flop (CMOS).
(C) Dual edge triggered D flip-flop (TTL).
(D) Dual edge triggered JK flip-flop (CMOS).

Ans: C
MSI chip 7474 dual edge triggered D Flip-Flop.

Q.57 Which of the following memories stores the most number of bits
(A) a $5M \times 8$ memory. **(B)** a $1M \times 16$ memory.
(C) a $5M \times 4$ memory. **(D)** a $1M \times 12$ memory.

Ans: A
5Mx8 = 5 x 220 x 8 = 40M (max)

Q.58 The process of entering data into a ROM is called
(A) burning in the ROM **(B)** programming the ROM
(C) changing the ROM **(D)** charging the ROM

Ans: B
The process of entering data into ROM is known as programming the ROM.

Q.59 When the set of input data to an even parity generator is 0111, the output will be
(A) 1 **(B)** 0
(C) Unpredictable **(D)** Depends on the previous input

Ans: B
In even parity generator if number of 1 is odd then output will be zero.

Q.60 The number 140 in octal is equivalent to

(A) $(96)_{10}$. **(B)** $(86)_{10}$.

(C) $(90)_{10}$. **(D)** none of these.

Ans: A

$(140)_8 = (96)_{10}$

$1 \times 8^2 + 4 \times 8 + 0 \times 1 = 64 + 32 = 96$

Q.61 The NOR gate output will be low if the two inputs are

(A) 00 **(B)** 01

(C) 10 **(D)** 11

Ans: B, C, or D

O/P is low if any of the I/P is high

Q.62 Which of the following is the fastest logic?

(A) ECL **(B)** TTL

(C) CMOS **(D)** LSI

Ans: A

Q.63 How many flip-flops are required to construct mod 30 counter

(A) 5 **(B)** 6

(C) 4 **(D)** 8

Ans: A

Mod - 30 counter +/- needs 5 Flip-Flop as $30 < 2^5$

Mod - N counter counts total ' N ' number of states.

To count 'N' distinguished states we need minimum n FlipFlop's as [$N = 2^n$]

For eg. Mod 8 counter requires 3 Flip-Flop's ($8 = 2^3$)

Q.64 How many address bits are required to represent a 32 K memory

(A) 10 bits. **(B)** 12 bits.

(C) 14 bits. **(D)** 16 bits.

Ans: D

$32K = 2^5 \times 2^{10} = 2^{15}$,

Thus 15 address bits are required, Only 16 bits can address it.

Q.65 The number of control lines for 16 to 1 multiplexer is

(A) 2. **(B)** 4.

(C) 3. **(D)** 5.

Ans: B

As $16 = 2^4$, 4 Select lines are required.

Q.66 Which of following requires refreshing?

(A) SRAM. **(B)** DRAM.

(C) ROM. **(D)** EPROM.

Ans: B

Q.67 Shifting a register content to left by one bit position is equivalent to
(A) division by two. **(B)** addition by two.
(C) multiplication by two. **(D)** subtraction by two.

Ans:C

Q.68 For JK flip flop with J=1, K=0, the output after clock pulse will be
(A) 0. **(B)** 1.
(C) high impedance. **(D)** no change.

Ans: B

Q.69 Convert decimal 153 to octal. Equivalent in octal will be
(A) $(231)_8$. **(B)** $(331)_8$.
(C) $(431)_8$. **(D)** none of these.

Ans: A
$(153)_{10} = (231)_8$

8	153	1
8	19	3
8	2	2

Q.70 The decimal equivalent of $(1100)_2$ is
(A) 12 **(B)** 16
(C) 18 **(D)** 20

Ans: A
$(1100)_2 = (12)_{10}$

Q.71 The binary equivalent of $(FA)_{16}$ is
(A) 1010 1111 **(B)** 1111 1010
(C) 10110011 **(D)** none of these

Ans: B
$(FA)_{16} = (11111010)_{10}$

Q.72 The output of SR flip flop when S=1, R=0 is
(A) 1 **(B)** 0
(C) No change **(D)** High impedance

Ans: A

As for the SR flip-flop S=set input R=reset input ,when S=1, R=0, Flip-flop will be set.

Q.73 The number of flip flops contained in IC 7490 is

(A) 2. **(B)** 3.
(C) 4. **(D)** 10.

Ans: A

Q.74 The number of control lines for 32 to 1 multiplexer is

(A) 4. **(B)** 5.
(C) 16. **(D)** 6.

Ans: B

The number of control lines for 32 (2^5) and to select one input among them total 5 select lines are required.

Q.75 How many two-input AND and OR gates are required to realize Y=CD+EF+G

(A) 2,2. **(B)** 2,3.
(C) 3,3. **(D)** none of these.

Ans: A

Y=CD+EF+G
Number of two input AND gates=2
Number of two input OR gates = 2
One OR gate to OR CD and EF and next to OR of G & output of first OR gate.

Q.76 Which of following can not be accessed randomly

(A) DRAM. **(B)** SRAM.
(C) ROM. **(D)** Magnetic tape.

Ans: D

Magnetic tape can only be accessed sequentially.

Q.77 The excess-3 code of decimal 7 is represented by

(A) 1100. **(B)** 1001.
(C) 1011. **(D)** 1010.

Ans: D

An excess 3 code is always equal to the binary code +3

Q.78 When an input signal A=11001 is applied to a NOT gate serially, its output signal is

(A) 00111. **(B)** 00110.
(C) 10101. **(D)** 11001.

Ans: B

As A=11001 is serially applied to a NOT gate, first input applied will be LSB 00110.

Q.79 The result of adding hexadecimal number A6 to 3A is

(A) DD. **(B)** E0.
(C) F0. **(D)** EF.

Ans: B

Q.80 A universal logic gate is one, which can be used to generate any logic function. Which of the following is a universal logic gate?

(A) OR **(B)** AND
(C) XOR **(D)** NAND

Ans: D

NAND can generate any logic function.

Q.81 The logic 0 level of a CMOS logic device is approximately

(A) 1.2 volts **(B)** 0.4 volts
(C) 5 volts **(D)** 0 volts

Ans: D

CMOS logic low level is 0 volts approx.

Q.82 Karnaugh map is used for the purpose of

(A) Reducing the electronic circuits used.
(B) To map the given Boolean logic function.
(C) To minimize the terms in a Boolean expression.
(D) To maximize the terms of a given a Boolean expression.

Ans: C

Q.83 A full adder logic circuit will have

(A) Two inputs and one output.
(B) Three inputs and three outputs.
(C) Two inputs and two outputs.
(D) Three inputs and two outputs.

Ans: D

A full adder circuit will add two bits and it will also accounts the carry input generated in the previous stage. Thus three inputs and two outputs (Sum and Carry) are there.

Q.84 An eight stage ripple counter uses a flip-flop with propagation delay of 75 nanoseconds. The pulse width of the strobe is 50ns. The frequency of the input signal which can be used for proper operation of the counter is approximately

(A) 1 MHz. **(B)** 500 MHz.
(C) 2 MHz. **(D)** 4 MHz.

Ans: A

Maximum time taken for all flip-flops to stabilize is 75ns x 8 + 50 = 650ns. Frequency of operation must be less than 1/650ns = 1.5 MHz.

Q.85 The output of a JK flipflop with asynchronous preset and clear inputs is '1'. The output can be changed to '0' with one of the following conditions.

(A) By applying J = 0, K = 0 and using a clock.
(B) By applying J = 1, K = 0 and using the clock.
(C) By applying J = 1, K = 1 and using the clock.
(D) By applying a synchronous preset input.

Ans: C

Preset state of JK Flip-Flop =1
With J=1 K=1 and the clock next state will be complement of the present state.

Q.86 The information in ROM is stored

(A) By the user any number of times.
(B) By the manufacturer during fabrication of the device.
(C) By the user using ultraviolet light.
(D) By the user once and only once.

Ans: B

Q.87 The conversation speed of an analog to digital converter is maximum with the following technique.

(A) Dual slope AD converter.
(B) Serial comparator AD converter.
(C) Successive approximation AD converter.
(D) Parallel comparator AD converter.

Ans: D

Q.88 A weighted resistor digital to analog converter using N bits requires a total of

(A) N precision resistors. **(B)** 2N precision resistors.
(C) N + 1 precision resistors. **(D)** N – 1 precision resistors.

Ans: A

Q.89 The 2's complement of the number 1101110 is

(A) 0010001. **(B)** 0010001.
(C) 0010010. **(D)** None.

Ans: C

1's complement of 1101110 is = 0010001
Thus 2's complement of 1101110 is = 0010001 + 1 = 0010010

Q.90 The decimal equivalent of Binary number 10101 is

(A) 21 **(B)** 31
(C) 26 **(D)** 28

Ans: A

$1x2^4 + 0x2^3 + 1x2^2 + 0x2^1 + 1x2^0$
$= 16 + 0 + 4 + 0 + 1 = 21.$

Q.91 How many two input AND gates and two input OR gates are required to realize Y = BD+CE+AB

(A) 1, 1 **(B)** 4, 2
(C) 3, 2 **(D)** 2, 3

Ans: A

There are three product terms, so three AND gates of two inputs are required.
As only two input OR gates are available, so two OR gates are required to get the logical sum of three product terms.

Q.92 How many select lines will a 32:1 multiplexer will have

(A) 5. **(B)** 8.
(C) 9. **(D)** 11.

Ans: A

For 32 inputs, 5 select lines will be required, as $2^5 = 32$.

Q.93 How many address bits are required to represent 4K memory

(A) 5 bits. **(B)** 12 bits.
(C) 8 bits. **(D)** 10 bits.

Ans: B

For representing 4K memory, 12 address bits are required as
$4K = 2^2 \times 2^{10} = 2^{12}$ $(1K = 1024 = 2^{10})$

Q.94 For JK flipflop J = 0, K=1, the output after clock pulse will be

(A) 1. **(B)** no change.
(C) 0. **(D)** high impedance.

Ans: C

J=0, K=1, these inputs will reset the flip-flop after the clock pulse. So whatever be the previous output, the next state will be 0.

Q.95 Which of following are known as universal gates

(A) NAND & NOR. **(B)** AND & OR.
(C) XOR & OR. **(D)** None.

Ans: A

NAND & NOR are known as universal gates, because any digital circuit can be realized completely by using either of these two gates.

Q.96 Which of the following memories stores the most number of bits

(A) 64K×8 memory. **(B)** 1M×8 memory.

(C) $32M \times 8$ memory. **(D)** 64×6 memory.

Ans: C

32M x 8 stores most number of bits

$2^5 \times 2^{20} = 2^{25}$ $(1M = 2^{20} = 1K \times 1K = 2^{10} \times 2^{10})$

Q.97 Which of following consume minimum power

(A) TTL. **(B)** CMOS.

(C) DTL. **(D)** RTL.

Ans: B

CMOS consumes minimum power as in CMOS one p-MOS & one n-MOS transistors are connected in complimentary mode, such that one device is ON & one is OFF.

PART – II

NUMERICALS

Q.1 Convert the octal number 7401 to Binary. **(4)**

Ans:

Conversion of Octal number 7401 to Binary:

Each octal digit represents 3 binary digits. To convert an octal number to binary number, each octal digit is replaced by its 3 digit binary equivalent shown below.

7 4 0 1

↓ ↓ ↓ ↓

111 100 000 001

Thus, $(7401)_8 = (111100000001)_2$

Q.2 Find the hex sum of $(93)_{16} + (DE)_{16}$. **(4)**

Ans:

Hex Sum of $(93)_{16} + (DE)_{16}$

Convert Hexadecimal numbers 93 and DE to its binary equivalent shown below:-

93 → 10010011

DE → 11011110

101110001 → 171

Thus $(93)_{16} + (DE)_{16} = (171)_{16}$

Q.3 Perform 2's complement subtraction of $(7)_{10} - (11)_{10}$. **(4)**

Ans:

2's Complements Subtraction of $(7)_{10} - (11)_{10}$

First convert the decimal numbers 7 and 11 to its binary equivalents.

$(7)_{10} = (0111)_2$

$(11)_{10} = (1011)_2$ in 4-bit system

Then find out the 2's complement for 1011 i.e.,

1's Complement of 1011 is 0100

2's Complement of 1011 is 0101

So, $(7)_{10} - (11)_{10}$ = 0111

0101

1100

Since there is no carry over flow occurring in the summation, the result is a negative number, to find out its magnitude, 2's Complement of the result must be found.
2's Complement of 1100 is 0011

```
      1
--------
 0100
--------
```

Here the answer is $(-4)_{10}$ (or) in 2's complement it is 1100.

Q.4 What is the Gray equivalent of $(25)_{10}$. **(2)**

Ans:

Gray equivalent of $(25)_{10}$
The binary equivalent of Decimal number 25 is $(00100101)_2$

1. The left most bit (MSB) in gray code is the same as the left most in binary
2. Add the left most bit to the adjacent bit
3. Add the next adjacent pair and so on., Discard if we get a carry.

0 + 0 + 1 + 0 + 0 + 1 + 0 + 1
↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
0 0 1 1 0 1 1 1 ← Gray Number

Q.5 Evaluate x = $\overline{A}.B + C(\overline{A.D})$ using the convention A = True and B = False. **(4)**

Ans:

Evaluate x = $\overline{A}.B + C(\overline{A.D})$
= $\overline{A}B + C(\overline{A} + \overline{D})$ (Since $\overline{A.D} = \overline{A} + \overline{D}$ by using Demorgan's Law)
= $\overline{A}.B + C.\overline{A} + C.\overline{D}$
By using the given convention, A = True = 1; B = False = 0
= $\overline{1}.0 + C.\overline{1} + C.\overline{D} = 0 + 0 + C.\overline{D} = C.\overline{D}$

Q.6 Simplify the Boolean expression F = C(B + C)(A + B + C). **(6)**

Ans:

Simplify the Boolean Expression F = C (B +C) (A+B+C)
F = C (B+C) (A+B+C)
= CB + CC [(A+B+C)]
= CB + C [(A+B+C)] (∵ CC = C)
= CBA + CBB + CBC + CA + CB + CC
= ABC + CB + CB + CA + CB + CC (∵ CBB =CB & CBC = CB)
= ABC + CB + CA + C (∵ CB+CB+CB = CB; CC = C)
= ABC + BC + C (1+A)
= ABC + BC + C (∵ 1+A = 1)
= ABC + C (1+B)

$= ABC + C \quad (\because 1+B = 1)$
$= C\,(1+AB) = C\ \{\because (1+AB)=1\}$

Q.7 Simplify the following expression into sum of products using Karnaugh map $F(A,B,C,D) = \sum(1,3,4,5,6,7,9,12,13)$ **(7)**

Ans:

Simplification of the following expression into sum of products using Karnaugh Map:

$F(A,B,C,D) = \Sigma\,(1,3,4,5,6,7,9,12,13)$

Karnaugh Map for the expression $F(A,B,C,D) = \Sigma\,(1,3,4,5,6,7,9,12,13)$ is shown in Fig.4(a). The grouping of cells is also shown in the Figure.

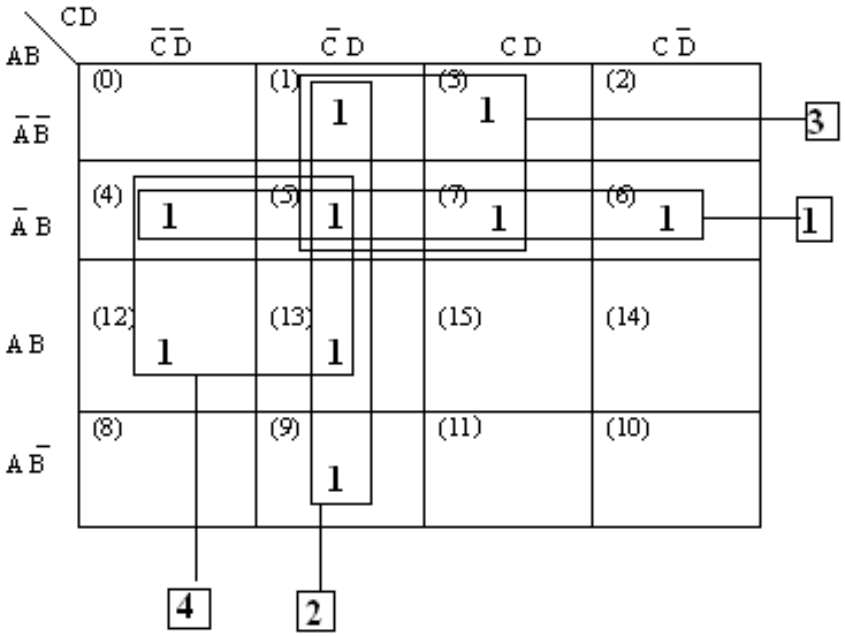


Fig 4(a)

The equations for (1) is $\bar{A}B$; (2) is $\bar{C}D$; (3) is $\bar{A}D$; (4) is $B\bar{C}$

Hence, the Simplified Expression for the above Karnaugh map is

$F(A,B,C,D) = \bar{A}B + \bar{C}D + \bar{A}D + B\bar{C}$

$= \bar{A}(B + D) + \bar{C}(B + D)$

Q.8 Simplify and draw the logic diagram for the given expression

$F = \overline{ABC} + \overline{AB}C + \bar{A}B\bar{C} + A\overline{BC} + A\bar{B}C$. **(7)**

Ans:

Simplification of the logic expression

$F = \overline{ABC} + \overline{AB}\,C + \bar{A}B\bar{C} + A\,\overline{BC} + A\bar{B}C$

$F = \overline{ABC} + \overline{AB}\,C + \bar{A}B\bar{C} + A\,\overline{BC} + A\bar{B}C$

$F = \overline{A} + \overline{B} + \overline{C} + (\overline{A} + \overline{B})C + \overline{A}\,B\,\overline{C} + A\,(\overline{B} + \overline{C}) + A\,\overline{B}\,C$

$(\because \overline{ABC} = \overline{A} + \overline{B} + \overline{C}$ and $\overline{AB} = \overline{A} + \overline{B}$ by using Demorgan's Law)

$= \overline{A} + \overline{B} + \overline{C} + \overline{A}\,C + \overline{B}\,C + \overline{A}\,B\,\overline{C} + A\,\overline{B} + A\,\overline{C} + A\,\overline{B}\,C$

$= \overline{A} + \overline{A}\,C + \overline{B} + \overline{B}\,C + \overline{C} + A\,\overline{C} + \overline{A}\,B\,\overline{C} + A\,\overline{B} + A\,\overline{B}\,C$

$= \overline{A}\,(1+C) + \overline{B}\,(1 + C) + \overline{C}\,(1 + A) + \overline{A}\,B\,\overline{C} + A\,\overline{B} + A\,\overline{B}\,C$

$= \overline{A} + \overline{B} + \overline{C} + \overline{A}\,B\,\overline{C} + A\,\overline{B} + A\,\overline{B}\,C$ $\{\because (1+C) = 1$ and $(1+A) = 1\}$

$= (\overline{A} + A\,\overline{B}) + \overline{B}\,(1 + AC) + \overline{C}\,(1 + \overline{A}\,B)$

$= (\overline{A} + \overline{B}) + \overline{B} + \overline{C}$ $\{\because (\overline{A} + A\,\overline{B}) = (\overline{A} + \overline{B});$ $(1+AC) = 1$ and $(1 + \overline{A}\,B) = 1\}$

$F = (\overline{A} + \overline{B} + \overline{C})$ $(\because \overline{B} + \overline{B} = \overline{B})$

The logic diagram for the simplified expression $F = (\overline{A} + \overline{B} + \overline{C})$ is given in fig.5(a)

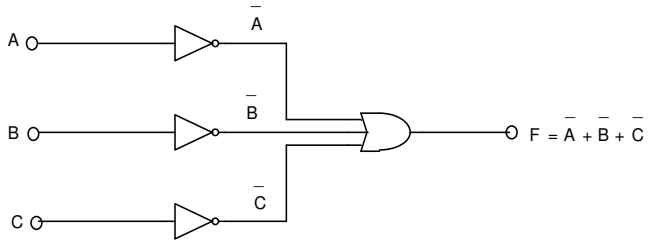


Fig.5(a) Logic diagram for the expression $F = (\overline{A} + \overline{B} + \overline{C})$

Q.9 Determine the binary numbers represented by the following decimal numbers. **(6)**
(i) 25.5 (ii) 10.625 (iiii) 0.6875

Ans:

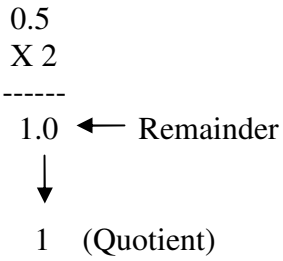
(i) Conversion of decimal number 25.5 into binary number:

Here integer part is 25 and fractional part is 0.5. First convert the integer part 25 into its equivalent binary number i.e., divide 25 by 2 till the quotient becomes 0 shown in table 2(a)

	Quotient	Remainder
$\frac{25}{2}$	12	1
$\frac{12}{2}$	6	0
$\frac{6}{2}$	3	0
$\frac{3}{2}$	1	1
$\frac{1}{2}$	0	1

Table 2(a)

So, integer part $(25)_{10}$ is equivalent to the binary number 11001. Next convert fractional part 0.5 into binary form i.e., multiply the fractional part 0.5 by 2 till you get remainder as 0



The decimal fractional part 0.5 is equivalent to binary number 0.1. Hence, the decimal number 25.5 is equal to the binary number 11001.1

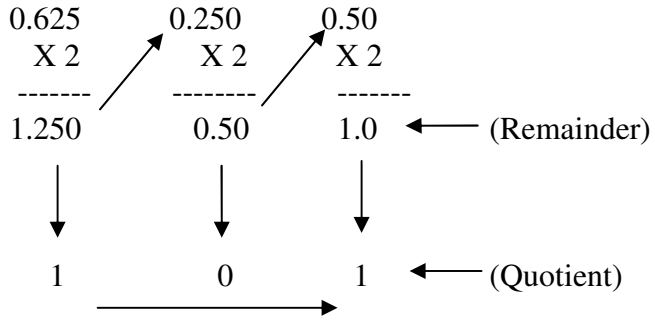
(ii) Conversion of decimal number 10.625 into binary number:

Here integer part is 10 and fractional part is 0.625. First convert the decimal number 10 into its equivalent binary number i.e., divide 10 by 2 till the quotient becomes 0 shown in table 2(b)

	Quotient	Remainder
$\frac{10}{2}$	5	0
$\frac{5}{2}$	2	1
$\frac{2}{2}$	1	0
$\frac{1}{2}$	0	1

Table 2(b)

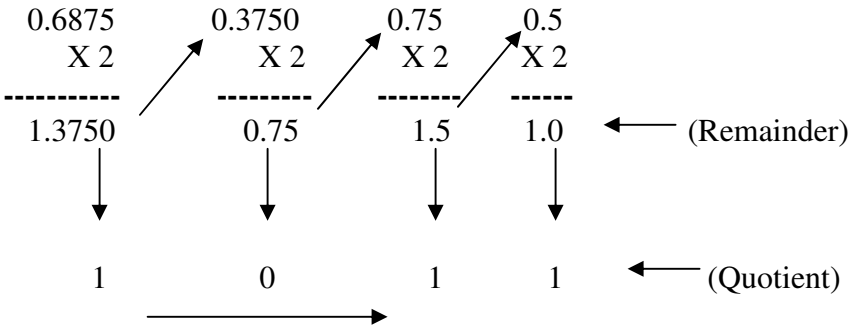
So, the integer part 10 is equal to binary number 1010. Next convert the decimal fractional part 0.625 into its binary form i.e., multiply 0.625 by 2 till the remainder becomes 0



So, the decimal fractional part 0.625 is equal to binary number 0.101. Hence the decimal number **10.625** is equal to binary number **1010.101**.

(iii)Conversion of fractional number 0.6875 into its equivalent binary number:

Multiply the fractional number 0.6875 by 2 till the remainder becomes 0 i.e.,



So, the decimal fractional number **0.6875** is equal to binary number **0.1011**.

Q.10 Perform the following subtractions using 2's complement method. **(8)**
(i) 01000 – 01001 (ii) 01100 – 00011 (iii) 0011.1001 – 0001.1110

Ans:

(i) Subtraction of 01000-01001: 1's complement of 01001 is 10110 and 2's complement is
10110+ 1 =10111. Hence

```
  01000 =  01000
- 01001 =  +10111   (2's complement)
-------------------------
          11111     (Summation)
-------------------------
```

Since the MSB of the sum is 1, which means the result is negative and it is in 2's complement form. So, 2's complement of 1111 =00001= $(1)_{10.}$ Therefore, the result is – 1.

(ii) Subtraction of 01100-00011: 1's complement of 00011 is 11100 and 2's complement is 11100 + 1 = 11101. Hence

```
  01100  =  01100
– 00011  = + 11101 (2's complement)
-----------------------------------------------
         1 01001 = + 9
           ↑
         Ignore
-----------------------------------------------
```

If a final carry is generated discard the carry and the answer is given by the remaining bits Which is positive i.e., $(1001)_2 = (+ 9)_{10}$

(iii) Subtraction of 0011.1001 – 0001.1110: 1's complement of 0001.1110 is 1110.0001 and its 2's complement is 1110.0010.

```
  0011.1001 =  0011.1001
- 0001.1110 = + 1110.1011 (2's complement)
-----------------------------------------
          1 0001.101I = + 1 .68625
            ↑
          Ignore
```

If a final carry is generated discard the carry and the answer is given by the remaining bits which is positive i.e., $(0001.1011)_2 = (+ 1.68625)_{10}$

Q.11 Simplify the expressions using Boolean postulates **(9)**

(i) $\overline{\overline{X\overline{Y} + XYZ} + X(Y + X\overline{Y})}$ **(ii)** Y = (A + B)($\overline{A}$ + C)(B + C)

(iii) XY + $\overline{XZ}$ + X$\overline{Y}$Z (XY + Z)

Ans:

(i) $\overline{\overline{X\overline{Y} + XYZ} + X(Y + X\overline{Y})}$

$= \overline{\overline{X\overline{Y} + XYZ} + X(Y + X\overline{Y})}$

$= \overline{\overline{X(\overline{Y} + YZ)} + X(Y + X\overline{Y})}$

$= \overline{\overline{X(\overline{Y} + Z)} + X(Y + X)}$

(Because $\overline{Y} + YZ = \overline{Y} + Z$ and $Y + X\overline{Y} = Y + X$)

$= \overline{\overline{X\overline{Y} + XZ} + XY + XX}$

$= \overline{\overline{X\overline{Y} + XZ} + XY + X}$ (Because XX=X)

$= \overline{\overline{X\overline{Y} + XZ} + X(1 + Y)}$

$= \overline{\overline{X\overline{Y} + XZ} + X}$ (Because (1+Y=1)

$= \overline{(\overline{X} + Y)(\overline{X} + \overline{Z}) + X}$ (Because $\overline{XY} = \overline{X} + \overline{Y}$)

$= \overline{\overline{X}\,\overline{X} + \overline{X}\,\overline{Z} + Y\overline{X} + Y\overline{Z} + X}$

$= \overline{\overline{X} + \overline{X}\,\overline{Z} + Y\overline{X} + Y\overline{Z} + X}$ (Because $\overline{X}\ \overline{X} = \overline{X}$)

$= \overline{\overline{X}(1 + \overline{Z} + Y) + Y\overline{Z} + X}$

$= \overline{\overline{X} + Y\overline{Z} + X}$

$= \overline{(\overline{X} + X) + Y\overline{Z}}$

$= \overline{1 + Y\overline{Z}}$ (Because $\overline{X} + X = 1$)

$= \overline{1} = 0$ (Because $1 + Y\overline{Z} = 1$)

(ii) Y = (A + B)($\overline{A}$ + C)(B + C)

Y = (A + B)($\overline{A}$ + C)(B + C)

= (A $\overline{A}$ + AC + B $\overline{A}$ + BC) (B + C)

= (AC + B $\overline{A}$ + BC) (B + C) (Because A $\overline{A}$ = 0)

= ABC + BB $\overline{A}$ + BBC + ACC + B $\overline{A}$ C + BCC

= ABC + B $\overline{A}$ + BC + AC + B $\overline{A}$ C + BC (Because BB = B)

= ABC + AC + B $\overline{A}$ + B $\overline{A}$ C + BC (Because BC + BC = BC)

=AC (B+1) + B $\overline{A}$ + BC ($\overline{A}$ +1)

= AC + B $\overline{A}$ + BC (Because B + 1 = 1 and $\overline{A}$ + 1 = 1)

= AC + B $\overline{A}$ + BC (A + $\overline{A}$) (Because A + $\overline{A}$ = 1)

= AC + B $\overline{A}$ + BCA + BC $\overline{A}$

= AC(1 + B) + B $\overline{A}$ (1 + C)

$= AC + B\overline{A}$ {Because (1 + B) = 1 and (1 + C) =1}

(iii) $XY + \overline{XZ} + X\overline{Y}Z\,(XY + Z)$

$= XY + \overline{XZ} + X\overline{Y}Z\,(XY + Z)$

$= XY + \overline{XZ} + XXY\overline{Y}Z + X\overline{Y}ZZ$

$= XY + \overline{XZ} + X\overline{Y}Z$ (Because $Y\overline{Y} = 0$ & $ZZ = Z$)

$= XY + \overline{X} + \overline{Z} + X\overline{Y}Z$ (Because $\overline{XZ} = \overline{X} + \overline{Z}$)

$= \overline{X} + XY + \overline{Z} + X\overline{Y}Z$

$= \overline{X} + X(Y + \overline{Y}Z) + \overline{Z}$

$= \overline{X} + X(Y + Z) + \overline{Z}$ (Because $Y + \overline{Y}Z = Y + Z$)

$= \overline{X} + XY(Z + \overline{Z}) + XZ + \overline{Z}$ (Because $Z + \overline{Z} = 1$)

$= \overline{X} + XYZ + XY\overline{Z} + XZ + \overline{Z}$

$= \overline{X} + XZ(1 + Y) + \overline{Z}(1 + XY)$

$= \overline{X} + XZ + \overline{Z}$ (Because $1 + Y = 1$ & $1 + XY = 1$)

$= \overline{X} + XZ) + \overline{Z}$

$= (\overline{X} + Z) + \overline{Z}$ (Because $\overline{X} + XZ = \overline{X} + Z$

$= \overline{X} + (Z + \overline{Z})$

$= \overline{X} + 1$ (Because $Z + \overline{Z} = 1$)

$= 1$ (Because $\overline{X} + 1 = 1$)

Q.12 Minimize the logic function $Y(A, B, C, D) = \sum m(0,1,2,3,5,7,8,9,11,14)$. Use Karnaugh map. Draw logic circuit for the simplified function. **(9)**

Ans:

Fig. 4(a) shows the Karnaugh map. Since the expression has 4 variables, the map has 16 cells. The digit 1 has been written in the cells having a term in the given expression. The decimal number has been added as subscript to indicate the binary number for the concerned cell. The term $ABC\overline{D}$ cannot be combined with any other cell. So this term will appear as such in the final expression. There are four groupings of 4 cells each. These correspond to the min terms (0, 1, 2, 3), (0, 1, 8, 9), (1, 3,5,7) and (1, 3, 9, 11). These are shown in the map. Since all the terms (except 14) have been included in groups of 4 cells, there is no need to form groups of two cells.

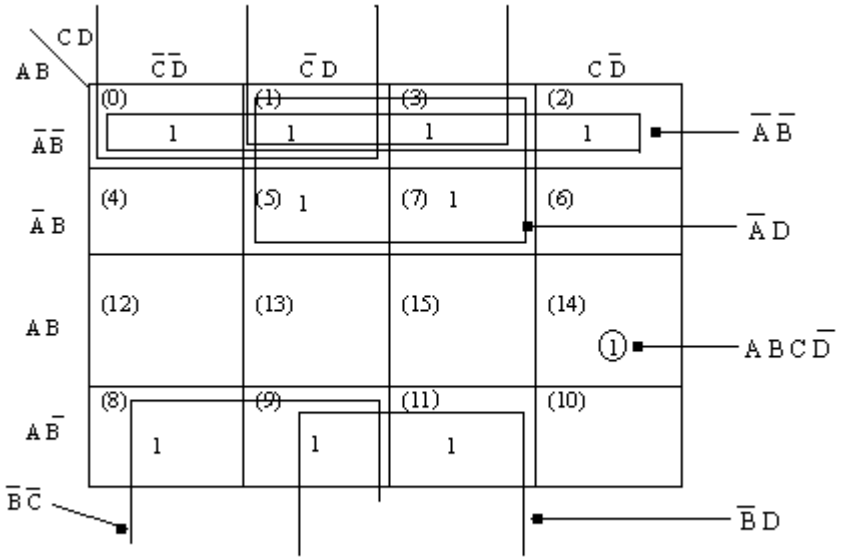


Fig. 4(a)

The simplified expression is Y (A,*B,*C,*D)* $= ABC\overline{D} + \overline{A}\ \overline{B} + \overline{B}\ \overline{C} + \overline{B}$ D+ $\overline{A}$ D

Fig.4 (b) shows the logic diagram for the simplified expression

Y (A,*B,* C, D*)* $= ABC\overline{D} + \overline{A}\ \overline{B} + \overline{B}\ \overline{C} + \overline{B}$ D+ $\overline{A}$ D

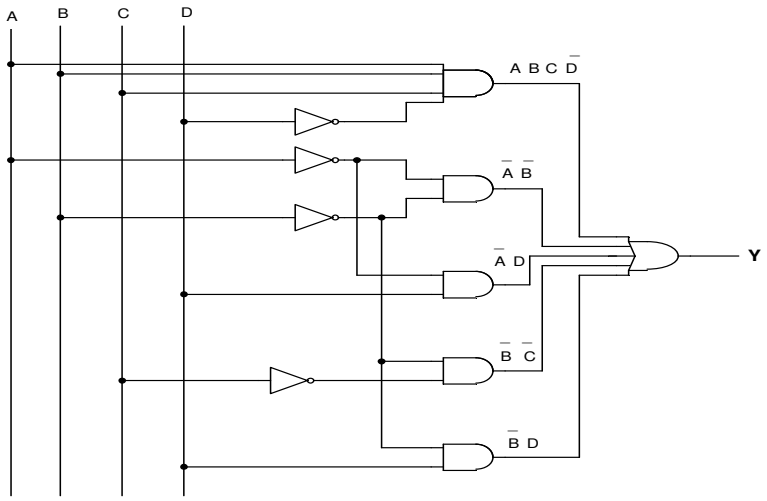


Fig.4(b) Logic diagram for Y

Q.13 Simplify the given expression to its Sum of Products (SOP) form. Draw the logic circuit for the simplified SOP function $Y = (A + B)(A + \overline{AB})C + \overline{A}(B + \overline{C}) + \overline{A}B + ABC$ **(5)**

Ans:

Simplification of given expression

Y = (A + B) (A + $\overline{AB}$) C + $\overline{A}$ (B + $\overline{C}$) + $\overline{A}$ B + ABC

in some of products (SOP) form:-

$Y = (A + B)(A + \overline{AB})C + \overline{A}(B + \overline{C}) + \overline{A}B + ABC$

$=(A + B)(A + \overline{AB})C + \overline{A}(B + \overline{C}) + \overline{A}B + ABC$

$=(A + B)(A + \overline{A} + \overline{B})C + \overline{A}(B + \overline{C}) + \overline{A}B + ABC$

$=(A + B)(1 + \overline{B})C + \overline{A}(B + \overline{C}) + \overline{A}B + ABC$ (Because $A + \overline{A} = 1$)

$= (A + B)(C + \overline{B}C) + \overline{A}B + \overline{A}\,\overline{C} + \overline{A}B + ABC$

$= (A + B)(C + \overline{B}C) + \overline{A}B + \overline{A}\,\overline{C} + \overline{A}B + ABC$

$=AC + A\overline{B}C + BC + B\overline{B}C + \overline{A}B + \overline{A}\,\overline{C} + \overline{A}B + ABC$

$= AC + AC(\overline{B} + B) + BC + 0 + \overline{A}B + \overline{A}\,\overline{C} + \overline{A}B$ (Because $B\overline{B} = 0$)

$= AC + AC + BC + \overline{A}B + \overline{A}\,\overline{C}$ (Because $\overline{B} + B = 1$)

$= AC + BC + \overline{A}B + \overline{A}\,\overline{C}$ (Because $AC + AC = AC$)

$= C(A + B) + \overline{A}(B + \overline{C})$

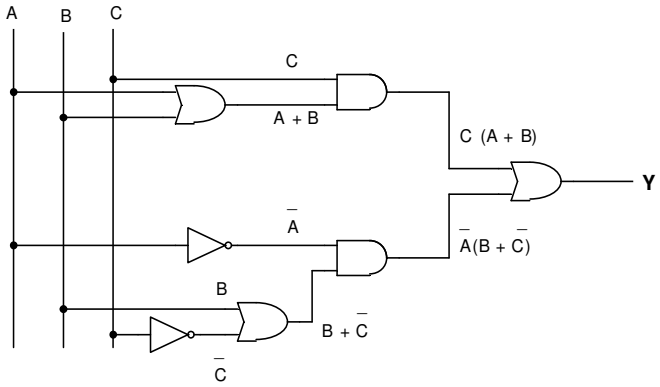


Fig.4(c) Simplified Logic Circuit

Q.14 Design a 8 to 1 multiplexer by using the four variable function given by $F(A,B,C,D) = \sum m(0,1,3,4,8,9,15)$. **(10)**

Ans:

Design of 8 to 1 Multiplexer: This is a four-variable function and therefore we need a multiplexer with three selection lines and eight inputs. We choose to apply variables *B,* C, and *D* to the selection lines. This is shown inTable 8.1. The first half of the minterms are associated with *A′* and the second half with *A*. By circling the minterms of the function and applying the rules for finding values for the multiplexer inputs, the implementation shown in Table.8.2.

The given function can be implemented with a 8-to-1 multiplexer as shown in fig.8(a). Three of the variables, B, C and D are applied to the selection lines in that order i.e., B is connected to s_2, C to s1 and D to s_0. The inputs of the multiplexer are 0, 1, A and A'. When BCD = 000,001 & 111 output F = 1 since I_0 & I_8 = 1 for BCD(000), I_1 = 1and I_9 =1 respectively. Therefore, minterms m_0 = A' B' C' m_1 = A' B' C, m_8 = A', B', C' and m_9 = A' B' C produce a 1 output. When BCD = 010, 101 and 110, output F = 0, since I_2, I_5 and I_6 respectively are equal to 0.

Minterm	A	B	C	D	F
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	1

Table .8.1 Truth Table for 8-1 Multiplexer

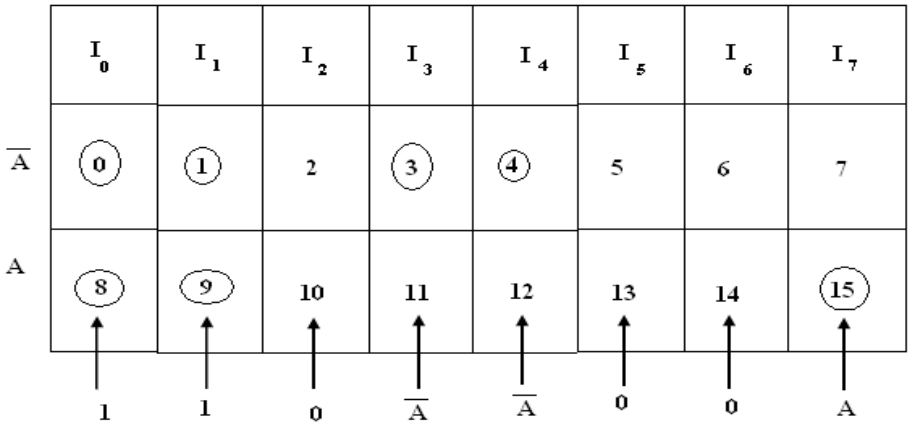


Table 8.2 Implementation Table for 8 to 1 MUX

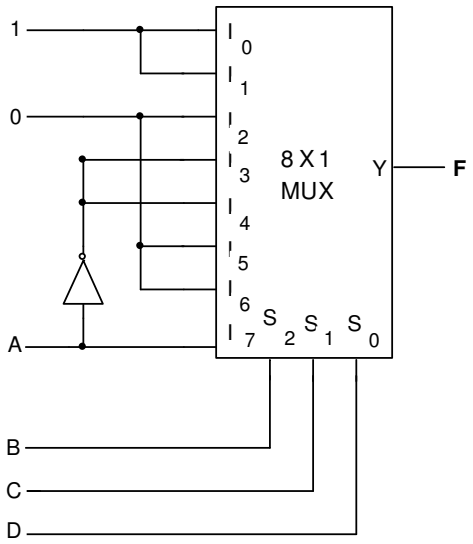


Fig.8(a) Logic circuit for 8-to-1 Multiplexer

Q.15 Convert the decimal number 82.67 to its binary, hexadecimal and octal equivalents. **(6)**

Ans:

(i)Conversion of Decimal number 82.67 to its Binary Equivalent

Considering the integer part 82 and finding its binary equivalent

2	82	
2	41	Remainder ----- 0 (LSB)
2	20	Remainder ----- 1
2	10	Remainder ----- 0
2	5	Remainder ------0
2	2	Remainder ----- 1
2	1	Remainder ---- 0
	0	Remainder ---- 1 (MSB)

The Binary equivalent is $(1010010)_2$

Now taking the fractional part i.e., 0.67

Fraction	Fraction X 2	Remainder New Fraction	Integer
0.67	1.34	0.34	1
0.34	0.68	0.68	0
0.68	1.36	0.36	1
0.36	0.72	0.72	0
0.72	1.44	0.44	1
0.44	0.88	0.88	0
0.88	1.76	0.76	1
0.76	1.52	0.52	1

It is seen that, it is not possible to get a zero as remainder even after 8 stages. The process continued further on an approximation can be made and the process is terminated here.

The binary equivalent is 0.10101011

Therefore, the binary equivalent of decimal number **82.67 is $(1010010.10101011)_2$**

(ii)Conversion of the binary equivalent of decimal number 82.67 into Hexadecimal:

The binary equivalent of decimal number 82.67 is $(1010010.10101011)_2$

Convert each 4-bit binary into an equivalent hexadecimal number i.e.

0101	0010	.1010	1011
5	2	A	B

Therefore, the hexadecimal equivalent of decimal number **82.67 is $(52.AB)_{16}$**

(iii)Conversioin of the binary equivalent of decimal number 82.67 into Octal number:

The binary equivalent of decimal number 82.67 is $(1010010.10101011)_2$

Convert each 3-bit binary into an equivalent octal number i.e.

001	010	010	.101	010	110
1	2	2	. 5	2	6

Therefore, the Octal equivalent of decimal number 82.67 is $(122.526)_8$

Q.16 Add 20 and (-15) using 2's complement. **(4)**

Ans:

Addition of 20 and (-15) using 2's Complement:

2	20	
2	10	Remainder ----- 0 (LSB)
2	5	Remainder ----- 0
2	2	Remainder ----- 1
2	1	Remainder ------0
	0	Remainder ----- 1(MSB)

2	16	
2	8	Remainder ----- 0 (LSB)
2	4	Remainder ----- 0
2	2	Remainder ----- 0
2	1	Remainder ------0
	0	Remainder ----- 1(MSB)

$(20)_{10}$ = 1 0 1 0 0

$(16)_{10}$ = 1 0 0 0 0

$(-16)_{10}$ = 0 1 1 1 1(1's Complement)
+1(2's Complement)
---
1 0 0 0 0

Therefore, 20 = 1 0 1 0 0
-16 = 1 0 0 0 0
---
1 0 0 1 0 0
↑
(Neglect)

Since the MSB of the sum is 0, which means **the result is positive i.e +4**

Q.17 Add 648 and 487 in BCD code. **(4)**

Ans:

Addition of 648 and 487 in BCD Code:

6 4 8 = 0 1 1 0 0 1 0 0 1 0 0 0
4 8 7 = 0 1 0 0 1 0 0 0 0 1 1 1
---
1 0 1 0 1 1 0 0 1 1 1 1

10 12 15

In the above problem all the three groups are invalid, because the four bit sum is more than 9. In such cases, add +6(i.e. 0110) to the four bit sum to skip the six invalid states. If a carry is generated when adding 6, add the carry to the next four bit group i.e.

```
6 4 8  =  0 1 1 0   0 1 0 0   1 0 0 0
4 8 7  =  0 1 0 0   1 0 0 0   0 1 1 1
 ------------------------------------------------
           1 0 1 0   1 1 0 0   1 1 1 1
         0 1 1 0   0 1 1 0   0 1 1 0
       1   1 1   1   1     1   1 1
 ------------------------------------------------------
     0001   0 0 0 1   0 0 1 1   0 1 0 1
            ↓         ↓         ↓         ↓
     1         1         3         5
  ------------------------------------------------------
```

Addition of 648 and 487 in BCD Code is 1135.

Q.18 Prove the following Boolean identities. **(4)**

(i) $XY + YZ + \overline{Y}Z = XY + Z$

(ii) $A.B + \overline{A}.B + \overline{A}.\overline{B} = \overline{A} + B$

Ans:

(i) Prove the Boolean Identity $XY + YZ + \overline{Y}Z = XY + Z$

L.H.S $= XY + YZ + \overline{Y}Z$

$= XY(Z + \overline{Z}) + YZ + \overline{Y}Z \; (\because Z + \overline{Z} = 1)$

$= XYZ + XY\overline{Z} + YZ + \overline{Y}Z$

$= YZ(1+X) + XY\overline{Z} + \overline{Y}Z$

$= YZ + XY\overline{Z} + \overline{Y}Z \; (\because 1+X = 1)$

$= Z(Y + \overline{Y}) + XY\overline{Z}$

$= Z + XY\overline{Z} \; (\because Y + \overline{Y} = 1)$

$= Z + XY \; (\because Z + XY\overline{Z} = Z + XY)$

= R.H.S (Hence Proved)

(ii) Prove the Boolean Identity $A\,B + \overline{A}\,B + \overline{A}\,\overline{B} = \overline{A} + B$

R.H.S $= \overline{A} + B$

$= \overline{A}(B + \overline{B}) + B(A + \overline{A}) \; (\because B + \overline{B} = 1 \;\&\; A + \overline{A} = 1)$

$= \overline{A}(B + \overline{B}) + B(A + \overline{A})$

$= \overline{A}B + \overline{A}\,\overline{B} + B\,A + B\,\overline{A}$

$= \overline{A}B + \overline{A}\overline{B} + B\,A \; (\overline{A}B + \overline{A}B = \overline{A}B)$

= L.H.S (Hence Proved)

Q.19 For $F = A.B.C + B.C.\overline{D} + \overline{A}.B.C$, write the truth table. Simplify using Karnaugh map and realize the function using NAND gates only. **(10)**

Ans:

Simplification of Logic Function $F = A\,B\,C + B\,\overline{C}\,D + \overline{A}\,B\,C$

(i)The Truth Table is given in Table 4.1

Inputs				Output (F)
A	B	C	D	
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Table 4.1

(ii) The Karnaugh Map is shown in fig.4(a).
The simplified expression is F = BC + BD

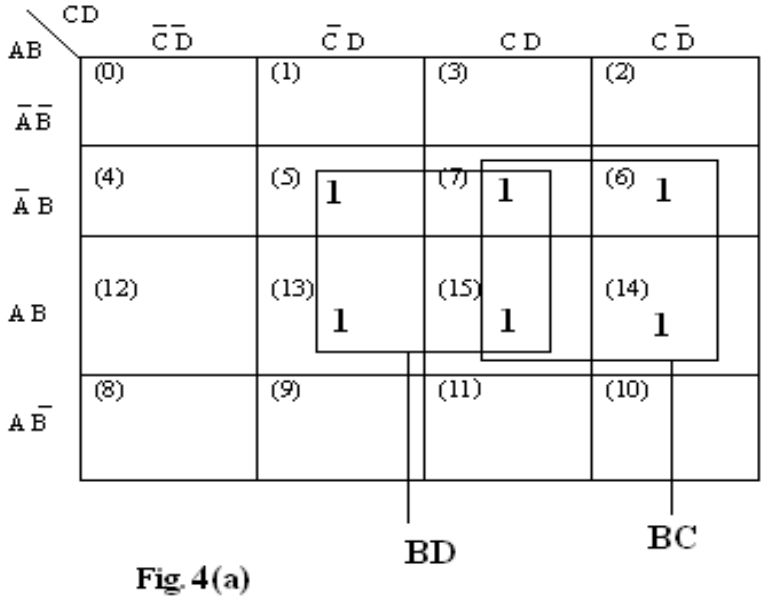


Fig. 4(a)

(iii) The NAND-NAND Realization is shown in fig.4(b)

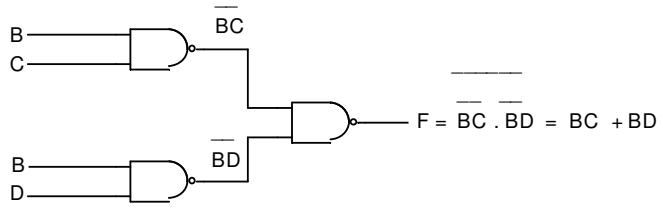


Fig. 4(b) NAND-NAND Realization

Q.20 Determine the analog output voltage of 6-bit DAC (R-2R ladder network) with V_{ref} as 5V when the digital input is 011100. **(10)**

Ans:

For 6-bit R-2R DAC ladder network, the output voltage is given by

$$V_0 = \frac{V_R}{2^n}\left(a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \cdots + a_1 2^1 + a_0 2^0\right)$$

Given Data : $V_R = 5V$, n = 6, $a_5=0$, $a_4=1$,$a_3=1$,$a_2=1$,$a_1=0$,$a_0=0$

$$V_0 = \frac{5}{2^6}\left(a_5 2^5 + a_4 2^4 + a_3 2^3 + a_2 2^2 + a_1 2^1 + a_0 2^0\right)$$

$$V_0 = \frac{5}{2^6}\left(0\times 2^5 + 1\times 2^4 + 1\times 2^3 + 1\times 2^2 + 0\times 2^1 + 0\times 2^0\right)$$

$$V_0 = 5\times\frac{28}{64} = \mathbf{2.1875\ V}$$

Q.21 Solve the following equations for X **(6)**

(i) $23.6_{10} = X_2$ (ii) $65.535_{10} = X_{16}$

Ans:

(i) Solve the equation $23.6_{10} = X_2$ for X

$23.6_{10} = X_2$

In order to find X, convert the Decimal number 23.6_{10} into its Binary form.

First take the decimal integer part 23 to convert into its equivalent binary form

2	23	▲
2	11 ------- 1	
2	5 --------1	
2	2 --------1	
2	1 --------0	
	0 --------1	

Hence $23_{10} = 10111_2$

Next take the decimal fractional part 0.6 to convert into its equivalent binary form.

Fraction	Fraction X 2	Remainder new fraction	Integer
0.6	1.2	0.2	1
0.2	0.4	0.4	0
0.4	0.8	0.8	0
0.8	1.6	0.6	1
0.6	1.2	0.2	1
0.2	0.4	0.4	0
0.4	0.8	0.8	0

It is seen that it is not possible to get a zero as remainder even after 7 stages. The process can be continued further or an approximation can be made and the process terminated here. The binary equivalent is 0.1001100.
Hence $\mathbf{23.6_{10} = 10111.1001100_{2.}}$
(ii) In order to find X, convert the Decimal number 65.535 into its equivalent Hexadecimal form. First taking the integer part 65 to convert into its equivalent Hexadecimal form.

16	65	
16	4	---- 1
	0	---- 4

Hence $65_{10} = 41_{16}$
Next take the decimal fractional part 0.6 to convert into its equivalent binary form.

Fraction	Fraction X 16	Remainder new fraction	Integer
0.535	8.56	0.56	8
0.56	8.96	0.96	8
0.96	15.36	0.36	15 (F)
0.36	5.76	0.76	5
0.76	12.16	0.16	12(C)
0.16	2.56	0.56	2
0.56	8.96	0.96	8

It is seen that it is not possible to get a zero as remainder even after 7 stages. The process can be continued further or an approximation can be made and the process terminated here. The Hexadecimal equivalent is 0.88F5C28.
Hence $\mathbf{65.535_{10} = 41.88F5C28_{16.}}$

Q.22 Perform the following additions using 2's complement **(5)**
(i) -20 to +26 (ii) +25 to -15

Ans:

(i) First convert the two numbers 20 and 26 into its 8-bit binary equivalent and find out the 2's complement of 20, then add -20 to +26.

20 = 0 0 0 1 0 1 0 0 (8-bit binary equivalent of 20)

$\overline{20}$ = 1 1 1 0 1 0 1 1 (1's complement)

+1

$\overline{20}$ = -20 = 1 1 1 0 1 1 0 0 (2's complement of 20)

+26 = 0 0 0 1 1 0 1 0 (8-bit binary equivalent of 26)

Addition of -20 to +26

= +6 = 0 0 0 0 0 1 1 0

Hence -20 to +26 = $(6)_{10}$ = $(0110)_{2.}$

(ii) First convert the two numbers 25 and 15 into its 8-bit binary equivalent and find out the 2's complement of 15, then add +25 to -15.

15 = 0 0 0 0 1 1 1 1 (8-bit binary equivalent of 15)

$\overline{15}$ = 1 1 1 1 0 0 0 0 (1's complement)

+1

$\overline{15}$ = -15 = 1 1 1 1 0 0 0 1 (2's complement of 15)

+25 = 0 0 0 1 1 0 0 1 (8-bit binary equivalent of 25)

Addition of -15 to +25

= +10 = 0 0 0 0 1 0 1 0

Hence -15 to +25 = $(10)_{10}$ = $(1010)_{2.}$

Q.23 (i) Convert the decimal number 430 to Excess-3 code: **(6)**

(ii) Convert the binary number 10110 to Gray code:

Ans:

(i) Excess 3 is a digital code obtained by adding 3 to each decimal digit and then converting the result to four bit binary. It is an unweighted code i.e., no weights can be assigned to any of the four digit positions.

4 3 0

+ 3 + 3 + 3

---

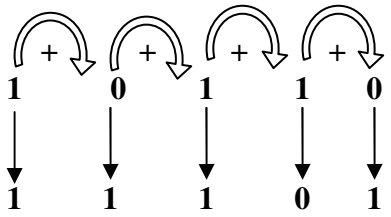
7 6 3

↓ ↓ ↓

0111 0110 0011 (Excess-3 Code)

(ii) The rules for changing binary number 10110 into its equivalent Gray code are, the left most bit (MSB) in Gray code i.e., 1 is the same as the left most bit in binary and

add the left most bit (1) to the adjacent bit (0) then add the next adjacent pair and discard the carry. Continue this process till completion.



Hence Gray equivalent of Binary number 10110 is 11101.

Q.24 Verify that the following operations are commutative but not associative **(6)**
(i) NAND (ii) NOR

Ans:

(i) Commutative Law is $\overline{AB} = \overline{BA}$. To verify whether the NAND operation is Commutative or not, prepare truth table shown in Table No.3.1

A	B	$\overline{AB}$	$\overline{BA}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

Table No.3.1

From the Table No.3.1, we observe that the last two columns are identical, which means

$$\overline{AB} = \overline{BA}$$

Associative Law is $\overline{A.\overline{(B.C)}} = \overline{\overline{(A.B)}.C}$

To verify whether the NAND operation is Associative or not, prepare truth table shown in Table No.3.2

A	B	C	$\overline{A.\overline{(B.C)}}$	$\overline{\overline{(A.B)}.C}$
0	0	0	1	1
0	0	1	1	0
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	1
1	1	1	1	1

Table No.3.2

From the Table No.3.2, we observe that the last two columns are not identical, which means

$$\overline{A.\overline{(B.C)}} \neq \overline{\overline{(A.B)}.C}$$

(ii) Commutative Law is $\overline{A+B} = \overline{B+A}$. To verify whether the NOR operation is Commutative or not, prepare truth table shown in Table No.3.3

A	B	$\overline{A+B}$	$\overline{B+A}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	1	1

Table No.3.3

From the Table No.3.3, we observe that the last two columns are identical, which means

$$\overline{A+B} = \overline{B+A}$$

Associative Law is $\overline{A+\overline{(B+C)}} = \overline{\overline{(A+B)}+C}$

To verify whether the NOR operation is Associative or not, prepare truth table shown in Table No.3.4

A	B	C	$\overline{A+\overline{(B+C)}}$	$\overline{\overline{(A+B)}+C}$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	1
1	1	1	0	0

Table No.3.4

From the Table No.3.4, we observe that the last two columns are not identical, which means

$$\overline{A.+\overline{(B+C)}} \neq \overline{\overline{(A+B)}+C}$$

Q.25 Prove the following equations using the Boolean algebraic theorems: **(5)**

(i) $A + \overline{A}.B + A.\overline{B} = A + B$ (ii) $\overline{A}BC + A\overline{B}C + AB\overline{C} + ABC = AB + BC + AC$

Ans:

(i) Given equation is $A + \overline{A}.B + A.\overline{B} = A + B$

L.H.S. $= A + \overline{A}.B + A.\overline{B}$

$= (A + A.\overline{B}) + \overline{A}.B$

$= A(1+\overline{B}) + \overline{A}.B$

$= A + \overline{A}.B \ (\because\ 1+\overline{B} = 1)$

$= (A + \overline{A})(A + B)$

$= (A + B) \ (\because A + \overline{A} = 1)$

$=$ **R.H.S**

Hence Proved

(ii) Given equation is $\overline{A}BC + A\overline{B}C + AB\overline{C} + ABC = AB + BC + AC$

L.H.S = $\overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$

$= \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$

$= \overline{A}BC + A\overline{B}C + AB(C + \overline{C})$

$= \overline{A}BC + A\overline{B}C + AB$ ($\because C + \overline{C} = 1$)

$= \overline{A}BC + A(B + \overline{B}C)$

$= \overline{A}BC + A(B + C)$ ($\because B + \overline{B}C = B + C$)

$= \overline{A}BC + AB + AC$

$= C(A + \overline{A}B) + AB + AC$

$= C(A + B) + AB + AC$ ($\because A + \overline{A}B = A + B$)

$= AC + BC + AB + AC$

$= AB + BC + AC$ ($\because AC + AC = AC$)

$=$ **R.H.S**

Hence Proved

Q.26 A staircase light is controlled by two switches one at the top of the stairs and another at the bottom of stairs **(5)**

(i) Make a truth table for this system.

(ii) Write the logic equation is SOP form.

(iii) Realize the circuit using AND-OR gates.

Ans:

A staircase light is controlled by two switches S_1 and S_2, one at the top of the stairs and another at the bottom of the stairs. The circuit diagram of the system is shown in fig.4(a).

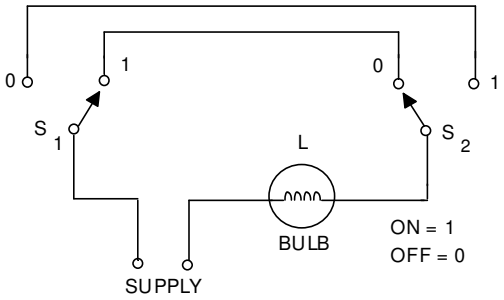


Fig.4(a) Circuit diagram

(i) The truth table for the system is given in truth table 4.1

S_1	S_2	L
0	0	0
0	1	1
1	0	1
1	1	0

Table 4.1

(ii) The logic equation for the system is given by $L = \overline{S_1}\,S_2 + S_1\,\overline{S_2}$

(iii) Realization of the circuit using AND-OR gates is shown in fig 4(b)

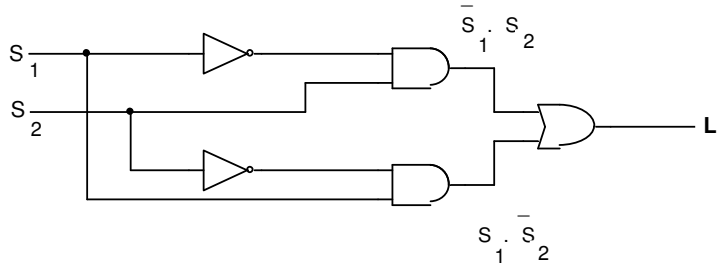


Fig.4(b) Logic Diagram for the system

Q.27 Minimize the following logic function using K-maps and realize using NAND and NOR gates.
$F(A,B,C,D) = \sum m(1,3,5,8,9,11,15) + d(2,13)$ **(9)**

Ans:

Minimization of the logic function F(A, B, C, D) = ∑ m(1,3,5,8,9,11.15) + d(2,13) using K-maps and Realization using NAND and NOR Gates

(i) Karnaugh Map for the logic function is given in table 4.1

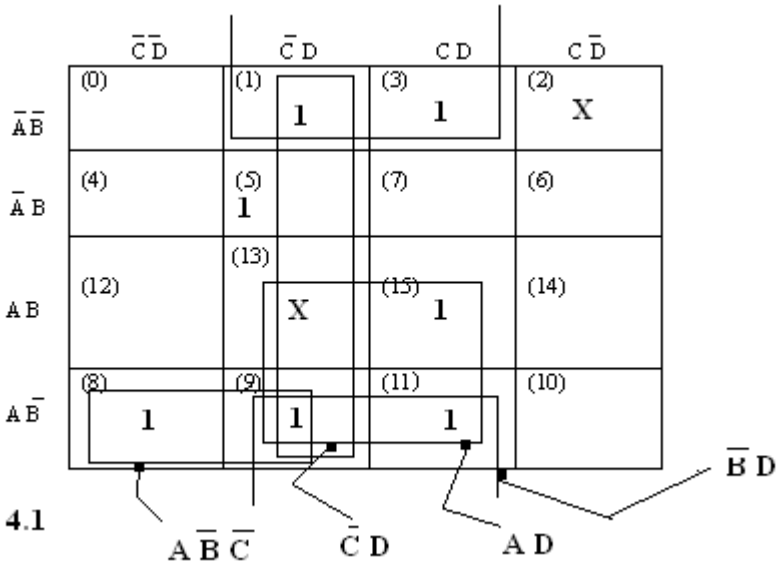


Table 4.1

The minimized logic expression in SOP form is F = $A\bar{B}\bar{C} + \bar{C}D + \bar{B}D + AD$

The minimized logic expression in POS form is F = $(A + \bar{B} + \bar{C})(\bar{C} + D)(\bar{B} + D)(A + D)$

(ii) Realization of the expression using NAND gates:

The minimized logic expression in SOP form is F = $A\bar{B}\bar{C} + \bar{C}D + \bar{B}D + AD$ and the logic diagram for the simplified expression is given in fig.4(c)

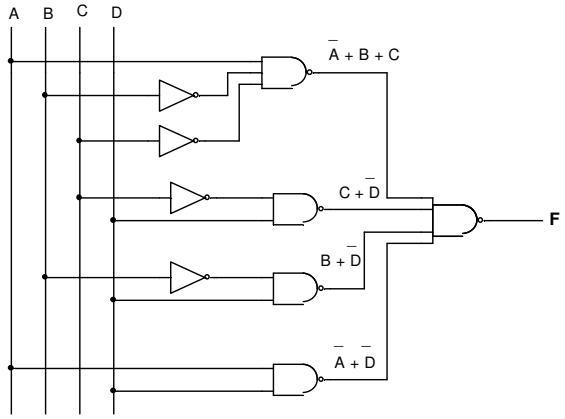


Fig.4(c) Logic Diagram

(iii) Realization of the expression using NOR gates:

The minimized logic expression in POS form is F = (A + $\overline{B}+\overline{C}$) ($\overline{C}$+D) ($\overline{B}$+D) (A+D) and the logic diagram for the simplified expression is given in fig.4(d)

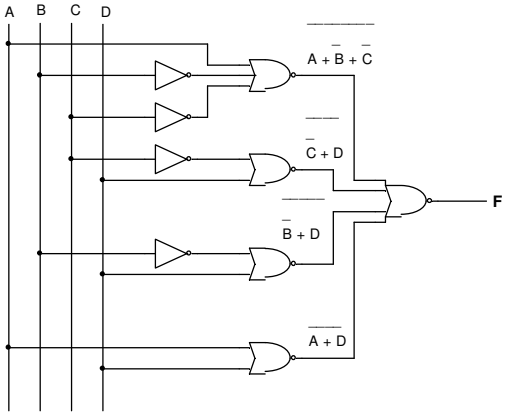


Fig.4(d) Logic Diagram

Q.28 Design a 4 to 1 Multiplexer by using the three variable function given by $F(A,B,C)=\sum m(1,3,5,6)$ **(7)**

Ans:

Design of 4 to 1 Multiplexer by using the three variable function given by
$F(A,B,C)=\sum m(1,3,5,6)$

The function **$F(A,B,C)=\sum m(1,3,5,6)$** can be implemented with a 4-to-1 multiplexer as shown in Fig.7(a). Two of the variables, B and C are applied to the selection lines in that order, i.e., *B* is connected to S_1 and C to S_0. The inputs of the multiplexer are 0, I, *A,* and *A′.*

When *BC* = 00, output *F* = 0 since $I_0 = 0$. Therefore, both minterms $m_0 = A'\,B'\,C'$ and *m4* = *A B′* C' produce a 0 output, since the output is 0 when *BC* = 00 regardless of the value of *A*.

When *BC* = 01, output *F* = 1, since $I_1 = 1$. Therefore, both minterms $m_1 = A'\,B'C$ and

$m_5 = AB'C$ produce a 1 output, since the output is 1. when $BC = 01$ regardless of the value of A.

When $BC = 10$, input I_2 is selected. Since A is connected to this input, the output will be equal to 1 only for minterm $m_6 = ABC'$, but not for minterm $m_2 = A'BC'$, because when $A' = I$, then $A = 0$, and since $I_2 = 0$, we have $F = 0$.

Finally, when $BC = 11$, input I_3 is selected. Since A' is connected to this input, the output will be equal to 1 only for minterm $m_3 = A'BC$, but not for $m_7 = ABC$. This is given in the Truth Table shown in Table No 7.1

Minterm	A	B	C	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

Table 7.1 Truth Table

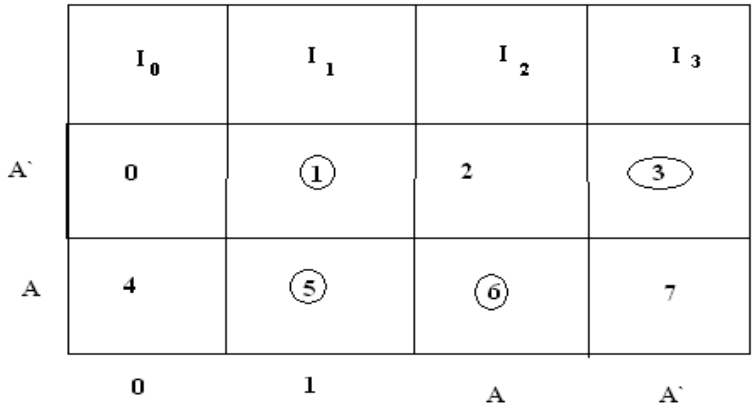


Fig.7(a) Implementation Table

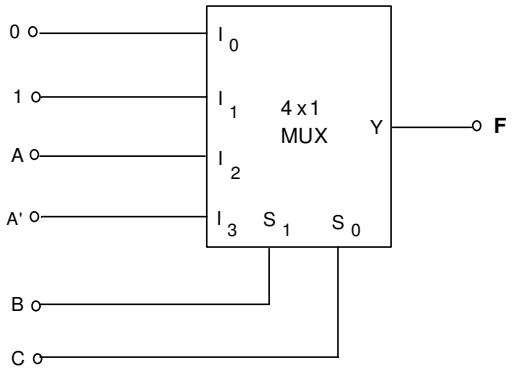


Fig.7(b) Logic Diagram of 4X1 Multiplexer

Q.29 Find the conversion time of a Successive Approximation A/D converter which uses a 2 MHz clock and a 5-bit binary ladder containing 8V reference. What is the Conversion Rate? **(4)**

Ans:

Given data:

Frequency of the clock (F) = 2 MHZ

Number of bits (n) = 5

(i) Conversion Time (T) = $\frac{n}{clockrate} = \frac{5}{2X10^{6}} = 2.5\ \mu\text{sec}$

(ii) Conversion Rate = $\frac{1}{T} = \frac{1}{2.5X10^{-6}} = 400{,}000$ conversions/sec

Q.30 A 6-bit R-2R ladder D/A converter has a reference voltage of 6.5V. It meets standard linearity. Find

(i) The Resolution in Percent.

(ii) The output voltage for the word 011100. **(4)**

Ans:

Given Data Number of Bits (n) = 6

Reference Voltage (V_R) = 6.5 V

For R-2R Ladder D/A Converter,

(i)The Resolution in Percent is given by $\frac{1}{2^n - 1} = \frac{1}{2^6 - 1} = \frac{1}{63} = 1.59\ \%$

(ii)The Output Voltage (V_O) of 6-bit R-2R Ladder D/A Converter for the word 011100 is given by

$$V_O = \frac{V_R}{2^n}\left[a_{n-1}.2^{n-1} + a_{n-2}.2^{n-2} + - - - - + a_1 2^1 + a_0 2^0\right]$$

$$V_O = \frac{6.5}{2^6}\left[0.2^{6-1} + 1X2^{5-1} + 1X2^{4-1} + 1X2^{3-1} + 0X2^{2-1} + 0X2^0\right]$$

$$V_O = \frac{6.5}{2_6}\left[2^4 + 2^3 + 2^2\right]$$

$$V_O = 2.84\ \text{V}.$$

Q.31 Convert 2222 in Hexadecimal number. **(4)**

Ans:

	2222		
16	138	14	↑
16	8	10	↑ =8AE
	0	8	↑

Q.32 Subtract –27 from 68 using 2's complements. **(6)**

Ans:

68-(-27)=68-(-27)using 2's complement

2's complement representation of 68=01000100(64+4)

2's complement representation of - (-27) = 00011011 =+ 27

11100101 =-27 in 2's complement

Now add 68 and 27

68	0 1 0 0 0 1 0 0
-(-27)	+ 0 0 0 1 1 0 1 1
95	0 1 0 1 1 1 1 1 1

Which is equal to +95

Q.33 Divide $(101110)_2$ by $(101)_2$. **(4)**

Ans:

```
1 0 1 | 1 0 1 1 1 0 | 1 0 0 1
      | 1 0 1       |
      -------       |
      0 0 0 1 1 0   |
          1 0 1     |
          -------   |
          0 0 1     |
```

Quotient -1001

Remainder -001

Q.34 Prove the following identities using Boolean algebra:

(i) $(A + B)(A + \overline{AB})C + \overline{A}(B + \overline{C}) + \overline{A}B + ABC = C(A + B) + \overline{A}(B + \overline{C})$.

(ii) $\overline{A(A \cdot B)} \cdot \overline{B(A \cdot B)} = A \oplus B$.

(iii) $\overline{\overline{AB} + \overline{A} + AB} = 0$. **(9)**

Ans:

(i) (A+B)(A+A'B')C+A'(B+C')+A'B+ABC

=C(A+B)+A'(B+C')

LHS (A+B)(A+A'+B')C+A'B+A'C'+A'B+ABC

= (A+B)(1+B')C+A'B+ A'C'+ABC as (A+A'=1)

= (A+B).1.C+A'B+ A'C'+ABC

= AB+AC+A'B+ A'C'+ABC

= ABC+AB+ABC+AC+A'B+A'C'

AB(C+1)+AC(B+1) +A'B+ A'C'

= AB+AC+A'B+ A'C'

= C(A+B) + A'(B+C') = **RHS**

Hence Proved

(ii) $\overline{\overline{A(\overline{A.B})}.\overline{B(\overline{A.B})}} = A \oplus B$

Let us take $X = \overline{A(\overline{A.B})}$

$Y = \overline{B(\overline{A.B})}$

So we have $\overline{X.Y} = A \oplus B$ -------3

Also $X = \overline{A(\overline{A.B})}$

$= \overline{A(A' + B')}$

By using DeMorgan's Law (AB)'=A'+B'

X = (A(A'+B'))'=(AA'+AB')'=(AB')'=(A'+B) ------1

Now Y = (B(AB)')'=[B(A'+B')]'= [A'B+BB']'=(A'B)'=(A+B') ------2

Now Combining X & Y from 1 & 2 above, we have L.H.S in 3 as :

((A+B')(A'+B))'

=[AA'+BB+A'B'+AB]'

=(AB+A'B')'

=A XOR B = **RHS**

Hence Proved

(iii) ((AB)'+A'+AB)'=0

LHS $\overline{(\overline{AB} + \overline{A} + AB)}$

$= (1 + A')'$ since $\overline{AB} + AB = 1$

$= 1'$ since $1 + A' = 1$

$= 0$ **= RHS Hence Proved**

Q.35 A combinational circuit has 3 inputs A, B, C and output F. F is true for following input combinations

A is False, B is True

A is False, C is True

A, B, C are False

A, B, C are True

(i) Write the Truth table for F. Use the convention True=1 and False = 0.
(ii) Write the simplified expression for F in SOP form.
(iii) Write the simplified expression for F in POS form.
(iv) Draw logic circuit using minimum number of 2-input NAND gates. **(7)**

Ans:

(i) Making the truth table

A	B	C	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

A is false b is true → For both value of c F is true.

(ii) Simplified expression for F can be found by K-map

A/BC	00	01	11	10
0	1	1	1	1
1	0	0	1	0

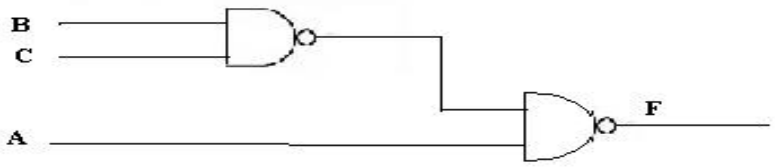
In SOP Form
F = A'+BC

(iii) Simplified expression for F in POS form

I. In POS Form MINIMIZE ZEROS
F'=AB'+AC'

II. F=A'+BC taking complement twice
F'=(A'+BC)'=(A.(BC)')
F"=F=(A.(BC)')'

(iv) Logic circuit by using minimum number of 2-input NAND gates



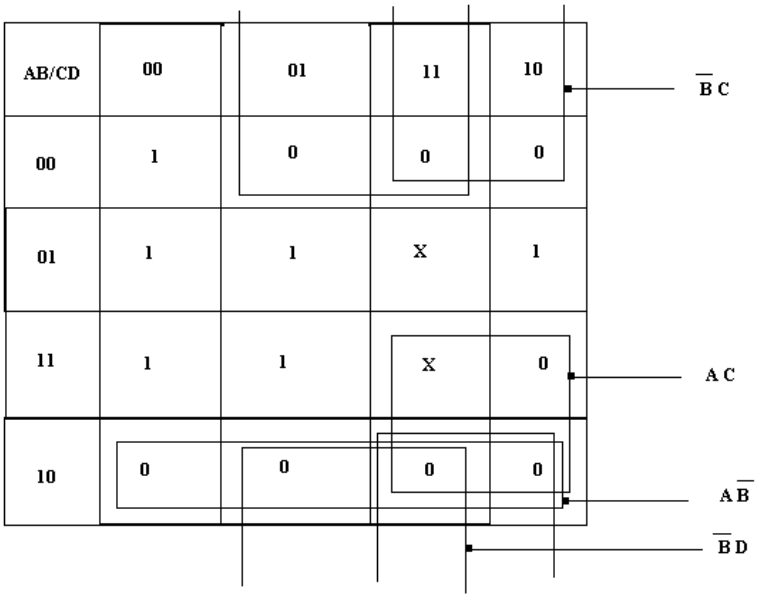
Q.36 Minimise the logic function

$$F(A,B,C,D) = \Pi M(1,2,3,8,9,10,11,14) \cdot d(7,15)$$

Use Karnaugh map. Draw the logic circuit for the simplified function using NOR gates only. **(7)**

Ans:

F=∏M(1,2,3,8,9,10,11,14).d(7, 15)



F'=B'D+B'C+AC+AB'

By Complementing F

F=(B'D+B'C+AC+AB')'

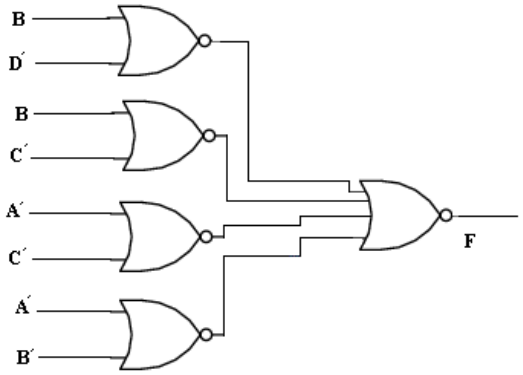
= [(B'D)'(B'C)'(AC)'(AB')']'

= (B+D')(B+C')(A'+C')(A'+B)

Taking complement twice and without opening the bracket

F=[(B+D')+(B+C')'('A'+C')+(A'+B)]'

The logic circuit for the simplified function using NOR gates



Q.37 The capacity of 2K × 16 PROM is to be expanded to 16 K × 16. Find the number of PROM chips required and the number of address lines in the expanded memory. **(4)**

Ans:

Required capacity =16k x 16

Available chip (PROM) =2k x 16

The no of chip $= \frac{16k \times 16}{2k \times 16} = 8$

In the chip total word capacity = 2×2^{10}

Thus the address line required for the single chip = 11
In the expanded memory the word capacity 16k = 2^{14}
Now the address lines required are 14. Among then 11 will be common and 3 will be connected to 3 x8 decoder.

Q.38 Perform following subtraction
(i) 11001-10110 using 1's complement
(ii) 11011-11001 using 2's complement **(8)**

Ans:
(i) 11001 - 10110
1' s Compliment of 10110 = 01001

```
   1 1 0 0 1
+  0 1 0 0 1
-----------------
 1 0 0 0 1 0
```

Add 1 and ignore carry.
Ans is 00011 = 3.
(ii) 11011 – 11001 = A – B
2's complement of B = 00111

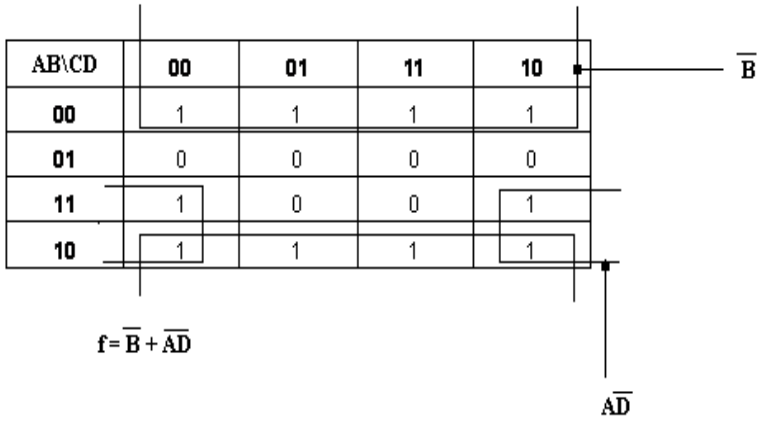
```
   1 1 0 1 1
+  0 0 1 1 1
-------------
 1 0 0 0 1 0
```

Ignore carry to get answer as 00010 = 2.

Q.39 Reduce the following equation using k-map
$Y = \overline{A}\overline{B}\overline{C} + A\overline{C}\overline{D} + A\overline{B} + AB\overline{C}\overline{D} + \overline{A}\overline{B}C$ **(8)**

Ans:

$$Y = \overline{A}\overline{B}\overline{C} + A\overline{C}\overline{D} + A\overline{B} + \overline{A}\overline{B}C$$
$$Y = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C}D + A\overline{B}\overline{C}\overline{D} + AB\overline{C}\overline{D} + A\overline{B}CD + A\overline{B}\overline{C}D$$
$$+ A\overline{B}C\overline{D} + A\overline{B}\overline{C}\overline{D} + ABC\overline{D} + \overline{A}\overline{B}C\overline{D} + \overline{A}\overline{B}CD$$



Q.40 Write the expression for Boolean function

F (A, B, C) = ∑m (1,4,5,6,7) in standard POS form. **(8)**

Ans:

$f(A,B,C) = \Sigma M(1,4,5,6,7)$ in standard POS form

$F = m_1 + m_4 + m_5 + m_6 + m_7$

$F = \Sigma m(1,4,5,6,7)$

$= \prod M(0,2,3)$

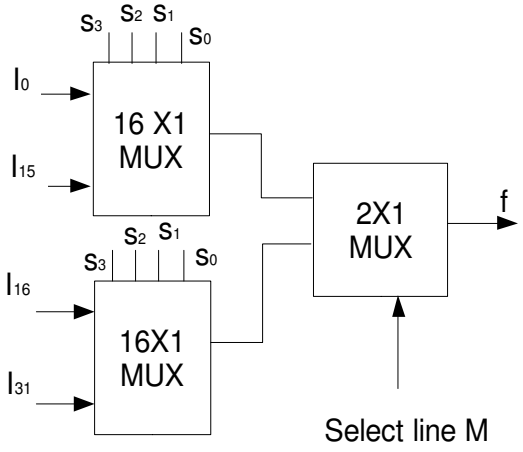
$= M_0 M_2 M_3$

$= (A+B+C)(A+\overline{B}+C)(A+\overline{B}+\overline{C})$

Q.41 Design a 32:1 multiplexer using two 16:1 multiplexers and a 2:1 multiplexer. **(8)**

Ans:

To design a 32 X 1 MUX using



Two 16 X 1 MUX & one 2 X 1

There are total 32 input lines and one O/P line. The 2 X 1 MUX will transmit one of the two I/P to output depending upon its select line M. For M = 0 upper MUX (I_0 – I_{15}) will be selected and M = 1 lower MUX (I_{16} – I_{31}) will be selected.

Q.42 Implement the following function using a 3 line to 8 line decoder.

S (A,B,C) = ∑ m(1,2,4,7)

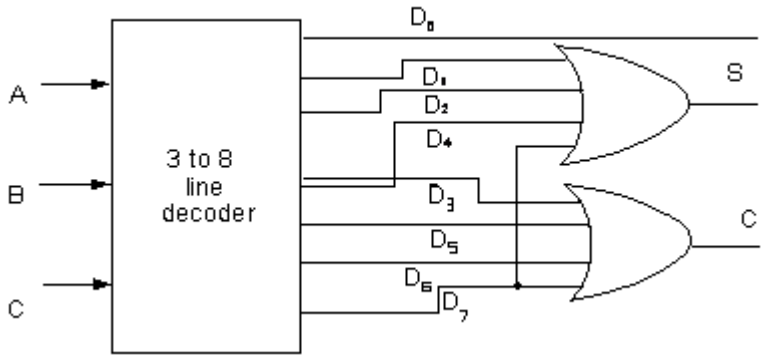
C (A,B,C) = ∑ m (3,5,6,7) **(8)**

Ans:

S (A,B,C) = m (1,2,4,7)

C (A,B,C) = m (3,5,6,7)

These are full adder's output as sum (S) and carry (C). We know that 3 to 8 line decoder generates all the minterms from 0 to 7. In the decoder shown in the figure, Do correspond to minterm m_o, and so on. So by ORing appropriate outputs of the decoder we can implement these functions.



Q.43 Perform the following operations using the 2's complement method:
(i) 23 – 48 (ii) – 48 – 23 **(4)**

Ans:

(i) 23 - 48

add them

23	0 1 0 1 1 1
- (- 48)	+ 0 1 0 0 0 0
71	1 0 0 1 1 1

(ii) – 48 - 23 = - 48 + (-23)

-48 = 1 1 0 1 0 0 0 0
-23 = 1 1 1 0 1 0 0 1

1 1 0 1 1 1 0 0 1 = -71

↓

Carry is discarded

Q.44 Prove the following Boolean identities using the laws of Boolean algebra:
(i) $(A+B)(A+C)=A+BC$
(ii) $ABC+A\overline{B}C+AB\overline{C}=A(B+C)$ **(4)**

Ans:

(i) (A+B)(A+C)=A+BC

LHS AA+AC+AB+BC=A+AC+AB+BC
OR A((C+1)+A(B+1))+BC
OR A+A+BC
OR A+BC = **RHS**
Hence Proved

(ii) ABC+AB'C+ABC'=A(B + C)

LHS AC(B+B')+AB(C+C')
OR AC+AB
OR A(B+C)= **RHS**
Hence Proved

Q.45 The Karnaugh map for a SOP function is given below in Fig.1. Determine the simplified SOP Boolean expression. **(5)**

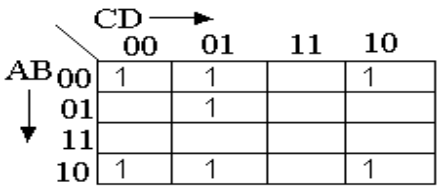
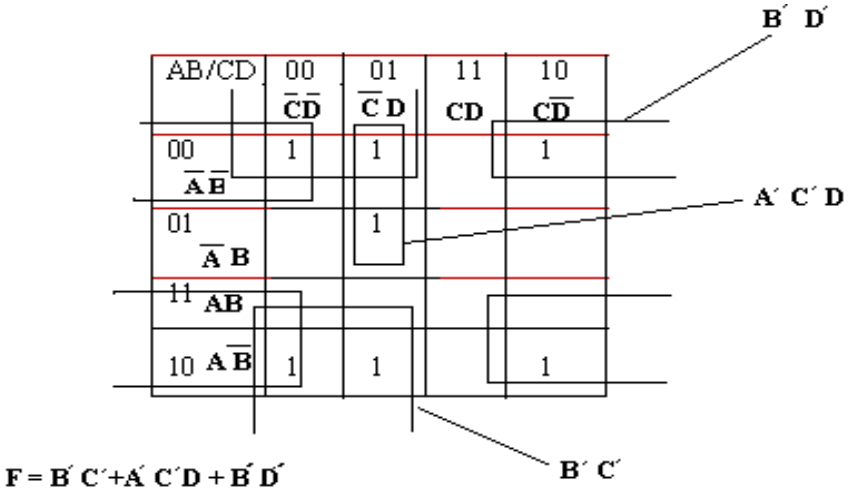


Fig.1

Ans:



Q.46 A certain memory has a capacity of 4K×8

(i) How many data input and data output lines does it have?

(ii) How many address lines does it have?

(iii) What is its capacity in bytes? **(5)**

Ans:

(i) available capacity =4Kx8

$= 2^{10} \times 2^{10} \times 8$

$= 2^{12} \times 8$

As in the 4Kx8 ,the second number represents the number of bits in each word so the number of data input lines will be 8(also the data output lines) .

(ii) It has total 4K (2^{12}) address line which are required to address 2^{12} locations.

(iii) Its capacity in bytes is 4K byte**s.**

Q.47 A 5-bit DAC produces an output voltage of 0.2V for a digital input of 00001. Find the value of the output voltage for an input of 11111. What is the resolution of this DAC? **(6)**

Ans:

For the Digital output of 00001

Output voltage is =0.2 volt =Resolution

The output=.2x31=15.5volts

Resolution=(0.2volt)/(15.5v)x100=1.290

Q.48 An 8-bit successive approximation ADC has a resolution of 20mV. What will be its digital output for an analog input of 2.17V? **(4)**

Ans:

Resolution =20mv
Analog input =2.17v
Equivalent value=(2.17)/(2.17)=108.5
Equivalent Binary value=1101100.1

Q.49 A microprocessor uses RAM chips of 1024×1 capacity.

(i) How many chips will be required and how many address lines will be connected to provide capacity of 1024 bytes.
(ii) How many chips will be required to obtain a memory of capacity of 16 K bytes. **(5)**

Ans:

(i) Available chips = 1024 x 1 capacity
Required capacity = 1024 x 8 capacity

$$No.\ of\ Chips = \frac{1024X8}{1024X1} = 8$$

Number of address lines are required = 10 (i.e. $1024 = 2^{10}$)
As the word capacity is same (1024) so same address lines will be connected to all chips.

(ii)

$$No.\ Of\ Chips\ Required = \frac{16X1024X8}{1024X1} = 128$$

Q.50 Find the Boolean expression for logic circuit shown in Fig.1 below and reduce it using Boolean algebra. **(6)**

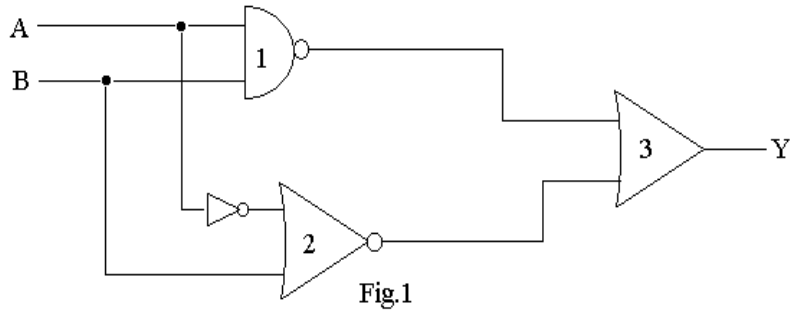


Fig.1

Ans:

Y = (AB)' + (A' + B)'
= A' + B' + AB' Using Demorgan's Theorem.
= A' + B'(1+A)
= A' + B' Since 1+A=1

Q.51 Implement the following function using 4-to-1 multiplexer.

$$Y(A, B, C) = \sum (2,3,5,6)$$ **(8)**

Ans:

Y(A,B,C)=∑(2,3,5,6)

Let us take B,C as the select bits and A as input. To decide the input we write.

Y = A'BC'+A'BC+AB'C+ABC'

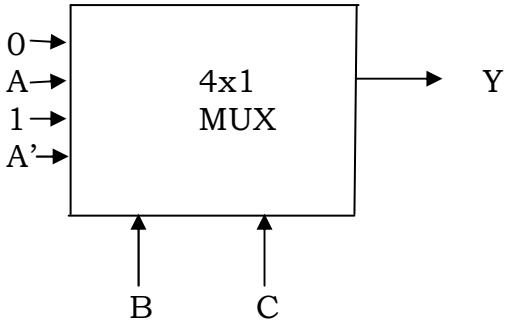
= 0 if B=0, C=0

= A if B=0, C=1

= 1 if B=1, C=0

= A' if B=1, C=1

The corresponding implementation is shown in the figure. Thus



Q.52 Design a mod-12 Synchronous up counter. **(8)**

Ans:

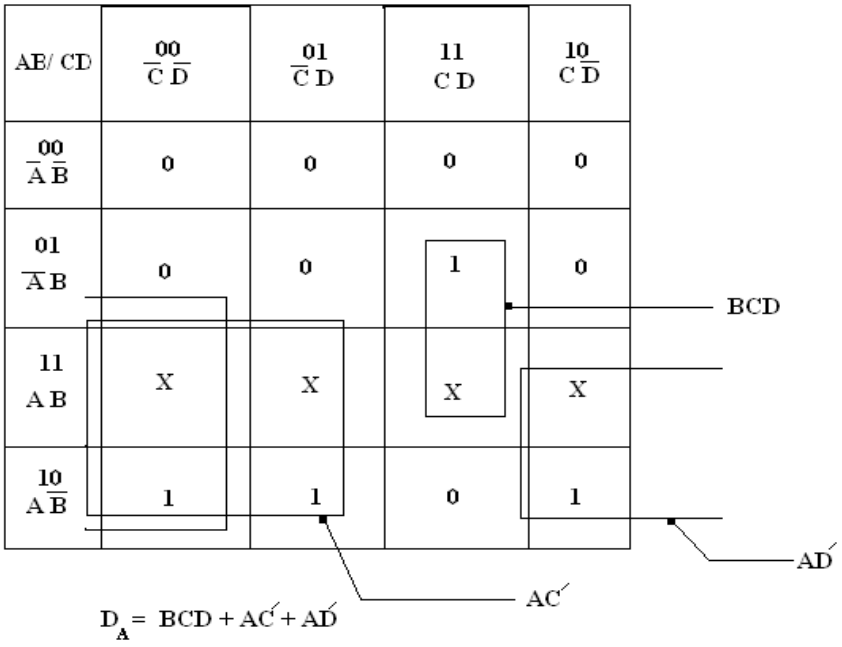
Design a mod 12 synchronous counter using D-flipflops.

I state table

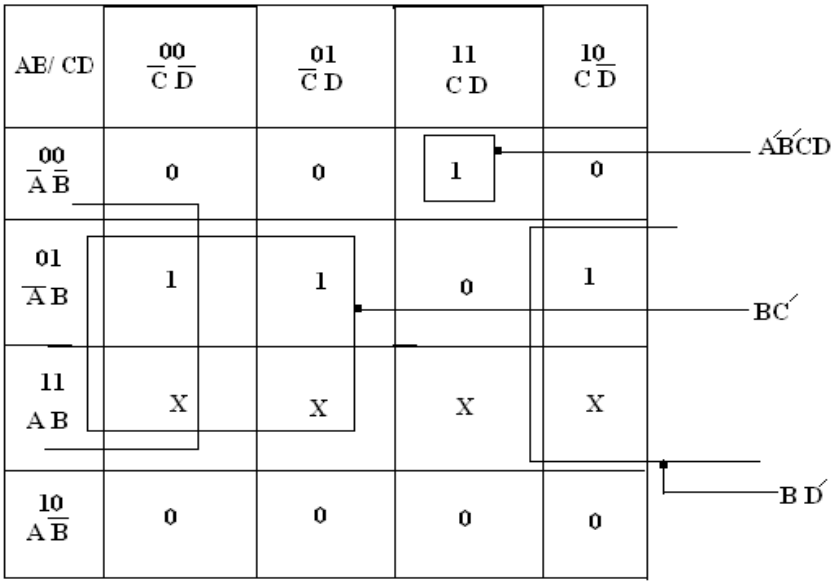
Present state				Next state				Required D Inputs			
A	B	C	D	A	B	C	D	D_A	D_B	D_C	D_D
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	0	0	0	1	0
0	0	1	0	0	0	1	1	0	0	1	1
0	0	1	1	0	1	0	0	0	1	0	0
0	1	0	0	0	1	0	1	0	1	0	1
0	1	0	1	0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	1	0	1	1	1
0	1	1	1	1	0	0	0	1	0	0	0
1	0	0	0	1	0	0	1	1	0	0	1
1	0	0	1	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	1	1	0	1	1
1	0	1	1	0	1	0	0	0	1	0	0

First draw the state table having present state, next state and required flip-flop input to give the transition. D flip flop gives the output same as the next state itself. Then solve by using K maps to find out D_A D_B D_C D_D for all states.

Unused states are 1100,1101,1110,1111 they can be treated as don't care conditions from the table. Draw Karnaugh-maps for D_A, D_B, D_C and D_D as follows and obtain Boolean expressions for them.



$D_A = BCD + AC' + AD'$



$D_B = A\bar{B}CD + B\bar{D} + B\bar{C}$

AE/ CD	$\overline{C}\,\overline{D}$ (00)	$\overline{C}D$ (01)	CD (11)	$C\overline{D}$ (10)
$\overline{A}\,\overline{B}$ (00)	0	1	0	1
$\overline{A}B$ (01)	0	1	0	1
AB (11)	X	X	X	X
$A\overline{B}$ (10)	0	1	0	1

$C\overline{D}$

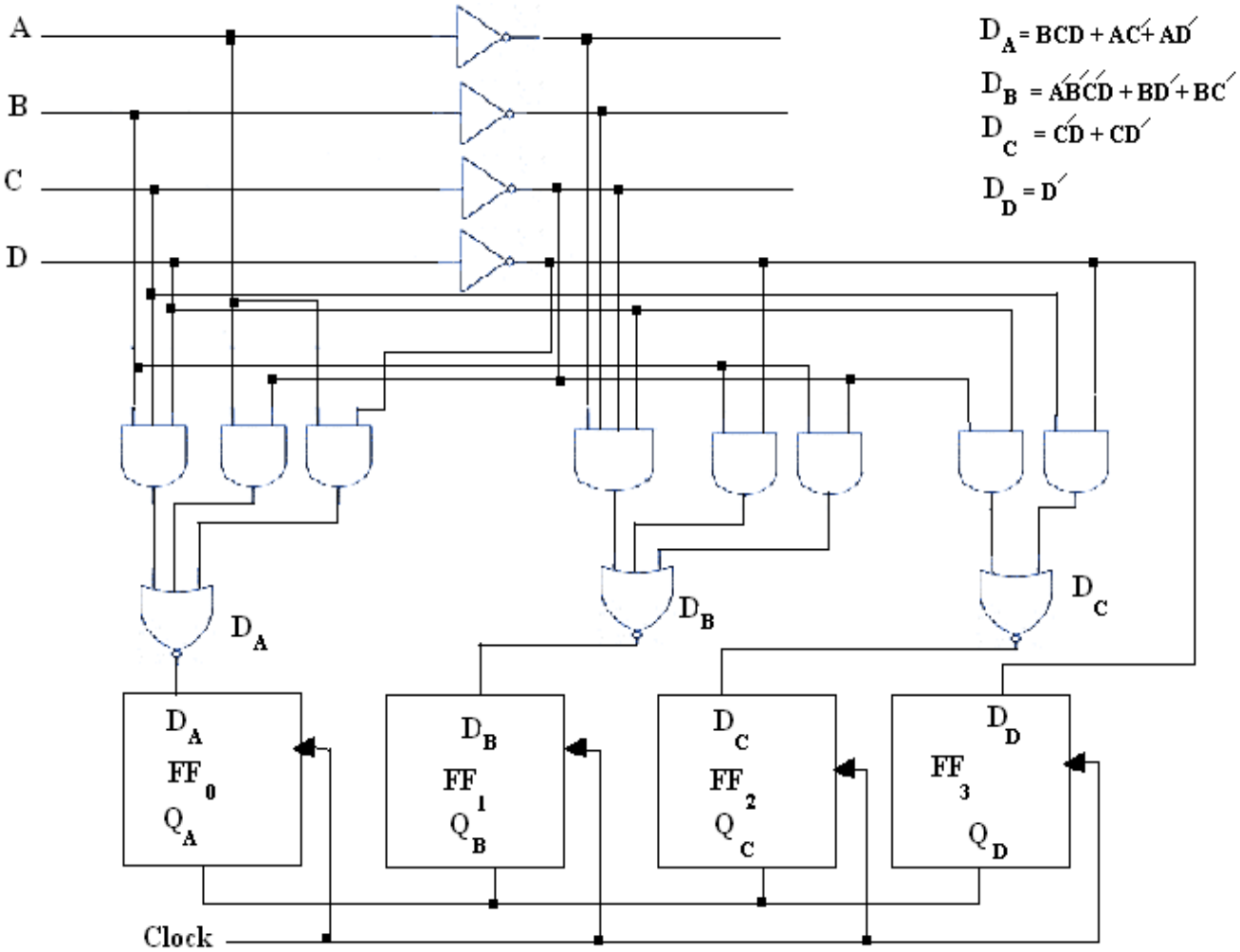
$\overline{C}D$

$$D_C = \overline{C}D + C\overline{D}$$

AE/ CD	$\overline{C}\,\overline{D}$ (00)	$\overline{C}D$ (01)	CD (11)	$C\overline{D}$ (10)
$\overline{A}\,\overline{B}$ (00)	1	0	0	1
$\overline{A}B$ (01)	1	0	0	1
AB (11)	X	X	X	X
$A\overline{B}$ (10)	1	0	0	1

$\overline{D}$

$$D_D = \overline{D}$$



Logic diagram for mod-12 Synchronous up-counter

Q.53 Find how many bits of ADC are required to get an resolution of 0.5 mV if the maximum full scale voltage is 10 V. **(8)**

Ans:

Resolution=.5mv

Full scale output=+10v

%resolution =(5mv)/10x100=0.05%

No of bits =$Log_2(2x1000) = 20$

Q.54 Convert the decimal number 45678 to its hexadecimal equivalent number. **(4)**

Ans:

$(45678)_{10}=(B26E)_{16}$

16	45678	
16	2854	14 → E
16	178	6 → 6
16	11	2 → 2
	0	11 → B

$(45678)_{10}=(B26E)_{16}$

Q.55 Write the truth table of NOR gate. **(4)**

Ans:

A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

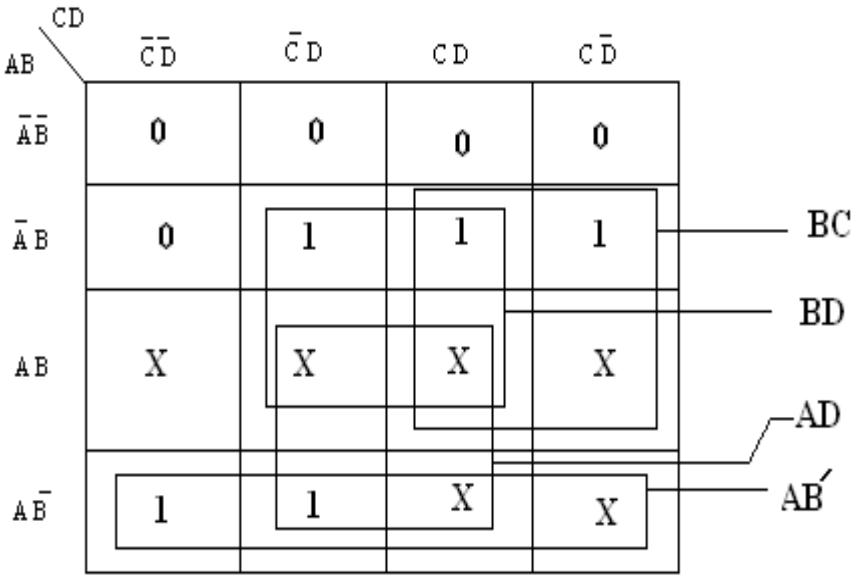
Q.56 Design a BCD to excess 3 code converter using minimum number of NAND gates. Hint: use k map techniques. **(8)**

Ans:

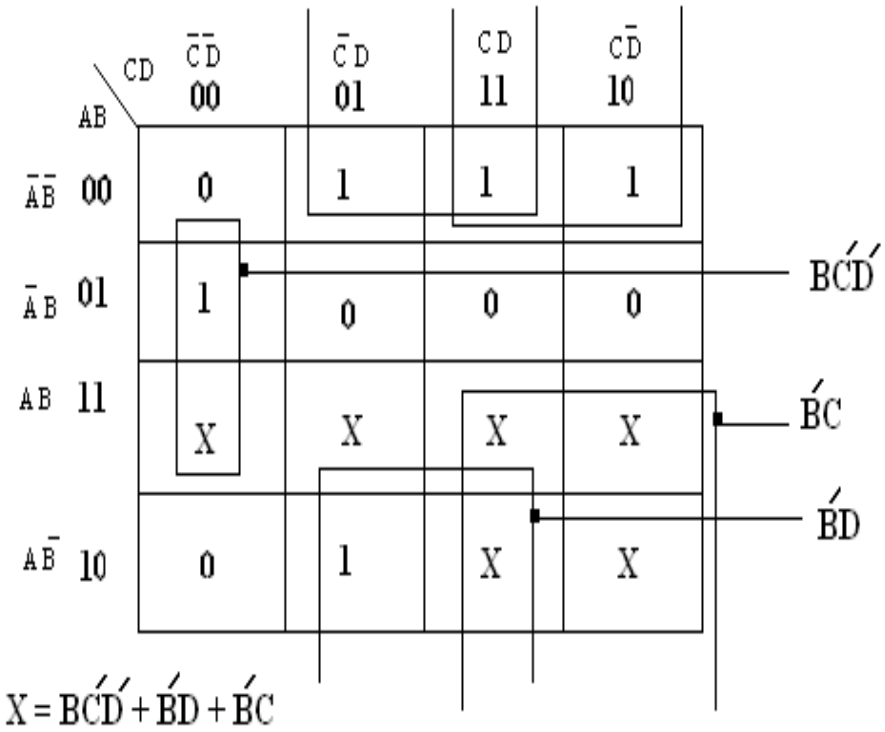
First we make the truth table

BCD no A B C D	EXCESS-3 NO W X Y Z
0 0 0 0	0 0 1 1
0 0 0 1	0 1 0 0
0 0 1 0	0 1 0 1
0 0 1 1	0 1 1 0
0 1 0 0	0 1 1 1
0 1 0 1	1 0 0 0
0 1 1 0	1 0 0 1
0 1 1 1	1 0 1 0
1 0 0 0	1 0 1 1
1 0 0 1	1 1 0 0

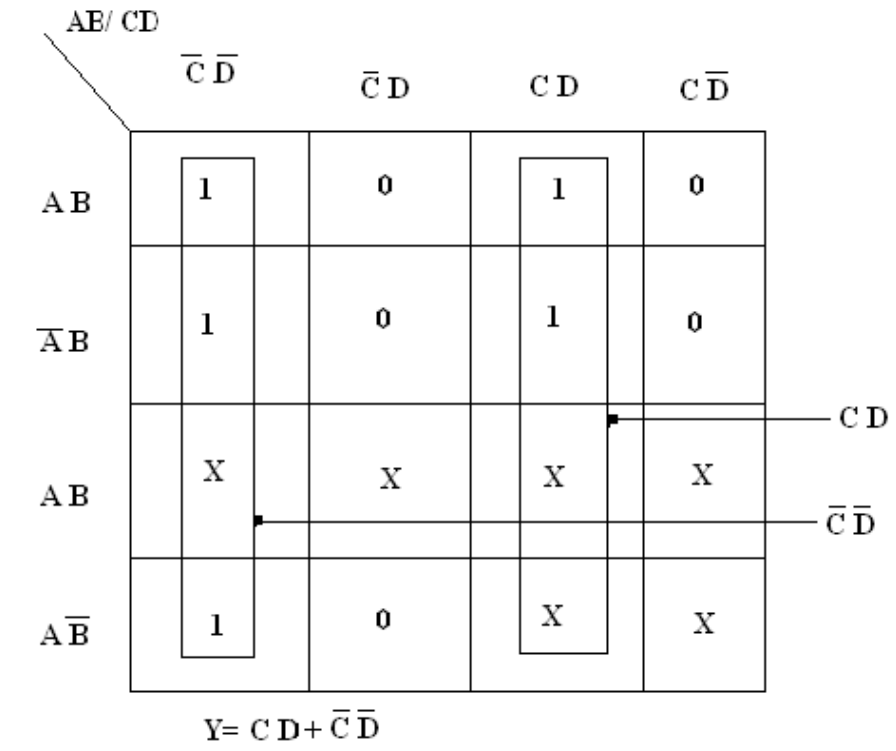
Then by using K maps we can have simplified functions for w, x, y, z as shown below:

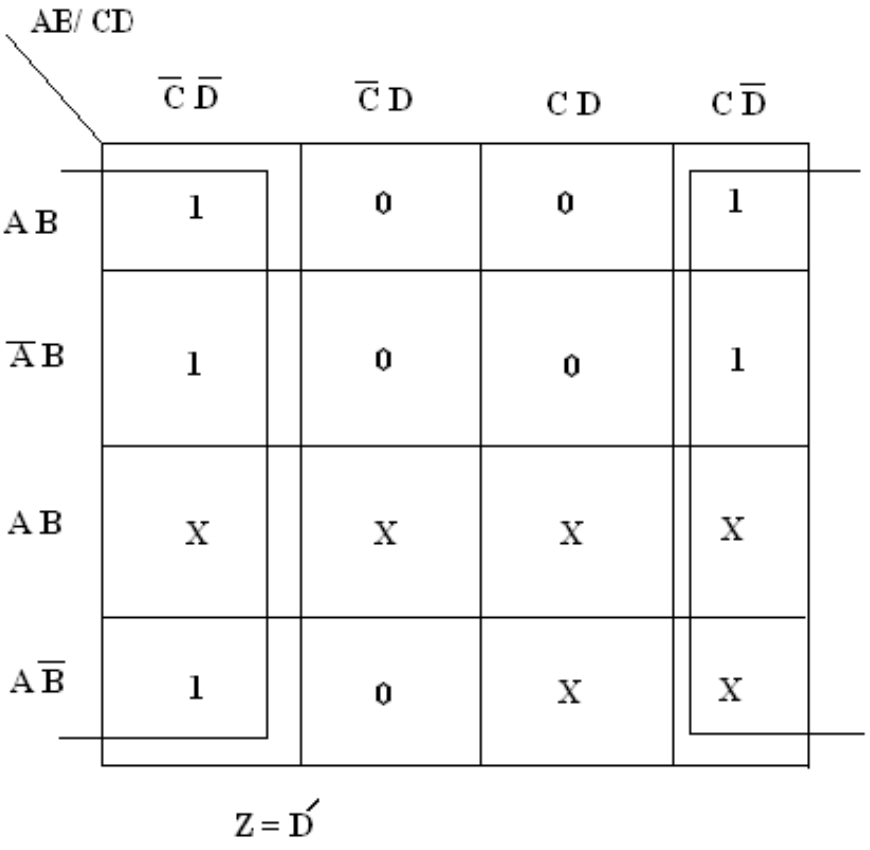


$W = BD + AD + A\acute{B} + BC$



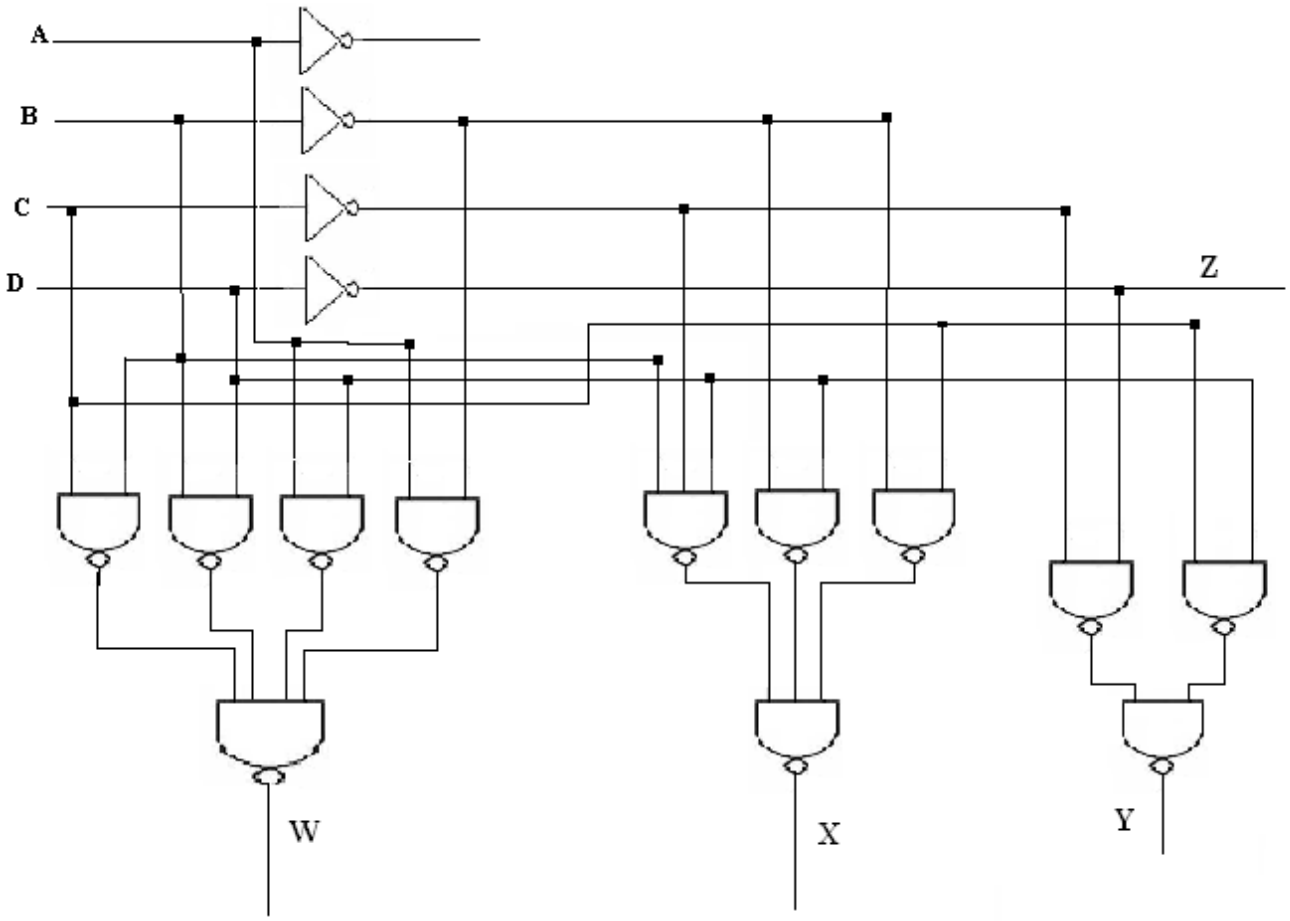
$X = B\acute{C}\acute{D} + \acute{B}D + \acute{B}C$





NAND gate implementation for simplified function

W = BD + AD + AB' + BC
By complementing twice we get
W = ((BD + AD + AB' + BC)')'
= ((BD)' . (AD)' . (AB')' . (BC)')'
X = BC'D + B'D + B'C
By complementing twice we get
X = BC'D + B'D + B'C
= ((BC'D)' . (B'D)' . (B'C)')'
Y = C'D' + CD
= ((C'D')' + (CD)')'
Z = D'



Logic diagram for BCD to excess 3 code converter by using minimum number of NAND gates

Q.57 With the help of a suitable diagram, explain how do you convert a JK flipflop to T type flipflop. **(4)**

Ans:

Given flip flop is JK flip flop and it is required to convert JK into T. First we draw the characteristic table of T flip flop and then relate the transition with excitation table of JK flip flop.

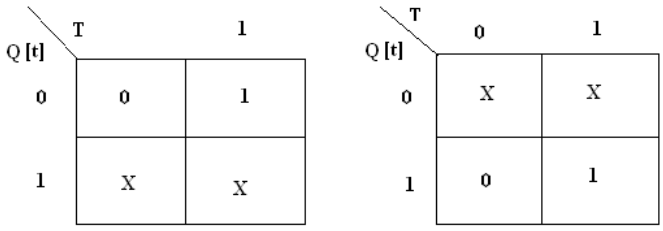
Q[t]	T	Q[T+1]	J	K
0	0	0	0	X
0	1	1	1	X
1	0	1	X	0
1	1	0	X	1

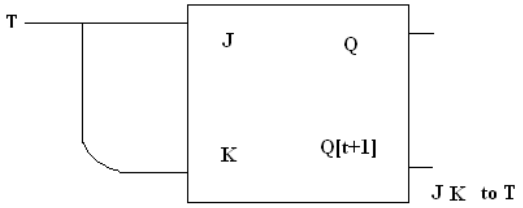
Excitation Table

JK Flip Flop

Q[t]	Q[t+1]	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Now we solve K maps for J and K by considering T and Q(t) as input.





Logic diagram convert a JK flipflop to T type flipflop.

Q.58 A number of 256 x 8 bit memory chips are available. To design a memory organization of 2 K x 8 memory. Identify the requirements of 256 x 8 memory chips and explain the details. **(8)**

Ans:

Chips available=256x8

Required capacity=2048x8

Number of chips=(2048x8)/(256x8)=8=(256=2^8)

Address lines required for 2048x8chip=11(2048=2^{11})

Thus the size of the decoder=3x8

Q.59 Convert $(177.25)_{10}$ to octal. **(8)**

Ans:

$(177.25)_{10} = (\quad)_8$

First we take integer part

8	177	**1**
8	22	**6**
8	2	**2**
	0	

Thus $(177)_{10} = (261)_8$

Now as 0.25 x 8 = 2.00

and 0.00 x 8 = 0

Thus $(0.25)_{10} = (0.2)_8$

Therefore, Thus $(177.25)_{10} = (261.2)_8$

Q.60 Perform the following subtraction using 1's complement

(i) 11001 – 10110 (ii) 11011 - 11001 **(8)**

Ans:

(i) 11001 – 10110 = X – Y

X = 11001

1's complement of Y = 01001

Sum = 1 00010

End around carry = 1

So X-Y = 00011

(ii) 11011 – 11001 = X – Y

X = 11011

1's complement of Y = 00110

Sum = 1 00001

End around carry = 1

So X-Y = 00010

Q.61 Prove the following identities

(i) $\overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,B\,\overline{C} + A\overline{B}\,\overline{C} + A\,B\,\overline{C} = \overline{C}$

(ii) $A\,B + A\,B\,C + \overline{A}\,B + A\,\overline{B}\,C = B + A\,C$ **(8)**

Ans:

(i) LHS = A'B'C' + A'BC' + AB'C' + ABC'

= A'C' (B' + B) + AC' (B' + B)

= A'C' + AC' [as B'+B = 1]

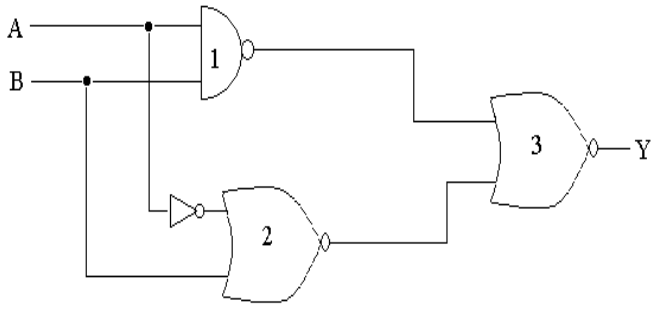
= C' (A' + A)

= C' [as A'+A =1]

= **RHS.**
Hence Proved

(ii) LHS = AB + ABC + A'B + AB'C = B + AC
= B (A + A') + AC (B + B')
= B + AC [as B + B' = A + A' = 1]
= B + AC
= **RHS.**
Hence Proved

Q.62 Find the boolean expression for the logic circuit shown below. **(8)**



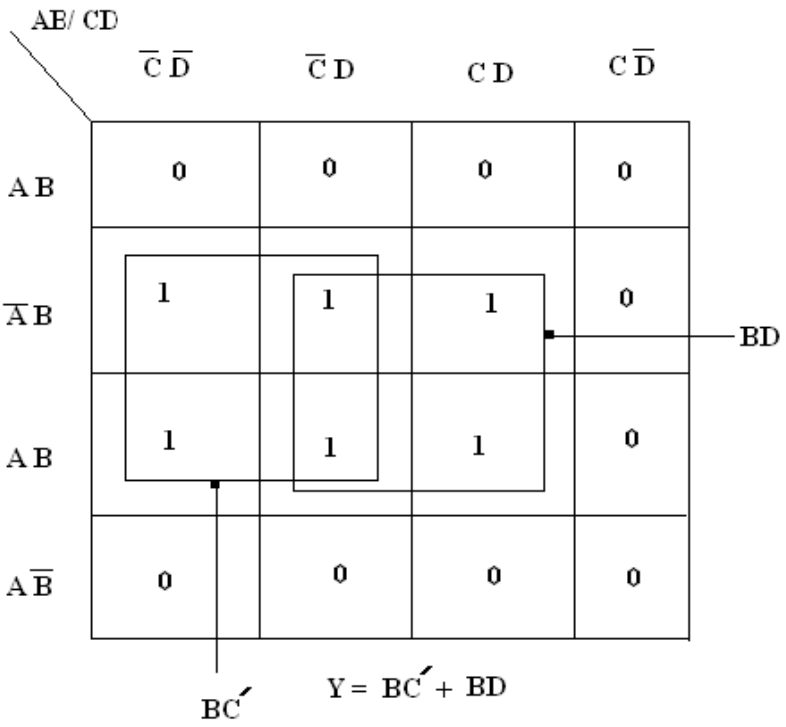
Ans:

Output of Gate-1 (NAND) = (AB)'
Output of Gate-2 (NOR) = (A'+B)'
Output of Gate-3 (NOR) = [(AB)' + (A'+B)']'
Now applying De-Morgans law, (X+Y)' = X'Y'
and (XY)' = (X'+Y')
[(AB)' + (A'+B)']' = [(AB)']' [(A'+B)']'
= (AB) (A'+B)
= AA'B + ABB
= ABB
= AB.

Q.63 Reduce the following equation using k-map
$Y = B\overline{C}\,\overline{D} + \overline{A}\,B\,\overline{C}\,D + A\,B\,\overline{C}\,D + \overline{A}\,B\,C\,D + A\,B\,C\,D$ **(8)**

Ans:

Multiplying the first term by (A+A')
Y = A'BC'D' + ABC'D' + A'BC'D + ABC'D + A'BCD + ABCD
$= \sum(4,12,5,7,15,13)$
= BC' + BD



Q.64 Implement the following function using 8 to 1 multiplexer

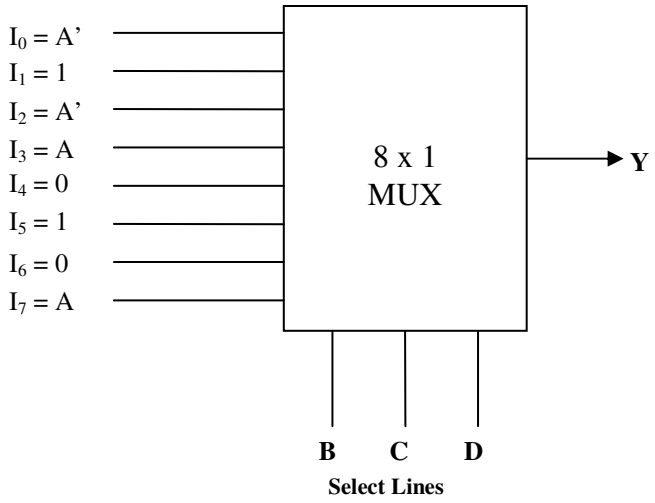
$$Y(A,B,C,D) = \sum (0,1,2,5,9,11,13,15)$$ **(8)**

Ans:

We will take three variables B,C & D at selection lines and A as input. Now there are eight inputs and they can be 0,1,A or A' depending on the Boolean function.

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
A'	0	1	2	3	4	5	6	7
A	8	9	10	11	12	13	14	15
	A'	1	A'	A	0	1	0	A

Now, the realization is:



Q.65 (i) How many 128×8 RAM chips are required to provide a memory capacity of 2048 bytes.

(ii) How many lines of address bus must be used to access 2048 bytes of memory. How many lines of these will be common to each chip?

(iii) How many bits must be decoded for chip select? What is the size of decoder?

(8)

Ans:

(i) Available RAM chips = 128 x 8

Required memory capacity = 2048 x 8

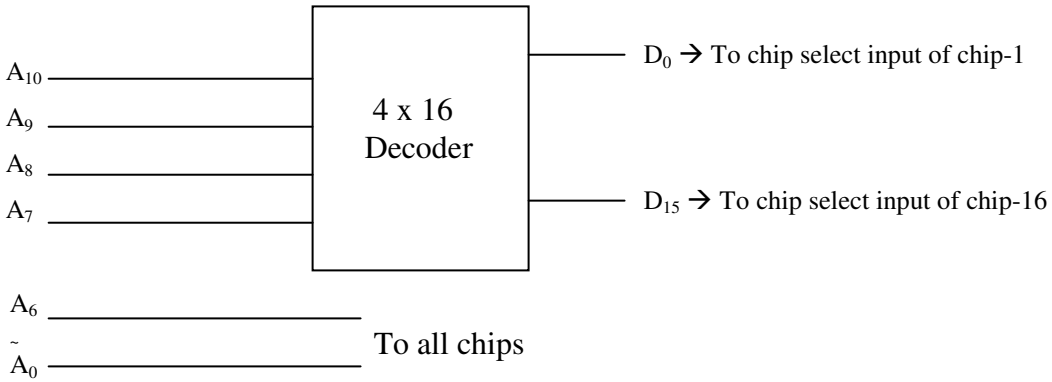
Number of chips required = (2048 x 8) / (128 x 8)

= 16.

(ii) Chips available are of 128 x 8 in size. It means that total 128 (2^7) locations are there and each location can store 8 bits. Thus the total number of address lines required to access 128 locations is 7. As seven address lines can address 2^7 locations. These seven lines are common to all chips.

Now to access 2048 locations, we require 11 address lines, as $2048 = 2^{11}$

(iii) These higher order lines will be applied to decoder input. The number of inputs to the decoder will be 11 - 7 = 4. The size of the decoder will be 4x16. These 16 decoder outputs will be connected to the chip select input of individual chips.



Q.66 How many bits are required at the input of a ladder D/A converter, if it is required to give a resolution of 5mV and if the full scale output is +5V. Find the %age resolution. **(8)**

Ans:

First we find out the ratio of Full scale output to Resolution = 5V / 5 mV = 1000.
Now number of bits = $\log_2 1000 = 10$.
Percentage Resolution = 5 mV / 5 V * 100 = 0.1%

Q.67 A 6-bit Dual Slope A/D converter uses a reference of –6V and a 1 MHz clock. It uses a fixed count of 40 (101000). Find Maximum Conversion Time. **(4)**

Ans

The time T_1 given by
$T_1 = 2^N T_C$ where N = no. of Bits, T_c = time period of clock pulse
Given N = 6, T_C = 1/ 1MHz = 1 µs.
Therefore $T_1 = 2^6 \text{ X } 10^{-6}$ s = 64 µs.

Q.68 A 2-digit BCD D/A converter is a weighted resistor type with $E_R = 1$ Volt, with $R = 1M\Omega$, $R_f = 10K\Omega$. Find resolution in Percent and Volts. **(5)**

Ans

Resolution = $1/2^2 = 0.25$ volts.
As the resolution is determined by number of input bits of D/A converter; For example two bit converter has 2^2 (4) possible output levels, therefore its resolution is 1 part in 4
In percent it will be ¼ X 100 = 25%
In volts, it will be 0.25 volts.

PART – III

DESCRIPTIVES

Q.1 Distinguish between min terms and max terms. **(6)**

Ans: Distinguish between Minterms and Maxterms:

(i) Each individual term in standard Sum Of Products form is called as minterm whereas each individual term in standard Product Of Sums form is called maxterm.
(ii) The unbarred letter represent 1's and the barred letter represent 0's in min terms, whereas the unbarred letter represent 0's and the barred represent 1's in maxterms.
(iii) If a system has variables A, B, C then the minterms would be in the form ABC, whereas the maxterm would be in the form A+B+C.
(iv) The minterm designation for three variable expression be

$Y=\sum m\ (1, 3, 5, 7)$

Where the capital $\sum$ represents the product and m stands for minterms.
Decimal number 1 corresponds to binary number 001 or $\overline{A}\ \overline{B}\,C$
Decimal number 3 corresponds to binary number 011 or $\overline{A}\,BC$
Decimal number 5 corresponds to binary number 101 or $A\,\overline{B}\,C$
Decimal number 7 corresponds to binary number 111 or ABC.
Whereas the Maxterm designation for three variable expression be

$Y=\prod M\ (0, 1, 3, 4)$

Where the capital $\prod$ represents the product and M stands for maxterms.
Decimal 0 means binary 000 and term is A+B+C
Decimal 1 means binary 001 and term is $A+B+\overline{C}$
Decimal 3 means binary 011 and term is $A+\overline{B}+\overline{C}$
Decimal 4 means binary 100 and term is $\overline{A}+B+C$

Q.2 What are universal gates. Construct a logic circuit using NAND gates only for the expression x = A . (B + C). **(7)**

Ans:

Universal Gates: NAND and NOR Gates are known as Universal gates. The AND, OR, NOT gates can be realized using any of these two gates. The entire logic system can be implemented by using any of these two gates. These gates are easier to realize and consume less power than other gates.

Construction of a logic circuit for the expression X = A (B + C) using NAND gates is Shown in fig.4 (b)

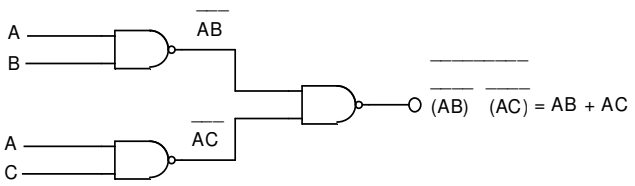


Fig.4(b) Logic Diagram for the expression X = A (B + C)

Q.3 Mention the various IC logic families. **(7)**

Ans:

Various IC Logic Families: Digital IC's are fabricated by employing either the Bipolar or the Unipolar Technologies and are referred to as Bipolar Logic Family or Unipolar Logic Family

I Bipolar Logic Families:

There are two types of operations in Bipolar Logic Families

1. Saturated Logic Families
2. Non-saturated Logic Families

1. Saturated Logic Families: In Saturated Logic, the transistors in the IC are driven to saturation.

(i) Resistor-Transistor Logic (RTL).
(ii) Direct-Coupled Transistor Logic (DCTL)
(iii) Integrated-Injection Logic (I^2L)
(iv) Diode -Transistor Logic (DTL)
(v) High-Threshold Logic (HTL)
(vi)Transistor-Transistor Logic (TTL)

2. Non-saturated Logic: In Non-saturated Logic, the transistors are not driven into saturation.

(i) Schottky TTL
(ii) Emitter Coupled Logic (ECL)

II Unipolar Logic Families:

MOS devices are Unipolar devices and only MOSFETs are employed in MOS logic circuits. The MOS logic families are

(i) PMOS
(ii) NMOS, and
(iii) CMOS

while in PMOS only p-channel MOSFETs are used and in NMOS only n-channel MOSFETs are used, in complementary MOS (CMOS), both P and N channel MOSFETs are employed and are fabricated on the same silicon chip.

Q.4 What is a half-adder? Explain a half-adder with the help of truth-table and logic diagram. **(10)**

Ans:

Half Adder: A logic circuit for the addition of two one-bit numbers is referred to as an half-adder. The addition process is illustrated in truth table shown in Table 6.1. Here A and B are the two inputs and S (SUM) and C (CARRY) are two outputs.

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Table 6.1 Truth Table for Half Adder

From the truth table, we obtain the logical expressions for S and C outputs as

$S = \overline{A}B + A\overline{B}$

$C = AB$

The logic diagram for an Half-adder using gates is shown in fig.6(a)

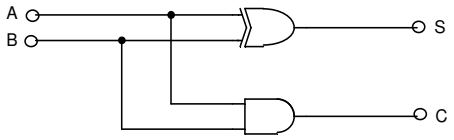


Fig.6(a) Logic Diagram for an Half-adder

Q.5 Using a suitable logic diagram explain the working of a 1-to-16 de multiplexer.**(7)**

Ans:

Working of a 1-to-16 Demultiplexer: A demultiplexer takes in data from one line and directs it to any of its N outputs depending on the status of the selected inputs. If the number of output lines is N (16), the number of select lines m is given by 2^m = N.i.e., 2^4 = 16. So, the number of select lines required for a 1-to-16 demultiplexer is 4. Table 7.1 shows the Truth Table of 1-to-16 Demultiplexer. The input can be sent to any of the 16 outputs, D_0 to D_{15}. If DCBA = 0000, the input goes to D_0. If DCBA = 0001, the input goes to D_1 and so on.

Fig.7(a) shows the logic diagram of a 1-to-16 demultiplexer, consists of 8 NOT gates, 16 NAND gates, one data input line(G), 4 select lines (A,B,C,D) and 16 output lines (D_0, D_1, D_2 ------D_{16}). The 8 NOT gates prevent excessive loading of the driving source. One data input line G is implemented with a NOR gate used as negative AND gate. A low level in each input $\overline{G_1}$ and $\overline{G_2}$ is required to make the output G high. The output G of enable is one of the inputs to all the 16 NAND gates. G must be high for the gates to be enabled. If the enable gate is not activated then all sixteen de multiplexer outputs will be high irrespective of the state of the select lines A,B,C,D.

Demulti-plexer Input	Selection Lines D C B A	Logic Function	Demultiplexer Outputs D_0 D_1 D_2 D_3 D_4 D_5 D_6 D_7 D_8 D_9 D_{10} D_{11} D_{12} D_{13} D_{14} D_{15}
0	0 0 0 0	$\overline{D}\,\overline{C}\,\overline{B}\,\overline{A}$	0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1	0 0 0 1	$\overline{D}\,\overline{C}\,\overline{B}$ A	1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
2	0 0 1 0	$\overline{D}\,\overline{C}\,B\,\overline{A}$	1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1
3	0 0 1 1	$\overline{D}\,\overline{C}$ B A	1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
4	0 1 0 0	$\overline{D}$ C $\overline{B}\,\overline{A}$	1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1
5	0 1 0 1	$\overline{D}$ C $\overline{B}$ A	1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1
6	0 1 1 0	$\overline{D}$ C B $\overline{A}$	1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1
7	0 1 1 1	$\overline{D}$ C B A	1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1
8	1 0 0 0	D $\overline{C}\,\overline{B}\,\overline{A}$	1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1
9	1 0 0 1	D $\overline{C}$ $\overline{B}$ A	1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1
10	1 0 1 0	D $\overline{C}$ B $\overline{A}$	1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1
11	1 0 1 1	D $\overline{C}$ B A	1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1
12	1 1 0 0	D C $\overline{B}\,\overline{A}$	1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1
13	1 1 0 1	D C $\overline{B}$ A	1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1
14	1 1 1 0	D C B $\overline{A}$	1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1
15	1 1 1 1	D C B A	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0

Table 7.1 Truth Table of 1-to-16 Demultiplexer

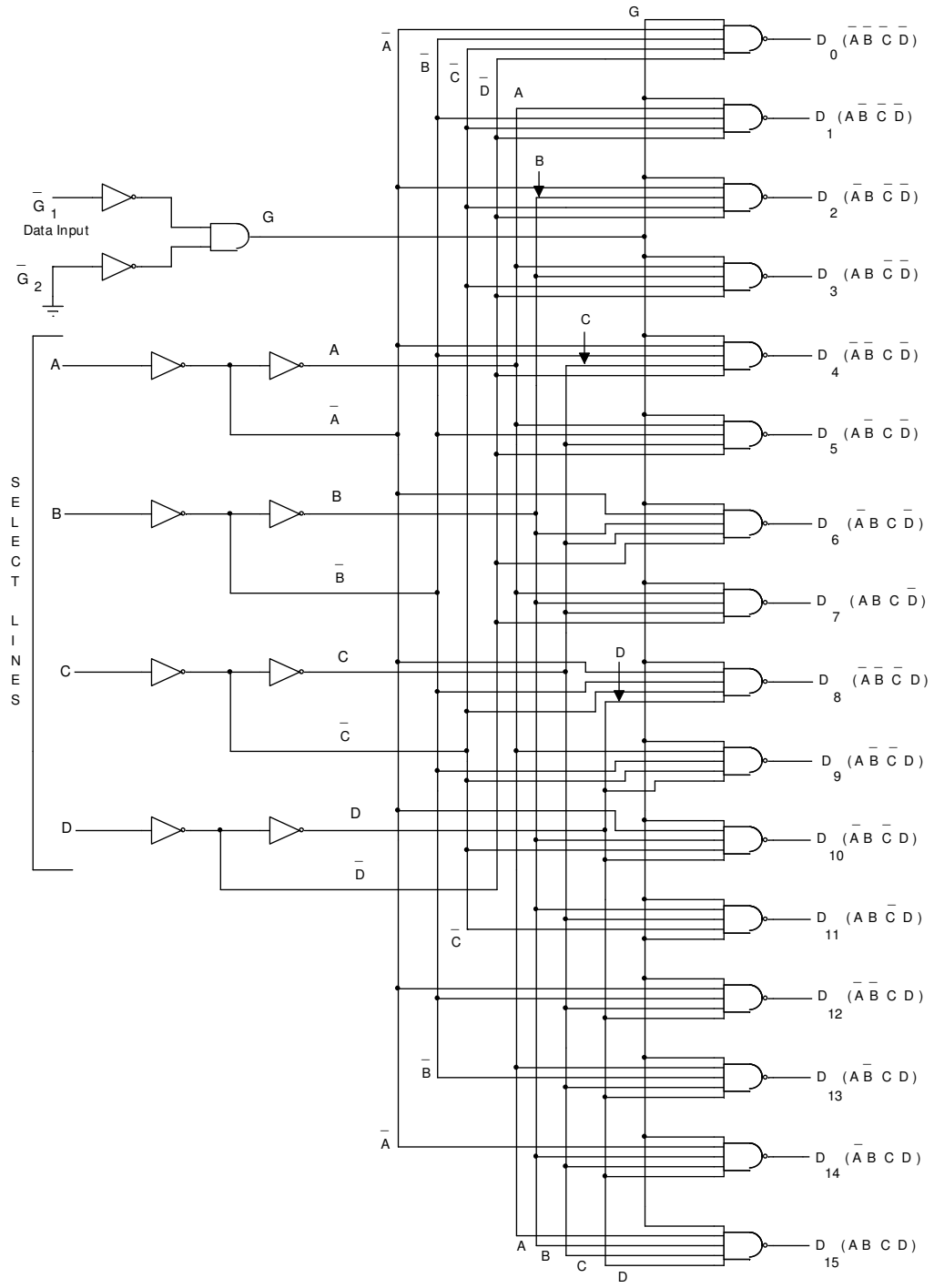


Fig.7(a) Logic Diagram of 1-to-16 De multiplexer

Q.6 . With relevant logic diagram and truth table explain the working of a two input EX-OR gate. **(7)**

Ans:

Two-Input EX-OR Gate: An Exclusive-OR (EX-OR) gate recognizes words which have an odd number of ones. Fig.7(b) shows the logic diagram of an EX-OR gate and Fig.7(c) shows the symbol of an EX-OR Gate. The upper AND gate gives an output $\overline{A}$ B and the lower AND gate gives an output A $\overline{B}$.

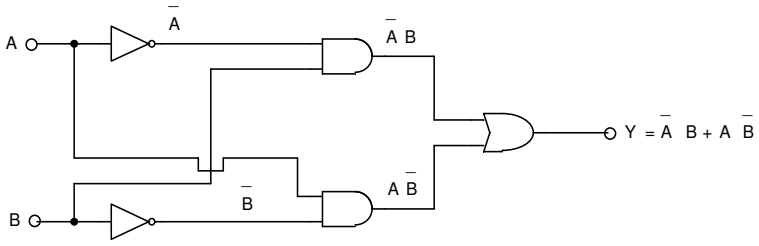


Fig.7(b) Logic Diagram of EX-OR Gate

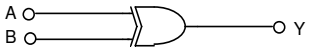


Fig.7(c) Symbol of EX-OR Gate

Therefore, the output equation becomes Y = $\overline{A}$ B + A $\overline{B}$ = A EX-OR B = A $\oplus$ B
If both A and B are low, the output is low. If either A or B (not both) are high (and the other is low), the output is high. If both A and B are high, output is low. Thus the output is 1 when A and B are different. Table 7.2 shows the Truth Table for EX-OR gate.

A	**B**	**Y ($\overline{A}$ B + A $\overline{B}$)**
0	0	0
0	1	1
1	0	1
1	1	0

Table 7.2 Truth Table of EX-OR Gate

Q.7 With the help of clocked JK flip flops and waveforms, explain the working of a three bit binary ripple counter. Write truth table for clock transitions. **(14)**

Ans:

3-Bit Binary Ripple Counter: In Ripple Counters, all the Flip-Flops are not clocked simultaneously and the flip-flops do not change state exactly at the same time. A 3-bit

Binary Counter has maximum of 2^3 states i.e., 8 states, which requires 3 Flip-Flops. The word Binary Counter means a counter which counts and produces binary outputs 000,001,010--111.It goes through a binary sequence of 8 different states (i.e, from 0 to 7). Fig.8(a) shows the logic circuit of a 3-bit Binary Ripple Counter consisting of 3 Edge Triggered JK flip-flops. As indicated by small circles at the CLK input of flip-flops, the triggering occurs when CLK input gets a negative edge. Q_0 is the Least Significant Bit (LSB) and Q_2 is the Most Significant Bit (MSB). The flip-flops are connected in series. The Q_0 output is connected to CLK terminal of second flip-flop. The Q_1 output is connected to CLK terminal of third flip-flop. It is known as a Ripple Counter because the carry moves through the flip-flops like a ripple on water.

Working: Initially, CLR is made Low and all flip-flops Reset giving an output Q = 000. When CLR becomes High, the counter is ready to start. As LSB receives its clock pulse, its output changes from 0 to 1 and the total output Q = 001. When second clock pulse arrives, Q_0 resets and carries (i.e., Q_0 goes from 1 to 0 and, second flip flop will receive CLK input). Now the output is Q = 010. The third CLK pulse changes Q_0 to 1 giving a total output Q = 011. The fourth CLK pulse causes Q_0 to reset and carry and Q_1 also resets and carries giving a total output Q = 100 and the process goes on. The action is shown is Table 8.1.The number of output states of a counter are known as Modulus (or Mod). A Ripple Counter with 3 flip-flops can count from 0 to 7 and is therefore, known as Mod-8 counter.

Counter State	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Table.8.1 Counting Sequence of a 3-bit Binary Ripple Counter

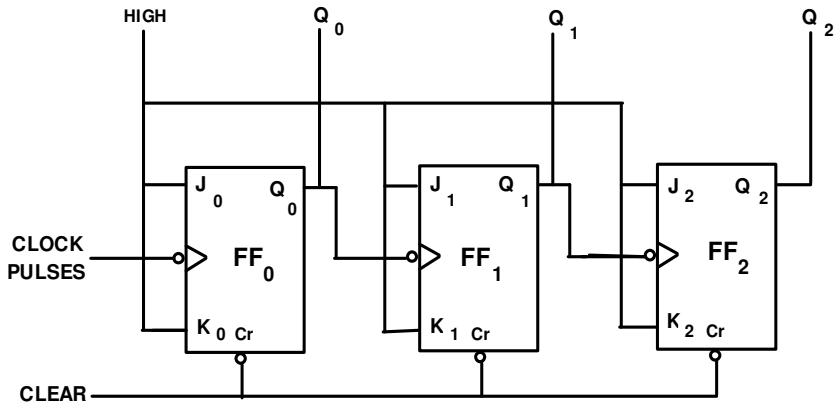


Fig.8(a) Logic Diagram of 3-Bit Binary Ripple Counter

Ripple counters are simple to fabricate but have the problem that the carry has to propagate through a number of flip flops. The delay times of all the flip flops are added. Therefore, they are very slow for some applications. Another problem is that unwanted pulses occur at the output of gates.

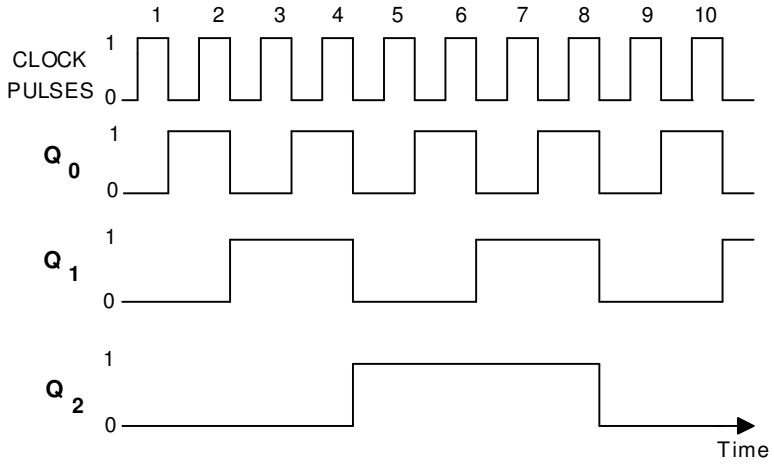


Fig.8(b) Timing Diagram of 3-bit Binary Ripple Counter

The timing diagram is shown in *Fig.8(b). FF_0* is LSB flip flop and *FF_2* is the MSB flip flop. Since *FF0* receives each clock pulse, Q_0 toggles once per negative clock edge as shown in Fig. 8*(b).*The remaining flip flops toggle less often because they receive negative clock edge from preceding flip flops. When Q_0 goes from 1 to 0, *FF_1* receives a negative edge and toggles. Similarly, when Q_1 changes from 1 to 0, *FF_2* receives a negative edge and toggles. Finally when Q_2 changes from 1 to 0, *FF_3* receives a negative edge and toggles. Thus whenever a flip flop resets to 0, the next higher flip flop toggles.

This counter is known as ripple counter because the 8th clock pulse is applied, the trailing edge of 8th pulse causes a transition in each flip flop. Q_0 goes from High to Low, this causes Q_1 go from High to Low which causes Q_2 to go from High to Low which causes Q_3

to go from High to Low. Thus the effect ripples through the counter. It is the delay caused by this ripple which result in a limitation on the maximum frequency of the input signal.

Q.8 Using D-Flip flops and waveforms explain the working of a 4-bit SISO shift register. **(14)**

Ans:

Serial In - Serial Out Shift Register: Fig.9(a) shows a 4 bit serial in - serial out shift register consisting of four D flip flops FF_0 , FF_1 , FF_2 and FF_3. As shown it is a positive edge triggered device. The working of this register for the data 1010 is given in the following steps.

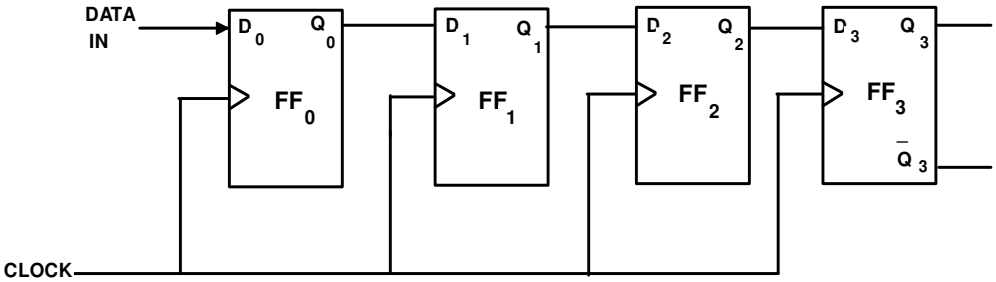


Fig.9(a) Logic Diagram of 4-bit Serial In – Serial Out Shift Register

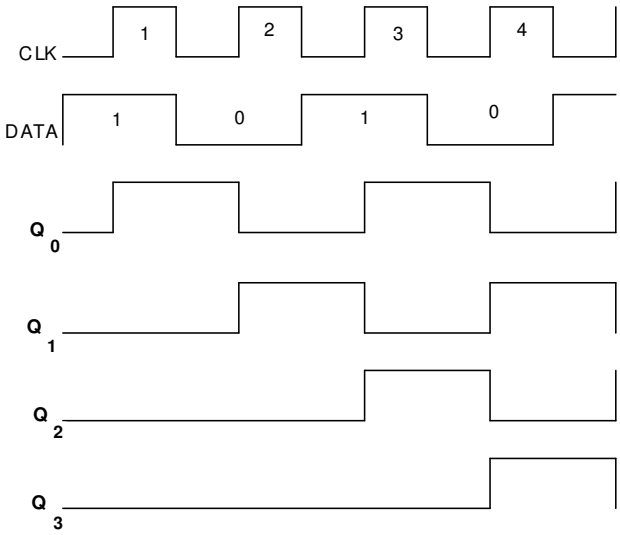


Fig.9(b) Output Waveforms of 4-bit Serial-in Serial-out Register

1. Bit 0 is entered into data input line. $D_0 = 0$, first clock pulse is applied, FF_0 is reset and stores 0.
2. Next bit 1 is entered. $Q_0 = 0$, since Q_0 is connected to D_1, D_1 becomes 0.
3. Second clock pulse is applied, the 1 on the input line is shifted into FF_0 because FF_0 sets. The 0 which was stored in FF_0 is shifted into FF_1.
4. Next bit 0 is entered and third clock pulse applied. 0 is entered into FF_0, 1 stored in FF_0 is shifted to FF_1 and 0 stored in *FF1* is shifted to FF_2.
5. Last bit 1 is entered and 4th clock pulse applied. 1 is entered into FF_0, 0 stored in FF_0 is shifted to FF_1, 1 stored in FF_1 is shifted to FF_2 and 0 stored in *FF2* is shifted to FF_3.

This completes the serial entry of 4 bit data into the register. Now the LSB 0 is on the output Q_3.

6. Clock pulse 5 is applied. LSB 0 is shifted out. The next bit 1 appears on Q_3 output.
7. Clock pulse 6 is applied. The 1 on Q_3 is shifted out and 0 appears on Q_3 output.
8. Clock pulse 7 is applied. 0 on Q_3 is shifted out. Now 1 appears on Q_3 output.
9. Clock pulse 8 is applied. 1 on Q_3 is shifted out.
10. When the bits are being shifted out (on CLK pulse 5 to 8) more data bits can be entered in.

Q.9 With the help of R-2R binary ladder, explain the working of a 4-bit D/A converter **(14)**

Ans:

R-2R Ladder network method: In a R-2R ladder network method of digital to analog conversion, irrespective of number of bits of the DAC only two convenient values of resistors are needed in the ratio of 1:2 as depicted in fig 10(a). An R-2R Ladder Network based on constant reference current. In the circuit of fig 10(a) points G are actual ground and points G' are virtual ground. Therefore the potential at all the G_s and G'_s is zero. Between ground (actual or virtual) and node A there are two resistors each of value 2R in parallel. Therefore this resultant resistance between ground and node A is R and the current through each of the 2R resistance connected to node A must be same. Let us say this current is I. Then the current flowing from A to B through the resistor R is 2I. Then the total resistance from ground to node B through the node A becomes 2R. Also the resistance directly connected between ground and B is also 2R. So between the node B and ground there are two equal resistances in parallel each of value 2R. Therefore, the resultant resistance is R and the current approaching to node B from both sides must be equal. Since current approaching from the side of node A is 2I, therefore the current approaching to the node B from the resistor 2R under it must also be 2I. Hence the total current approaching the node C from the side of node B is 4I. On the basis of the same logic the current approaching to node D from the side of node C must be 8I and the current approaching it form the 2R resistor under node D should also be 8I.

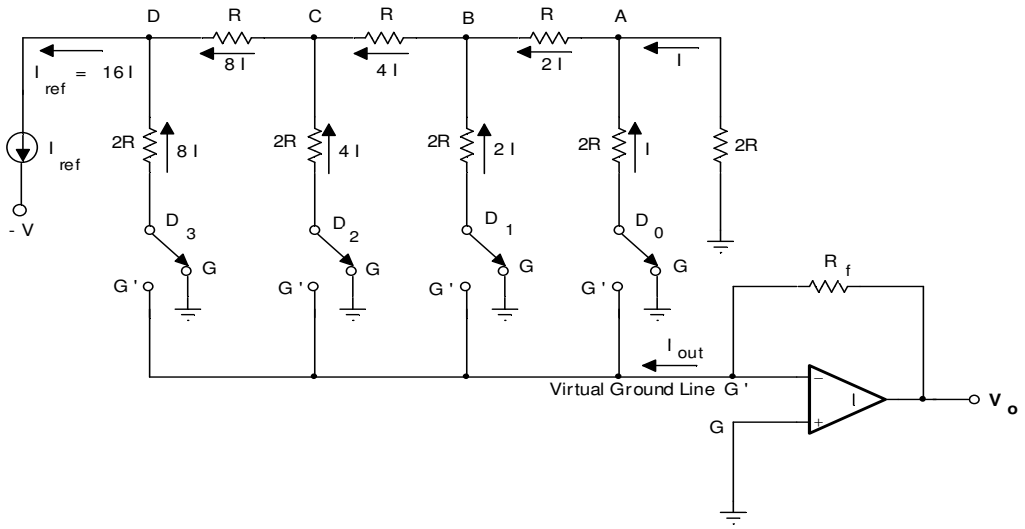


Fig.10(a) R-2R Ladder Network D/A Converter

Whenever any of the bit or bits of the digital input word D_3 D_2 D_1D_0 is high, the corresponding transistor switch is ON *i.e.* connected to virtual ground and the current of that vertical branch of the ladder comes from the output, otherwise the current of the vertical branch comes directly from the actual ground without any effect on the output. Hence the output current (lout) gives the analog current value corresponding to the digital input word. This analog current gets converted to the analog voltage V_o.

An R-2R 4-Bit Ladder Network DAC based on reference voltage: An R-2R 4-bit Ladder Network D/A Converter is shown in fig. 10(b)

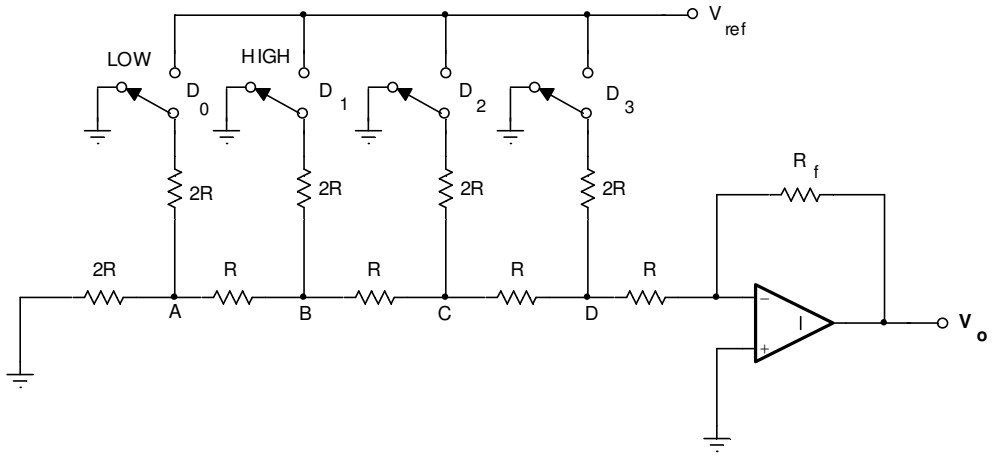


Fig.10(b) R-2R 4-bit Ladder Network D/A Converter

Proof:

Step 1: If the digital value to be converted to analog value is 0001 *i.e.*D_0 is on the high side connected to V_{ref} while D_1, D_2, and D_3 are connected to ground. Then the circuit redrawn as shown inFig.10(c).

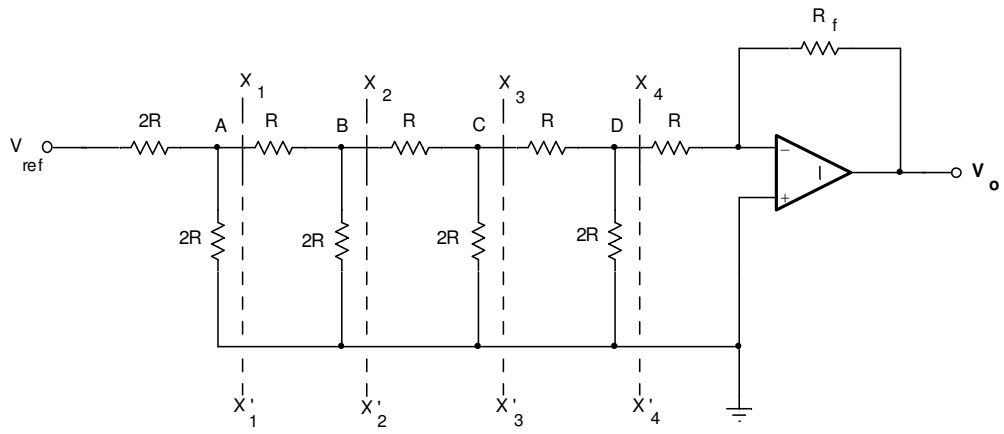


Fig.10(c) R-2R Ladder Network D/A Converter when D_0 is connected to V_{ref} and D_1,D_2,D_3 are connected to ground

Applying Thevenin's theorem at X_1,X_1' , the circuit of fig.10(c) becomes the equivalent circuit shown in fig.10(d)

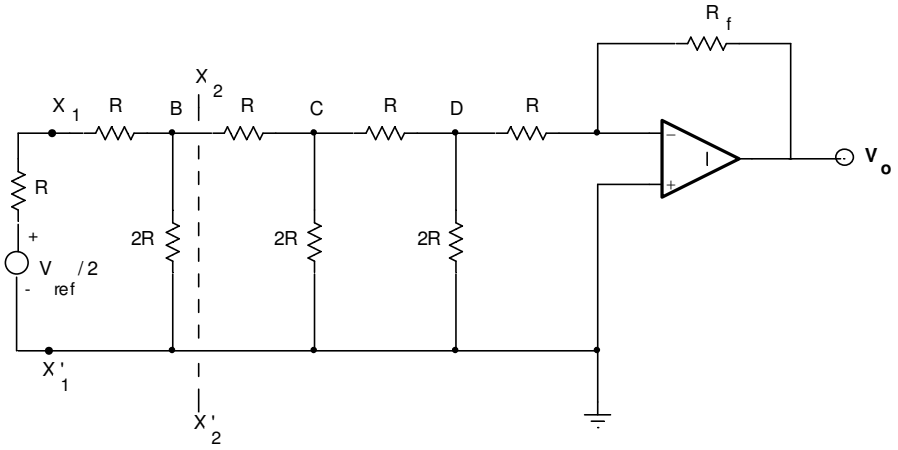


Fig.10(d) R-2R Ladder Network D/A Converter when Thevenin's Theorem applied at X_1 and X_1'

Again Applying Thevenin's Theorem at X_2,X_2', then the circuit of fig.10(d) becomes the equivalent circuit shown in fig.10(e).

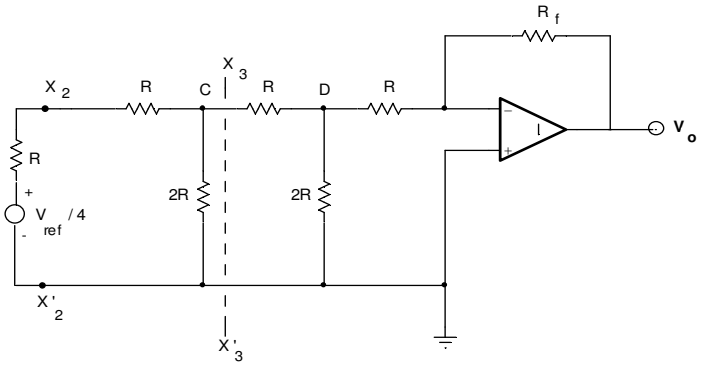


Fig.10(e) R-2R Ladder Network D/A Converter when Thevenin's Theorem applied at X_2 and X_2'

Again Applying Thevenin's Theorem at X_3,X_3' the circuit of fig.10(e) becomes the equivalent circuit shown in fig.10(f):

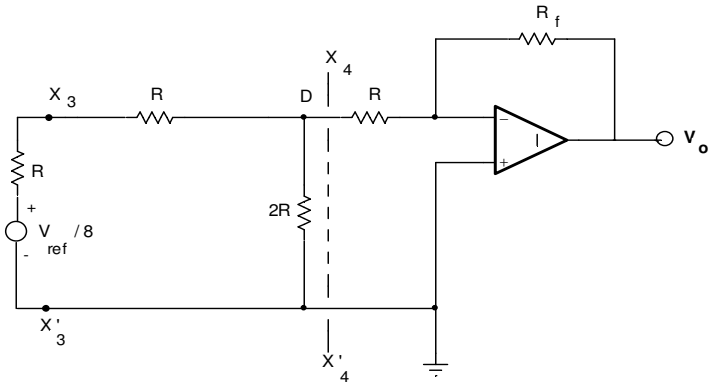


Fig.10(f) R-2R Ladder Network D/A Converter when Thevenin's Theorem applied at X_3 and X_3'

Once Again applying Thevenin's theorem at section X_4, X_4' the circuit of fig.10(f) finally becomes the equivalent circuit shown in fig.10(g)**.**

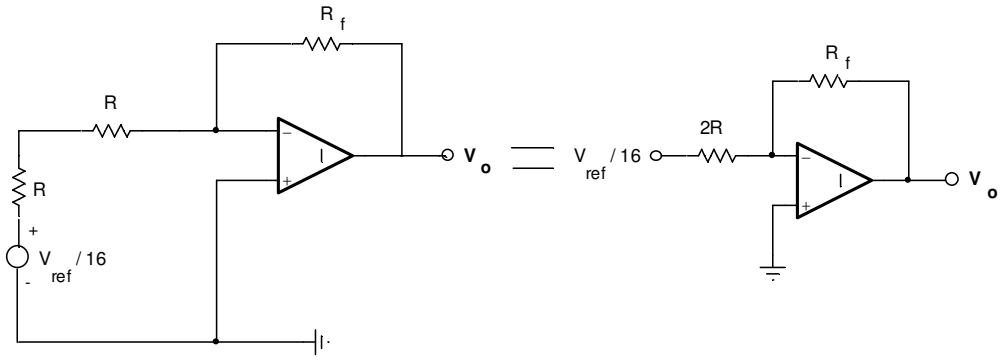


Fig.10(g) R-2R Ladder Network D/A Converter when Thevenin's Theorem applied at X_4 and X_4'

Step 2: If D_1 is high (connected to V_{ref}) and D_0, D_2, D_3 are all low (connected to ground), then the circuit becomes:

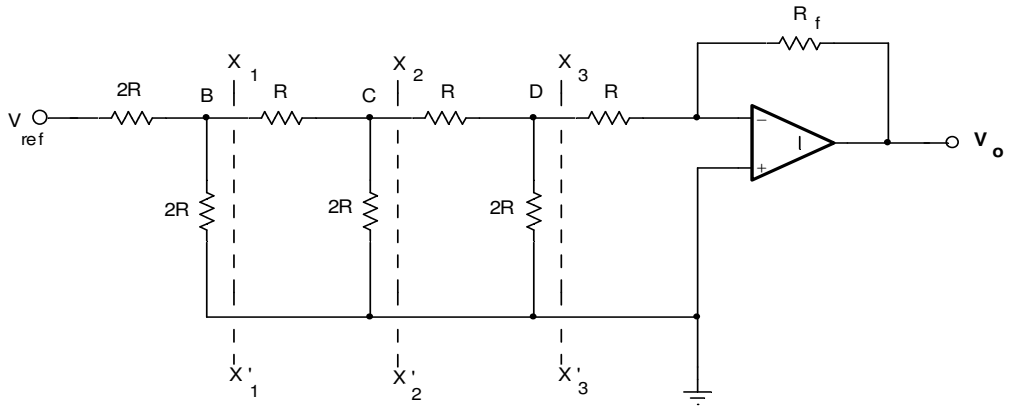


Fig.10(h) R-2R Ladder Network D/A Converter when D_1 is connected to V_{ref} and D_0,D_2,D_3 are connected to ground

Applying Thevenin's Theorem, three times and reducing the circuit each time at sections X_1, X_2, X_3 we finally get the circuit as shown in fig.10(i).

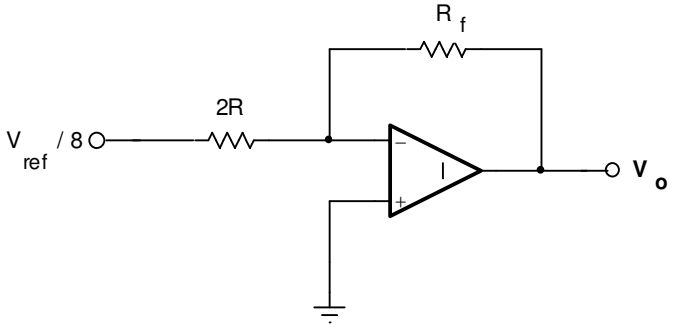


Fig.10(i) Equivalent Circuit when D_1 is connected to V_{ref} D_0,D_2,D_3 are connected to ground

Step 3: Repeating the same exercise of D_2 High and other bits Low, we get the finally reduced Circuit as shown in fig.10(j).

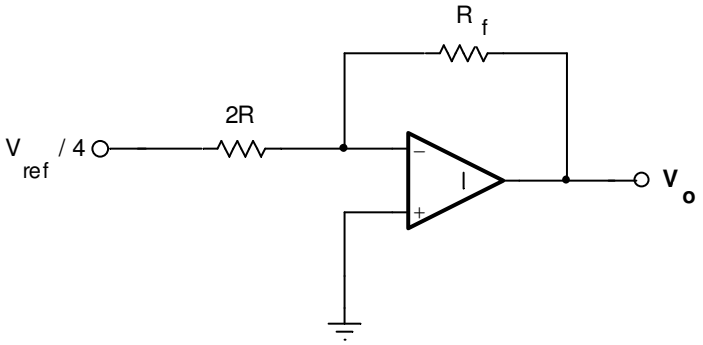


Fig.10(j) Equivalent Circuit when D_2 is connected to V_{ref} D_0,D_1,D_3 are connected to ground

Step 4: Repeating the same for D_3 High and other bits Low, we can reduce the circuit shown in fig.10(k).

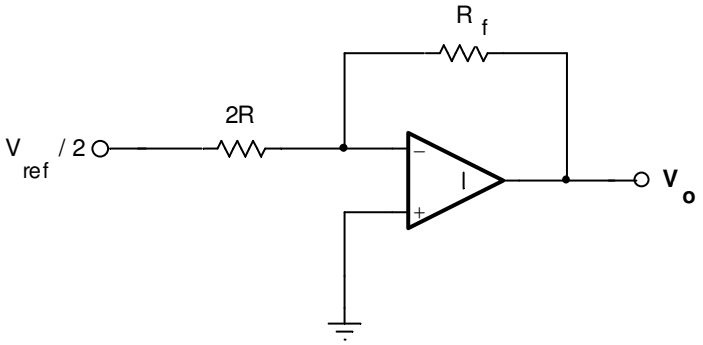


Fig.10(k) Equivalent Circuit when D_3 is connected to V_{ref} D_0,D_1,D_2 are connected to ground

Step 5: Compiling the reduced circuits of the above four steps by applying Superposition Theorem, then the network of fig 10(g),10(i),10(j),10(k) becomes the equivalent circuit shown in fig.10(m).

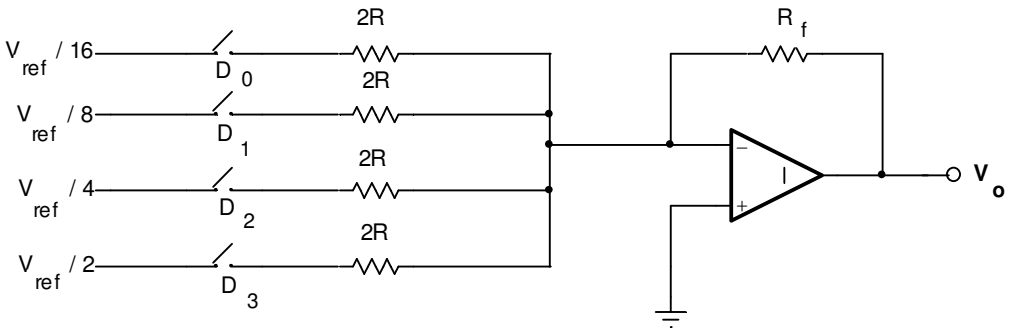


Fig.10(m) Equivalent circuit by applying Superposition Theorem for the circuits of fig.10(g),10(i),10(j),10(k)

Hence the derived equivalent circuit of the R-2R ladder network proves that the bits of the input digital word D_3, D_2, D_1, D_0 receive the applied voltages as per their binary weights and we get the corresponding analog value at V_o. Therefore,

$$V_O = \frac{V_{ref}}{2R}.R_f\left(\frac{D_3}{2}+\frac{D_2}{4}+\frac{D_1}{8}+\frac{D_0}{16}\right)$$

$$V_O = \frac{V_{ref}}{2R}.R_f\left(\frac{D_{n-1}}{2}+\frac{D_{n-2}}{4}+.....+\frac{D_0}{2^n}\right)$$

If R_f is also selected equal to 2R, then

$$V_O = V_{ref}.\left(\frac{D_{n-1}}{2}+\frac{D_{n-2}}{4}+.....+\frac{D_0}{2^n}\right)$$

V_O is independent of the numerical values of R. Thus any convenient value of R & 2R can be taken for the design of the D/A converter. The maximum output analog voltage is nearly equal to V_{ref}. The actual values of R-2R resistors influence only the maximum current handled by the op-amp. Voltage resolution of n-bit ladder network DAC is $V_{ref}/2^n$

Q.10 With relevant diagram explain the working of master-slave JK flip flop. **(9)**

Ans:

Master-Slave *J-K* FLIP-FLOP: A master-slave *J-K* FLIP-FLOP is a cascade of two *S-R* FLIP-FLOPS. One of them is known as Master and the other one is slave. Fig.11(a) shows the logic circuit. The master is positively clocked. Due to the presence of inverter, the slave is negatively clocked. This means that when clock is high, the master is active and the slave is inactive.

When the clock is low, the master is inactive and the slave is active. Fig.11(b) shows the symbol. This is a level clocked Flip-Flop. When clock is high, any changes in J and K inputs can affect S and R outputs. Therefore, J and K are kept constant during positive half of clock. When clock is low, the master is inactive and J and K inputs can be allowed to be changed. The different conditions are Set, Reset, and Toggle. The race condition is avoided because of feedback from slave to master and the slave being inactive during positive half of clock.

(i) **SET State**: Assume that Q is low (and $\overline{Q}$ is high). For high J, low K and high CLK, the Master goes to SET state giving High S and Low R. Since Slave is inactive, Q and $\overline{Q}$ do not change. When CLK becomes Low, the Slave becomes to Set state giving High Q (and low $\overline{Q}$).

(ii) **RESET State**: At the end of Set State Q is High (and $\overline{Q}$ low). Now if J is low, K is high and CLK is high, the Master Resets giving Low S and High R. Q and $\overline{Q}$ do not change because Slave is inactive. When CLK becomes Low, the Slave becomes active and resets giving Low Q (and High $\overline{Q}$).

(iii) Toggle State: If both J and K are High, the Slave copies the Master. When CLK is High, the Master toggles once. Then the Slave toggles once when CLK is low. If the Master toggles into Set state, the slave copies the Master and toggles into Set state. If the Master toggles into Reset state, the slave again copies the Master and toggles into Reset state. Since the second FLIP-FLOP simply follows the first one, it is referred to as the *slave* and the first one as the *master.* Hence, this configuration is referred to as *master-slave(M-S) FLIP*-FLOP.

Truth Table of JK Master Slave Flip-Flop in Table 11.1 shows that a Low PR and Low CLR can cause race condition. Therefore, PR and CLR are kept High when inactive. To clear, we make CLR Low and to preset we make PR Low. In both cases we change them to High when the system is to be run.

Low J and Low K produce inactive state irrespective of clock input. If K goes High, the next clock pulse resets the Flip-Flop. If J goes High by itself, the next clock pulse sets the Flip-Flop. When both J and K are High, each clock pulse produces one toggle.

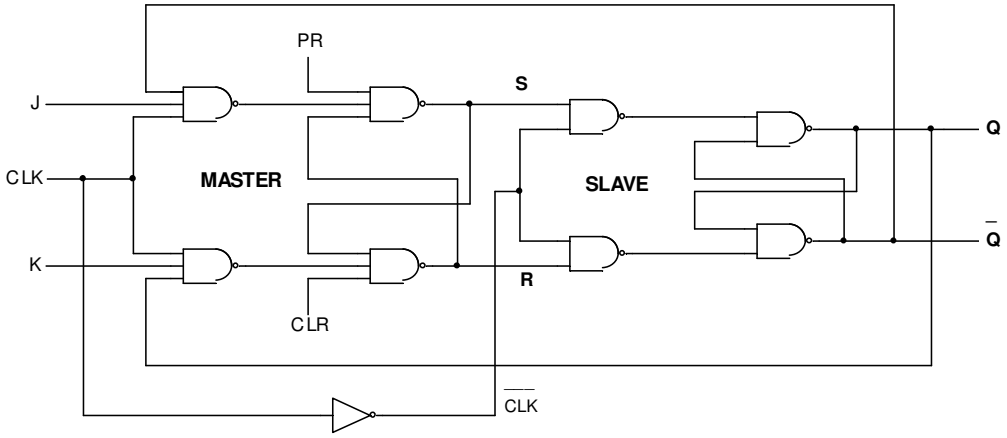


Fig.11(a) Logic Diagram of Master-Slave J-K FLIP-FLOP

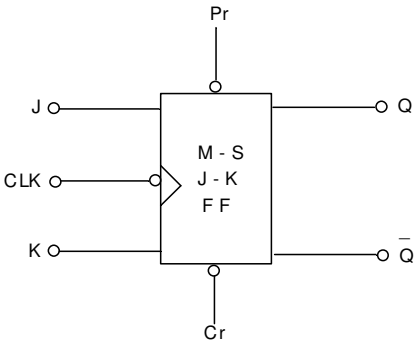


Fig.11(b) Logic Symbol of Master-Slave J-K FLIP-FLOP

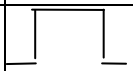
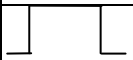
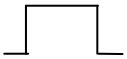
		Inputs			Output
PR	CLR	CLK	J	K	Q
0	0	X	X	X	Race Condition
0	1	X	X	X	1
1	0	X	X	X	0
1	1	X	0	0	No change
1	1		0	1	0
1	1		1	0	1
1	1		1	1	Toggle

Table 11.1 Truth Table of JK Master-Slave Flip-Flop

Q.11 Compare the memory devices RAM and ROM. **(5)**

Ans:

Comparison of Semi-conductor Memories ROM and RAM

The advantages of ROM are:

1. It is cheaper than RAM.
2. It is non-volatile. Therefore, the contents are not lost when power is switched off.
3. It is available in larger sizes than RAM. '
4. It's contents are always known and can be easily tested.
5. It does not require refreshing.
6. There is no chance of any accidental change in its contents.

The advantages of RAM are:

1. It can be updated and replaced.
2. It can serve as temporary data storage.
3. It does not require lead time (as in ROM) or programming time (as in PROM).
4. It does not require any programming equipment

Q.12 State and prove Demorgan's laws. **(5)**

Ans:

De Morgan's Theorems:

(i) Statement of First Theorem: $\overline{A+B} = \overline{A} \cdot \overline{B}$

Proof: The two sides of the equation $\overline{A+B} = \overline{A}\overline{B}$ is represented by logic diagrams shown in fig.3 (a) & 3(b)

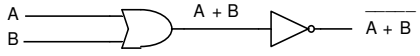


Fig.3(a) Logic diagram for $\overline{A+B}$

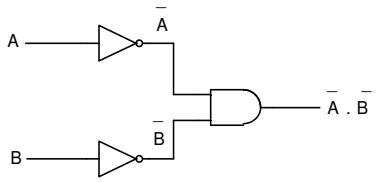


Fig.3(b) Logic diagram for $\overline{A}\,\overline{B}$

The equality of the logic diagrams of fig.3 (a) & 3(b) is proved by the truth table shown in table 2(c)

Inputs		Intermediate Values			Outputs	
A	B	A + B	$\overline{A}$	$\overline{B}$	$\overline{A+B}$	$\overline{A}\ \overline{B}$
0	0	0	1	1	1	1
0	1	1	1	0	0	0
1	0	1	0	1	0	0
1	1	1	0	0	0	0

Table 2(c)

(ii) Statement of second theorem: $\overline{AB} = \overline{A} + \overline{B}$

Proof: The two sides of the equation $\overline{AB} = \overline{A} + \overline{B}$ is represented by the logic diagrams shown in fig.3(c) & 3(d).

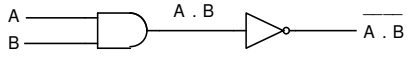


Fig.3(c) Logic diagram for $\overline{AB}$

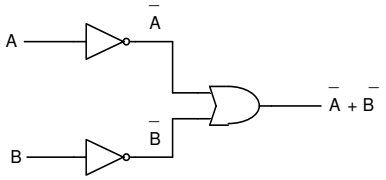


Fig.3(d) Logic diagram for $\overline{A} + \overline{B}$

The equality of the logic diagrams of fig.3(c) & 3(d) is proved by the truth table shown in table 2(d)

Inputs		Intermediate Values			Outputs	
A	B	A .B	$\overline{A}$	$\overline{B}$	$\overline{A.B}$	$\overline{A}$ + $\overline{B}$
0	0	0	1	1	1	1
0	1	0	1	0	1	1
1	0	0	0	1	1	1
1	1	1	0	0	0	0

Table 2(d)

Q.13 Discuss in detail, the working of Full Adder logic circuit and extend your discussion to explain a binary adder, which can be used to add two binary numbers. **(14)**

Ans:

Full-Adder: A half-adder has only two inputs and there is no provision to add a carry from the lower order bits when multibit addition is performed. For this purpose, a third input terminal is added and this circuit is used to add A_n, B_n, and C_{n-1}, where A_n and B_n are the nth order bits of the numbers, A and B respectively and C_{n-1} is the carry generated from the addition of (n-1)th order bits. This circuit is referred to as full-adder and its truth table is given in Table 5.1

Inputs			Outputs	
A_n	B_n	C_{n-1}	S_n	C_n
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Table 5.1 Truth Table of a Full-Adder

The K-maps for the outputs S_n and C_n are given in Fig.5(a) and Fig.5(b) respectively and the minimized expressions are given by

$S_n = \overline{A_n}\, B_n \overline{C_{n-1}} + \overline{A_n}\ \overline{B_n}\ C_{n-1} + A_n\ \overline{B_n}\ \overline{C_{n-1}} + A_n\, B_n\, C_{n-1}$

$C_n = A_n\, B_n + B_n\, C_{n-1} + A_n\, C_{n-1}$

	$\overline{A_n}\ \overline{B_n}$	$\overline{A_n}\, B_n$	$A_n\, B_n$	$A_n \overline{B_n}$
$\overline{C_{n-1}}$		1		1
C_{n-1}	1		1	

Fig.5(a) K-map for S_n

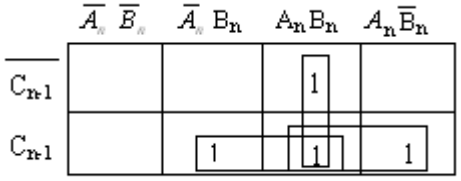


Fig.5(b) K-map for C_n

The logic diagrams for the Sn and Cn are shown in fig.5(c) & fig.5(d).

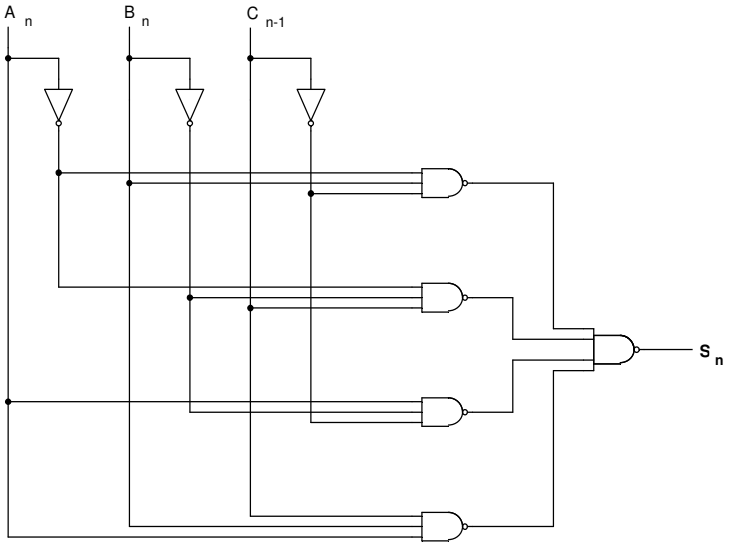


Fig.5(c) NAND-NAND Realization of S_n

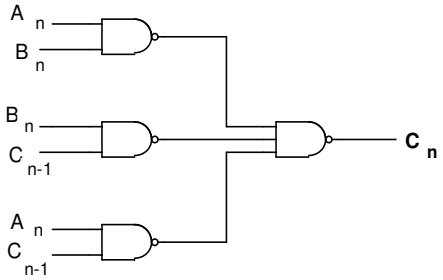


Fig.5(d) NAND-NAND Realization of C_n

Binary Adder: The full adder forms the sum of two bits and a previous carry. Two binary numbers of n bits each can be added by means of Binary Adder. If A = 1011 and B = 0011, whose sum is S = 1110. When pair of bits is added through a full-adder, the circuit produces a carry to be used with the pair of bits one significant position higher. This is shown in Table 5.2 The bits are added with full-adders, starting from the Least Significant Position (subscript 1), to form the sum bit and carry bit. The input carry C1 in the Least Significant position must be 0. The value of C_{i+1} in a given significant position is the output carry of the full-adder. This value is transferred into the input carry of the full-adder that adds the bits one higher significant position to the left. The sum bits are thus generated starting form the rightmost position and are available as soon as the corresponding previous carry bit is generated

Subscript i	4	3	2	1		Full-Adder
Input Carry	0	1	1	0	C_i	Z
Augend	1	0	1	1	A_i	X
Addend	0	0	1	1	B_i	Y
Sum	1	1	1	0	S_i	S
Output Carry	0	0	1	1	C_{i+1}	C

Table 5.2 Truth Table for Binary Adder

A Binary Parallel Adder is a digital function that produces the arithmetic sum of two binary numbers in parallel. It consists of full-adders connected in cascade, with the output carry from one full-adder connected to the input carry of the next full-adder. Fig.5(e) shows a 4-bit Binary Parallel Adder. The augend bits of A and the addend bits of the B are designated by subscript numbers from right to left, with subscript 1 denoting the low-order bit. The carries are connected in a chain through the full-adders. The input carry to the adder is C_1 and the output carry is C_5. The S outputs generate the required sum bits.

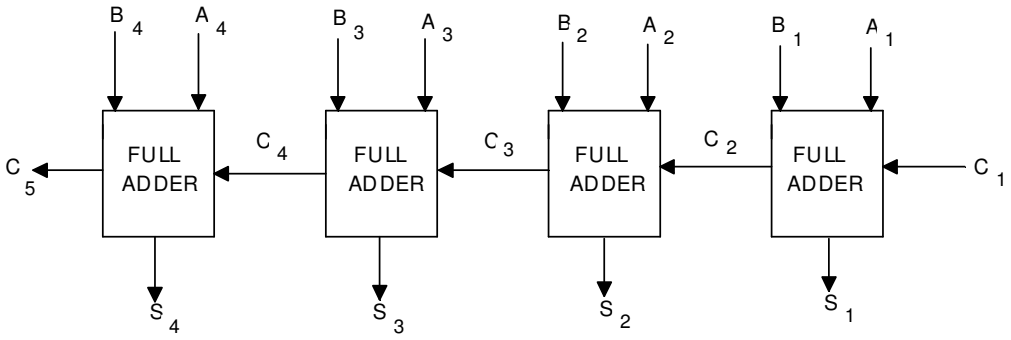


Fig.5(e) 4-bit Binary Parallel Adder using Full-Adders

Q.14 What is a decoder? Draw the logic circuit of a 3 line to 8 line decoder and explain its working. **(7)**

Ans:

Decoder: A Decoder is a combinational logic circuit that converts Binary words into alphanumeric characters. Thus the inputs to a decoder are the bits 1, 0 and their combinations. The output is the corresponding decimal number. It converts binary information from *n* input lines to a maximum of 2^n unique output lines. If the *n-bit* decoded information has unused or don't-care combinations, the decoder output will have less than 2^n outputs.

Working: The logic circuit of a 3 line to 8 line decoder is shown in fig.6 (a). The three inputs (x, y, z) are decoded into eight outputs (from D_0 to D_7), each output representing one of the minterms of the 3-input variables. The three inverters provide the complement of the inputs, and each one of the eight AND gates generate one of the minterms. A particular application of this decoder is a binary-to-octal conversion. The input variables may represent a binary number, and the outputs will then represent the eight digits in the octal number system. However, a 3-to-8 line decoder can be used for decoding any 3-bit code to provide eight outputs, one for each element of the code.

The operation of the decoder may be verified from its input-output relationships shown in Table 6.1.The table shows that the output variables are mutually exclusive because only one output can be equal to 1 at any one time. Consider the case when X=0, Y=0 and Z=0, the output line D_0 (X', Y', Z') is equal to 1 represents the minterm equivalent of the binary number presently available in the input lines.

Inputs			Outputs							
X	Y	Z	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	**1**	0	0	0	0	0	0	0
0	0	1	0	**1**	0	0	0	0	0	0
0	1	0	0	0	**1**	0	0	0	0	0
0	1	1	0	0	0	**1**	0	0	0	0
1	0	0	0	0	0	0	**1**	0	0	0
1	0	1	0	0	0	0	0	**1**	0	0
1	1	0	0	0	0	0	0	0	**1**	0
1	1	1	0	0	0	0	0	0	0	**1**

Table 6.1 Truth Table of 3-to-8 line Decoder

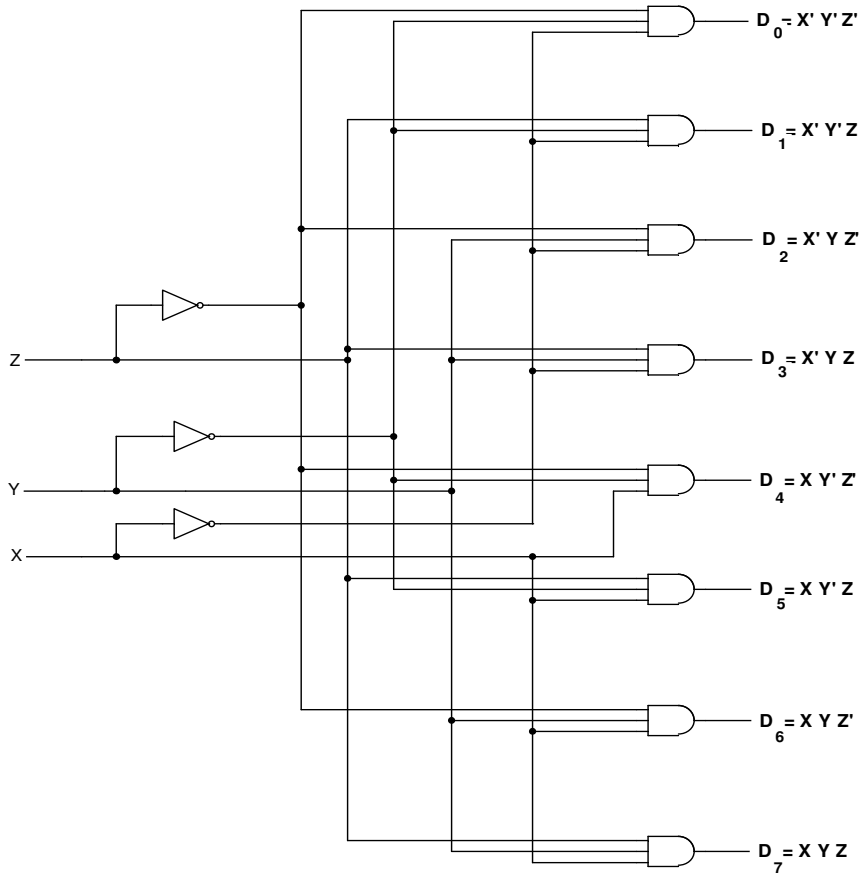


Fig. 6(a) Logic Circuit of 3-to-8 line Decoder

Q.15 What is an encoder? Draw the logic circuit of Decimal to BCD encoder and explain its working. **(7)**

Ans:

Encoder: An Encoder is a combinational logic circuit which converts Alphanumeric characters into Binary codes. It has 2n (or less) input lines and n output lines. An Encoder may be Decimal to Binary, Hexadecimal to Binary, Octal to BCD etc.

Decimal to BCD Encoder: This encoder has 10 inputs (for decimal numbers 0 to p) and 4 outputs for the BCD number. Thus it is a 10 line to 4 line encoder. Table 6(a) lists the decimal digits and the equivalent BCD numbers. From the table, we can find the relationship between decimal digit and BCD bit. MSB of BCD bit is Y_3. For decimal digits 8 or 9, $Y_3 = 1$. Thus we can write OR expression for Y_3 bit as

$$Y_3 = 8 + 9$$

Similarly , Bit Y_2 is 1 for decimal digits 4,5,6 and 7. Thus we can write OR expression

$$Y_2 = 4 + 5 + 6 + 7$$
$$Y_1 = 2 + 3 + 6 + 7$$
$$Y_o = 1 + 3 + 5 + 7 + 9$$

The logic circuit for the expressions (Y0, Y1, Y2, Y3) is shown in fig. 6(b). When a High appears on any of input lines the corresponding OR gates give the BCD output. For *e.g.,* if decimal input is 8, High appears only on output 3 (and LOW on Y_0, Y_1 , Y_2), thus giving the BCD code for decimal 8 as 1000. Similarly, if decimal input is 7, then High appears on outputs Y_0, Y_1, Y_2 (and LOW on *Y3),* thus giving BCD output as 0111.

Decimal Digit	**BCD Code**			
	Y_3	**Y_2**	**Y_1**	**Y_0**
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

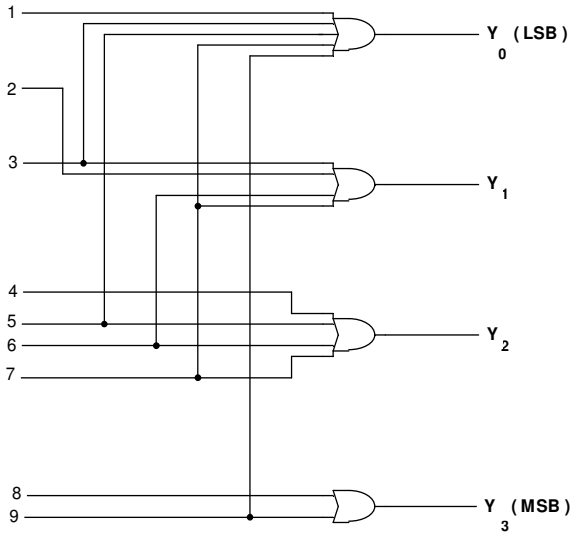


Fig.6(b) Logic diagram for Decimal to BCD Encoder

Q.16 What is a flip-flop? What is the difference between a latch and a flip-flop? List out the application of flip-flop. **(4)**

Ans:

Flip-Flop: A flip-flop is a basic memory element used to store one bit of information. Both Flip-flops and latches are bistable logic circuits and can reside in any of the two stable states due to a feedback arrangement. The main difference between them is in the method used for changing the state.

Applications of Flop-Flops:

(1) Bounce elimination switch
(2) Parallel Data Storage in Registers
(3) Transfer of Data from one bit to another.
(4) Counters
(5) Frequency Division

Q.17 Draw the circuit diagram of a Master-slave J-K flip-flop using NAND gates. What is race around condition? How is it eliminated in a Master-slave J-K flip-flop. **(10)**

Ans:

Logic Diagram of Master-Slave J-K Flip-Flop using NAND Gates: Fig.7(a) shows the logic diagram of Master-Slave J-K Flip-Flop using NAND gates.

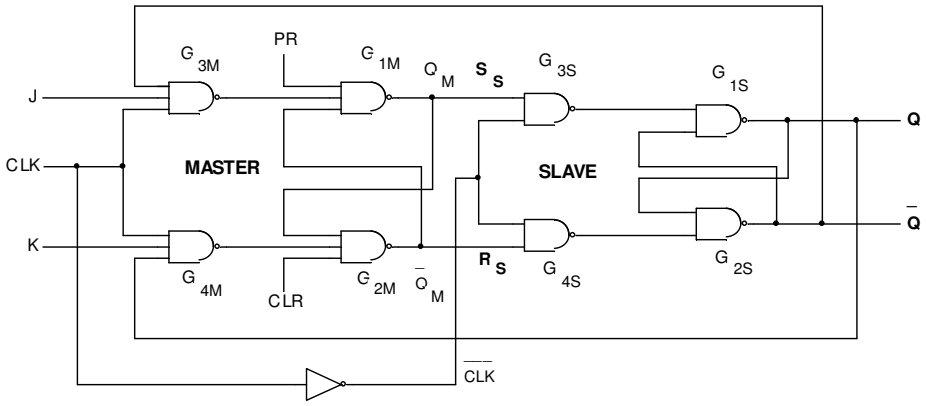


Fig.7(a) Logic Diagram of Master-Slave J-K FLIP-FLOP

The Race-around Condition: The difficulty of both inputs 1 (S = *R* = I) being not allowed in an *S-R* Flip-Flop is eliminated in a *J-K Flip-Flop* by using the feedback connection from outputs to the inputs of the gates. In R-S Flip-Flop, the inputs do not change during the clock pulse *(CK* = 1), which is not true in J-K Flip-Flop because of the feedback connections. Consider that the inputs are *J* = *K* = 1 and *Q* = 0 and a pulse as shown in Fig. 7(b) is applied at the clock input. After a time interval Δt equal to the propagation delay through two NAND gates in series, the output will change to *Q* = 1.

Now we have *J* = *K* = 1 and *Q* = 1 and after another time interval of Δt the output will change back to *Q* = O. Hence, for the duration t_p of the clock pulse, the output will oscillate back and forth between 0 and 1. At the end of the clock pulse, the value of *Q* is uncertain. This situation is referred to as the race-around condition.The race-around condition can be avoided if $t_p < \Delta t < T$. However, it may be difficult to satisfy this inequality because of very small propagation delays in ICs. A more practical method of overcoming this difficulty is the use of the master-slave *(M-S)* configuration.

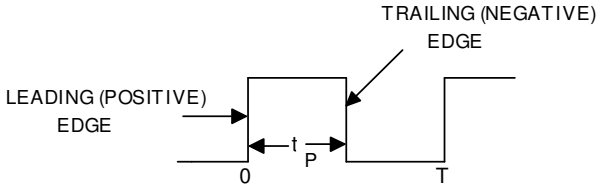


Fig.7 (b) a Clock Pulse

A master-slave J-K Flip-Flop is a cascade of two S-R Flip-Flops with feedback from the outputs of the second to the inputs to the first as illustrated in Fig.7(a). Positive clock pulses are applied to the first Flip-Flop and the clock pulses are inverted before these are applied to the second Flip-Flop. When CK=1, the first Flip-Flop is enabled and the outputs Q_M and $\overline{Q_M}$ respond to their inputs J and K according the Table 7.1. At this time, the second Flip-Flop is inhibited because its clock is LOW ($\overline{CK}$ = 0). When CK goes LOW ($\overline{CK}$ = 1), the first Flip-Flop is inhibited and the second Flip-Flop is enabled, because now its clock is HIGH ($\overline{CK}$ = 1). Therefore, the outputs Q and $\overline{Q}$ Follow the outputs Q_M and $\overline{Q_M}$ respective (second and third rows of Table 7.1). Since the second Flip-Flop simply follows the first one, it is referred to as the Slave and the first one as the Master. Hence, this

configuration is referred to as Master-Slave Flip-Flop. In this circuit, the inputs to the gates G_{3M} and G_{4M} do not change during the clock pulse, therefore the Race-around condition does not exist. The state of the Master-Slave Flip-Flop changes at the negative transition (trailing end).

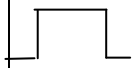
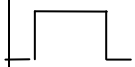
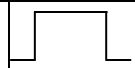
		Inputs			**Output**
PR	CLR	CLK	J	K	Q
0	0	X	X	X	Race Condition
0	1	X	X	X	1
1	0	X	X	X	0
1	1	X	0	0	No change
1	1		0	1	0
1	1		1	0	1
1	1		1	1	Toggle

Table 7.1 Truth Table of JK Master-Slave Flip-Flop

Q.18 What is a demultiplexer? Discuss the differences between a demultiplexer and a decoder. **(4)**

Ans:

Demultiplexer: It is a logic circuit that accepts one data input and distributes it over several outputs. A demultiplexer has one data input, m select lines, and n output lines, whereas a decoder does not have the data input but the select lines are used as input lines.

Q.19 What is a shift register? Can a shift register be used as a counter? If yes, explain how? **(4)**

Ans:

Shift Register: A register in which data gets shifted towards left or right when clock pulses are applied is known as a Shift Register. A shift register can be used as a counter. If the output of a shift register is fed back to serial input, then the shift register can be used as a Ring Counter.

Q.20 What are synchronous counters? Design a Mod-5 synchronous counter using J-K Flip-Flops. **(10)**

Ans:

Synchronous Counters: The term synchronous means that all flip-flops are clocked simultaneously. The clock pulses drive the clock input of all the flip-flops together so that there is no propagation delay.

Mod-5 Counter Synchronous Counter: The Mod-5 Synchronous Counter have five counter states. The counter design table for this counter lists the three flip-flop and their states (0 to 5 states), as shown in table 9(a), the six inputs required for the three flip-flops. The flip-flop inputs required to step up the counter from the present to the next state have been worked out with the help of the excitation table shown in the table.

Input pulse Count	**Counter States**			**Flip-Flop Inputs**					
	A	B	C	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	1	X	0	X	0	X
1	1	0	0	X	1	1	X	0	X
2	0	1	0	1	X	X	0	0	X
3	1	1	0	X	1	X	1	1	X
4	0	0	1	0	X	0	X	X	1
5(0)	0	0	0						

Table 9(a) counter Design Table for Mod-5 Counter

A flip-flop*:* The initial state is 0. It changes to 1 after the clock pulse. Therefore J_A should be 1 and K_A may be 0 or 1 (that is X).

B flip-flop: The initial state is 0 and it remains unchanged after the clock pulse. Therefore J_B should be 0 and K_B may be 0 or 1 (that is X)

C flip-flop: The state remains unchanged. Therefore *Jc* should be 0 and K_C should be X.The flip-flop input values are entered in Karnaugh maps shown in Table 9(b) [(i) (ii) (iii) (iv) (v) and (vi)] and a boolean expression is found for the inputs to the three flip-flops and then each expression is simplified. *As* all the counter states have not been utilized, X's (don't) are entered to denote un-utilized states. The simplified expressions for each input shown under each map. Finally, these minimal expressions for the flip-flop inputs are used to draw a logic diagram for the counter, which is shown in fig.9 (b).

	$\overline{B}\,\overline{C}$	$\overline{B}C$	BC	$B\overline{C}$
$\overline{A}$	1	0	X	1
A	X	X	X	X

(i) Map for J_A

$J_A = \overline{C}$

	$\overline{B}\,\overline{C}$	$\overline{B}C$	BC	$B\overline{C}$
$\overline{A}$	X	X	X	X
A	1	X	X	1

(ii) Map for K_A

$K_A = 1$

	$\overline{B}\,\overline{C}$	$\overline{B}C$	BC	$B\overline{C}$
$\overline{A}$	0	0	X	X
A	1	X	X	X

(iii) Map for J_B

$J_B = A$

	$\overline{B}\,\overline{C}$	$\overline{B}C$	BC	$B\overline{C}$
$\overline{A}$	X	X	X	0
A	X	X	X	1

(iv) Map for K_B

$K_B = A$

	$\overline{B}\,\overline{C}$	$\overline{B}C$	BC	$B\overline{C}$
$\overline{A}$	0	X	X	0
A	0	X	X	1

(v) Map for J_C

$J_C = AB$

	$\overline{B}\,\overline{C}$	$\overline{B}C$	BC	$B\overline{C}$
$\overline{A}$	X	1	X	X
A	X	1	X	X

(vi) Map for K_C

$K_C = 1$

Table 9(b) Karnaugh Maps for MOD-5 Synchronous Counter

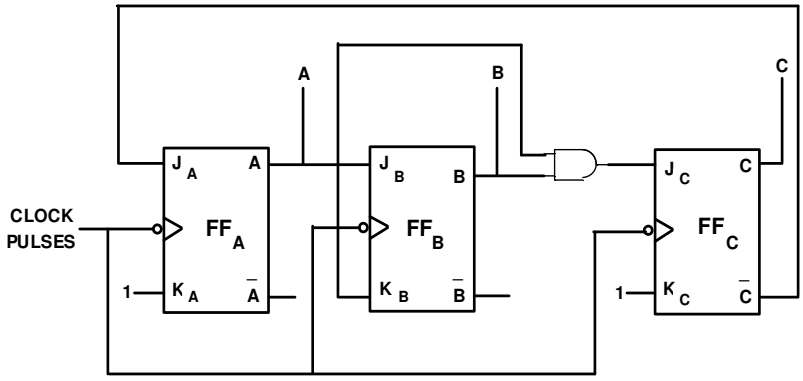


Fig.9 (b) Logic Diagram of MOD-5 Synchronous Counter

Q.21 With the help of a neat diagram, explain the working of a successive approximation A/D converter. **(14)**

Ans:

Successive Approximation ADC:

This is the most widely used A/D converter. As the name suggests, the digital output tends towards analog input through successive approximations. In Successive Approximation ADC, the comparison with the input analog voltage is done in descending order starting from maximum voltage. Fig.10 (a) shows the block diagram of SA A/D converter. The main components are Op-amp Comparator, Control Logic, SA register and D/A converter. It uses Digital to Analog converter as a feedback element. The control logic is the most important part of Successive Approximation Converter, as this decides the next step to be taken. The ring counter provides timing waveform to control the operation of the converter. The Digital to Analog Converter unit, n bit register and ring counter are all reset by the first pulse from the ring counter. The ring counter containing a single one sets the MSB of the Digital to Analog Converter to 1 and the other to 0

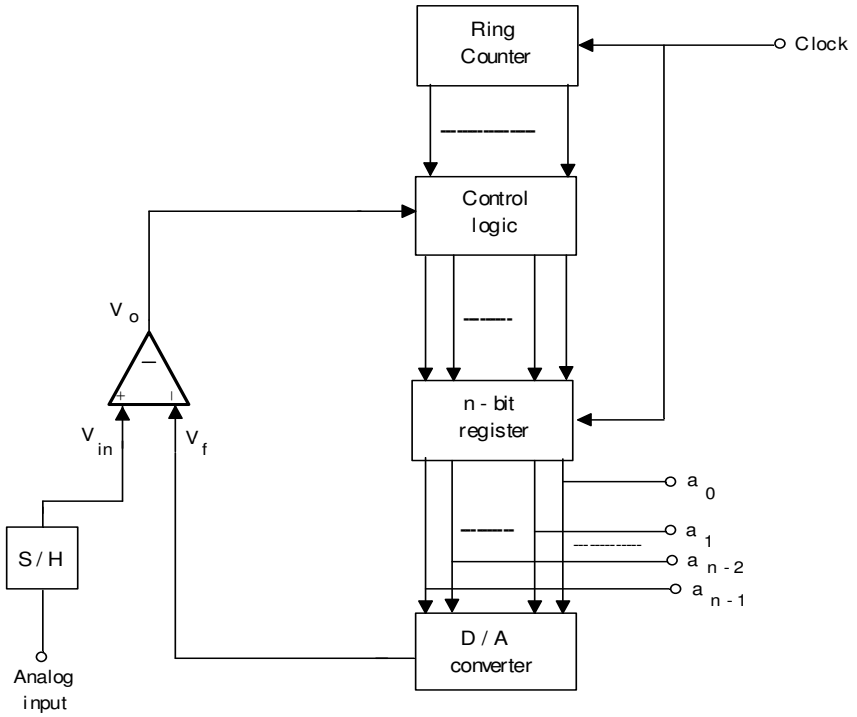


Fig.10 (a) Block Diagram of Successive Approximation A/D Converter

The basic operating principle of Successive Approximation Converter is that the voltage ouput of DAC corresponding to MSB is compared by the comparator with the input voltage and if the voltage is less, the bit 1 is retained. If the voltage is more, it is reset to 0 and counter moves to next position. Similar decisions are made at each bit position until the nearest value is reached.

Assume that the MSB of a unipolar 6 bit converter produces 10 V output and we have to measure an analog output voltage of 8.2 V. Each bit divides the voltage by 2 so that the voltages for the 6 bits from MSB downwards is

Bits	5	4	3	2	1	0
Voltage	10	5	2.5	1.25	0.625	0.3125
	MSB					LSB

The operation of SA A/D converter is shown in Table No.10(a). Let the analog input be 8.2 V. The SA register is first set to zero. Then 10 is placed in MSB. This is fed to D/A converter whose output goes to comparator. Since the analog input (8.2 V) is greater than D/A output (i.e.,10 V), the MSB is set to one. Then 1 is placed in bit next to MSB (i.e., 1 is placed in second position). Now the output of D/A is 5 V. Since analog input is less than 5 V, it is reset to 0. Next 0 is placed in third position. Now the D/A output is (5+2.5=7.5V) which is less than analog input. Therefore, this 0 bit is retained and 0 is placed in the next bit (i.e., fourth position). Now the D/A output is (7.5+1.25=8.75), which is more than analog input. Therefore, the 1 bit is placed in fifth position. Now the D/A output is (8.75+0.625=8.125) which is less than analog input, it is reset to 0. Now 0 is placed in LSB producing a D/A output of (8.125+0.3125=8.4375) which is more than analog input. Therefore, LSB is set to one.

The various steps and voltages are tabulated in Table No.10 (a).

Step	Register	DAC Output	Comparator decision w.r.t 8.2V.
Start	100000	10	High
2	010000	5	Low
3	011000	5+2.5=7.5	Low
4	011100	7.5+1.25=8.75	High
5	011010	7.5+0.625=8.125	Low
6	011011	8.125+0.3125=8.4375	High

Table 10(a)

The D/A converter waveform is shown in fig.10 (b)

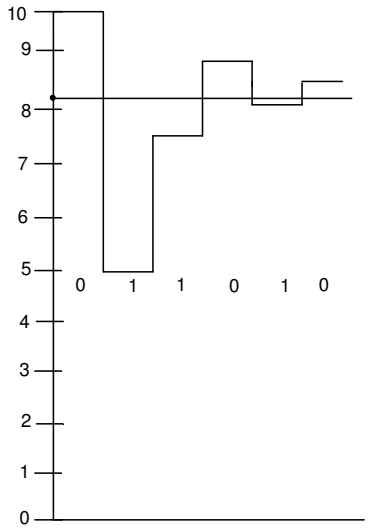


Fig.10 (b) Output Waveform of D/A Converter

Features:

(i) It is one of the most widely used ADC

(ii) Its conversion time is very next only to Flash or Parallel ADC

(iii) SACs have fixed value of conversion time that is not dependent on the value of analog input voltage.

(iv) Data can be taken out either in serial or in parallel.

(v) During the period of comparison the input analog voltage should be held constant and so the input to comparator is through a Sample Hold circuit.

Q.22 Difference between static and dynamic RAM. Draw the circuits of one cell of each and explain its working. **(10)**

Ans:

Differentiation between Static RAM and Dynamic RAM:

Static RAMs store ones and zeros using conventional FLIP-FLOPs. whereas, the memory cells of dynamic RAMs are basically charge storage capacitors with driver transistors.The presence or absence of charge in a capacitor is interpreted as Logic1 or 0.

Static RAMs do not require refreshing because there is no problem of charge leaking-off in FLIP-FLOPs whereas Dynamic RAMs require periodic charge refreshing to maintain data storage because the charge stored on capacitors leak-off with time.

Static RAMs are slower but easier to drive than dynamic memories, which generally require clock signals in addition to extra power supplies whereas dynamic circuits usually require externally generated clock voltages,

Advantages of Static RAMs over Dynamic RAMs:

(i)Higher speed of operation (faster)i.e, lower access –time.

(ii)Does not require refreshing.

Advantages of Static RAMs over Dynamic RAMs:

(i)Higher number of bits storage on a given silicon chip area. i.e, Higher packaging density.

(ii)Lower power consumption.

Static RAM Cell: A RAM memory cell consisting of two cross-coupled MOS inverters is shown in Fig.11 (a). It is addressed by setting A_X and A_Y to 1. When A_X = 1, the cell is connects to the data and $\overline{data}$ line. When A_Y = 1, T_7 and T_8 are ON.
To write into the cell, set W = 1, T_9 becomes ON. If data input is 1, the voltage at node D will correspond to level 1 making T_3 ON and level at $\overline{D}$ will be 0. On the other hand, if the data input is at logic 0, then T_3 will be OFF and $\overline{D}$ would be at 1. To read the state of the FLIP-FLOP, we set R = 1.This connects the data output to $\overline{D}$. Thus, the complement of the data level written into the cell is read at $\overline{data}$ output.

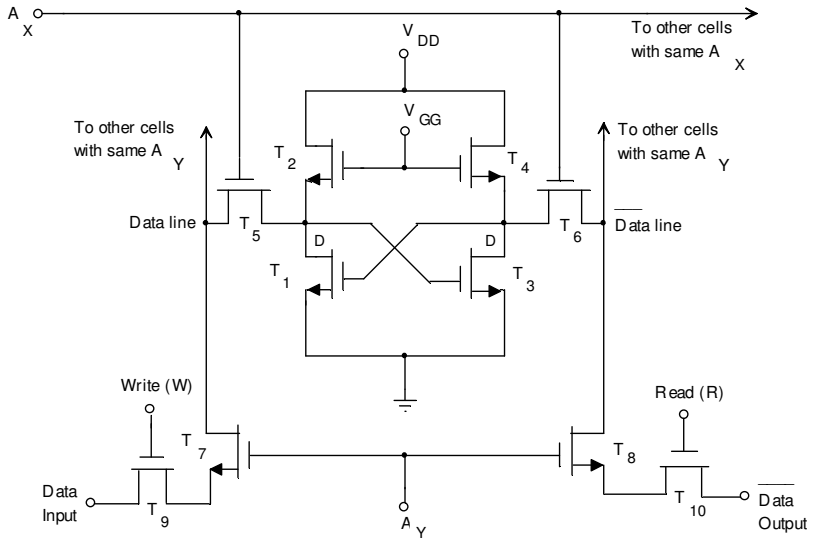


Fig.11(a) Logic Diagram of a Static MOS RAM Cell

Dynamic RAM Cell: A dynamic cell uses four transistors in place of the six used in a static cell. This reduces the silicon chip area and results in saving of power. The circuit of a 4-transistor dynamic MOS RAM cell is shown in Fig.11 (b).The state of the cell is stored on the stray capacitances C_1 and C_2, whose presence is essential. The cell is addressed by making $A_X = A_Y$ =1. In one state of the cell, the voltage across C_1 is large and T_1 is ON. Correspondingly, C_2 has zero voltage and T_2 is OFF. In the other state, the voltages on C_1 and C_2 and the conducting states of T_1 and T_2 are reversed. For writing into the cell, we set W = 1 and for reading from the cell we set R = 1.It is necessary to refresh the cell periodically, otherwise the charge stored on the capacitors leak off. The refreshing operation is accomplished by allowing brief access from the supply voltage V_{DD} to the cell. This is done by making A_X = 1 and the refresh terminal voltage corresponding to 1 level. This makes T_3, T_4, T_9, and T_{10} ON. Suppose initially T_1 is ON, T_2 is OFF. The voltage across C_1 is large and across C_2 it is zero volt. During the refresh interval, V_{DD} is applied through T_{10} and T_4 to C_1, since T_2 is OFF. Therefore, current from V_{DD} will flow through C_1, allowing C_1 to replenish any charge lost due to leakage. Since T1 is ON, hence C_2 will not charge as rapidly as C_1. Similarly V_{DD} is applied to C_2, which is in parallel to T_1 when T_1 is OFF and T_2 is ON.

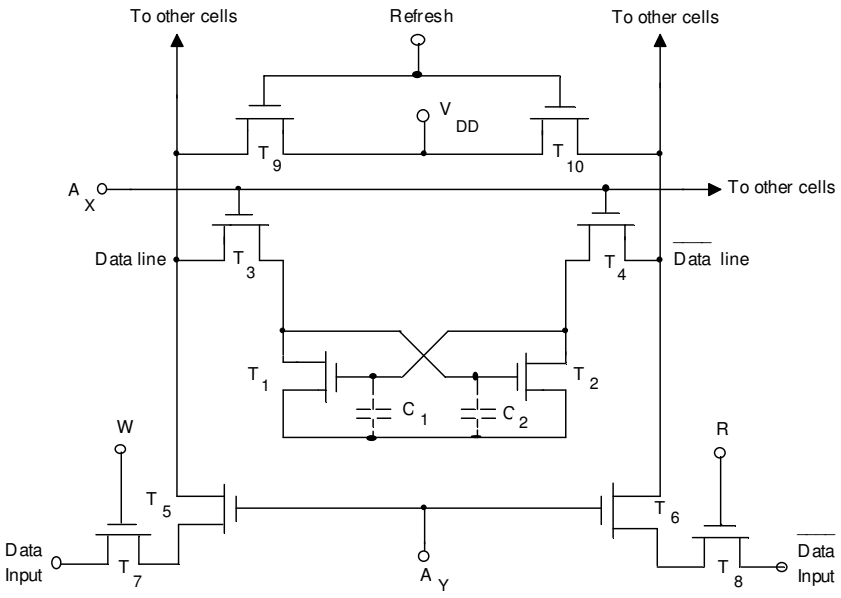


Fig.11(b) Logic Diagram of Dynamic MOS RAM Cell

Q.23 Distinguish between ROM, PROM, EPROM, EEPROM. **(4)**

Ans:

ROM: Read Only Memory is a Permanent Memory. In Permanent ROM, the data is permanently stored and cannot be changed. It can only be read from the memory. There cannot be a write operation because the specified data is programmed into the device by the manufacturer or the user. ROM is a Non-volatile memory. Some examples of ROM are conversion tables, pre-programmed instructions etc.

PROM: Programmable Read Only Memory allows user to store the data. An instrument PROM programmer is used to store the required data. The process used is opening the links at bit locations using high current (this process is called burning in). Once this process has been done, the data is permanently stored and no change is possible.

EPROM: EPROM means Erasable PROM. It can be reprogrammed by first erasing the existing program. EPROM uses N-MOSFET array with isolated gate structure. The isolated transistor gate has no electrical connection and can store an electrical charge indefinitely. The data bits in this memory array are represented by presence or absence of charge. Erasure is achieved by removing the gate charge. EPROM can be UV EPROM or EEPROM.

UV EPROM means Ultra Violet Erasable PROM. Erasure is achieved by using ultra violet light. The light passes through a window in the IC package to the chip where there are stored charges. Thus the stored contents are erased.

EEPROM: EEPROM means Electrically Erasable PROM. In this memory device, the erasure and programming is done by electrical pulses.

Q.24 What is a universal gate? Give examples. Realize the basic gates with any one universal gate. **(8)**

Ans:

Universal Gates: NAND and NOR are known as Universal gates The AND, OR, NOT gates can be realized using any of these two gates. The entire logic system can be implemented by using any of these two gates. These gates are easier to realize and consume less power than other gates.

Realizations of NOT, AND and OR gates using NAND gates

NOT GATE: Fig. 3(a) shows the realization of Inverter (NOT) gate using NAND gate. Both the inputs to the NAND gates are tied together so that the gate works as an inverter (NOT) gate.

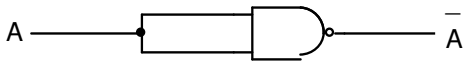
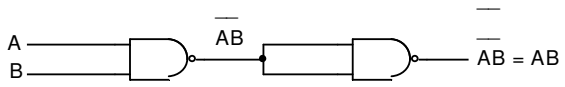


Fig. 3(a) Realization of Inverter (NOT) gate using NAND gate

AND GATE: Fig. 3(b) shows the realization of AND gate using two NAND gates. It has combination of two NAND gates gives AND operation. The first NAND gate has two inputs A and B. The two inputs to the second NAND gate are tied together and the output $\overline{AB}$ of the first gate is fed to this common terminal. The output is AB thus giving AND operation.



.

Fig. 3(b) Realization of AND gate using NAND gates

OR GATE: Fig. 3(c) shows the realization of OR gate using NAND gates. The two inputs from each of the first two NAND gates are tied together and fed by A and B as shown in the figure. The outputs are $\overline{A}$ and $\overline{B}$. They are fed to as inputs to third NAND gate. The final output is A OR B thus giving OR operation.

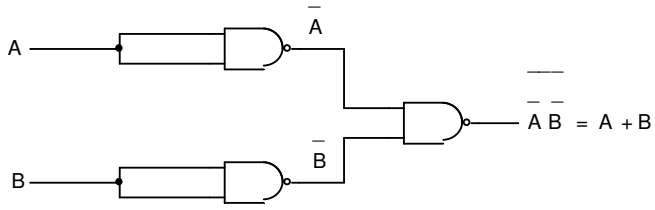


Fig. 3(c) Realization of OR gate using NAND gates

Q.25 Give the circuit of a TTL NAND gate and explain its operation in brief. **(6)**

Ans:

Operation of TTL NAND Gate: Fig.3(d) shows a TTL NAND gate with a totem pole output. The totem pole output means that transistor T_4 sits atop T_3 so as to give low output impedance. The low output impedance implies a short time constant RC so that the

output can change quickly from one state to another. T_1 is a multiple emitter transistor. This transistor can be thought of as a combination of many transistors with a common base and collector. Multiple emitter transistors with about 60 emitters have been developed. In the figure, T_1 has 3 emitters so that there can be three inputs A, B, C. The transistor T_2 acts as a phase splitter because the emitter voltage is out of phase with the collector voltage. The transistors T_3 and T_4 from the totem pole output, The capacitance CL represents the stray capacitance etc. The diode D is added to ensure that T_4 is cut off when output is low. The voltage drop of diode D keeps the base-emitter junction of T_4 reverse biased so that only T_3 conducts when output is low. The operation can be explained briefly by three conditions as given below:

Condition 1: At least one input is low (i.e., 0). Transistor T_1 saturates. Therefore, the base voltage of T_2 is almost zero. T_2 is cut off and forces T_3 to cut off. T_4 acts like an emitter follower and couples a high voltage to load. Output is high (i.e. Y=1).

Condition 2: All inputs are high. The emitter base junctions of T_1 are reverse biased. The collector base junction of T_1 is forward biased. Thus, T_1 is in reverse active mode. The collector current of T1 flows in reverse direction. Since this current is flowing into the base of T_2, the transistors T_2 and T_3 saturate and output Y is low.

Condition 3: The circuit is operating under II when one of the input becomes low. The corresponding emitter base junction of T_1 starts conducting and its base voltage drops to a low value. Therefore, T_1 is in forward active mode. The high collector current of T_1 removes the stored charge in T_2 and T_3 and therefore, T_2 and T_3 go to cutoff and T_1 saturates and output Y returns to high.

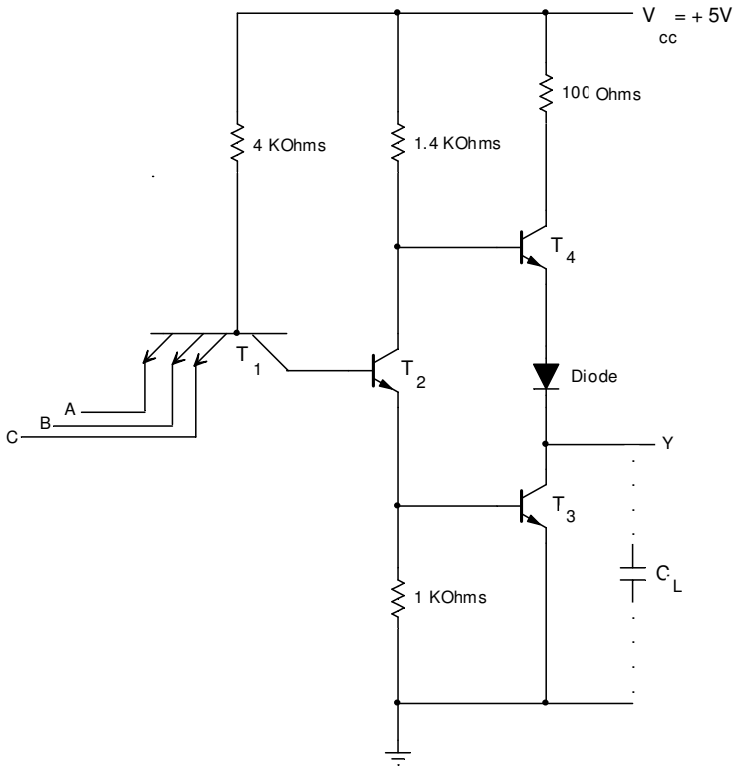


Fig.3(d) Logic Diagram of TTL NAND Gate with Totem Pole Output

Q.26 With the help of a truth table explain the working of a half subtractor. Draw the logic diagram using gates. **(8)**

Ans:

Half Subtractor: A logic circuit for the subtraction of B (subtrahend) from A (minuend) where A and B are 1-bit numbers is referred to as a Half-Subtractor**.** The truth table for half subtractor is given in Table No.5.1. Here A and B are the two inputs and D_i (difference) and B_o (borrow) are the two outputs. If B is larger than A (e.g., A=0 and B=1), a borrow is necessary,

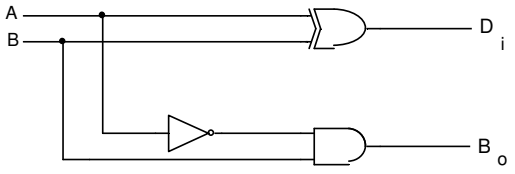
Inputs		**Outputs**	
A	**B**	**D_i(Difference)**	**B_o(Borrow)**
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Table 5.1

From the Truth Table, the logical expressions for D_i and B_o are obtained as

$$D_i = \overline{A}B + A\overline{B}$$

$$B_O = \overline{A}B$$



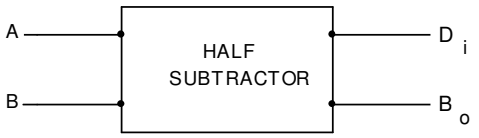


Fig.5(a) Logic Diagram of Half Subtractor **Fig.5(b) Block Diagram of Half Subtractor**

In Table 5.1, input variable *B* is subtracted from A to give output D_i(difference). If B is larger than A *(e.g., A* = 0 and *B* = 1), a borrow is necessary. In the Truth Table, inputs are *A* and *B,* Outputs are D_i (difference) *and* B_O *(borrow).* Hence, the Boolean expressions for the half subtractor from the Truth Table can be written as

$$D_i = A \oplus B \text{ ---------(1)}$$

$$B_O = \overline{A}B \text{ --------(2)}$$

By combining Boolean Expressions (1) & (2), we get the logic circuit for Half Subtractor shown in fig.5(a) and its block diagram is shown in fig.5(b).

Q.27 Draw the logic diagram of a full subtractor using half subtractors and explain its working with the help of a truth table. **(6)**

Ans:

Full Subtractor: A Full Subtractor has to take care of repeated borrow from the next higher bit. At any stage alongwith the two bits (one of which is to be subtracted from the other) is another input B_{in}, i.e., borrow bit from the D_i and borrow B_o. Table shows the truth table.

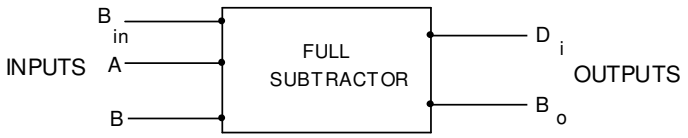


Fig.5(c) Block Diagram of Full Subtractor

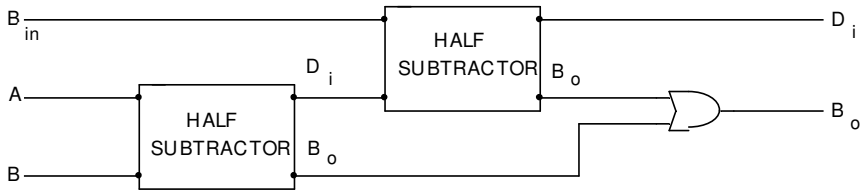


Fig.5(d) Block Diagram of Full Subtractor as Combination of two Half Subtractors and OR Gate

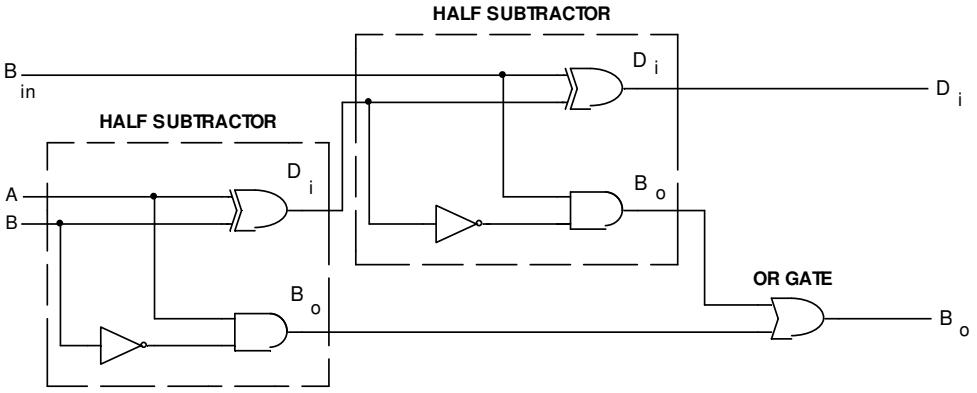


Fig.5(e) Logic Diagram of Full Subtractor

Fig.5(c) shows a block diagram for a full subtractor. It can be constructed from two Half Subtractors and an OR gate as shown in Fig.5(d). The logic diagram is shown in Fig.5(e). This logic diagram is as per the truth table of Table 5.1.

	Inputs		Outputs	
A	**B**	**B_{in}**	**D_i**	**B_O**
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Table 5.1 Truth Table for Full Subtractor

Q.28 Design a BCD to seven segment decoder that accepts a decimal digit in BCS and generates the appropriate output for segments in display indicator.**(14)**

Ans:

BCD-TO-7-Segment Decoder: A digital display that consists of seven LED segments is commonly used to display decimal numerals in digital systems. Most familiar examples are electronic calculators and watches where one 7-segment display device is used for displaying one numeral 0 through 9. For using this display device, the data has to be converted from some binary code to the code required for the display. Usually the binary code used is Natural BCD. Fig.6(a) shows the display device. Fig.6(b) shows the segments which must be illuminated for each of the numerals and Fig.6(c) gives the display system.

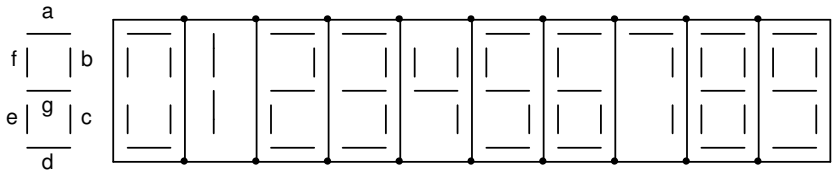


Fig.6(a) **Fig.6(b)**

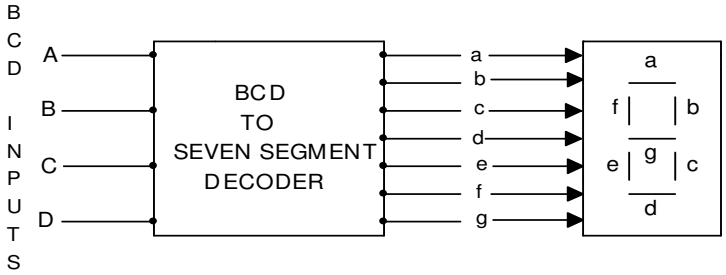


Fig.6(c)

Table 6.1 gives the truth table of BCD-to-7-segment Decoder. Here ABCD is the Natural BCD code for numerals 0 through 9. The K-maps for each of the outputs a through g are given in Fig.6(d), 6(f),6(h),6(j),6(l),6(n),6(p). The entries in the K-map corresponding to six binary combinations not used in the truth table are X –don't care.

Decimal Digit Displayed		**Inputs**						**Outputs**			
	A	**B**	**C**	**D**	**a**	**b**	**c**	**d**	**e**	**f**	**g**
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	0	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1

Table 6.1 Truth Table of BCD-to-7 Segment Decoder

(i) K-map and Logic Diagram for Digital Output 'a':

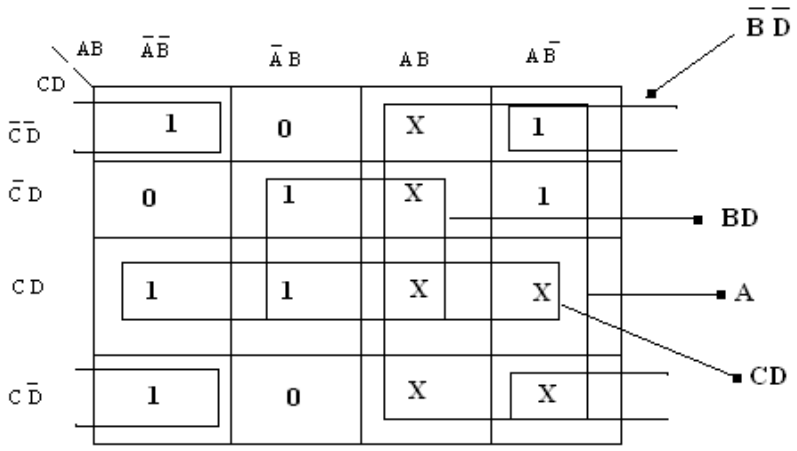


Fig.

6(d) K-map for Output `a'

The simplified expressions for the Fig.6(d) is given by a = $\overline{B}\ \overline{D}$ + BD + CD + A and the logic diagram is given in Fig.6(e)

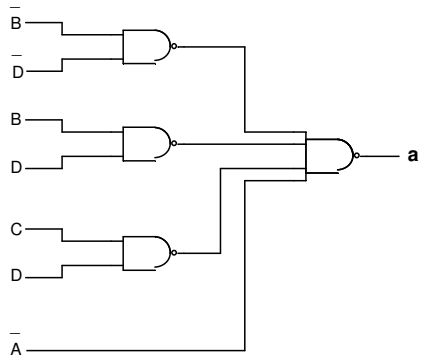


Fig.6(e) Logic Diagram for Output 'a'

(ii) K-map and Logic Diagram for Digital Output 'b':

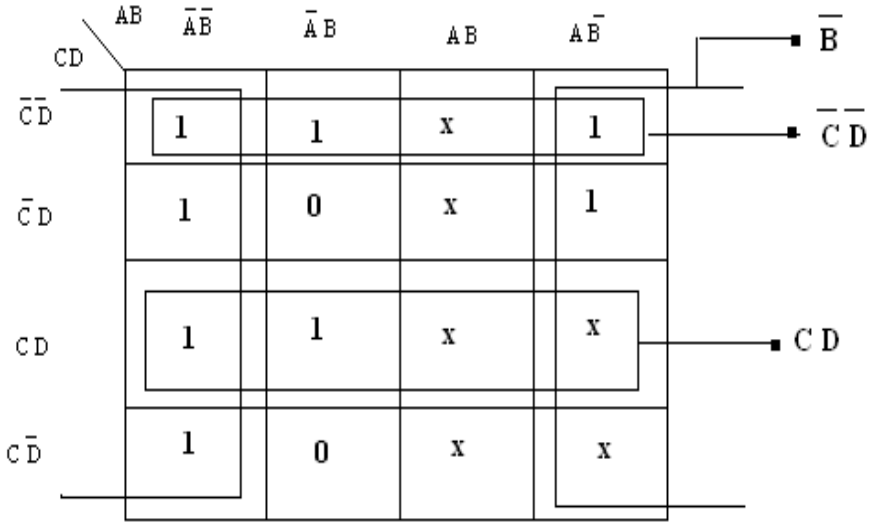


Fig. 6(f) K- map for output `b'

The simplified expressions for the Fig.6(f) is given by b = $\overline{B}$ + $\overline{C}\ \overline{D}$ + CD and the logic diagram is given in Fig.6(g)

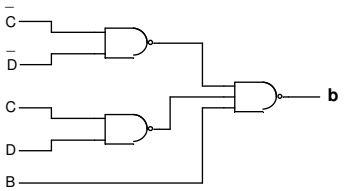


Fig.6(g) Logic Diagram for Output 'b'

(iii) K-map and Logic Diagram for Digital Output 'c':

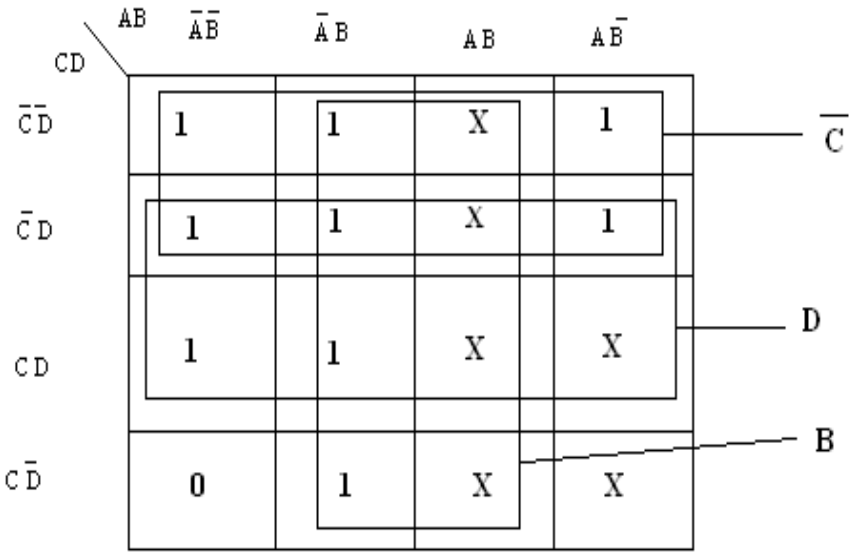


Fig. 6(h) K-map for output `C'

The simplified expressions for the Fig.6(h) is given by c = $B + \bar{C} + D$ and the logic diagram is given in Fig.6(i)

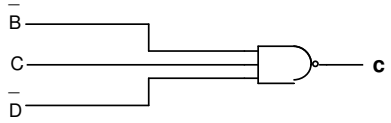


Fig.6(i) Logic Diagram for Output ' c '

(iv) K-map and Logic Diagram for Digital Output'd':

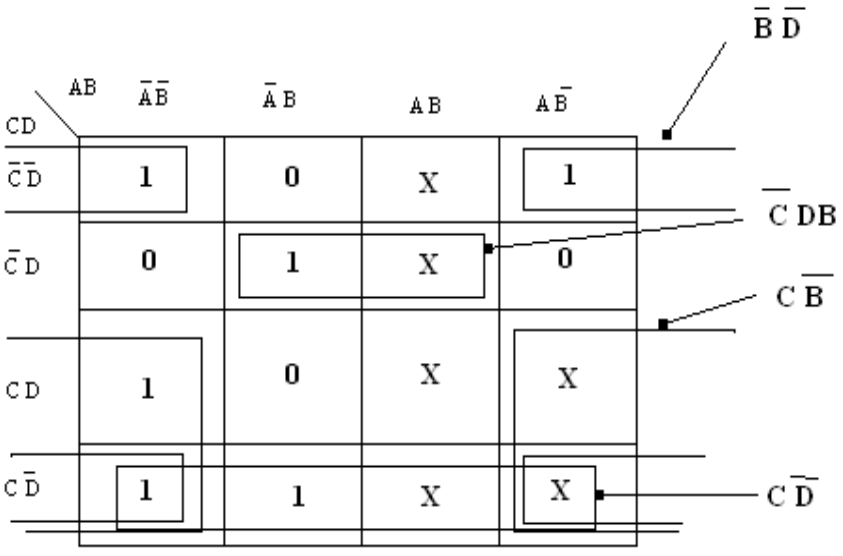


Fig. 6(j) K- map for Output `d'

The simplified expressions for the Fig.6(j) is given by
d = $\overline{B}\ \overline{D}$ + C $\overline{D}$ + $\overline{B}$ C +B $\overline{C}$ D and the logic diagram is given in Fig.6(k)

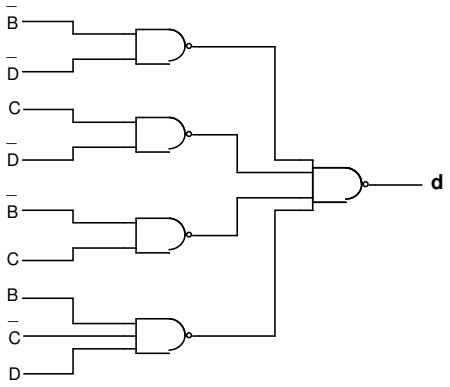


Fig.6(k) Logic Diagram for Output ' d '

(v) K-map and Logic Diagram for Digital Output 'e':

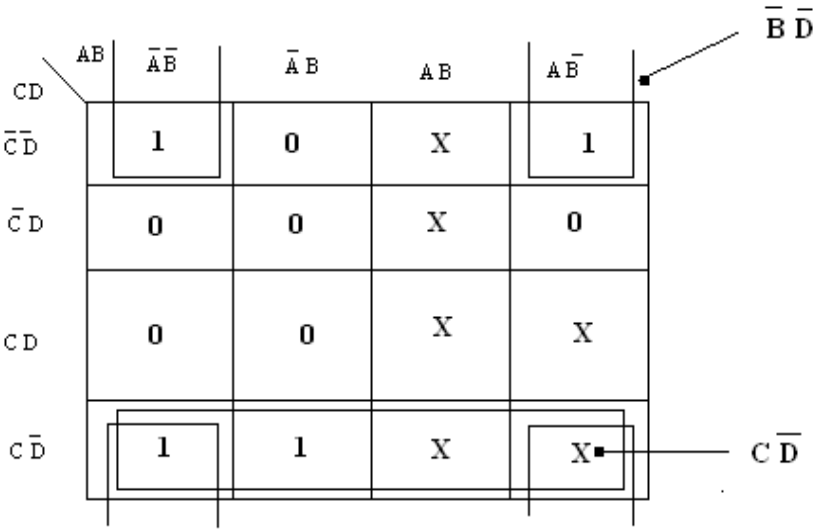


Fig. 6(l) K-map for output `e'

The simplified expressions for the Fig.6(l) is given by e = $\overline{B}\ \overline{D}$ + C $\overline{D}$ and the logic diagram is given in Fig.6(m)

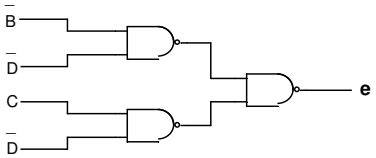


Fig.6(m) Logic Diagram for Output 'e'

(vi) K-map and Logic Diagram for Digital Output 'f':

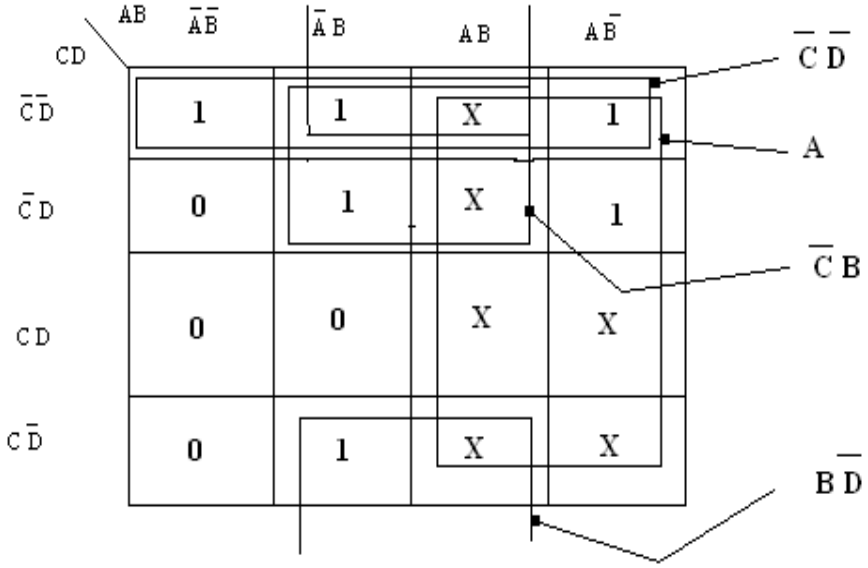


Fig. 6(n) K-map for Output `f'

The simplified expressions for the Fig.6(n) is given by f = A + $\bar{C}\ \bar{D}$ + B $\bar{C}$ +B $\bar{D}$ and the logic diagram is given in Fig.6(o)

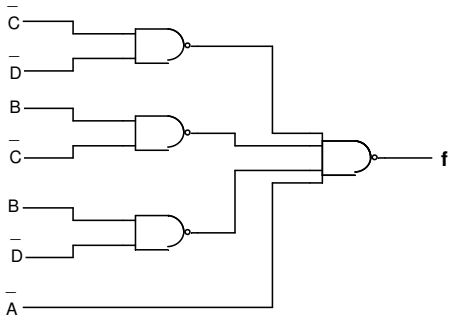


Fig.6(o) Logic Diagram for Output ' f'

(vii)K-map and Logic Diagram for Digital Output 'g':

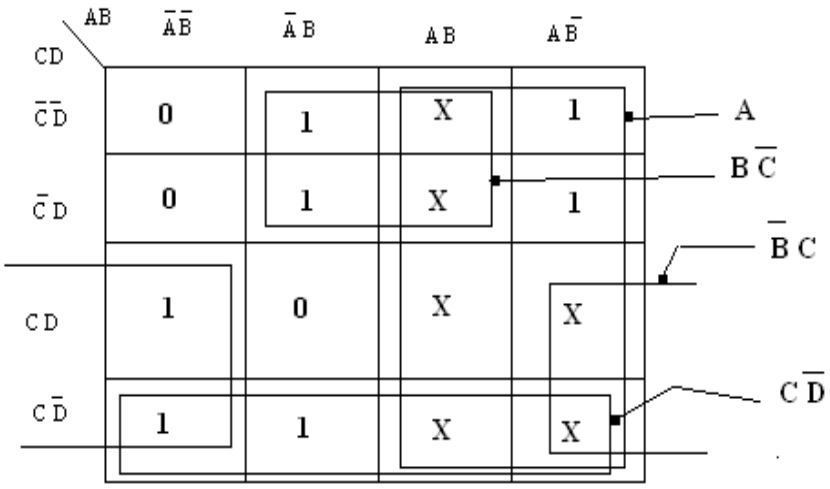


Fig. 6(p) K-map for Output ' g '

The simplified expressions for the Fig.6(p) is given by g = A + B $\bar{C}$ + $\bar{B}$C+ C $\bar{D}$ and the logic diagram is given in fig.6(q).

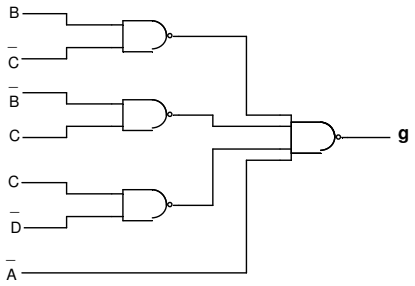


Fig.6(q) Logic Diagram for Output ' g '

Q.29 Explain the working of a demultiplexer with the help of an example. **(6)**

Ans:

1:4 Demultiplexer: Fig.7(a) shows the logic circuit of a 1:4 demultiplexer. It has two NOT gates, 4 AND gates, one data input line, 2 select lines (S_0,S_1) and four output lines *(D_0,* D_1,D_2,D_3*).* The data input line feeds all the AND gates. However, the two select lines enable only one gate at one time. If S_1S_0= 00 then the data goes to D_0. if S_1S_0= 01, then the data goes toD_1. If S_1S_0 = 10, then the data goes to D_2 and if S_1S_0= 1, then the data goes to D_3.

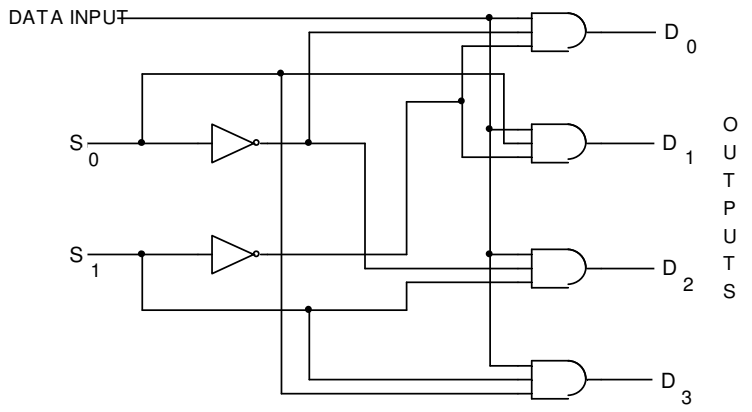


Fig.7(a) Logic Circuit of 1:4 Demultiplexer

Q.30 Give the truth table of S-R and D-flipflops. Convert the given S-R flipflop to a D-flipflop. **(8)**

Ans:

The Truth Table of S-R Flip-Flop is shown in Fig.7(b) and truth table of D Flip-Flop is shown in Fig.7(c)

Inputs		**Output**
S_n	**R_n**	**Q_{n+1}**
0	0	Qn
1	0	1
0	1	0
1	1	?

Fig.7(b) Truth Table for S-R Flip-Flop

Input **D_n**	**Output** **Q_{n+1}**
0	0
1	1

Fig.7(c) Truth Table for D-Flip-Flop

If we use only the middle two rows of the truth table of the S-R Flip-Flop shown in Fig.7(b) then we obtain a D-type Flip-Flop as shown in Fig.7(d) and 7(e). It has only one input referred to as D-input or Data Input. Its truth table is given in Fig.7(c) from which it is clear that the output Q_{n+1} at the end of the clock pulse equals the input D_n before the clock pulse. This is equivalent to saying that the input data appears at the output at the end of the clock pulse. Thus, the transfer of data from the input to the output is delayed and hence the name Delay (D) Flip-Flop. The D-type Flip-Flop is either used as a Delay Device or as a Latch to store 1-bit of binary information.

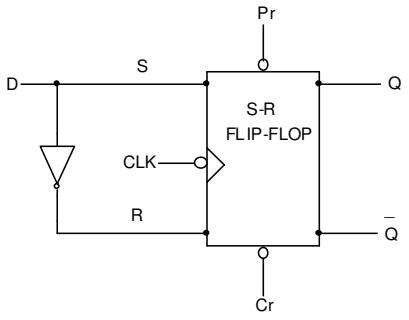


Fig.7 (d) S-R Flip-Flop converted into a D-Flip-Flop

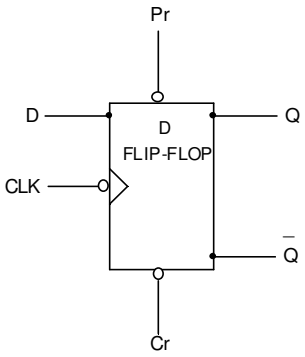


Fig.7 (e) Logic Symbol of D Flip-Flop

Q.31 Define a register. Construct a shift register from S-R flip-flops. Explain its working.**(8)**

Ans:

Register: A register consists of a group of flip-flops and gates that effect their transition. The flip flops hold the binary information and the gates control when and how new information is transformed into the register.

S-R Flip-Flop Shift Register: Shift registers can be built by using SR flip-flops.Fig.9(a) shows the 4-bit shift register, which uses RS flip-flops. It uses Four SR Flip-Flops in cascade and the inputs to the last three flip-flops in the chain receive complementary inputs, that is if S = 0, *R* = 1 and if S = 1, *R* = 0. The first flip-flop has complementary S and *R* inputs and, therefore, it behaves like a D-type flip-flop. Because of the Inverter in the clock line, data will be transferred to flip-flop outputs on the positive going edge of the clock pulse.

There are two inputs *A* and *B*. Any one of the inputs can be used. Since a 1 input at *A* or *B* will be a 1 input at S of the first flip-flop, as a result of double complementation, a positive going clock pulse will produce an output of 1 at Q of the first flip-flop. Normally both *A* and B inputs of the NAND gate are connected together when data is being fed and the NAND is not required to serve as a gate.

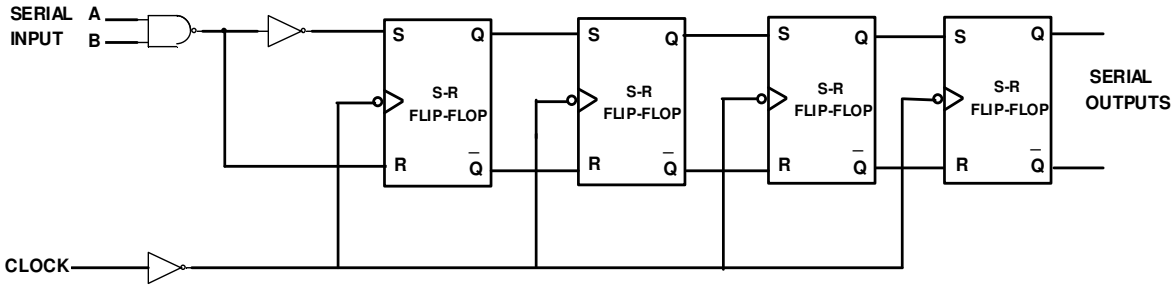


Fig.9(a)Logic Diagram of S-R Flip-Flop Shift Register

Q.32 Explain how a shift register can be used as a ring counter giving the wave forms at the output of the flipflops. **(6)**

Ans:

Shift Register as a Ring Counter: A Ring Counter is a Circular Shift Register with only one flip-flop being set at any particular time; all other are cleared. The single bit is shifted from one flip-flop to the other to produce the sequence of timing signals. Fig.9(b) shows a 4-bit shift register connected as a ring counter. The initial value of the register is 1000, which produces the variable T_0. The single bit is shifted right with every clock pulse and circulates back from T_3 to T_0. Each flip-flop is in the 1 state once every four clock pulses and produces one of the four timing signals shown in Fig.9(c). Each output becomes 1 after the negative-edge transition of a clock pulse and remains 1 during the next clock pulse.

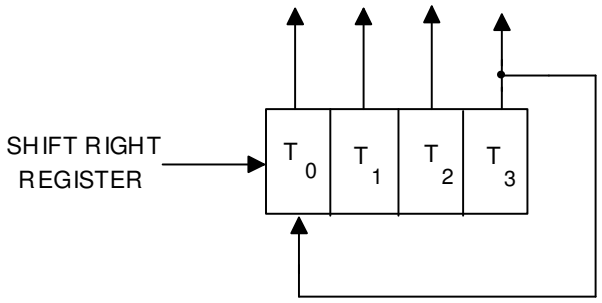


Fig.9(b) 4-bit shift register connected as a ring counter.

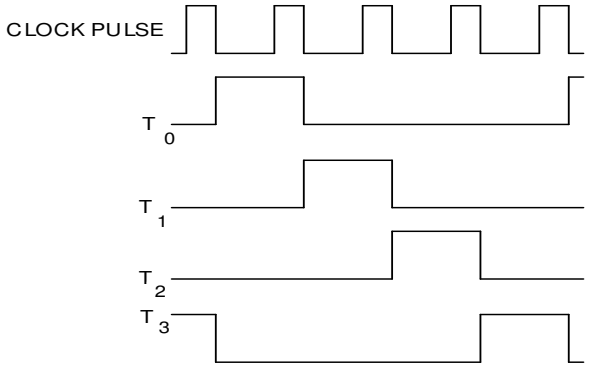


Fig.9(c) Waveforms at the output of Flip-Flops

Q.33 Differentiate between linear addressing and matrix addressing modes with examples. Which of them is the best method? **(4)**

Ans:

Linear Addressing: Addressing is the process of selecting one of the cells in a memory to be written into or to be read from. In order to facilitate selection, memories are generally arranged by placing cells in Linear form or Matrix form.

Linear Addressing Mode: A single column that has n rows and 1 column (such as the 16X1 array of cells) shown in fig.11(a) is frequently called Linear Addressing. Selection of a cell simply means selection of the corresponding row and the column is used.

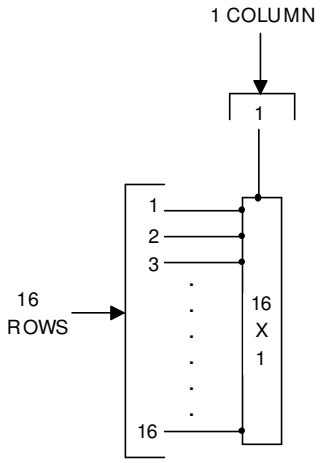


Fig.11 (a) Linear Addressing Mode

Matrix Addressing Mode: The arrangement that requires the fewest address lines is a square array of n rows and n columns for a total memory capacity of n x n = n^2 cells. This arrangement of n rows and n columns is frequently referred to as Matrix Addressing which is shown in fig.11(b).

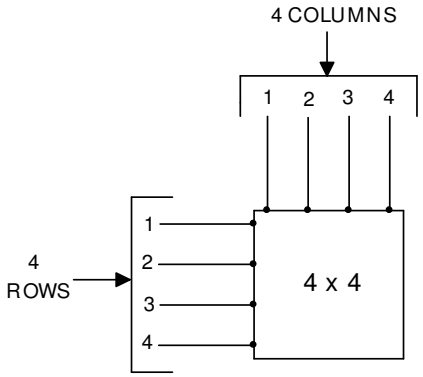


Fig.11(b) Matrix Addressing Mode

Best Method: Matrix Addressing is the best method, because this configuration only requires 8 address lines (i.e.,4 rows and 4 columns), whereas Linear Addressing method requires a total of 17 address lines (i.e., 1 column and 16 rows).The square configuration is so widely used in industry.

Q.34 Write short note on the following: Johnson counter. **(4)**

Ans:

Johnson Counter: Johnson Counter is an synchronous counter, where all flip-flops are clocked simultaneously and the clock pulses drive the clock input of all the flip-flops together so that there is no propagation delay. Fig.11(e) shows the circuit of Johnson counter. In this case the D input of FF_0 is driven by $\overline{Q}$ output of FF_3, *i.e.,* the complement of the output of the last flip flop is fed to the D of FF_0. This feedback arrangement produces the sequence of states shown in Table 11.2. The 4 bit sequence

has a total of 8 states (*n* bit sequence will have *2n* states). Thus an *n* bit Johnson counter will have a modulus of *2n.*

The Q output of each stage feeds the *D* input of next stage. But the $\overline{Q}$ output of the last stage feeds the *D* input of first stage. The counter fills up with 1's from left to right and then fills up 0s again as shown in Table 11.2. Fig. 11*(f)* shows the waveshapes/timing diagram of 4 bit Johnson counter.

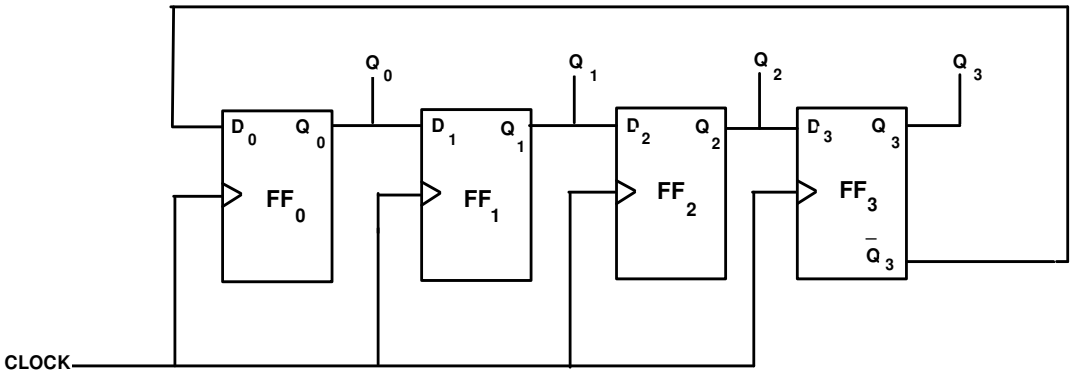


Fig.11(e) Logic Diagram of Johnson counter

Clock Pulse	Q_0	Q_1	Q_2	Q_3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1

Table 11.2 Sequence of states of 4 bit Johnson Counter

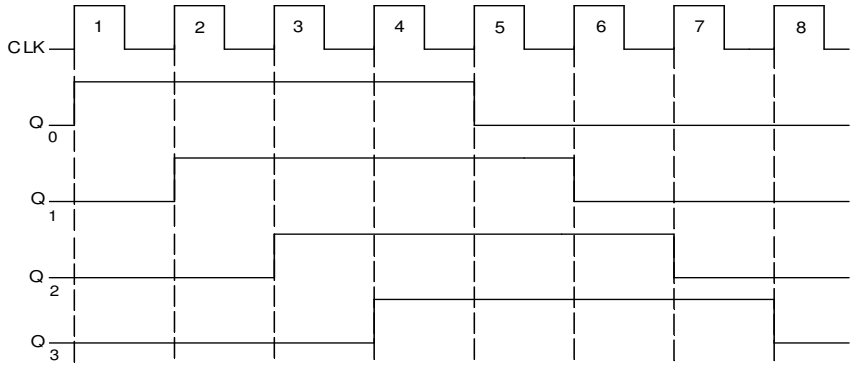
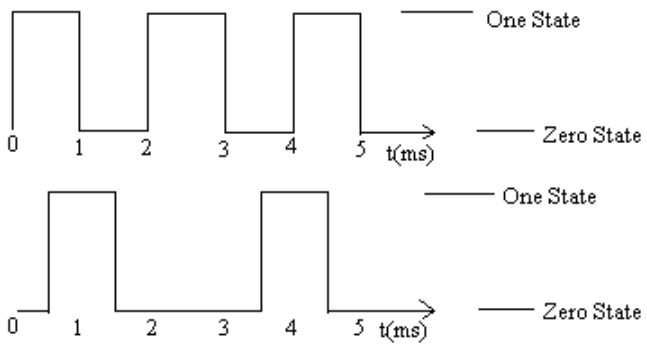


Fig.11(f) Timing Diagram of 4-bit Johnson Counter

Q.35 The voltage waveforms shown in Fig.1 are applied at the inputs of 2-input AND and OR gates. Determine the output waveforms. **(3)**



Ans:

The Output waveforms for AND and OR gates are shown in fig.3(a)

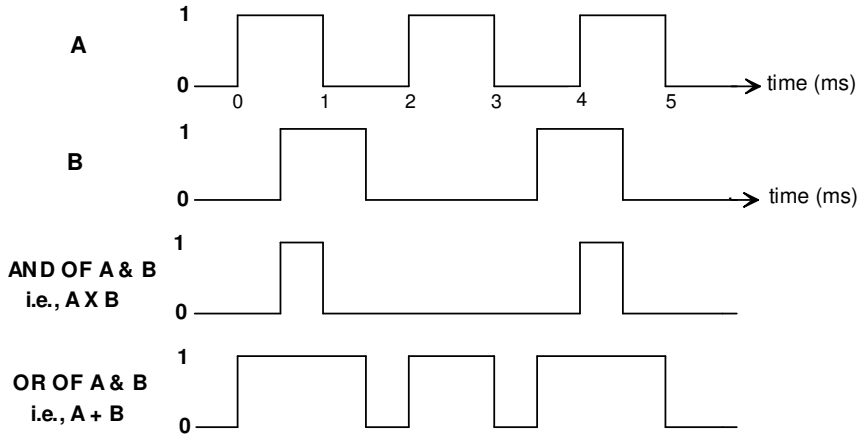


Fig.3(a) Output Waveforms

Q.36 What are the advantages of CMOS logic and explain CMOS Inverter with the help of a neat circuit diagram. **(7)**

Ans:

Advantages of CMOS Logic:

(i) The power dissipation is minimum of all the logic families

(ii) LSI & VLSI are possible

CMOS Inverter:

The basic CMOS logic circuit is an inverter shown in Fig.5(a). For this circuit the logic levels are 0 V (logic 0) and V_{CC} (logic 1). When $V_i = Vcc$, T_1 turns ON and T_2 turns OFF. Therefore $V_O \approx 0$ V and since the transistors are connected in series, the current I_D is very small. On the other hand, when $V_i = 0$ V, T_1 turns OFF and T_2 turns ON giving an output voltage $V_O \approx Vcc$ and I_D is again very small. In either logic state, T_1 or T_2 is OFF and the quiescent power dissipation which is the product of the OFF leakage current and V_{cc} is very low. More complex functions can be realized by combinations of inverters.

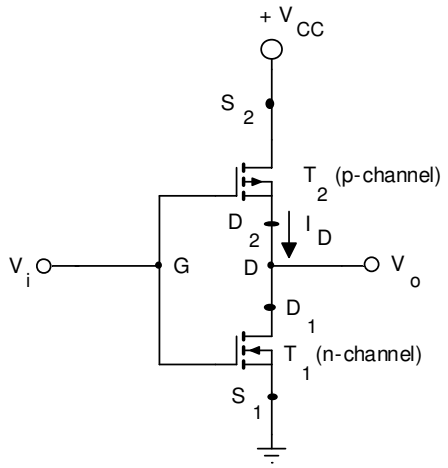


Fig.5(a) Logic Diagram of CMOS Inverter

Q.37 What is Tri-state logic and explain Tri-state logic inverter with the help of a circuit diagram. Give its Truth Table. **(7)**

Ans:

Tri-state Logic:

In normal logic circuits, there are two states of the output, LOW and HIGH. If the output is not in the LOW state, it is definitely in the other state (HIGH). Similarly, if the output is not in the HIGH state, it is definitely in the LOW state. In complex digital systems like microcomputers and microprocessors, a number of gate outputs may be required to be connected to a common line which is referred to as a bus which in turn may be required to drive a number of gate inputs.

When a number of gate outputs are connected to the bus, Totem pole TTL outputs leads to heating of the ICs which may get damaged and Open-collector TTL outputs causes the problems of loading and speed of operation. To overcome these difficulties, in addition to low impedance outputs 0 & 1, there is a third state known as the High-impedance state. Such logic circuits in which the output can have three states is called tri-state logic.

In the Tri-state Logic, in addition to low impedance outputs 0 & 1, there is a third state known as the High-impedance state. When the gate is disabled, it is in the third state.

Tri-state Logic Inverter:

The functional diagram of Tri-state Logic Inverter is shown in fig.5(b) and its logic diagram is shown in fig. 5(c). When the control input is LOW, the drive is removed from T_3 & T_4. Hence both T_3 & T_4 are cut-off and the output is in the third state. When the control input is HIGH, the output Y is Logic 1 or 0 depending on the data input. Truth table of Tri-state Logic Inverter is given in Table No.5.1

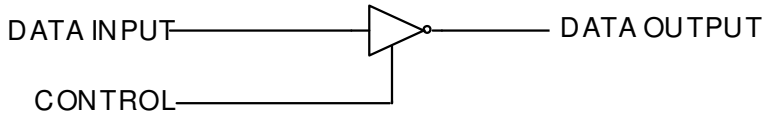


Fig.5(b) Functional Diagram of Tri-state Logic Inverter

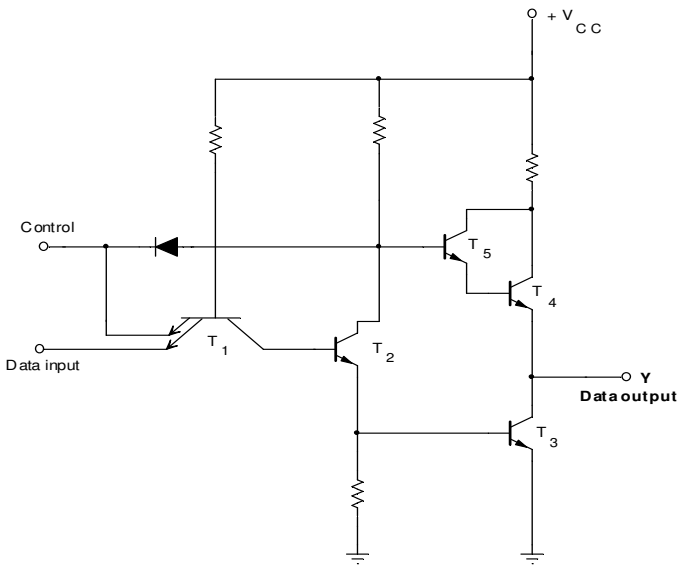


Fig.5(c) Logic Diagram of Tri-state Logic Inverter

Data Input	Control	Data Output
0	0	High - Z
1	0	High - Z
0	1	1
1	1	0

Table 5.1 Truth Table of Tri-state Logic Inverter

Q.38 What is a digital comparator. Explain the working of a 2-bit digital comparator with the help of Truth Table. **(6)**

Ans:

Digital Comparator: The comparison of two numbers is an operation that determines if one number is greater than, less than, or equal to the other number. A Digital comparator is a combinational circuit that compares two numbers, A and B, and determines their relative magnitudes. The outcome of the comparison is specified by three binary variables that indicate whether A > B, A = B, or A < B.

Comparators can be designed for comparing multibit numbers. Figure 6(e) shows the block diagram of an *n-bit* comparator. It receives two *n-bit* numbers *A* and *B* as inputs and the outputs are *A > B, A = B,* and *A < B*. Depending upon the relative magnitude of the two numbers, one of the outputs will be HIGH. Table 6.2 gives the truth table of a 2-bit comparator.

(I) If the magnitude of the inputs A and B are equal (i.e., A = B): Consider two numbers, A and B as inputs with two digits each i.e., A1, A0 and B1, B0. The two numbers are equal if all pairs of significant digits are equal i.e., if $A_1 = 0$, $A_0 = 0$, $B_1 = 0$, $B_0 = 0$, then $A_1 = B_1$ and $A_0 = B_0$. For example if $A_1 = 0$, $A_0 = 0$, $B_1 = 0$, $B_0 = 0$, then pairs of significant digits i.e., $A_1 = B_1 = 0$ and $A_0 = B_0 = 0$. Output for this combination becomes 1 for A = B and 0 for A < B and A > B. This is given in the Truth Table.

(II) If the magnitude of the input A is greater than or less than B (i.e., A > B or A < B): To determine if A is greater than or less than B, we inspect the relative magnitude of pairs of significant digits starting from the most significant position. If the two digits are equal, we compare the next lower significant pair of digits. This comparison continues until a pair of unequal digits is reached.

(i)If the input A is greater than B (i.e., A > B): If the corresponding digit of A is 1 and that of B is 0, we conclude that A > B. For example if $A_1 = 0$, $A_0 = 1$, $B_1 = 0$, $B_0 = 0$, then pairs of significant digits are $A_1 = B_1 = 0$, and A_0 (i.e., digit 1) > B_0(i.e., digit 0). This is shown in the Truth Table.

(ii) If the input A is less than B (i.e., A < B): If the corresponding digit of A is 0 and that of B is 1, we conclude that A < B. For example if $A_1 = 0$, $A_0 = 0$, $B_1 = 0$, $B_0 = 1$, then pairs of significant digits are $A_1 = B_1 = 0$, and A_0 (i.e., digit 0) < B_0 (i.e., digit 1). This is shown in the Truth Table.

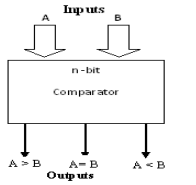


Fig.6(e) Block diagram of n-bit comparator

	Inputs				Outputs	
A_1	A_0	B_1	B_0	A > B	A = B	A < B
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

Table 6.2 Truth Table of a 2-Bit Comparator

Q.39 What is a Shift Register? What are its various types? List out some applications of Shift Register. **(6)**

Ans:

Shift Register: A register in which data gets shifted towards left or right when clock pulses are applied is known as a Shift Register.

Types of Shift Registers:
(i) Serial-In Serial-Out (SISO) Shift Register
(ii) Serial-In Parallel Out (SIPO) Shift Register
(iii) Parallel-In Serial Out (PISO) Shift Register
(iv) Parallel-In Parallel Out (PIPO) Shift Register

Applications of Shift Registers:
(i) Serial to Parallel Converter
(ii) Parallel to Serial Converter
(iii)Delay line
(iv)Ring Counter
(v) Twisted-ring Counter
(vi)Sequence Generator

Q.40 Design a MOD-6 synchronous counter using J-K Flip-Flops. **(8)**

Ans:

Design of Mod-6 Counter: The Mod-6 synchronous counter, have six counter states (i.e., from 0 to 6). The counter design table for this counter lists the three flip-flop and their states as 0 to 6 and the six inputs for the three flip-flops. The flip-flop inputs required to step up the counter from the present to the next state is worked out with the help of the excitation table. The desired counter states and *the J* K inputs required for counter flip-flops are given in the counter design table shown in Table No.8.1

Input pulse count	**Counter States**			**Flip-Flop Inputs**					
	A	B	C	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	1	X	0	X	0	X
1	1	0	0	X	1	1	X	0	X
2	0	1	0	1	X	X	0	0	X
3	1	1	0	X	1	X	1	1	X
4	0	0	1	1	X	0	X	X	0
5	1	0	1	X	1	0	X	X	1
6(0)	0	0	0						

Table 8.1 counter Design Table for Mod-6 Counter

Flip-Flop A:

The initial state is 0. It changes to 1 after the clock pulse. Therefore, J_A should be 1 and K_A may be 0 or 1 (that is X). In the next state 1 changes to 0 after the clock pulse. Therefore, J_A may be 0 or 1 (i.e., X)and K_A should be 1.

Flip-Flop B:

The initial state is 0 and it remains unchanged after the clock pulse. Therefore, J_B should be 0 and K_B may be 0 or 1 (that is X). In the next state 0 changes to 1 after the clock pulse. Therefore, J_B should be 1 and K_B may be 0 or 1 (i.e., X).

Flip-Flop C:

The initial state is 0 and it remains unchanged after the clock pulse. Therefore J_C should be 0 and K_C may be 0 or 1 (i.e., X). In the next state, it remains unchanged after the clock pulse . Therefore, J_C should be 0 and K_C may be 0 or 1 (i.e., X).The JK inputs required for this have been determined with the help of the excitation table, (Table 8.1). The flip-flop input values are entered in Karnaugh maps shown in Fig. 8b [(i), (ii), (iii), (iv), (v) and (vi)] and a Boolean expression is found for the inputs to the three flip-flops and then each expression is simplified. *As* all the counter states have not been utilized, Xs (don't) are entered to denote un-utilized states. The simplified expressions for each input have been shown under each map. Finally, these minimal expressions for the flip-flop inputs are used to draw a logic diagram for the counter shown in fig.8(c).

A*s* before, the JK inputs required for this have been determined with the help of the excitation table, (Table 8.1). These input values are entered in Karnaugh maps Fig. 8(b)[i to vi] and a Boolean expression is found for the inputs to the three flip-flops and then each expression is simplified. Xs have been entered in those counter states which have not been utilized. The simplified expressions for each input have been shown under each map and finally a logic diagram based on these expressions is drawn and is shown in fig.8(c).

	$\overline{B}\,\overline{C}$	$\overline{B}C$	BC	$B\overline{C}$
$\overline{A}$	1	1	X	1
A	X	X	X	X

Map for J_A
$J_A = 1$
Fig.(i)

	$\overline{B}\,\overline{C}$	$\overline{B}C$	BC	$B\overline{C}$
$\overline{A}$	X	X	X	X
A	1	1	X	1

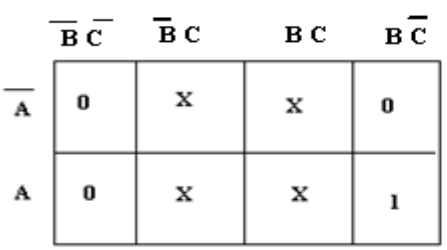
Map for K_A
$K_A = 1$
Fig.(ii)

	$\overline{B}\,\overline{C}$	$\overline{B}C$	BC	$B\overline{C}$
$\overline{A}$	0	0	X	X
A	1	0	X	X

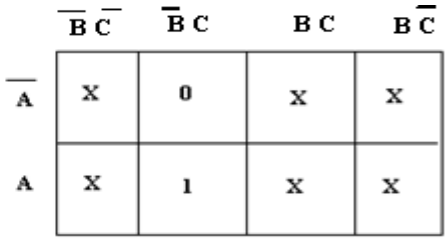
Map for J_B
$J_B = A\overline{C}$
Fig.(iii)

	$\overline{B}\,\overline{C}$	$\overline{B}C$	BC	$B\overline{C}$
$\overline{A}$	X	X	X	0
A	X	X	X	1

Map for K_B
$K_B = A$
Fig.(iv)



Map for J_C
$J_C = AB$
Fig.(v)



Map for K_C
$K_C = A$
Fig.(vi)

Fig.8(b) Karnaugh Maps for J_A,K_A,J_B,K_B,J_C,K_C

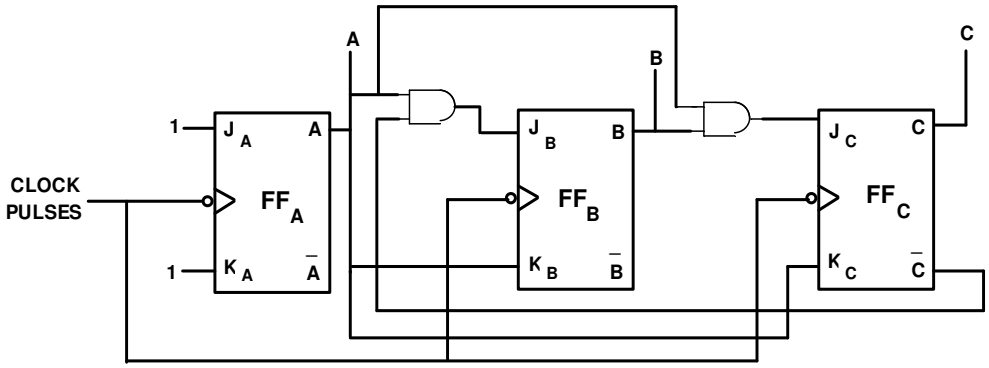


Fig.8(c) Logic Diagram for MOD-6 Synchronous Counter

Q.41 What is ROM? Is the ROM a volatile memory? Explain. **(3)**

Ans:

ROM: Read Only Memory is a Permanent or Semi-permanent Memory. In Permanent ROM, the data is permanently stored and cannot be changed. It can only be read from the memory. There cannot be a write operation because the specified data is programmed into the device by the manufacturer or the user. In Semi-permanent ROM also there is no write operation, but the data can be altered, to a limited extent, by special methods.

No. ROM is a Non-Volatile memory. Programming of ROM involves making of the required interconnections at the time of fabrication and therefore, its contents are unaffected, even when the power is OFF. Thus it is a Non-Volatile Memory.

Q.42 Draw the logic diagram of 16-bit ROM Array and explain its principle of operation. **(8)**

Ans:

16-bit ROM Array: A read-only memory is an array of selectively open and closed unidirectional contacts. .A 16-bit ROM array is shown in Fig. 9(b). To select any one of the 16 bits, a 4-bit address (A_3, A_2, A_1, A_0) is required. The lower order two bits (A1, A_0) are decoded by the decoder D_L which selects one of the four rows, whereas the higher order

two bits (A_3, A_2) are decoded by the decoder D_H which activates one of the four column sense amplifiers.

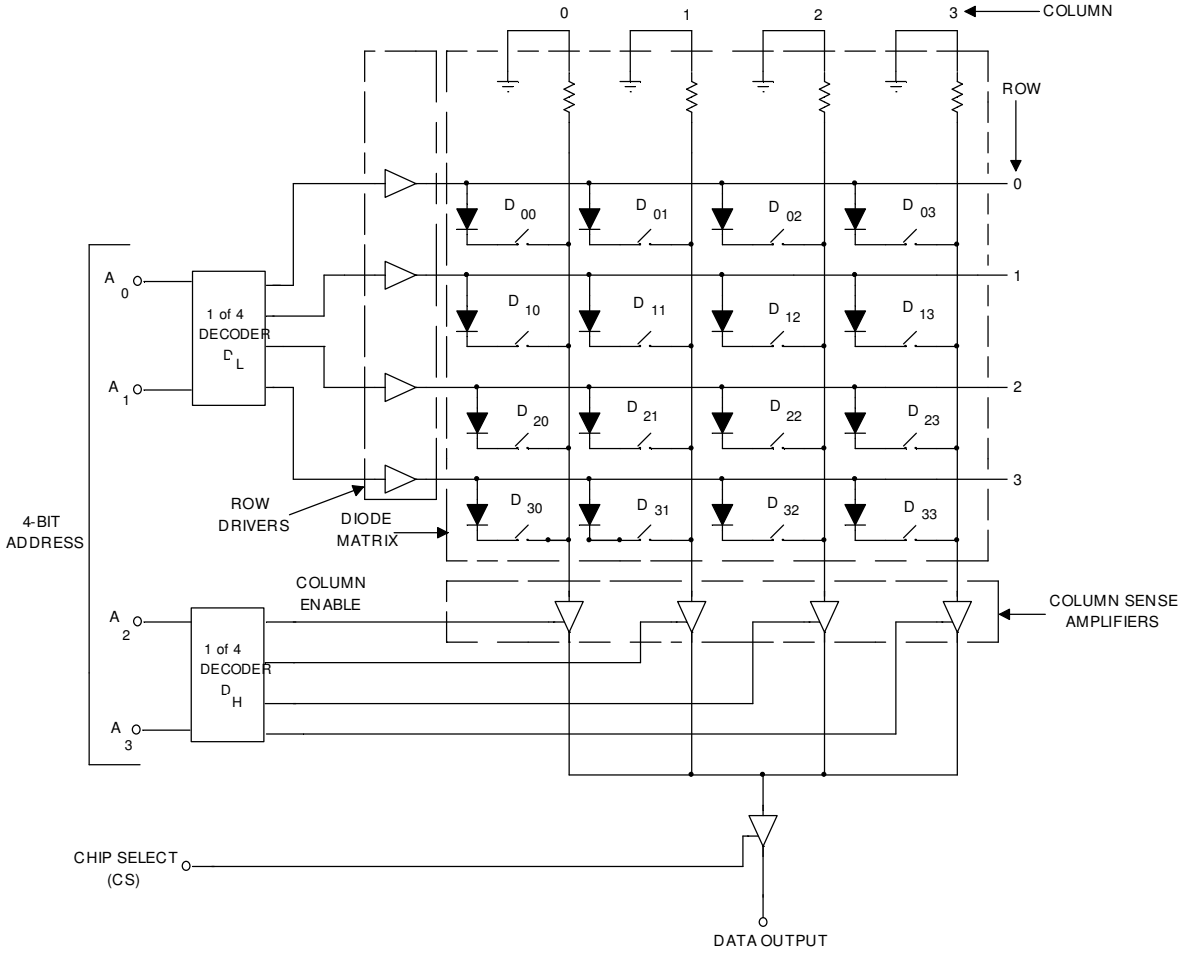


Fig.9(b) Logic Diagram of 16-bit ROM array

The diode matrix is formed by connecting one diode along with a switch between each row and column. For example the diode D_{21} is connected between row 2 and column 1.The output is enabled by applying logic 1 at the chip select *(CS)* input. Programming a ROM means to selectively open and close the switches in series with the diodes. For example, if the switch of diode D_{21} is in closed position and if the address input is 0110, the row 2 is activated connecting it to the column 1. Also the sense amplifier of column 1 is enabled which gives logic 1 output if the chip is selected *(CS* = 1). This shows that a logic1 is stored at the address 0110. On the other hand if the switch of diode D_{21} is open, logic 0 is stored at the address 0110.

Q.43 Explain briefly, why dynamic RAMs require refreshing? **(3)**

Ans:

Because of the charge's natural tendency to distribute itself into a lower energy-state configuration (i.e., the charge stored on capacitors leak-off with time), dynamic RAMs require periodic charge refreshing to maintain data storage.

Q.44 Draw the schematic circuit of an Analog to Digital converter using Voltage-to Frequency conversion and explain its principle of operation. Draw its relevant Waveforms. **(10)**

Ans:

Analog to Digital Converter Using Voltage-to-Frequency Conversion:An analog voltage can be converted into digital form by producing pulses whose frequency is proportional to the analog voltage. These pulses are counted by a counter for a fixed duration and the reading of the counter will be proportional to the frequency of the pulses and hence to the analog voltage.

A voltage-to-frequency converter is shown in Fig. 10(a). The analog voltage V_a is applied to an integrator whose output is applied at the inverting input terminal of a comparator. The non-inverting input terminal of the comparator is connected to a reference voltage $-V_R$. Initially, the switch S is open and the voltage v_o decreases linearly with time ($v_o = V_a t/\tau$) which is shown in Fig. 10(b). When the decreasing v_o reaches $-V_R$ at $t = T$, the comparator output V_C goes HIGH. This is used to close the switch S through a monostable multivibrator. When the switch S is closed, the capacitor C discharges, thereby returning the integrator output v_o to 0. Since the pulse width of the waveform V_C is very small, therefore, a monostable multivibrator is used to keep the switch S closed for a sufficient time to discharge the capacitor completely. The rate at which the capacitor discharges depends upon the resistance of the switch.

Let the pulse width of the monostable multivibrator be T_d. Therefore, the switch S remains closed for T_d after which it opens and v_o starts decreasing again.

If the integration time $T >> T_d$, the frequency of the waveforms v_o and V_C is given by

$$f = \frac{1}{T + T_d} \cong \frac{1}{T} = \frac{1}{\tau}\frac{V_a}{V_R}$$

Thus we obtain an output waveform whose frequency is proportional to the analog input voltage. An *A/D* converter using the voltage-to-frequency *(V/F)* converter is shown in Fig. 10(c). The output of the *V/F* converter is applied at the clock *(CK)* input of a counter through an AND gate. The AND gate is enabled for a fixed time interval T_1. The reading of the counter at $t = T_1$ is given by

$$n = fT_1 = \frac{1}{\tau}\frac{V_a}{V_R}T_1 \text{ which is proportional to } V_a.$$

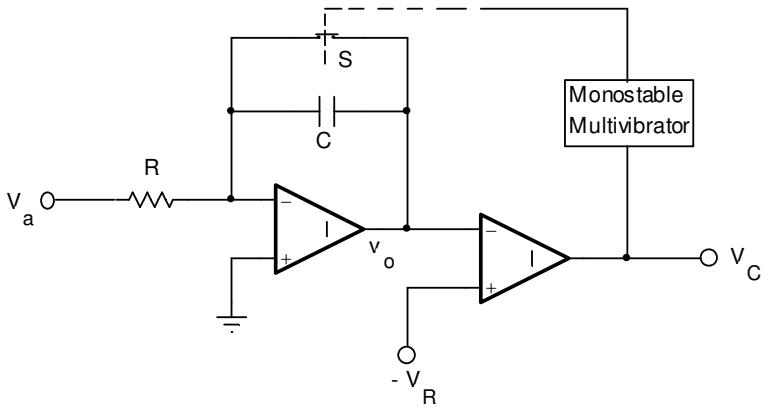


Fig.10(a) Logic diagram of Voltage-to-Frequency Converter

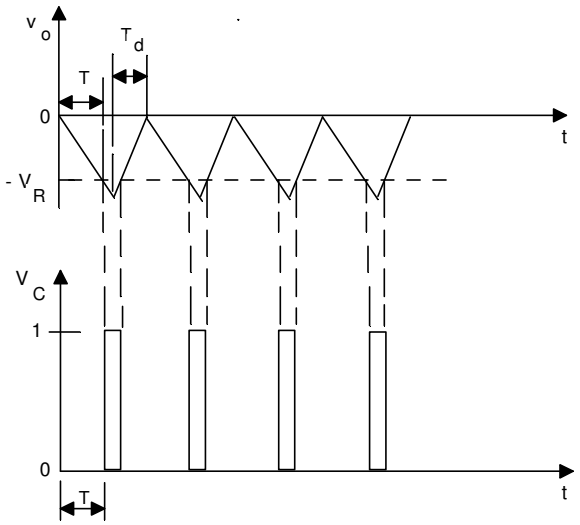


Fig.10(b) Waveforms of Voltage-to-Frequency Converter

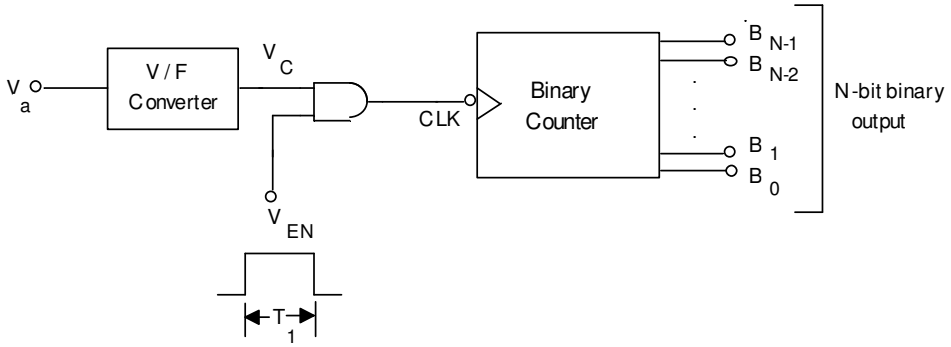


Fig.10(c) Schematic circuit of A/D converter using a V/F converter

Q.45 With the help of R-2R binary network, explain the working of a 3-bit D/A converter and derive an expression for the output voltage. **(10)**

Ans:

R-2R ladder D/A converter: An *R-2R* ladder D/A converter is shown in Fig.11(a). It uses resistors of only two values R and *2R*. The inputs to the resistor network are applied through digitally controlled switches. A switch is in 0 or 1 position corresponding to the digital input for that bit position being 0 or 1 respectively. Now, we consider a 3-bit *R-2R* ladder *D/A* network shown in Fig.11(b).In this circuit we have assumed that the digital input as 001.

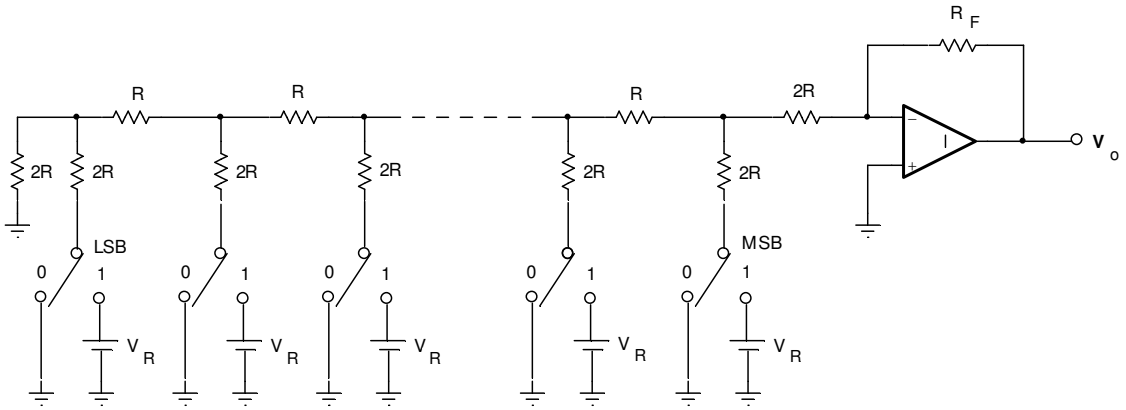


Fig.11(a) Logic Diagram of R-2R Ladder D/A Converter

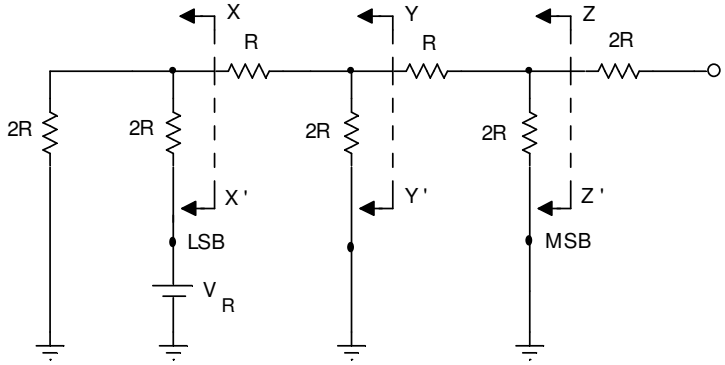


Fig.11(b) 3 bit R-2R Ladder D/A Network

The circuit is simplified using Thevenin's theorem. Applying Thevenin's theorem at *XX',* we obtain the circuit of Fig. I1(c). Similarly, applying Thevenin's theorem at *YY'* and *ZZ',* we obtain the circuits of Fig.11(d) and 11(e) respectively. Here, LSB is assumed as 1 and the equivalent voltage obtained is $V_R / 2^3$.

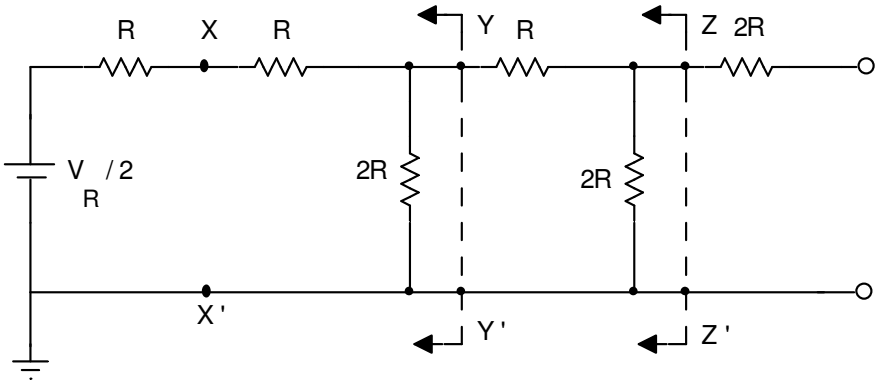


Fig.11(c) Equivalent circuit after applying Thevenin's Theorem at XX'

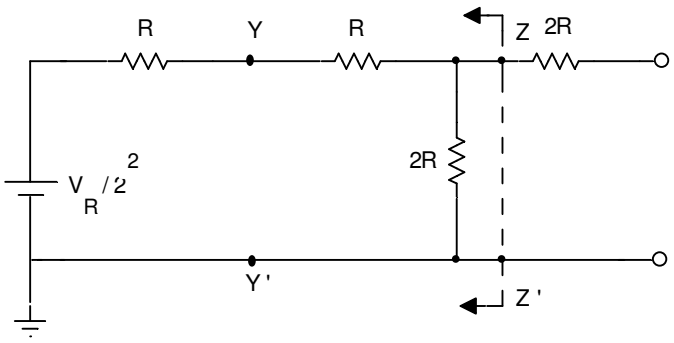


Fig.11(d) Equivalent circuit after applying Thevenin's Theorem at YY'

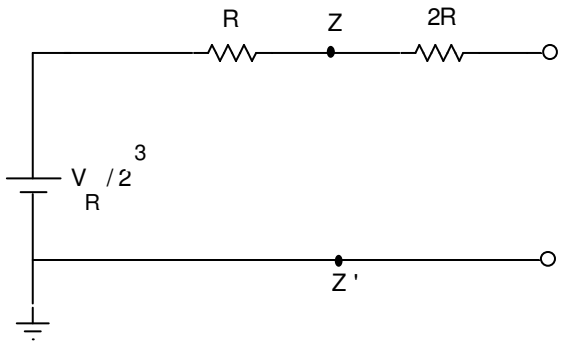


Fig.11(e) Equivalent circuit after applying Thevenin's Theorem at ZZ'

Similarly for the digital input of 010 and 100 the equivalent voltages are $V_R/2^2$ and $V_R/2^1$ respectively. The value of the equivalent resistance is 3R in each case. Therefore, we obtain an equivalent circuit of 3-bit R-2R Ladder D/A Converter which is given in Fig.11(f). The output analog voltage V_O is given by

$$V_O = -\left(\frac{R_F}{3R}.\frac{V_R}{2^3}b_0 + \frac{R_F}{3R}.\frac{V_R}{2^2}b_1 + \frac{R_F}{3R}.\frac{V_R}{2^1}b_2\right)$$

$$V_O = -\left(\frac{R_F}{3R}\right).\left(\frac{V_R}{2^3}\right)\left[4b_2 + 2b_1 + 1b_0\right]$$

Hence the above equation shows that the analog output voltage is proportional to the digital input.

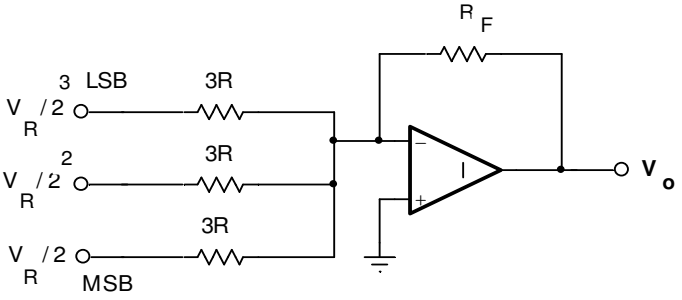


Fig.11(f) Equivalent circuit of 3-bit R-2R Ladder D/A Converter

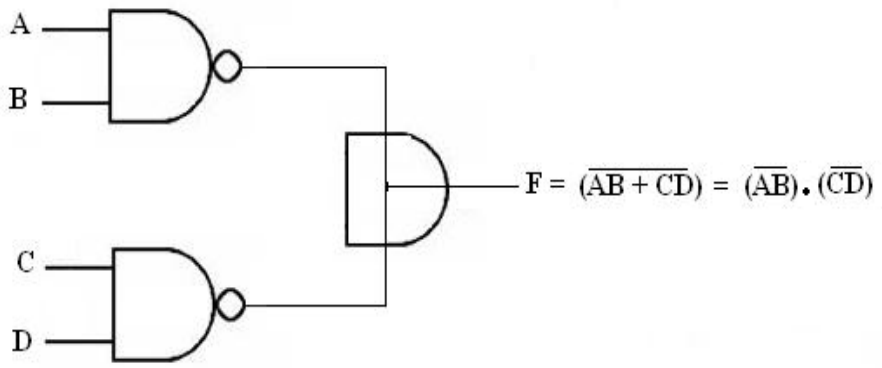
Q.46 What is meant by Wired-AND connection of digital ICs? What are its advantages and disadvantages? Draw a circuit of TTL gates with Wired-AND connection and explain its operation. **(10)**

Ans:

Wired AND digital IC: If input F and F' at two DTL NAND gates connected, the output can be considered as AND operations between the logic output. Because when both the output corresponds to cut-off stages of the transistors, the output will be unaffected and logic 1.when any of the outputs corresponds to the saturation condition approx 0.2 volt, the output from common point will become 0.2 volt. If A and B both input are DTL NAND gate and the C,D, are input for another ,NAND the output Y on joining F and F' at common terminal as follows: Y=(A.B)'.(C.D)'=(A.B+C.D)'

Wired – AND Connection

In digital IC`s NAND and NOR gates are most often used. For this reason NAND and NOR logic implementation are the most important from the practical point of view, some NAND & NOR gates are realized using wire connections between the o/p`s of two gates to provide a specific logic function. This type of logic is called as wired logic.



Wired – AND in open collector TTL gates

Advantages and disadvantages

- In this IC additional logic is performed without additional hardware.
- There is an effective reduction in the fan out of the gate.
- In the wired- AND connection speed of operation increases.
- Power dissipation in low output state in P(O) increases because of reduction in effective collector resistor.
- Current dissipation in logic 0 state will increases when two TTL gates with passive pull ups are ANDed by wired logic.
- The TTL fates with missing pull up circuit at the collector are also called open collector gates. These are more suitable for the wired connections.

Q.47 What is the necessity of Interfacing in digital ICs and what are the points to be kept in view, while interfacing between TTL gate and CMOS gate? **(4)**

Ans:

To achieve the optimum performance in digital system, device from more than one logic families can be used ,which takes advantages of the superior characteristics of each logic families. For example, CMOS logic ICs can be used in those parts of the system where low power dissipation is required, and TTL can be used where high speed of operation is required. When CMOS derives TTL, the following conditions are required to be satisfied.

$V_{OH(CMOS)} \geq V_{IH(TTL)}$

$V_{OL(CMOS)} \leq V_{IL(TTL)}$

$-I_{OH(CMOS)} \geq NI_{IH(TTL)}$

$-I_{OH(CMOS)} \geq -NI_{IL(TTL)}$

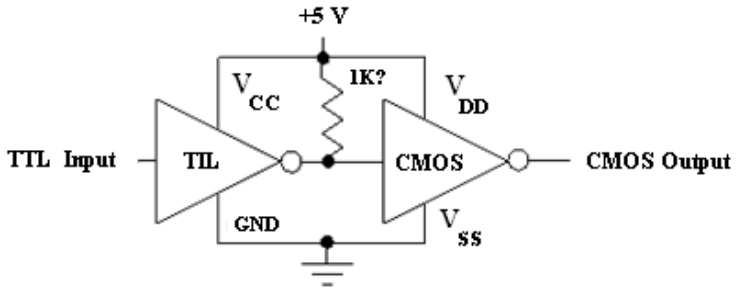


Figure 1: TTL-to-CMOS interfacing using pull-up register.

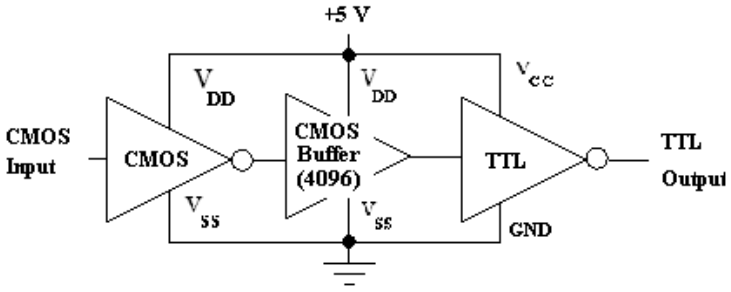
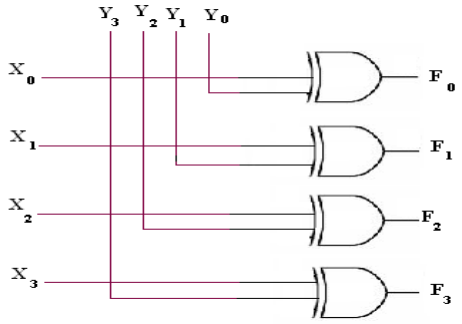


Figure 2: CMOS-to-TTL interfacing using a CMOS buffer IC

Q.48 Draw the logic diagram of 4-bit odd parity checker using EX-NOR gates and explain its operation with the help of Truth table. **(7)**

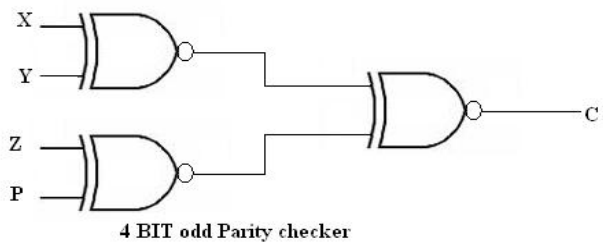
Ans:

4 bit odd parity checker using XNOR circuit:-The concept of parity checker ,Wherein the additional bit is known as parity. It can be either even or odd. The following circuit will give the 4 bit parity checker circuit.



logic diagram of 4-bit odd parity checker using EX-NOR gates

Parity checker networks are logic circuits with exclusive – OR functions. Ex OR operation of parity bit is a scheme for detecting errors during transmission of binary information. It is an extra bit transmitted and then checked at the receiving end for errors.
In 4 bit odd parity checker, the three bits X,Y,Z constitute the message and `P` is the parity bit. For odd parity bit `P` is generated, so as to make the total number of 1`s odd (including P). The three bit message and the parity bit are transmitted to their destination; they are applied to a parity checker circuit. An error occurs during transmission if the parity of the four bits received is even, since binary information transmitted was originally odd. The output "C" of the parity checker should be "1"when an error occurs i.e. when the number of 1's in the four input is even.



4 BIT odd Parity checker

Truth Table

Four bits received			Parity error check	
x	y	z	P	C
0	0	0	0	---------- 1
0	0	0	1	---------- 0
0	0	1	0	---------- 0
0	0	1	1	---------- 1
0	1	0	0	---------- 0
0	1	0	1	---------- 1
0	1	1	0	---------- 1
0	1	1	1	---------- 0
1	0	0	0	---------- 0
1	0	0	1	---------- 1
1	0	1	0	---------- 1
1	0	1	1	---------- 0
1	1	0	0	---------- 1
1	1	0	1	---------- 0
1	1	1	0	---------- 0
1	1	1	1	---------- 1

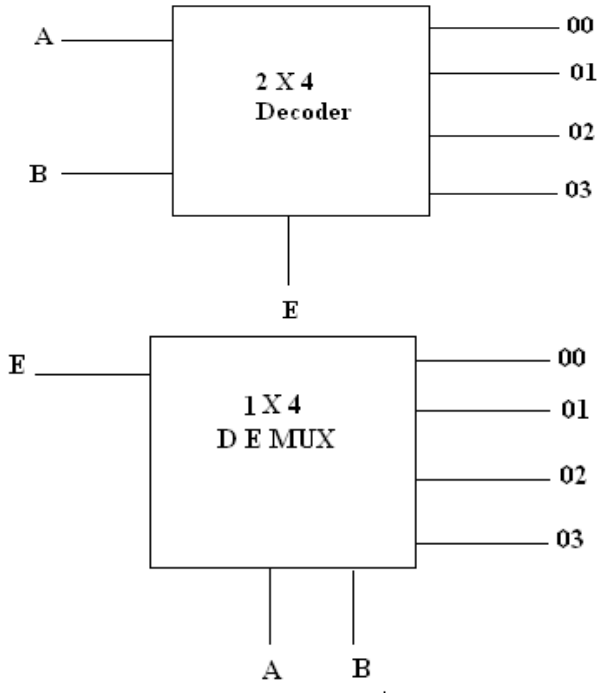
Q.49 What is a Decoder? Compare a decoder and a demultiplexer with suitable block diagrams. **(4)**

Ans:

Decoder:- It decodes the information. The decoders have n inputs 7 at the end maximum 2^n outputs because n bit no can decode max 2^n information , Now 1 enable input 'E ' is connected to the decoder. If it is high then only the circuit will be enabled and it will work as a decoder. If 'E' is low then the circuit will be disabled.

Demultiplexer has the same circuit as decoder but here e is taken as the single input line, the output lines are same as decoder (i.e max 2^n). The information at E will be transmitted to one of the output line and the output line will be selected by bit combination of n select lines.

Block diagrams of a decoder and a demultiplexer

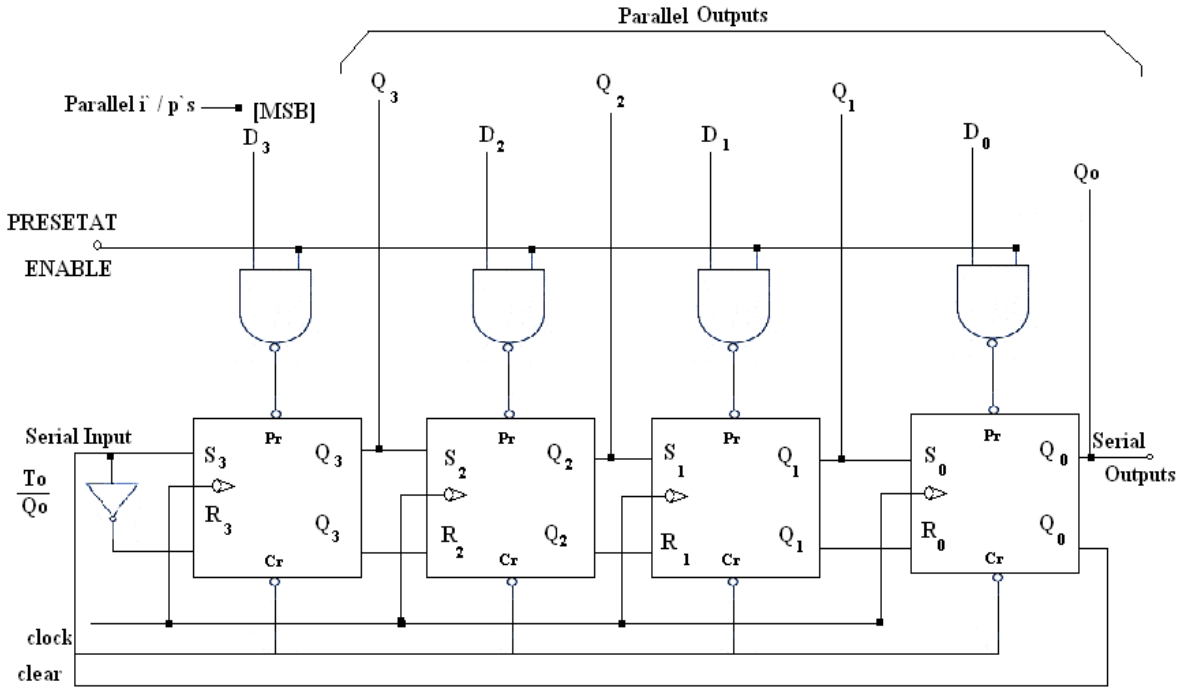


Q.50 Draw the logic diagram of 4-bit Twisted Ring counter and explain its operation with the help of timing diagram. **(6)**

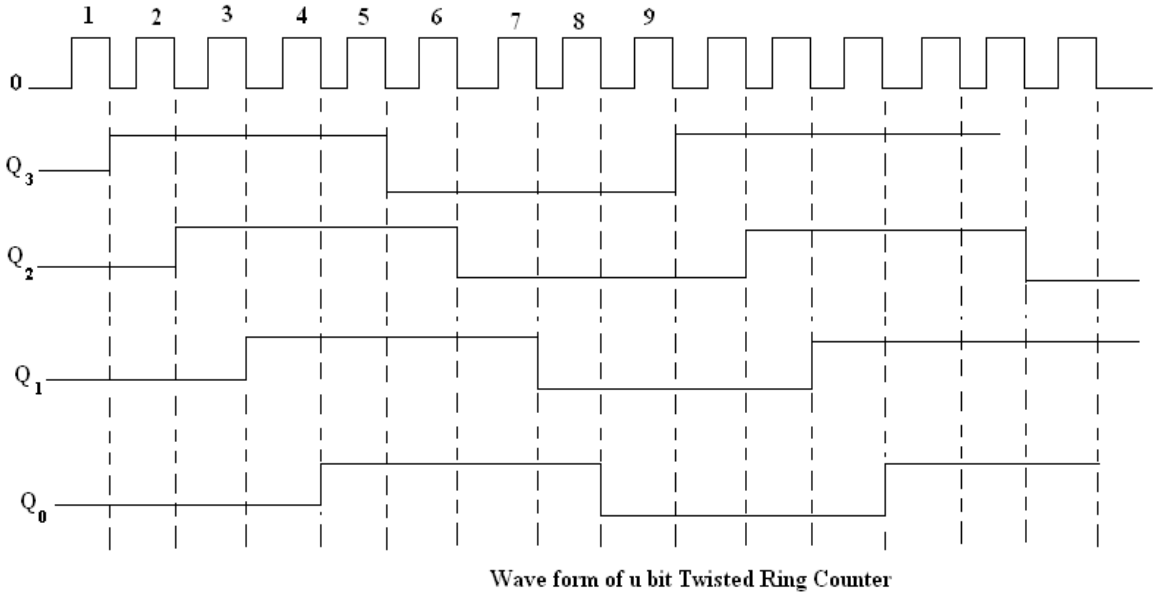
Ans:

Twisted ring counter (4 BIT) We know that shift registers can operate in 4 different modes that is SISO, SIPO, PISO and PIPO.

Following is the 4 BIT register which can operate in any of the mode. If $\overline{Q}_0$ is applied to the serial input , the resulting circuit is called twisted ring or Johnson Counter. If the clock pulse are applied after clearing the Flip Flops, square wave form is obtained at the Q output.

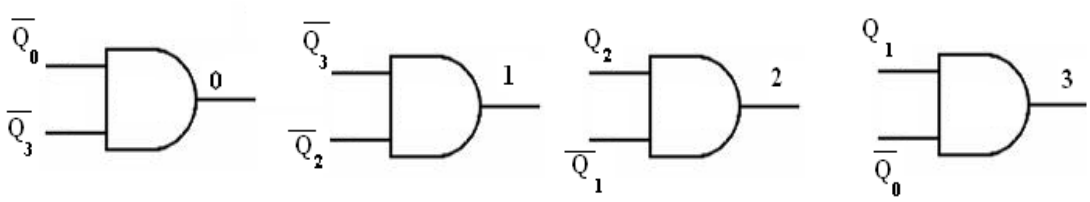


The logic diagram of 4-bit Twisted Ring counter



Wave form of u bit Twisted Ring Counter

For decoding the count, two input AND Gates are required Decoding logic for 4 stage twisted ring counter are



Q.51 Explain the following characteristics for digital IC's. **(8)**
(i) Propagation delay (ii) Power dissipation

Ans:

Propagation Delay:- The speed of operation of a digital IC is specified in terms of propagation delay time. The delay time is measured between the 50% voltage levels of input & output wave forms. There are two delay times.

a) t_{phl} = When the O/P goes from HIGH state to LOW state.
b) t_{plh} = When the O/P goes from Low state to HIGH state.

The propagation delay time of the logic gate is taken as the average of these two delay times.

Power Dissipation:- This is amount of power dissipated in an IC. It is determined by the current I_{CC} , that it draws from the V_{cc} supply and is given by V_{CC} X I_{CC}. This is specified in milliwatts. I_{cc} is the average value of I_{cc} (o) and I_{cc} (1)

Q.52 How will you form an 8 bit adder using 2 four bit adder IC's 7483? **(8)**

Ans:

IC 7483 is a 4 bit adder IC. It has two four bit data inputs and output carry, 4 bit data output carry. These two IC's should be connected in cascade, the first IC will add lower order bits and it generate sum and carry. This carry should be the input of second IC, The inputs of second IC will be the higher order bits of number A & B

Q.53 Distinguish between combinational logic circuits and sequential logic circuits. How are the design requirements of combinational circuits specified? **(7)**

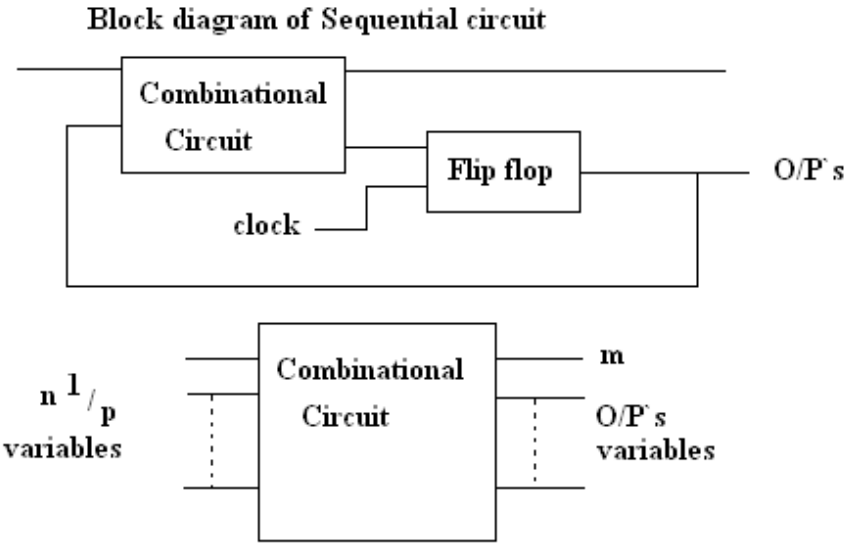
Ans:

Combinational logic circuits:-

(i) Outputs only depends upon present state of the input.
(ii) No memory element present or no feedback connection.

Sequential logic circuit:-

(i) Output not only depends on the present state of the input but also depend on the previous state of the output.
(ii) Memory element is present or a feedback connection is there.



Block diagram of Combinational circuit

Design Requirements of Combinational Logic:-

(i) From the specifications of circuit, we determine the no of inputs & outputs.

(ii) Derive the truth table which contains all possible combination of the inputs and corresponding outputs.

(iii) Minimize the output function using K-Map.

(iv) Draw the logic diagram.

Design Requirements of Sequential circuit:

(i) The circuit specifications translated into a state diagram.

(ii) The state diagram is then converted into state table.

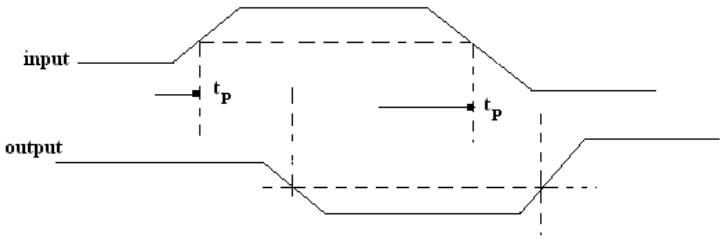
(iii) From state table, information for obtaining logic circuit diagram is obtained.

Q.54 What are the characteristics of digital ICs used to compute their performance? **(11)**

Ans:

Characteristics of Digital Integrated Circuits

1. **Speed of operation:** The Speed of a digital circuit is specified in terms of the propagation delay time. The input and output delay times can be shown as:



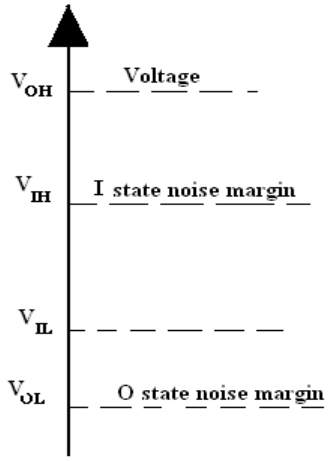
The delay times are measured between the 50 percent voltage levels of input and output wave forms. There are two delay times t_{phl}, when the O/P goes from the high state to low state and t_{phl}, when O/P goes from low state to high state.

2 **Power Dissipation:** This is the amount of power dissipated in an IC. It is determined by the current, I_{CC} that it draws from the V_{cc} supply and is given by V_{cc} X I_{cc}. I_{cc} is the av value of I_{CC} [O] and I_{cc} [1]. It is specified in mW.

3 **Figure of merit:** For digital IC, it is defined as the product of speed and power. It is specified in Pico joules [as ns x mw= pj]. A low value of speed power product is desirable.

4 **Fan Out:** This is the no of similar gates which can be driven by a gate. High fan out is advantageous, as it reduces the need for additional drivers to drive more gates.

5 **Noise Immunity:** Stray electric and magnetic fields induce unwanted voltages known as noise, on the connecting wires between logic circuits. This may cause the voltage at the I/P to a logic circuit to drop below V_{ih} or fuse above V_{il} and may produce undesired operation. The circuit`s ability to tolerate noise signals is referred to as the noise immunity.



6. **Operating Temperature:** The temperature range in which an IC functions properly must be known. The accepted temperature range for consimer IC` s are 0 to 70 degree C and for industrial applications [-55° C to +125° C for military applications].

Q.55 What is a digital multiplexer? Illustrate its functional diagram. Write the scheme of a 4-input multiplexer using basic gates (AND/OR/NOT) and explain its operation. **(8)**

Ans:

Multiplexer: MUX or data selector is a logic circuit selects binary information from one of many input and directs it to a single output line. Selection of the particular input line is controlled by a set of selection lines. Normally there are 2^n input lines and correspondingly n selection lines.

There are 4 inputs I_1 I_0 I_2 I_3 and two selection line S_0 and S_1. Depending upon the bit combination of S_0 and S_1 one of the input is transferred to the output. Basically there is a decoder circuit with one input for each bit of information and one OR gate connected to

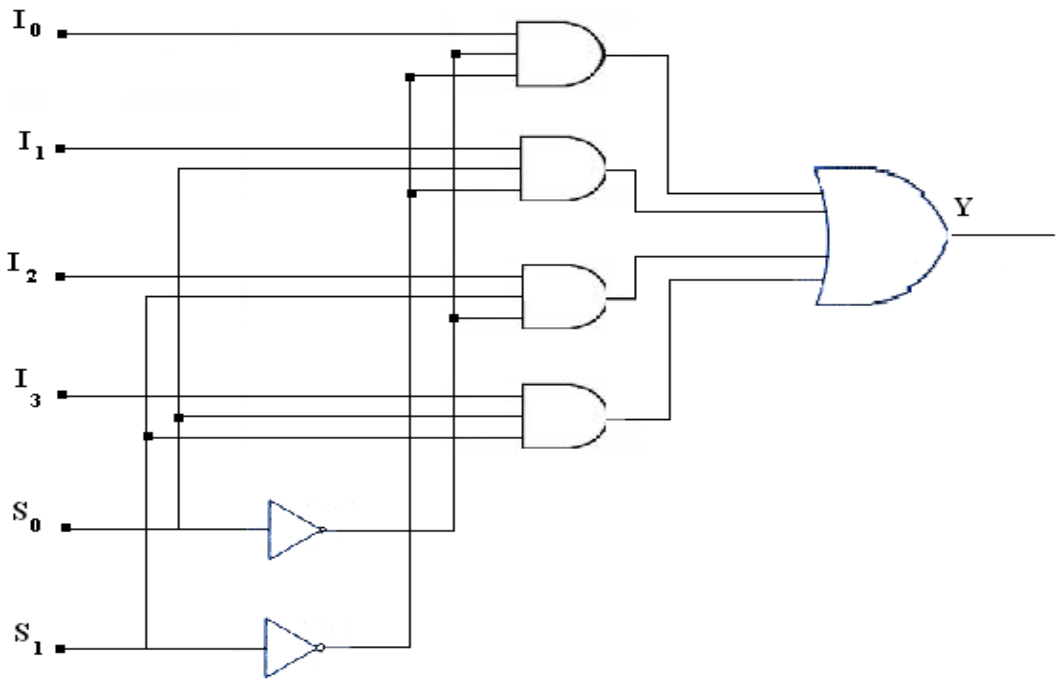
the output. If S_o, S_i = 00 ,then first AND gate will have the two inputs as one output will depend on I_0. At the same time outputs of all other AND gates are Zero.
The multiplexer is a combinational circuit which is one of the most widely used standard circuit in digital design. It has N select lines 2^N inputs and a single output.

Multiplexer:-

$$Y = \overline{S}_1\overline{S}_0 I_0 + \overline{S}_1 S_0 I_1 + S_1\overline{S}_0 I_2 + S_1 S_0 I_3$$

Truth table of 4x1 Mux

Select inputs		Output
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

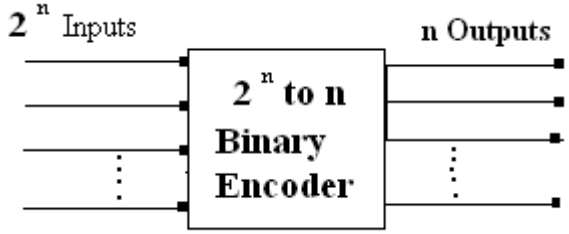


Circuit Diagram of 4 X 1 MUX using basic gate

Q.56 What is meant by a priority encoder? Name the 7400 series TTL chip which is a priority encoder. Write its truth table. Illustrate how it can be used as a decimal-to-BCD encoder. **(8)**

Ans:

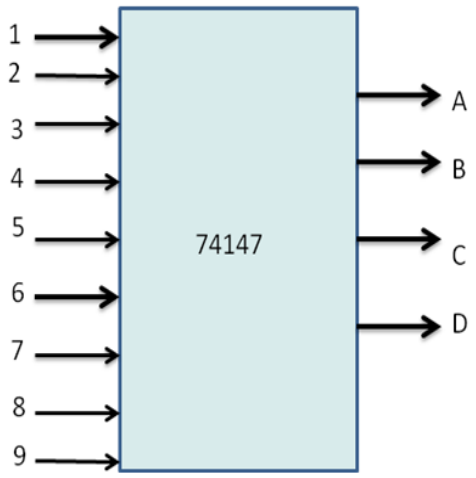
Priority encoder- An encoder is a combinational circuit that performs the inverse operation of a decoder. If a device output code has fewer bits than the input code has, the device is usually called an encoder. e.g. 2^n-to-n, priority encoders. The simplest encoder is a 2^n-to-n binary encoder, where it has only one of 2^n inputs = 1 and the output is the n-bit binary number corresponding to the active input.



One of the most commonly used input device for a digital system is a set of 10 switches, one for each numeral between 0 & 9. These switches generate 1 or 0 logic levels in response to turning them off or on. When a particular number is to be fed to the digital circuit in BCD code, the switch corresponding to that number is pressed. Available IC in 74 series is 74147 which is a priority encoder. This IC has active low inputs and outputs. The meaning of the word priority can be understood from the truth table. For example if 2 & 5 are low, the output will be corresponding to 5 which has a higher priority than 2 i.e. the highest numbered I/P has priority over lower numbered input's.

Truth table of 74147

Active low decimal input`s								**Active low BCD outputs**			
1	2	3	4	5	6	7	8	9	D	C	B	A
1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	0
X	0	1	1	1	1	1	1	1	1	1	0	1
X	X	0	1	1	1	1	1	1	1	1	0	0
X	X	X	0	1	1	1	1	1	1	0	1	1
X	X	X	X	0	1	1	1	1	1	0	1	0
X	X	X	X	1	0	1	1	1	1	0	0	1
X	X	X	X	1	1	0	1	1	1	0	0	0
X	X	X	X	X	X	X	0	1	0	1	1	1
X	X	X	X	X	X	X	X	0	0	1	1	0



Q.57 What is a flip-flop? Write the truth table for a clocked J-K flip-flop that is triggered by the positive-going edge of the clock signal. Explain the operation of this flip-flop for the following conditions.
Initially all inputs are zero and assume the 'Q' output to be 1. **(10)**

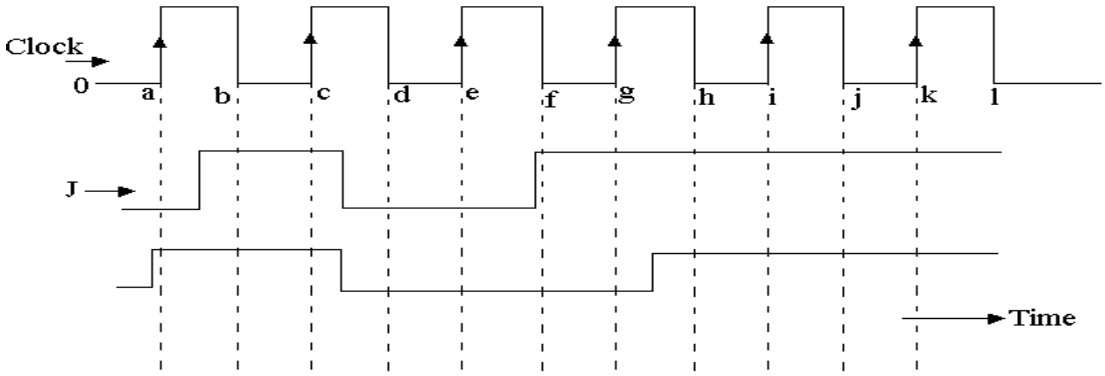


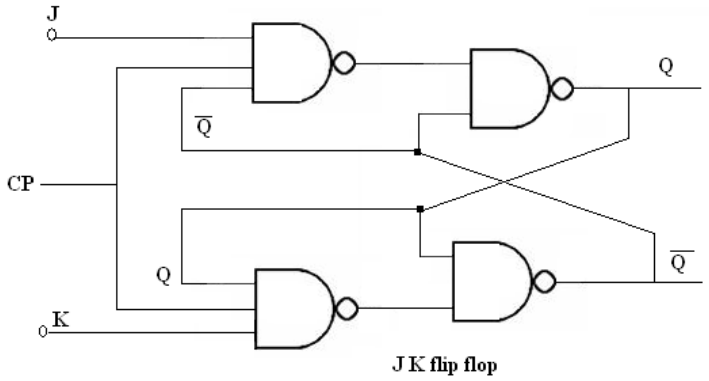
Fig.2

Ans:

Flip –flop is single bit memory cell. It stores single bit information in its true and compliment form. This is the fundamental block of any sequential circuit.

Truth table for clocked J K Flip- flop

clock	J	K	Q(t+1)
0	X	X	Q(t)
1 / 0 / 0 → 1	0	0	Q(t)
-do-	1	0	1
-do-	0	1	0
-do-	1	0	Q`(t)



J K flip flop

Let Q = 0 initially

At 1st (1) edge Q = 0

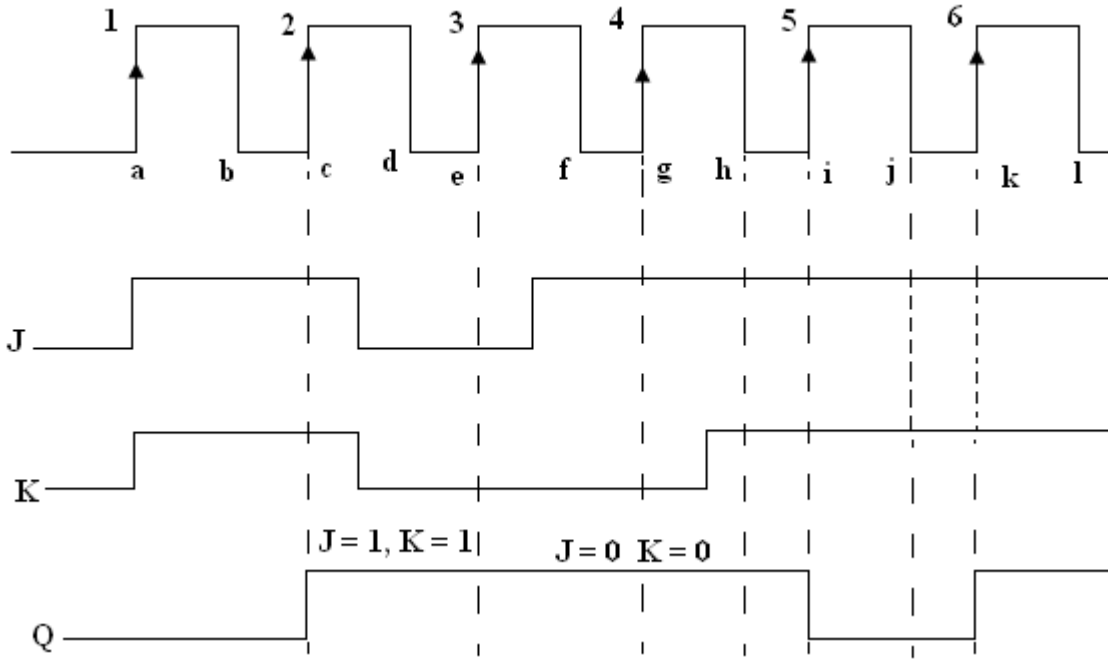
At 2 nd (2) edge J = 1, K = 1 ⇨ Q = 1

At 3 rd (3) edge J = 0, K = 0 ⇨ Q = 1

At 4 rth (4) edge J = 1, K = 0 ⇨ Q = 1

At 5 th (5) edge J = 1, K = 1 ⇨ Q = 0

At 6 th (6) edge J = 1, K = 1 ⇨ Q = 1



Q.58 How is it possible to make a modulo 2^n counter using N-flipflops? Name the two types of such counters. **(4)**

Ans:

Module 2^n counter counts total 2^n distinguishable states we know that n-bit can represent 2^n unique combinations for eg. Mod-8 counter will count total 8 states and as $8=(2^3)$ each state will have combination of 3 bits.

Two types of such counters are:

- Mod 8 counter
- Mod 16 counter

Q.59 In applications where the required memory capacity cannot be satisfied by a single available memory IC chip, what should the designer do to meet this requirement? **(10)**

Ans:

If the single memory chip can not be specified the required memory capacity then the designer should do the followings.

(1) Find out the no of single chip required to full fill the total capacity by

$$\text{No of chip} = \frac{\text{Required capacity}}{\text{Available capacity}}$$

(2) There are two type of expression

(i) Increasing memory location or words

(ii) Increasing word size, i.e. no of bits in each word.

(3) In case (i) the number will be of same as the address lines of available chip. The difference of the address lines of the capacity 7 availability will give the size of the decoder and the output of the decoder will decode among the chips.

In case (ii) address line data lines will be common to all chips because all chips at the same location collectively make a single word.

Q.60 Explain the operation of 8:1 multiplexer. **(8)**

Ans:

There are 8 Inputs & 1 Output and three select lines S2, S1, S0. Any one of the inputs will be selected & transmitted to the output depending upon the combination of the select lines, for e.g. If $S_2S_1S_0 = 001$ then information present on I1 line will be transmitted to the output.

Q.61 What is race around condition? How it can be avoided? **(8)**

Ans:

Race Around Condition:-

Jn	Kn	Q(n+1) output
0	0	Q(n)
1	0	1
0	1	0
1	1	Q(n)'

In JK flip-flop, When J=k=1 then output will be the complement of the previous state. Suppose the output Q_n is 0 and clock pulse is high. After the time interval Δt equal to the propagation delay through two NAND gates the output will change to the Q_{n+1}=1 (if J=K=1). Now we have=K=1 and Q=1 and after another Δt interval the output, Q will change to 0 from 1. Hence after every Δt duration of the output will flip between 0 and 1. At the end of the clock pulse the value of Q is uncertain because the value of Δt is not known exactly. This situation is known as race around condition .

The race around condition can be avoided if

1 Duration of clock pulse being high is small as compare to the delay of the gates.
This is difficult because of very small propagation delay in IC's.

2 A master slave JK flipflop is used. In this 2 SR flip-flops are there. The feedback from the output of the second to the input of the first flip-flop. Positive clock pulses are applied to the first clock pulse and clock pulse are inverted at the second flip-flop when clk=1 first flip-flop is enabled and second is disabled clk'=0.

Q.62 Draw the circuit diagram of Asynchronous decade counter and explain its working. **(8)**

Ans:

To design a decade asynchronous counter first we draw the circuit for MOD 16 asynchronous counter which counts from 0 to 15 using four flip-flop (JK or T flipflop). It should count from 0 to 9 and then come to 0. The first state to be skipped is 1010 (10)here Q_3 and Q_1 are 1 and Q_2 and Q_0 are 0 if we take Q_3 and Q_1 and applied these to a NAND gate then the output of the NAND gate will be low only where Q_3 and Q_1 are high. This signal can be used to asynchronously clear all flipflops to make the counting state 0000. In this way MOD 16 counter will be restricted to count 10 state that is from 0 to 9.

Q.63 Explain the following for an ADC

(i) Input stage. (ii) Resolution.
(iii) Accuracy. (iv) Quantization error. **(8)**

Ans:

(i) Input Stage- In A-D Converter at the input stage, analog voltage can have any value in a range but the digital output can have only 2^N discrete values for an n bit A-D converter.

(ii) Resolution- This is the smallest possible change in input voltage as the fraction of percentage of the full scale output range.

(iii)Accuracy-The accuracy of D/A converter is the difference between actual output voltage and the expected output voltage in D/A converter.

(iv)Quantization error- An analog voltage is in the range of 0 to 1V and for 3 bit output, the size of each interval is S=1/8.Each interval is assigned a 3 bit binary value .We observe that the

whole range of voltage in an interval is represented by only one digital value .This error is referred to an quantization error which is because of process of quantization.

Q.64 Give the details of excess 3 code and gray code using four binary digits. Compare the two codes. **(8)**

Ans:

Binary no	Excess3	Gray code
0 0 0 0	0 0 1 1	0 0 0 0
0 0 0 1	0 1 0 0	0 0 0 1
0 0 1 0	0 1 0 1	0 0 1 1
0 0 1 1	0 1 1 0	0 0 1 0
0 1 0 0	0 1 1 1	0 1 1 0
0 1 0 1	1 0 0 0	0 1 1 1
0 1 1 0	1 0 0 1	0 1 0 1
0 1 1 1	1 0 1 0	0 1 0 0
1 0 0 0	1 0 1 1	1 1 0 0
1 0 0 1	1 1 0 0	1 1 0 1
1 0 1 0		1 1 1 1
1 0 1 1		1 1 1 0
1 1 0 0		1 0 1 0
1 1 0 1		1 0 1 1
1 1 1 0		1 0 0 1
1 1 1 1		1 0 0 0

Excess 3 Code

1. It is another from of BCD code. Each decimal digit is coded in 4 bit binary code.
2.The code for each decimal digit is obtained by adding decimal 3 to the natural BCD code of the digit.
3.The code is obtained by adding 3 to the decimal no
4.Self complementing code-useful in subtraction.

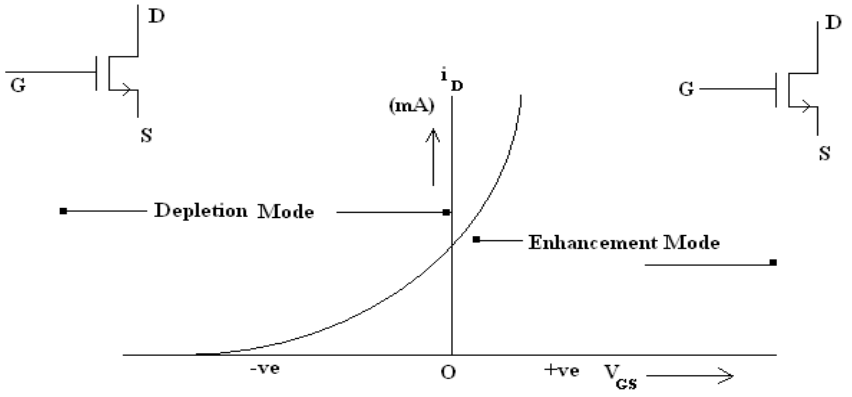
Gray Code

1. Very useful code. Also called reflected code.
2. Each gray code differs from the preceding and succeeding codes by a single bit.
3. Used in shaft encoders.

Q.65 Distinguish between enhancement mode and depletion mode metal oxide semiconductor field effect transistors giving their characteristics. **(6)**

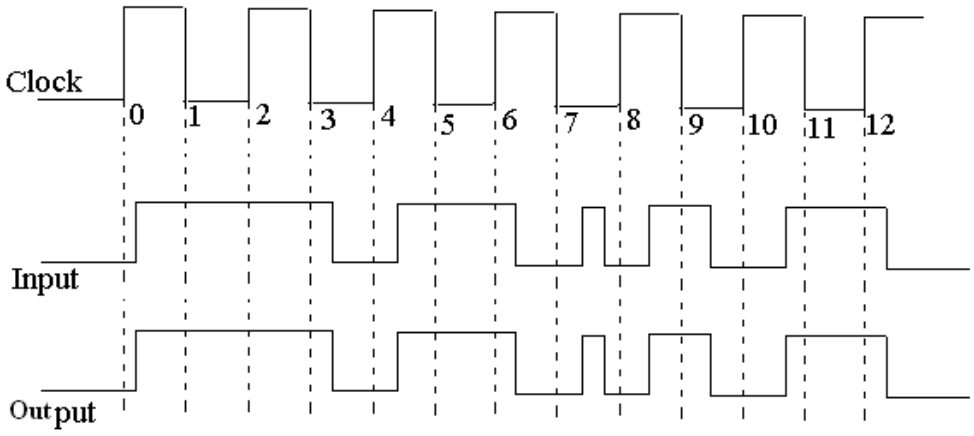
Ans:

E Mode MOSFET	Depletion Mode MOSFET
01. No channel exists between rain and source at $V_{GS} = 0$	01. Channel exists at $V_{GS} = 0$ [in fabrication n type impurity is diffused between two n+ regions
02. Threshold voltage is positive for nMOS Device.	02. Threshold voltage is negative for nMOS Device.
03. No current flows for negative V_{GS} [nMOS]	03. Current flows even for negative V_{GS}



Q.66 The clock and the input waveforms shown below are applied to the D input of a positive edge triggered D flipflop. Sketch the output waveforms. **(6)**

Ans:



As it is D Flip Flop at the positive edge ,output will be same as the input.

Q.67 What are the specifications/ characteristics used by the manufacturers to describe a digital to analog converter. Explain each one briefly. **(8)**

Ans:

The characteristics of D/A converter are

(i)Resolution:- This is the smaller possible change in output voltage as a function of percentage of full scale output voltage.

(ii) Linearity:-In a D/A converter equal increments in the numerical significance of the digital circuit the input-output relationship is not linear.

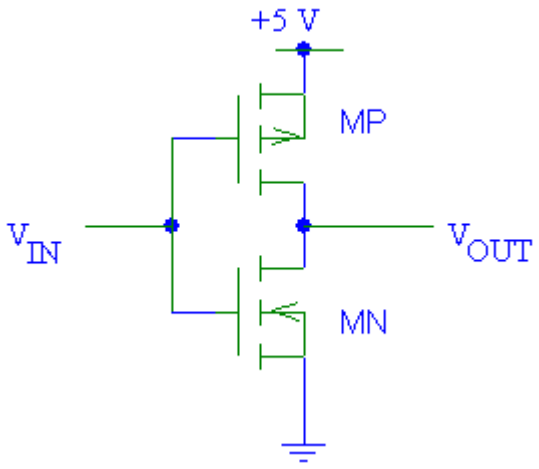
(iii)The accuracy of D/A converter is a measure of the difference between the actual output voltage and the expected output voltage.

(iv) Settling time:-when the digital input to a D/A Converter changes the analog output voltage does not change absolutely. Because of the presence of switches, active devices, stray capacitances and inductances associated with passive circuit components. The transient appears in the output voltages and oscillations may also occur the time required for the analog output to settle within +- ½ LSB of the final value after a change in the digital input is known as settling time.

Q.68 Describe CMOS inverter and state advantages of CMOS. **(8)**

Ans:

CMOS inverters (Complementary MOSFET Inverters) are some of the most widely used and adaptable MOSFET inverters used in chip design. They operate with very little power loss and at relatively high speed. Furthermore, the CMOS inverter has good logic buffer characteristics, in that, its noise margins in both low and high states are large.A CMOS inverter contains a PMOS and a NMOS transistor connected at the drain and gate terminals, a supply voltage V_{DD} at the PMOS source terminal, and a ground connected at the NMOS source terminal, were V_{IN} is connected to the gate terminals and V_{OUT} is connected to the drain terminals.(See diagram). It is important to notice that the CMOS does not contain any resistors, which makes it more power efficient that a regular resistor-MOSFET inverter. As the voltage at the input of the CMOS device varies between 0 and 5 volts, the state of the NMOS and PMOS varies accordingly. If we model each transistor as a simple switch activated by V_{IN}, the inverter's operations can be seen very easily:



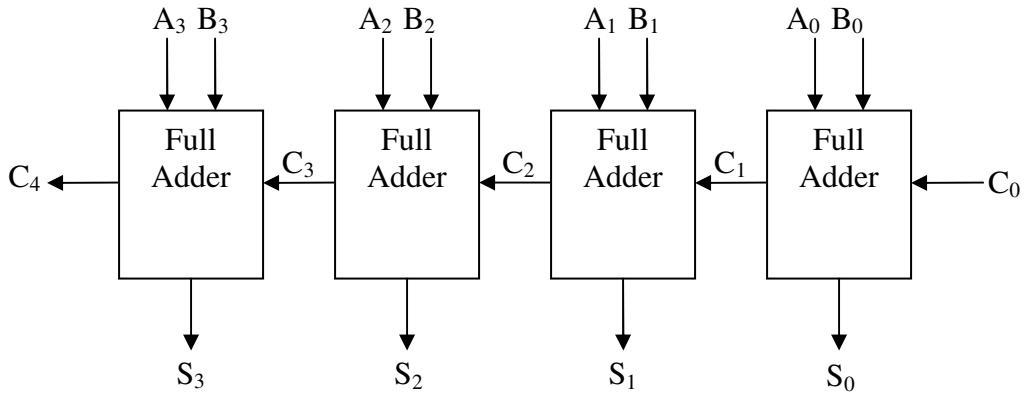
Following are the advantages of CMOS:

- Both n-channel & p-channel devices are fabricated on the same substrate.
- Low power dissipation, so more efficiency.
- Good noise immunity.
- High packing density.

Q.69 What is parallel adder? Draw and explain block diagram for 4 bit parallel adder. **(8)**

Ans:

By using full adder circuit, any two bits can be added with third input as carry. If numbers of bits are more than one, then full adder circuits are cascaded. Addend & Augend bits are applied simultaneously at inputs to the full adders. Carry generated in the lower significant stage is transferred to the next higher stage so that it can be added there.



Q.70 What is parity generator and checker? Describe five bit even parity checker. **(8)**

Ans:

When a digital signal is transmitted, it may not be received correctly by the receiver. At the receiving end it may or may not be possible to detect the error. To overcome this problem, an extra bit is attached to the n-bit code word to make the number of bits (n+1) in such a way so as to make the number of ones in the resulting (n+1) bit code even or odd. Then it will be an error detecting code. So for detection of error this extra bit is known as parity bit. Parity term is used to specify the number of ones in a word as odd or even. A logic circuit that checks the parity of a binary word is called as parity checker. Similarly a logic circuit that generates an additional bit to make the digital word of desired parity (even or odd) is known as parity generator.

Five bit even parity checker:

EX-OR gates are used for checking the parity as they produce output 1, when the input has an odd number of 1's. Therefore an even parity input to an EX-OR gate produces a low output.

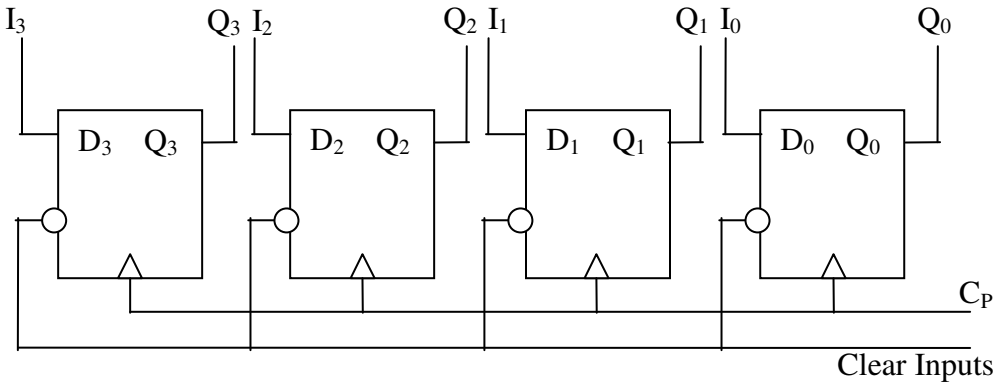
Truth table

W	X	Y	Z	P	C
0	0	0	0	0	1
0	0	0	0	1	1
0	0	0	1	0	1
0	0	0	1	1	0
0	0	1	0	0	1
0	0	1	0	1	0
0	0	1	1	0	0
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	0	1	0
0	1	0	1	0	0
0	1	0	1	1	1
0	1	1	0	0	0
0	1	1	0	1	1
0	1	1	1	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	0	0	1	0
1	0	0	1	0	0
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	0	1	1
1	0	1	1	0	1
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	0
1	1	1	0	0	1
1	1	1	0	1	0
1	1	1	1	0	0
1	1	1	1	1	1

Q.71 Describe the operation of parallel in parallel out (PIPO) shift register. **(8)**

Ans:

Parallel In Parallel Out



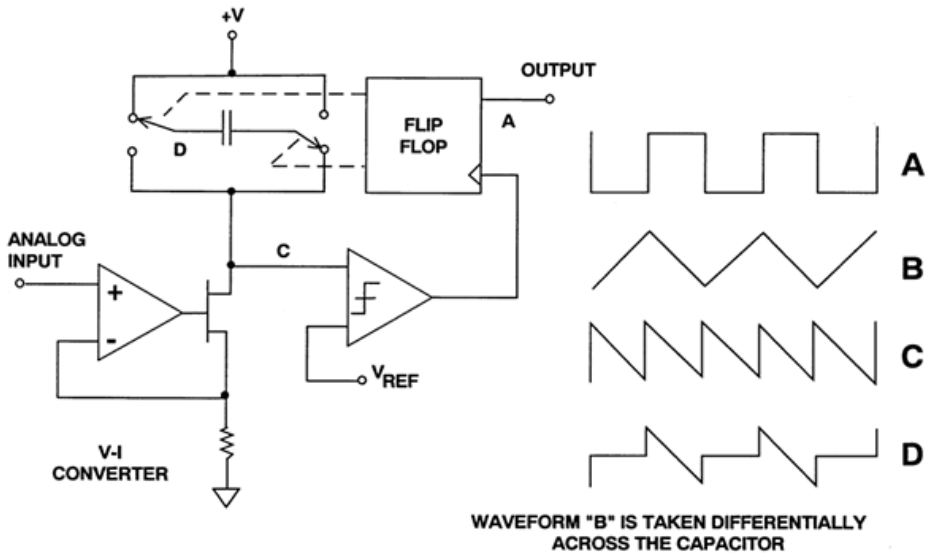
As the name suggests, in parallel in parallel out (PIPO), inputs are given in parallel, and outputs are also taken in parallel fashion. For synchronization same clock pulse is connected to all flip-flops. Thus any state change will take place simultaneously. Clear inputs are also connected to all flip-flops. So that the register can be cleared if required.

Q.72 Describe the operation of voltage to frequency ADC. **(8)**

Ans:

A voltage-to-frequency converter (VFC) is an oscillator whose frequency is linearly proportional to a control voltage. The VFC/counter ADC is monotonic and free of missing codes, integrates noise, and can consume very little power.

The current-steering multivibrator VFC is actually a current to-frequency converter rather than a VFC, but, as shown in Figure below, practical circuits invariably contain a voltage to-current converter at the input. The principle of operation is evident: the current discharges the capacitor until a threshold is reached, and when the capacitor terminals are reversed, the half cycle repeats itself. The waveform across the capacitor is a linear triangular wave, but the waveform on either terminal with respect to ground is the more complex waveform shown.



A_3	A_2	A_1	A_0	d_6	d_5	d_4	d_3	d_2	d_1	d_0		x
-	-	-	-	-	-	-	-	-	-	-	---	-
0	0	0	-	-	-	-	-	-	-	0		0
0	0	0	-	-	-	-	-	-	-	1		1
0	0	1	-	-	-	-	-	-	0	-		0
0	0	1	-	-	-	-	-	-	1	-		1
0	1	0	-	-	-	-	-	0	-	-		0
0	1	0	-	-	-	-	-	1	-	-		1
0	1	1	-	-	-	-	0	-	-	-		0
0	1	1	-	-	-	-	1	-	-	-		1
1	0	0	-	-	-	0	-	-	-	-		0
1	0	0	-	-	-	1	-	-	-	-		1
1	0	1	-	-	0	-	-	-	-	-		0
1	0	1	-	-	1	-	-	-	-	-		1
1	1	0	-	0	-	-	-	-	-	-		0
1	1	0	-	1	-	-	-	-	-	-		1

Q.73 Draw and explain the function of dual slope analogue to digital converter. Derive the equations used. **(8)**

Ans.

Dual slope A to D converter: It has 4 major blocks.

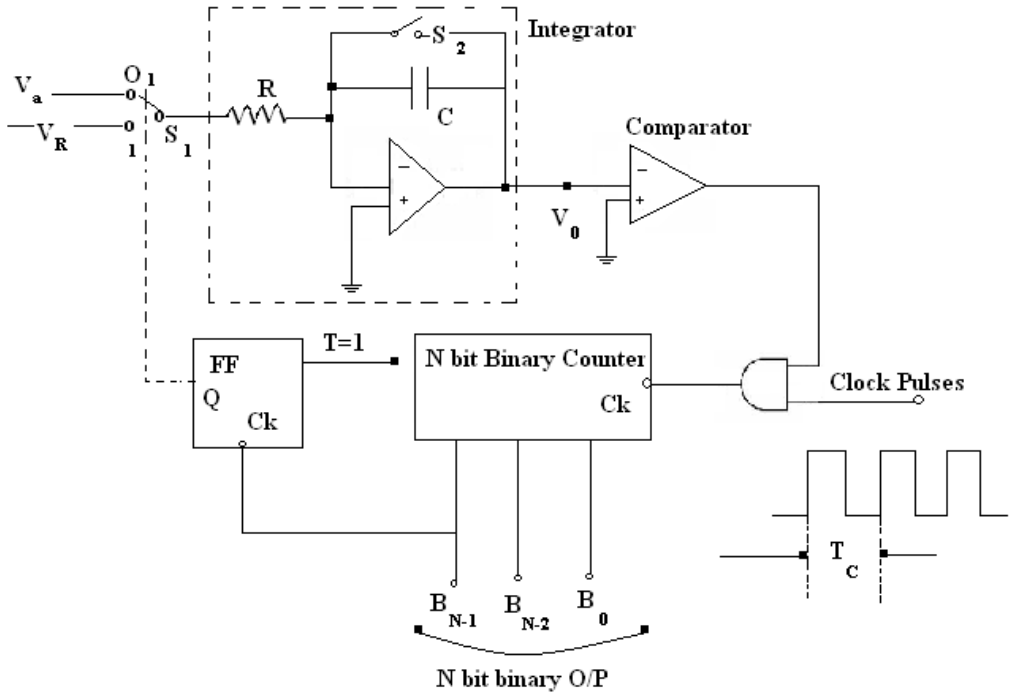
1. An integrator
2. A Comparator
3. A binary counter
4. A switch driver

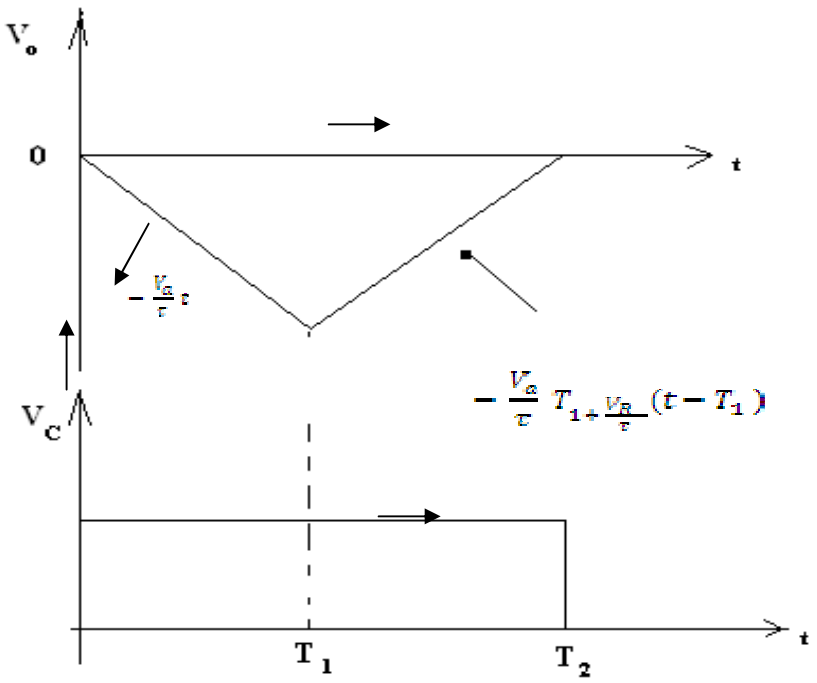
The conversion process at T=0 with switch S1 in position 0. This connects the analogue voltage Va to the input of the integrator. The output of the integrator will be

$$V_0 = -\frac{1}{\tau}\int_0^t V_a\, dt = -\left[\frac{V_a}{\tau}\right]t$$

This results in high V_c. This enables the AND Gate and the clock pulse reaches the ck input of the counter, which was initially clear. The counter counts from 00……00 to 11…..11 when

2^n -1 clock pulses are applied. At the next clock pulse 2^n the counter is cleared and Q becomes 1. This controls the state of S_1 which now moves to position 1 at T_1, thereby connecting $-V_R$ to the input of the integrator. The output of the integrator now starts to move in the positive direction. The counter continues to count until V_0 is less than 0. As soon as V_0 goes positive at T_2, V_C goes LOW disabling the AND Gate.





Wave form of dual slope A/D convertor

The time T1 is given by

$T_1 = 2^N T_C$ where T1 is time period of clock pulse.

When the switch S1 is in position 1, the output voltage of the integrator is given by

$$V_0 = -\frac{V_a}{\tau} T_1 + \frac{V_R}{\tau}(t - T_1)$$

$V_0 = 0$ at $t = T_2$

Therefore, $T_2 - T_1 = \frac{V_a}{V_R} T_1 = \frac{V_a}{V_R} 2^N T_C$

Let the count recorded in the counter be n at T_2 therefore $T_2 - T_1 = n T_C = \frac{V_a}{V_R} 2^N T_C$

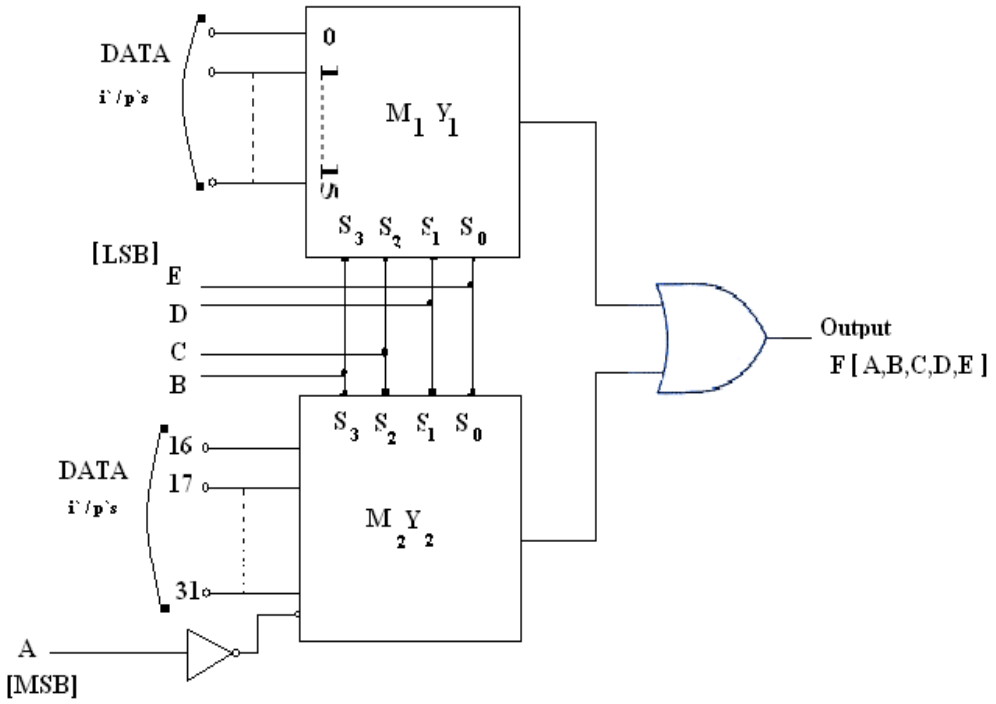
which gives $n = \frac{V_a}{V_R} 2^N$

Q.74 What is a Multiplexer Tree? Why is it needed? Draw the block diagram of a 32:1 Multiplexer Tree and explain how input is directed to the output in this system.**(10)**

Ans

Multiplexer Tree: The largest available MUX IC is 16 to 1. To meet the larger input needs there should be a provision to expand it. This can be achieved with the help of Strobe Inputs and so MUX trees are designed.

One of the possible method is shown for 32 to 1 MUX, by using two 16 to 1 MUX and OR Gate.

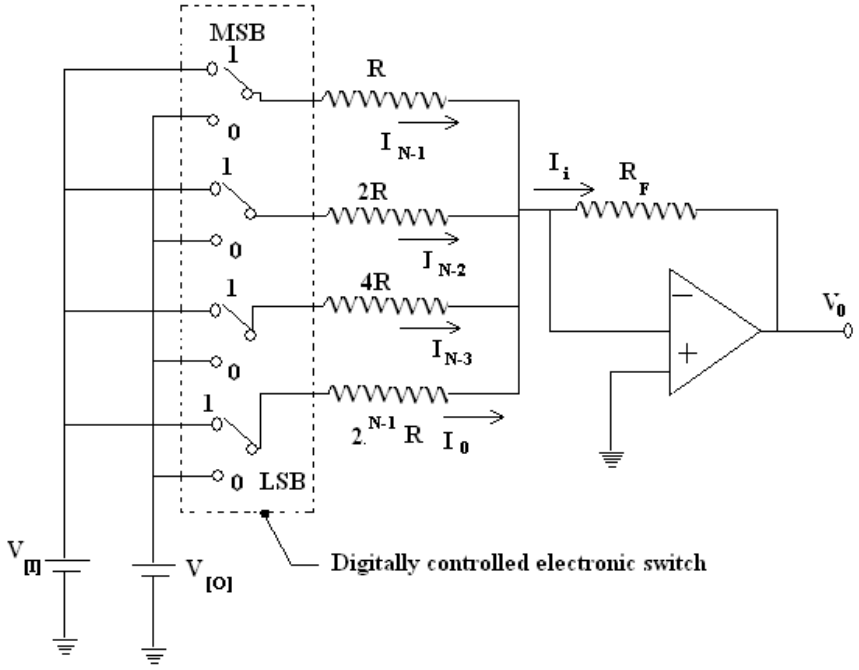


There are two 16 to 1 MUX M_1 and M_2 having data inputs 0…..15 and 16…..31 respectively. The selection lines are S_3 S_2 S_1 S_0 , which are able to select one input among 16 inputs. Now the strobe pin is used as fifth selection line that is if it is 0 than one input among the upper MUX is selected and if A = 1, than one among the data input of lower MUX is selected. The output of both the MUX are O Red.

Q. 75 With the help of a neat diagram, explain the working of a weighted-resistor D/A converter. **(9)**

Ans

Weighted Register D/A



Converter:

N Bit digital input is applied to a register network through electronic switch. This electronic switch produces current I at MSB (corresponding to Logic 1), I/2 at the next lower significant position. The total current produced will be proportional to digital input. This current can be converted to corresponding voltage by using an op-ampere. This circuit is referred to as weighted register converter since the resistance values are weighted in accordance with the binary weights.

The current I_i is given by

$I_i = I_{N-1} + I_{N-2} + \ldots\ldots\ldots + I_0$ where $I_{N-1} = V_{N-1}/R$, $I_{N-2} = V_{N-2}/2R$, $I_{N-3} = V_{N-3}/4R$

also $V_N = V(1)$ if bn = 1, V(O) if bn = 0

For straight binary inputs V(0) = 0 and V(1) = - V_R and the output voltage is given by

$$V_0 = -[-V_R]\left[\frac{R_F}{R}\,b_{n-1} + \frac{R_F}{2R}\,b_{n-1} + \frac{R_F}{2^2R}\,b_{n-1} + \text{\_\_\_\_\_} + \frac{R_F}{2^{n-1}R}\,b_0\right]$$

Q. 76 Briefly explain the following:

(i) Binary number system.

(ii) Signed binary numbers **(7)**

Ans

(i) Binary Number System

The number of system with base or Radix two is known as the Binary Number System. To represent the number, 0 & 1 are used. These are known as bits. It is a positional system that is every place carries specific weight. As the base is two, the coefficients can take only two value i.e. 0 & 1.

$$(N)_b = d_{n-1}\, d_{n-2} \ldots\ldots\ldots\ldots\ldots d_0 * d_{-1}\, d_{-2} \ldots\ldots\ldots\ldots\ldots\ldots\ldots d_{-m}$$

integer portion Radix Point Fraction

b = 2 (Radix)

d_{n-1} = Most significant bit

d_{-m} = Least significant bit

& $.0 \leq (d_i \text{ or } d_{-f}) \leq b_{-1}$

(ii) Signed Binary Numbers: In decimal number system positive numbers are denoted by (+) sign and negative numbers are denoted by –ve sign Digital circuits understand only the language of 0`s and 1`s. Thus normally an additional bit is used for sign and it is placed at the most significant position.

1. A `O` is used for +ve nos. and 1 is for –ve numbers. For example an eight bit signed number 00000100 represents +4 and 10000100 represents (-4). This representation is known as sign magnitude number. There are three different ways by which signed numbers are presented.

2. **One's compliment representation:** In this system the +ve numbers are represented by their Binary equivalent with a 0 placed at most significant position to represent the –ve numbers compliment is taken and than a `1` is placed as MSB to represent the –ve sign. For example + 7 = $(0111)_2$

$$-7 = (1000)_2$$

3. **Two's Compliment presentation**: If 1 is added number is known as I`s compliment of the binary No. For example 2`s compliment representation of 0101 is 1011. Since 0101 represents $(+5)_{10}$ therefore 1011 represents $(-5)_{10}$ in two's compliment representation.

Q.77 What is chattering as applied to mechanical switches used in digital systems and why do they occur? What is its effect on the functioning of a sequential circuit? **(6)**

Ans

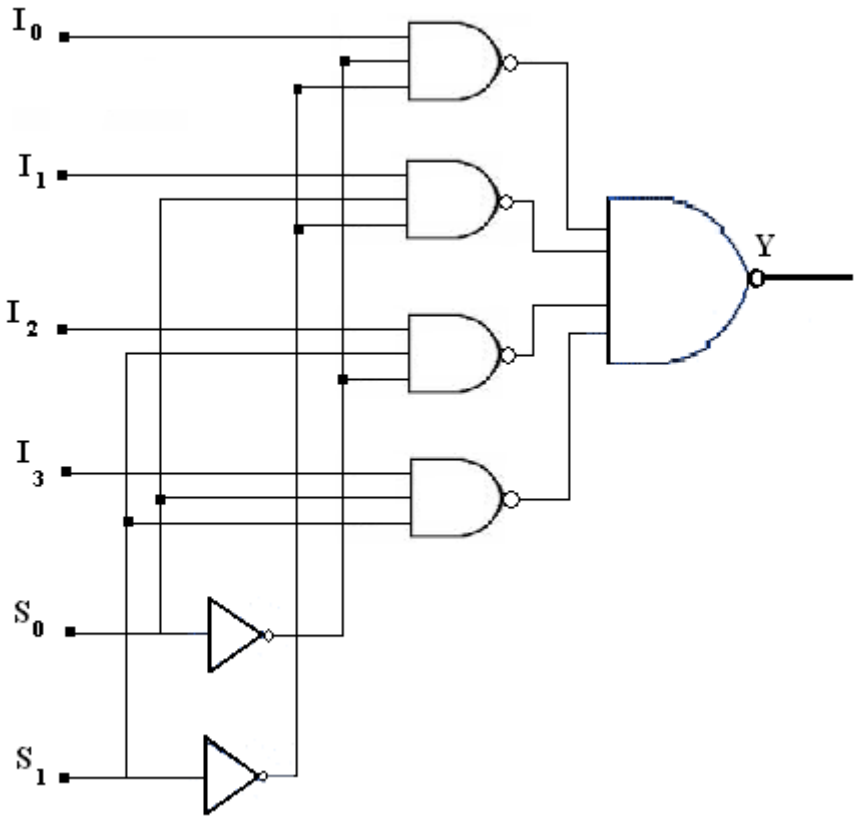
Chattering:

Mechanical switches are employed in digital systems as input devices by which digital information (0 or 1) is entered into the system. When the arm of the switch is thrown from one position to another, it chatters or bounces several times before finally coming to the root in the position of contact. This is known as bouncing or chattering. This bounce is result of the spring loaded impact of the switch through contact and the pole

contacts. In a sequential circuit, if a 1 is to be entered through a switch then the switch is thrown to the corresponding position, as soon as it is thrown to this position, the output is 1 but the output oscillates between 0 & 1 for some times due to make and break (bouncing) of the switch at the point of contact before coming to rest. This changes the output of the sequential circuit and creates difficulties in the operation of the system. This problem is eliminated by using bounce – free elimination switches

Q.78 Design a 4 : 1 multiplexer with strobe input using NAND gates. **(5)**

Ans



Design of 4 : 1 multiplexer with strobe input using NAND gates.

Q.79 Explain the operation of octal to binary encoder. **(8)**

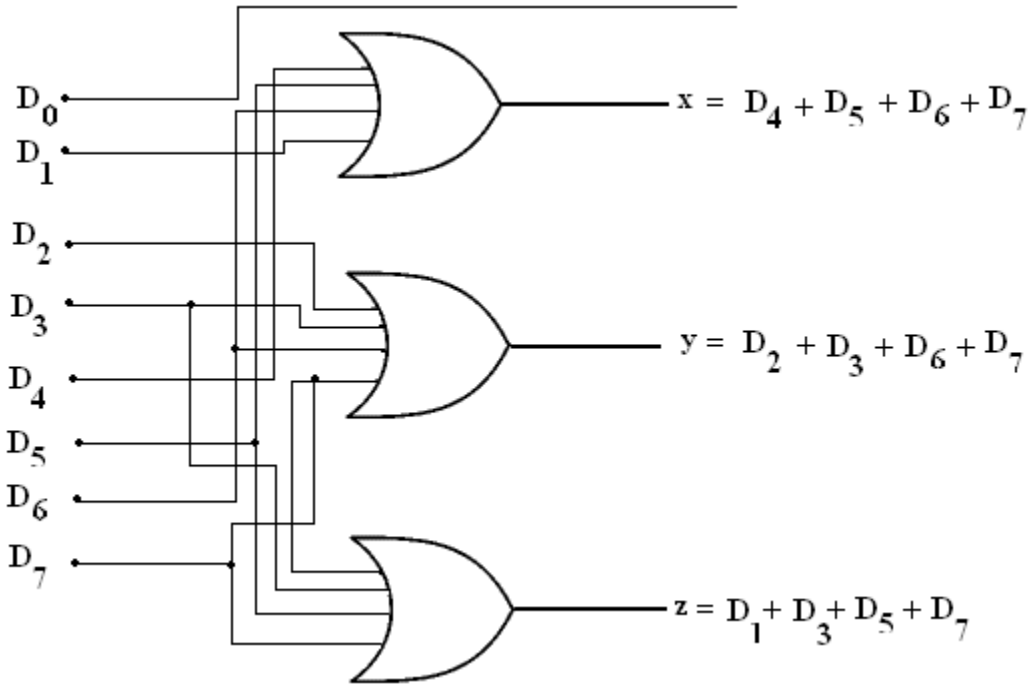
Ans

Octal to binary encoder consists of eight inputs, one for each of eight digits and three outputs that generate the corresponding binary number. For example: low order output bit Z is if the input octal digit is odd.

Here D_O input is not connected to any O R gate; the binary output must be all zeroes in this case and all 0's output is also obtained, when all inputs are zeroes. This discrepancy can be resolved by providing one more output to indicate the fact that all inputs are not zeroes.

Truth table

Inputs								Outputs		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1



Logic diagram of octal to binary encoder

TYPICAL QUESTIONS & ANSWERS

OBJECTIVE TYPE QUESTIONS

Each question carries 2 marks.

Choose the correct or best alternative in the following:

Q.1 Which of the following is not an scripting language ?

(A) HTML **(B)** XML
(C) Postscript **(D)** Javascript

Ans: C Postscript

Q.2 Which of the following is a platform free language

(A) Fortran **(B)** Assembly
(C) C **(D)** Java

Ans: D Java

Q.3 A digital signature is

(A) scanned signature **(B)** signature in binary form
(C) encrypting information **(D)** handwritten signature

Ans: C encrypting information

Q.4 Mechanism to protect private networks from outside attack is

(A) Firewall **(B)** Antivirus
(C) Digital signature **(D)** Formatting

Ans: A Firewall

Q.5 A computer system that permits multiple users to run programs at same time

(A) Real time system **(B)** Multi programming system
(C) Time sharing system **(D)** Multi tasking system

Ans: D Multi tasking system

Q.6 A computer communication technology that provides a way to interconnect multiple computer across short distance is

(A) LAN
(B) MAN
(C) WAN
(D) Wireless network

Ans: A LAN

Q.7 Telnet is a service that runs

(A) Television on net
(B) Remote program
(C) Cable TV network
(D) Telenext

Ans: B Remote program

Q.8. A device that forwards data packet from one network to another is called a

(A) Bridge
(B) Switch
(C) Hub
(D) Gateway

Ans: B Switch

Q.9 Which of the following is the fastest media of data transfer

(A) Co-axial Cable
(B) Untwisted Wire
(C) Telephone Lines
(D) Fibre Optic

Ans: D Fiber Optic.

Q.10 Tool that is used to transfer data/files among computers on the Internet

(A) FTP
(B) Archie
(C) TCP
(D) Gopher

Ans: C TCP

Q.11 HTML is a

(A) Programming Language
(B) Scripting Language
(C) Web Browser
(D) Network Protocol

Ans: B Scripting Language

Q.12 Secret-key encryption is also known as

(A) Asymmetric encryption **(B)** Symmetric encryption
(C) Secret-encryption **(D)** Private encryption

Ans: D Private encryption

Q.13 The concept of electronic cash is to execute payment by

(A) Credit Card **(B)** ATM Card
(C) Using computers over network **(D)** Cheque

Ans: C Using computers over network.

Q.14 SMTP is a

(A) Networking Protocol
(B) Protocol used for transferring message between end user & Mail Server
(C) Protocol used for smart card message interchange
(D) Encryption Standard

Ans: B Protocol used for transferring message between end user & Mail Server.

Q.15 Digital Signature is

(A) Scanned Signature on Computer
(B) Code number of the sender.
(C) Public Key Encryption.
(D) Software to recognize signature.

Ans: D Software to recognize signature

Q.16 Telnet is a

(A) Network of Telephones **(B)** Television Network
(C) Remote Login **(D)** Remote Login.

Ans: C Remote Login.

Q.17 The internet is

(A) Network of networks **(B)** Web site.
(C) Host **(D)** Server

Ans: A Network of networks

Q.18 An e-business that allows consumer to name their own price for products and services is following which e-business model?

(A) B2B
(B) B2G
(C) C2C
(D) C2B

Ans: D C2B

Q.19 Kerberos is an encryption-based system that uses

(A) Secret key encryption
(B) Public key encryption
(C) Private key encryption
(D) Data key encryption

Ans: A Secret key encryption.

Q.19 The method(s) of payment for online consumers are

(A) Electronic cash
(B) Credit/debit
(C) Electronic checks
(D) All of the above

Ans: D All of the Above.

Q.20 DNS is

(A) The distributed hierarchical naming system
(B) The vertical naming system
(C) The horizontal naming system
(D) The client server system

Ans: C The horizontal naming system.

Q.21 A firewall is

(A) An established network performance reference point.
(B) Software or hardware used to isolate a private network from a public network.
(C) A virus that infects macros.
(D) A predefined encryption key used to encrypt and decrypt data transmissions.

Ans: B Software or hardware used to isolate a private network from a public network.

Q.22 A router

(A) Screens incoming information.
(B) Distributes information between networks
(C) Clears all viruses from a computer system
(D) Is a work virus.

Ans: B Distributes information between networks

Q.23 LDAP stands for

(A) Light weight Data Access Protocol.
(B) Light weight Directory Access Protocol.
(C) Large Data Access Protocol.
(D) Large Directory Access Protocol.

Ans: B -> Light weight Directory Access Protocol.

Q.24 E-Commerce is not suitable for

(A) Sale/Purchase of expensive jewellery and antiques.
(B) Sale/Purchase of mobile phones.
(C) Sale/Purchase of branded clothes.
(D) Online job searching.

Ans: D Online job searching

Q.25 Amazon.com comes under the following model

(A) B2B **(B)** B2C
(C) C2C **(D)** C2B

Ans: B B2C

Q.26 Hubs are present in the network

(A) to diagnose line failures, measure and manage traffic flow and simplify re configuring of LANs.
(B) to interconnect the LAN with WANs.
(C) to interconnect the WANs with WANs.
(D) to interconnect the WANs with LANs.

Ans: B to interconnect the LAN with WANs.

Q.27 Firewalls operate by

(A) The pre-purchase phase.
(B) isolating Intranet from Extranet.
(C) Screening packets to/from the Network and provide controllable filtering of network traffic.
(D) None of the above.

Ans: C Screening packets to/from the Network and provide controllable filtering of network traffic.

Q.28 The mercantile process model consists of the following pahase(s):

(A) The pre-purchase phase.
(B) Purchase consummation phase.
(C) Post-purchase Interaction phase.
(D) All of the above.

Ans: D All of the Above.

Each Question carries 1 mark.

State which of the statement is true and which are false. Write 'T' or 'F' in the answer book.

Q.29 One disadvantage to online buyers is lack of trust when dealing with unfamiliar sellers

TRUE	**FALSE**

Ans: T

Lack of trust is there obviously as buyer cannot physically see the seller. As Trust is a qualitative function, it develops with time & goodwill.

Q.30 Multimedia contents are not important to e-business applications.

TRUE	**FALSE**

Ans: F Rather Multimedia contents are most important as visual graphics & animations form the core of e-business applications.

Q.31 While making payment using electronic check, credit and debit cards, the server authenticates the customers and verifies with the bank that funds are adequate before purchase.

TRUE	**FALSE**

Ans: T The server authenticates that the customer has enough available balance to carry out transaction.

Q.32 Trojan horse is a program that performs not only a desired task but also includes unexpected malicious functions

TRUE	FALSE

Ans: T Trojan horse is a program that performs not only a desired task but also includes unexpected malicious functions.

Q.33 Home Banking is not an example of consumer oriented applications

TRUE	FALSE

Ans: F Home Banking is one of the most important examples of consumer-oriented applications as using Home Banking business is carried out just by sitting at home.

Q.34 Electronic checks are another form of electronic tokens.

TRUE	FALSE

Ans: T Electronic checks is a part of e-token. When giving e-token some authorizing co. verifies that the no. is active i.e. its run by Certifying authority (CA),Verifying Authority(VA) & Digital Signature(DS).

Code: T-16 **Subject: E-COMMERCE**

PRT – II, VOL - I

DESCRIPTIVES

Q.1 a. Define e-commerce? What are the benefits of using e-commerce? **(7)**

Ans:

The term 'electronic commerce' has evolved from electronic shopping, to imply all aspects of business and market processes enabled by the Internet and World Wide Web technologies.

According to Philip Kotler :
E-commerce can be defined as a general term for buying and selling process that is supported by electronic means.

Electronic commerce, also known as e-business, a term for all kinds of business that are established electronically especially over the Internet. This includes both electronic sale (internet shops) and B2B transactions, i.e. business between two companies. It is any on-line transaction of buying and selling where business is done via Electronic Data Interchange (EDI). E-Commerce can be defined from different perspectives – **1.** Communications perspective, **2.** Business process perspective, **3.** Service perspective and **4.** Online perspective.

Basic Benefits of E-Commerce
The major benefits are increasing sales and decreasing costs. The other benefits are as follows:

1. Increased accessibility to customers

i)Allows people to carry out operations without barriers of time i.e. 24 hours a day, seven days a week.
ii) To reach out to global consumers easily and is also cost effective.
iii) It helps business to reach out new markets.
iv) Consumers and suppliers can be directly approached over the Internet.
v) Acquisition of new consumers over the internet is considerably cheaper..

2. Convenience of making comparisons:
E-commerce helps consumers to make comparisons while shopping . Automated online shopping assistants called hopbots score are available to find deals on anything from flowers to perfume

3. Increased Profitability
i) The direct cost to sale for an order taken from an web site is lower as compared to traditional means. Moreover processing errors are virtually eliminated in e-selling besides being faster and more convenient to visitor.
ii) It provides the solution by decimating the costs, which are incurred.

4. Innovation:
E-commerce enables business organization to create new products or services.

5.Improvement in consumer service:
There is a direct benefit in improvement of consumer service. High levels of customer satisfaction generate increased sales and increased profits.

6. Tangible advantages:
From the buyer's perspective e-commerce provides a lot of tangible advantages:
i. Reduction in buyers sorting out time
ii Better buyer decisions.
iii.Less time spent in resolving invoice and order discrepencies.
iv. Increased opportunities for buying alternative products.

7. Stratergic Benefits:
It helps to reduce delivery time,labour cost and also the cost incurred in the following areas:
i) Document prepration.
ii) Error detection and correction.
iii) Reconciliation.
iv) Mail Prepration.
v) Telephone calling.
vi) Data Entry.
vii)Overtime.
viii)Supervision Expenses.

b. What do you mean by the followed types of e-commerce **(7)**
i) B2B (Business to Business) **ii)** B2C (Business to Customer

Ans:

(i) B2B - Business to Business

It is a mode of conducting business between two or more companies over the Internet, rather than more traditional modes such as telephone, mail, and face to face.

In the past EDI was conducted on a direct link of some form between the two businesses where as today the most popular connection is the Internet.

The two businesses pass information electronically to each other. B2B e-commerce currently makes up about 94% of all e-commerce transactions.

Some of the advantages of B2B are:

i) Improved customer satisfaction

ii) Improved inventory system
iii) Easy and cost effective marketing
iv) Coordination between manufacturers, distributors and dealers.
v) Better management of business

(ii) B2C -Business to Consumer

This is where the consumer accesses the system of the supplier. It is still a two-way function but is usually done solely through the Internet.

In B2C e-commerce companies sell goods to consumers online in a dynamic environment. Each transaction under B2C represents an individual buying online.

Some examples:- Conducting individual stock trades, a co. offering lots of books for sale on its web site.

An example of B2C model is Amul.com which sells Amul branded products online.

Q.2 **b.** Discuss in brief the following terms
i) Telnet **ii)** http (hypertext transfer protocol) **iii)** Remote login **(3 x 3)**

Ans:

i) Telnet:
It is a terminal emulation program for TCP/IP networks such as the Internet. The Telnet program runs on your computer and connects your PC to a server on the network. One can then enter commands through the Telnet program and they will be executed as if you were entering them directly on the server console. This enables one to control the server and communicate with other servers on the network. To start a Telnet session, one must log in to a server by entering a valid username and password. Telnet is a common way to remotely control Web servers.

(ii) Http :
Hyper Text Transfer Protocol; The WWW protocol that performs the request and retrieve functions of a server. Commonly seen as the first part of a website address.
It is the communication protocol used to connect to servers on the World Wide Web. The primary function of HTTP is to establish a connection with a Web server and transmit HTML pages to the user's browser.

(iii) Remote Login
A login that allows a user terminal to connect to a host computer via a network or direct telecommunications link, and to interact with that host computer as if the user terminal were directly connected to that host computer.

Gives the same functionality of telnet, with the added functionality of not requiring a password from trusted clients, which can also create security concerns.
Protocols such as TELNET and RLOGIN were developed for terminal users to use their terminals as if they were directly connected to a remote system. UNIX systems, with their predominately terminal-oriented interface, still make heavy use of these protocols.

Q.3 **a.** Explain briefly how firewalls protect network. **(7)**

Ans

A firewall is simply a program or hardware device that filters the information coming through the Internet connection into your private network or computer system. If an incoming packet of information is flagged by the filters, it is not allowed through the network.

A firewall gives a company tremendous control over how people use the network.

Firewalls use one or more of three methods to control traffic flowing in and out of the network:

- **Packet filtering** - Packets (small chunks of data) are analyzed against a set of **filters**. Packets that make it through the filters are sent to the requesting system and all others are discarded.
- **Proxy service** - Information from the Internet is retrieved by the firewall and then sent to the requesting system and vice versa
- **Stateful inspection** - A newer method that doesn't examine the contents of each packet but instead compares certain key parts of the packet to a database of trusted information.

b. Explain the use of SSL to secure the network. **(7)**

Ans:

SSL (***S**ecure **S**ockets **L**ayer),* is a protocol developed by Netscape for transmitting private documents via the Internet. SSL works by using a private key to encrypt data that's transferred over the SSL connection. Both Netscape Navigator and Internet Explorer support SSL, and many Web sites use the protocol to obtain confidential user information, such as credit card numbers.

The SSL standard is not a single protocol, but rather a set of accepted data transfer routines that are designed to protect the integrity of transmitted messages.

SSL relies on certificates - digital identification cards - and keys.

Two types of keys are used as ciphers to encrypt and decrypt data. Private keys are issued to entities and are never given out. Public keys are given out freely. Both keys are necessary for authentication routines. Data encrypted with the public key cannot be decrypted with the same key. The private key must be used.

Q.4 a. Discuss the process of data mining? What are the advantages of data mining. **(5)**

Ans:

The process of data mining consists of three stages:

(1)The initial exploration

(2) Model building or pattern identification with validation/verification,

(3) Deployment (i.e., the application of the model to new data in order to generate predictions).

Stage 1: Exploration. This stage usually starts with data preparation which may involve cleaning data, data transformations, selecting subsets of records and - in case of data sets with large numbers of variables ("fields") - performing some preliminary feature selection operations to bring the number of variables to a manageable range (depending on the statistical methods which are being considered).

Stage 2: Model building and validation. This stage involves considering various models and choosing the best one based on their predictive performance (i.e., explaining the variability in question and producing stable results across samples).

There are a variety of techniques developed to achieve that goal - many of which are based on so-called "competitive evaluation of models," that is, applying different models to the same data set and then comparing their performance to choose the best.

Stage 3: Deployment. That final stage involves using the model selected as best in the previous stage and applying it to new data in order to generate predictions or estimates of the expected outcome.

Advantages offered by Data Mining :

1. Facilitates discovery of knowledge from large, massive data sets.
2. Can be used within different application areas via. Fraud detection, risk assessment , market analysis and customer segmentation etc.
3. Technique to develop effective CRM applications.
4. Returns best results when used within data warehouse environment.
5. Used extensively in telecommunications, retail and banking & finance industry.

b. Explain the use of following devices used in networking **(3 x 3)**
i) Digital Switches **ii)** Routers **iii)** Ramps

Ans:

(i) Digital switch

It is a device that handles digital signals generated at or passed through a telephone company's central office and forwards them across the company's backbone network.

It receives the digital signal from the office's channels banks that have been converted from users' analog signals and switches them with other incoming signals out to the wide area network.

(ii) Router

A router is used to route (transfer) data between two or more similar networks.
It determines the next network point to which a data packet should be forwarded. The router is connected to at least two networks and determines which way to send each data packet based on its current understanding of the state of the networks it is connected to. Routers create or maintain a table of available routes and use this information to determine the best route for a given data packet.

(iii) Ramps

A network planning method that makes the most efficient use of manpower, materials and cash resources among several projects going on simultaneously.

Q.5 a. Explain network security. What are the types of security features used in client server types of network? **(7)**

Ans:

Network security means the protection of networks and their services from unauthorized access, modification, destruction or disclosure. It provides for assurance that a network performs its critical functions correctly and there are no harmful side effects.

Security features used in Client-Server types of network are as follows :

i) Digital Signatures

ii) Encryption / Decryption

iii) Secure Socket Layer (SSL)

iv) Firewalls.

b. What is Public Key Cryptography? What are its advantages and disadvantages? **(7)**

Ans:

Public-key cryptography is a form of modern cryptography which allows users to communicate safely without previously agreeing on a shared secret key

There are a number of significant practical difficulties in this approach to distributing keys.

Public-key cryptography was invented to address these drawbacks — with public-key cryptography, users can communicate with safety over an insecure channel without having to agree upon a key beforehand.

Public-key algorithms typically use a pair of two related keys — one key is private and must be kept secret, while the other is made public and can be widely distributed; it should not be possible to deduce one key of a pair given the other.

Advantages

(i) Increased security and convenience

(ii) Electronic records may be authenticated by affixing digital signatures

Disadvantages

Used to encrypt a secret key which is used to encrypt the bulk of a file or message. Such a protocol is called a digital envelope

Public-key cryptography is not necessary and secret-key cryptography alone is sufficient. This includes environments where secure secret-key agreement can take place, for example by users meeting in private.

Q.6 a. Explain the application of E-Commerce in any one of the following fields
i) Home banking **ii)** Home Entertainment **iii)** Home Shopping **(3 x 3)**

Ans:

i) Home Banking:

E-commerce is used in Home Banking as one call or one click.
Online banking (Internet banking) is a term used for performing transactions, payments etc. over the internet through a bank's secure website.
This can be very useful, especially for banking outside bank hours (which tend to be very short) and banking from anywhere where internet access is available.

ii) Home Entertainment:
E-commerce has led to HOME ENTERTAINMENT. The video aspect usually involves a large-screen and/or high definition television or a projection system.

"Home cinema" has become something of a buzzword. Technically, a home cinema could be as basic as a simple arrangement of a Television, VCR, and a set of speakers. It is therefore difficult to specify exactly what distinguishes a "home cinema" from a "television and stereo".

iii) Home Shopping
TV broadcast of goods for purchase, sent directly to a viewer . This online shopping is available due to e-commerce.

b. Differentiate between data warehousing and data mining. **(5)**

Ans:

Data warehouse means

- Subject oriented
- Integrated
- Time variant
- Nonvolatile collection of data for management's decisions.

It contains the bedrock data that forms the single source for all DSS processing. Data warehouses contain historical data and detailed data, they are eternally large.

Different types of data warehouses.
- Financial Data Warehouses
- Insurance Data Warehouses
- Human Resources Data Warehouses
- Global Data Warehouses

Data Mining is an analytic process designed to explore data and then to validate the findings by applying the detected patterns to new subsets of data. The ultimate goal of data mining is prediction. The process of data mining consists of three stages:

1. the initial exploration
2. model building
3. deployment

The difference between data warehousing and data mining is that data warehousing refers to the data storage whereas data mining is a process of extracting useful knowledge from the data warehouse. Different techniques are used for implementation of these two concepts.

Q.7 a. What is on electronic payment system? What are its types and advantages? **(7)**

Ans:

Electronic payment systems are alternative cash credit payment methods using various electronic technologies to pay for products and services in electronic commerce.
Types: The most Internet payment method for B2C is credit cards.

The concern for customers is security while sending credit card information including name, card number and expiry date through Internet.

At present most of the companies use SSL (Secured Socket layer) protocol to provide security and privacy.
Visa and MasterCard have jointly developed a more secure protocol, called SET (Secure Electronic Transmission)

Typical Electronic payment system for EC- Electronic credit card, EFT, debit card, stored-value card, and e-check.

Advantages

It provides good security schemes.

Four essential security requirements for safe e-payments are Authentication, Encryption, Integrity, and Nonrepudiation.

SET(Secure Electronic Transmission) which is theoretically a perfect protocol.

b. Write notes on following
i) E-Cash **ii)** Electronic Cheques. **(7)**

Ans:

i) E-Cash:

E-cash is cash represented by two models. One is the on-line form of e-cash (introduced by DigiCash) which allows for the completion of all types of internet transactions. The other form is off-line; essentially a digitially encoded card that could be used for many of the same transactions as cash.
The primary function of e-cash is to facilitate transactions on the Internet. Many of these transactions may be small in size and would not be cost efficient through other payment mediums such as credit cards

These types of payments, turning the Internet into a transaction oriented forum, require mediums that are easy, cheap (from a merchants perspective), private (see Privacy), and secure (see Security). Electronic Cash is the natural solution, and the companies that are pioneering these services claim that the products will meet the stated criteria.

ii) Electronic Cheques

Another mechanism for Internet payment is electronic cheques. With electronic cheques, the payer (either an individual consumer or a business) instructs his financial institution to pay a specific amount to another part, the payee.
A cheque in the electronic form means a cheque which contains the exact mirror image of a paper cheque, and is generated, written and signed by a secure system ensuring the minimum safety standards with the use of digital signature (with or without biometrics signature) and asymmetric crypto system

Q.8 a. Define EDI. Explain the layered architecture of EDI. **(7)**

Ans:

Electronic Data Interchange (EDI) is used by organizations for transactions that occur on regular basis to a pre-determined format. It is one of the electronic commerce technologies.

It is used in number of trade sectors for inter-organization, regular, repeat transactions. These systems require EDI standards, EDI software, an EDI network and trading community.

Layered Architecture of EDI:

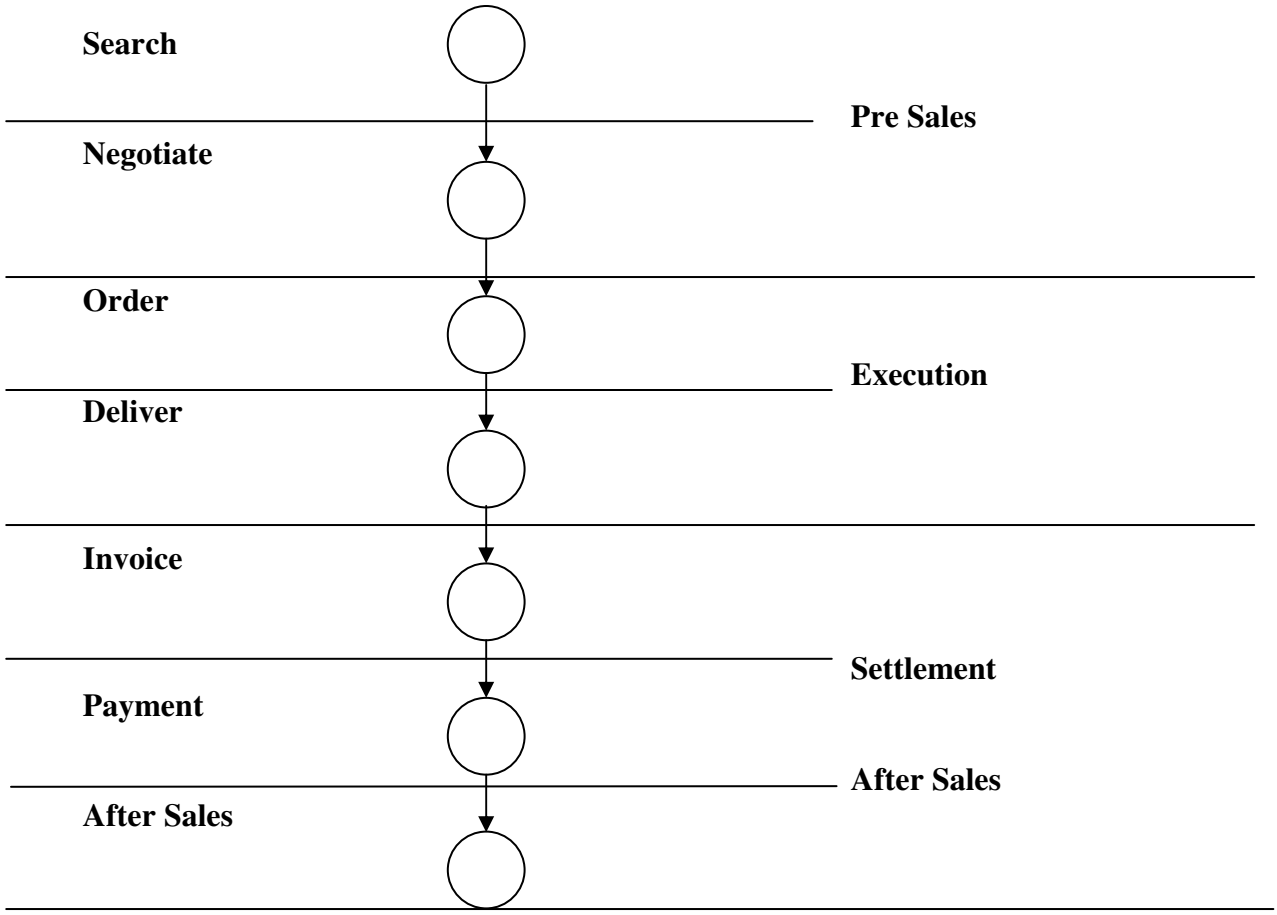
EDI is most commonly applied in the Execution and settlement phases of the trade cycle. In execution of a simple trade exchange, the customers' orders can be sent by EDI and the delivery notification from the supplier can be electronic.

For settlement the supplier can use EDI to send the invoice and the customer can finish the cycle with an electronic funds transfer via the bank and an EDI payment notification to the supplier.

This whole cycle may be complex and other electronic messages can be included.

EDI can be used for Pre-Sales transactions; there have been EDI messages for transactions such as contract but are not wisely implemented.

EDI can be used for After -Sales transactions but only if they were in a standardized format and frequent enough to justify system costs, transactions such as dealer claiming payment for warrantee work could be possible application.



b. What are the applications of EDI in business? **(7)**

Ans:

A. Organistions that use EDI

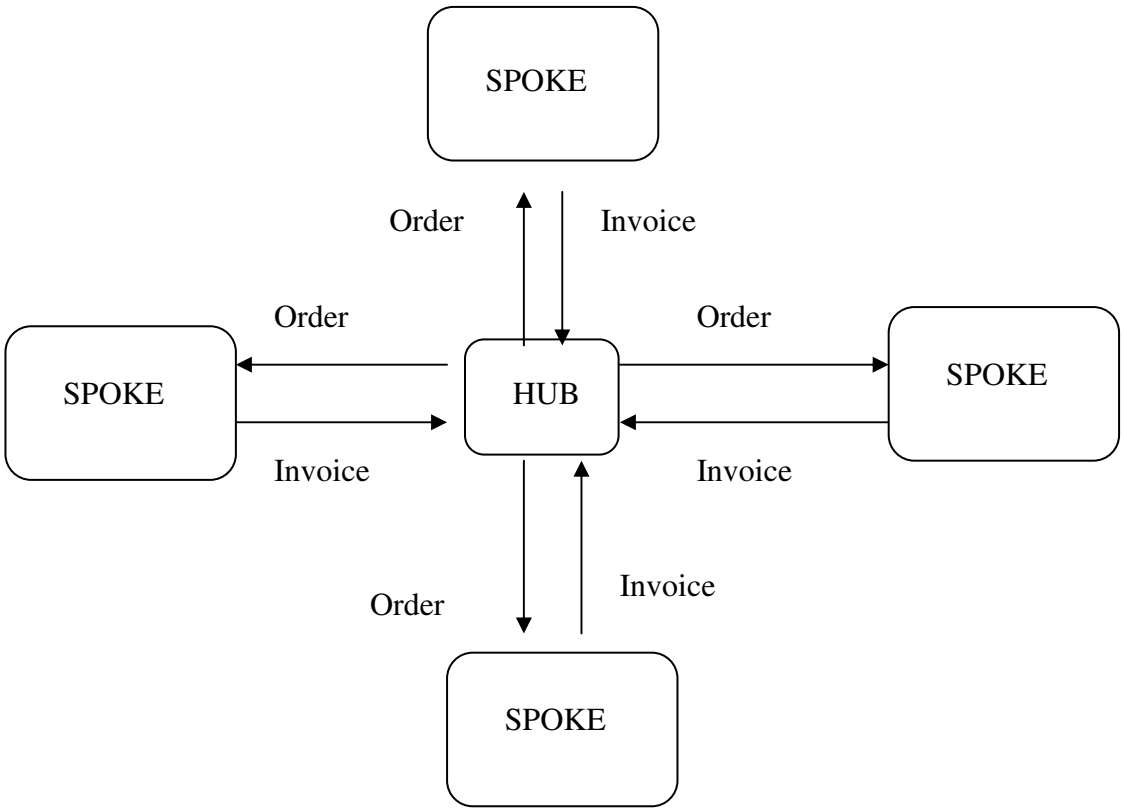
Extensive users of EDI include:
BHS- is a UK and European retailer dealing mainly in apparel(fashion) goods.
Lucas Rist : manufacturers the 'main harness' for Rover Cars. All volume car manufacturers make extensive use of EDI as a facilitator of just-in-time manufacturing systems.

Teleordering: The EDI system for the book trade is called TeleOrdering.

B. EDI Trading Patterns

1. Hubs and Spokes

Many of the prime movers in the adoption of EDI have been large retail organizations, such as BHS and component assembly manufacturers such as the Rover Group.



Hub and Spoke Trading Patterns

2. Overlapping User Communities

The user community looks like a hub and spoke network to the hub but more like a spider's web to the spoke organization , entrapped by the conflicting requirements of a no. of powerful and demanding customer's organizations.

Q.9 **a.** What is Organizational Structure? Differentiate between Vertical and Horizontal organization. **(7)**

Ans:

A business organization may be structured in many different ways, depending upon the environment within which it operates.

There are always problems with any organizational structure. Traditional organizations based on departments often tend to be bureaucratic and slow in distributing information, while organizations which are more aware of the external environment often lack the formality and control of the traditional organization.

Difference between Vertical and Horizontal Organization

Vertical Organizations

It is a traditional approach which is typified by a functional approach to work in which departments work on tasks relevant to their particular function.

The company is organized on a hierarchical basis with managers matching tasks with appropriate individuals.

When things go wrong, managers are so familiar with this type of organization that they know what questions to ask and what action to take

The main advantage of this kind of structure is seen as functional excellence

Coordination across tasks, departments and functions is the Limitation of Vertical Organizations.

Horizontal Organizations.

Main characteristics:

- The work is structured around a small number of business processes or work flows
- The activities of employees is linked to the needs and capabilities of suppliers and customers enhancing performance all round
- Work is done mainly by teams
- A flatter hierarchy replaces the old steeper functional hierarchy
- Management focus more on continuous improvement than the traditional management activities of decision making, evaluation and resource allocation

b. Explain the Structure of Virtual Enterprise. **(7)**

Ans:

The virtual enterprise can be an appropriate structure to explore the emerging opportunities for creating value in the information society.

They also illustrate the impact of the virtual enterprise on specialization within an organization.

That is, when value drives the restructuring of virtual operations, a new pattern of specialization for the individual company can be expected.

Within the virtual network, specialized partners provide services.

Despite their contingent network-structure, virtual enterprises have to build up their own identity if they want to survive.

Q.10 **a.** Describe in brief the history of E-Commerce. **(7)**

Ans:

History of E-commerce.

E-commerce began before personal computers were prevalent and has grown into a multi-billion dollar industry. By looking at the evolution of e-commerce, it will be easier to judge its trends for the future.

Year	Event
1984	EDI, or electronic data interchange, was standardized through ASC X12. This guaranteed that companies would be able to complete transactions with one another reliably.
1992	CompuServe offers online retail products to its customers. This gives people the first chance to buy things off their computer.
1994	Netscape arrived. Providing users a simple browser to surf the Internet and a safe online transaction technology called Secure Sockets Layer.
1995	Two of the biggest names in e-commerce are launched: Amazon.com and eBay.com.
1998	DSL, or Digital subscriber Line, provides fast, always-on Internet service to subscribers across California. This prompts people to spend more time, and money, online.
1999	Retail spending over the Internet reaches $20 billion, according to Business.com.
2000	The U.S government extended the moratorium on Internet taxes until at least 2005.

b. Explain briefly the generic framework for e-commerce. **(7)**

Ans:

Generic framework of e-commerce includes the Applications of EC (such as banking, shopping in online stores and malls, buying stocks, finding a job, conducting an auction, and collaborating electronically on research and development projects). To execute these applications, it is necessary to have Supporting Information and Organizational Infrastructure and System, which includes

- Common Business Services Infrastructure (security smart cards/authentication, electronic payment, directories/catalogs)
- Messaging and Information Distribution Infrastructure (EDI, e-mail, Hypertext Transfer Protocol)
- Multimedia Content and Network Publishing Infrastructure (HTML, Java, World Wide Web, VRML)
- Network Infrastructure (Telecom, cable TV, wireless, Internet, VAN, WAN, LAN, Intranet, Extranet) And their implementation is dependent on four major areas as supporting pillars:
- People (Buyers, Sellers, Intermediaries services, IS People, and Management),
- Public policy (Taxes, Legal, Privacy Issues, Free Speech, Domain Names)
- Technical Standards (for Documents, Security and Network Protocols, Payments)
- Organizations (Partners, competitors, Associations, Govt. Services)

The EC management coordinates the applications, Infrastructures and pillars.

Q.11 **a.** What are the features of Client/Server Computing? How do TP systems affect performance of e-commerce sites? **(7)**

Ans:

Although there are various different configurations, different hardware and software platforms and even different network protocols in Client-Server Architecture, they all possess certain characteristics and features that distinguish them from traditional mainframe computing environment. Some of them are listed below:

- Consists of a networked webs of small and powerful machines (both servers and clients)
- Open Systems
- Modularity
- Cost Reduction and Better Utilization of Resources
- Complexity

Transaction Processing System affects performance of e-commerce sites

Transaction-Processing applications Typical OLTP (Online-Transaction Processing Applications), also known as mission-critical applications, include order entry, inventory, and point-of-sale systems

This kind of mission-critical system must run continuously, if it is unavailable even for a brief moment, the organizations will experience server repercussions.

Examples are stock exchange system, air traffic control networks and airline reservation systems.

In a simple client/server system, many clients issue requests and server responses.

b. What are IP addresses? How are IP addresses allocated to Computer? **(7)**

Ans:

IP address refers to the name of a computer on a network, like the Internet.

- An Identifier for a computer or device on a TCP/IP network, like the Internet.
- The format of an IP address is a 32-bit numeric address written as four numbers separated by periods. Each number can be zero to 255.
- The IP addresses can be in different classes/categories.

CLASS A, CLASS B, CLASS C, CLASS D, CLASS E IP ADDRESSES ARE ALLOCATED AS FOLLOWS:
Every machine on the Internet has a unique identifying number, called an IP Address. A typical IP address looks like this:

216.27.61.137

To make it easier for us to remember, IP addresses are normally expressed in decimal format as a "dotted decimal number" like the one above. But computers communicate in binary form. Look at the same IP address in binary.

11011000.00011011.00111101.10001001

The four numbers in an IP address are called octets, because they each have eight positions when viewed in binary form. It can have two different states (1or0). If you add all the positions the total number of possible combinations per octet is 28 or 256. So each octet can contain any value between 0 and 255. Combine the four octets and you get 2342 or a possible 4,294,967,296 unique values:

There are five IP classes plus certain special addresses:

Default Network-The IP address of 0.0.0.0 is used for the default network.

Class A- This class is for very large networks. IP addresses with a first octet from 1 to 126 are part of this class. The other three octets are used to identify each host. This means that there are 126 Class A networks each with 16,777,214 (2^{24}-2) possible hosts for a total of 2,147,483,648 (2^{31}) unique IP addresses. Class A networks account for half of the total available IP addresses. In class A networks, the high order bit value (the very first binary number) in the first octet is always 0.

Net	Host or Node
115	24.53.107

Loop back- The IP address 127.0.0.1 is used as the loop back address. This means that it is used by the host computer to
send a message back to itself. It is commonly used for troubleshooting and network testing.

Class B- Class B is used for medium-sized networks. A good example is a large college campus. IP addresses with a first octet from 128 to 191 are part of this class. Class B addresses also include the second octet as part of the Net identifier. The other two octets are used to identify each host. This means that there are 16,384(2^{14}) Class B networks each with 65, 534 (2^{16}-2) possible hosts for a total of 1,073,741,824(2^{30}) unique IP addresses. Class B networks make up a quarter of the total available IP addresses. Class B networks have a first bit value of 1 and a second bit value of 0 in the first octet.

Net	Host or Node
145.24	53.107

Class C- Class C addresses are commonly used for small to mid-size businesses. IP addresses with a first octet from 192 to 223 are part of this class. Class C addresses also include the second and third octets as part of the Net identifier. The last octet is used to identify each host. This means that there are 2,097,152(2^{21}) Class C networks each with 254 (2^{8}-2) possible hosts for a total of 536,870,912 (2^{29}) unique IP addresses. Class C Networks make up an eighth of the total available IP addresses. Class C networks have a first bit value of 1, second bit value of 1 and a third bit value of 0 in the first octet.

Net	Host or Node
195.24.53	107

Class E- Class E is used for experimental purposes only. Like Class D, it is different from the first three classes. It has a first bit value of 1, second bit value of 1, third bit value of 1 and fourth bit value of 1. The other 28 bits are used to identify the group of computers the multicast message is intended for. Class E accounts 1/16th (268,435,456,or 2^{28}) of the available IP addresses.

Net	Host or Node
240	24.53.107

Broadcast- Message that are intended for all computers on a network are sent as broadcasts. These messages always use IP address 255.255.255

Q.12 Explain the role of World Wide Web in the field of e-commerce**.** **(5)**

Ans:

In the 1990s, the advent of the World Wide Web on the Internet represented a turning point in e-commerce by providing an easy-to-use technological solution to the problems of information publishing and dissemination. The web made e-commerce a cheaper way of doing business and enabled more diverse business activities. Use of the Internet and web-site is to allow vendors and service providers to do business on-line. The Internet supports e-commerce and it also requires an infrastructure which allows customers to visit virtual shopping malls for items on sale. This concept was developed in 1999. E-commerce requires a secure way to transfer the money electronically.

Internet promotes e-selling in fraction of seconds. Thus, it promotes the growth of a customer base. It promotes interactive surveys. The users opinions can be gathered anywhere as it provides real time statistics to the uses.

Q.13 **a.** What is non-repudiation? How can it be achived in designing e-cash based system? Give a suitable algorithm. **(7)**

Ans:

Non Repudiation: Assurance that the sender is provided with proof of delivery and that the recipient is provided with proof of the sender's identity so that neither can later deny having processed the data.

E-cash is essentially an online solution. The buyer must validate the coins by the issuer in order to get the purchase done. The coins used in the system contain the value of the coin, the serial number that is unique for every coin and identity strings connecting the coin to the user withdrawing it. However, neither the issuer nor the merchant can deduce the identity of the customer by examining the coins if the coin is used only once.

The user of the e-cash system must have an account in a bank that's a certified eCash minter. The blind signature and secret splitting techniques are used, which ensures that the coins are anonymous until used twice. The suitable algorithm is RSA & DES.

The algorithm is as follows:

<u>Key generation in RSA</u>

1. Select p, q where p and q are both prime and $p \neq q$ (private, chosen)
2. Calculate $n = pXq$
3. Calculate $\phi(n)=(p-1)(q-1)$ where n is public calculated and $\phi(n)$is Euler totient funtion
4. Select integer e, with gcd $(\phi(n),e)=1$; $1<e<\phi(n)$ where e is public, chosen
5. Calculate d where $d = e^{-1} \bmod \phi(n)$ and d is private, calculated
6. Public key KU={e,n}
7. Private key KR={d,n}

<u>Encryption in RSA:</u>

Plaintext M where M<n
Ciphertext C where $C=M^e \pmod n$
Decryption in RSA:
Ciphertext C

Plaintext M where $M = C^d \pmod n$

b. Explain the use of the following terms:
i) Public Key Encryption
ii) Secret Key Encryption **(7)**

Ans:

A cryptographic system that uses two-keys—a public key known to everyone and a private or secret key known only to the recipient of the message.

An important element to the public key system is that the public and private keys are related in such a way that only the public key can be used to encrypt messages and only the corresponding private key can be used to decrypt them. Moreover, it is virtually impossible to deduce the private key if you know the public key.

They are extremely secure and relatively simple to use.
These algorithms lead to several varieties of public key encryption(PKE). PKE addresses three issues that flaw many encryption schemes;
PKE is computationally difficult to decode.
PKE does not require a secure channel to send the key; the key is, in fact, public.

Q.14 **a.** How does an authentication system differ from a firewall in functioning? **(6)**

Ans:

Authentication vs firewall
User Authentication and Authorization

An important advanced firewall security feature is user-oriented authentication, which is the ability to allow or deny certain connections based on user name and password combination or some other, more advanced identification scheme.
Various authentication technologies are available. The simplest forms require typing a user name and a reusable password.
This method is suitable for controlling only outbound Internet access, because password guessing and eavesdropping attacks are likely on inbound access attempts.
Firewalls can log network activity in detail, filter the log to produce meaningful reports, and alert a network administrator when the network has reached a predefined threshold.

The firewall software supports at least Internet services:
HTTP
FTP

Gopher
SMTP
Telnet

DNS name resolution, preferably by letting you run DNS on the firewall and on an internal system.

b. Give the types of firewalls and explain any one type in detail. **(8)**

Ans:

Conceptually, there are two types of firewalls:

1. Network Level
2. Application Level

Network Level Firewall/Packet Filters

The Network level firewalls operate on the mechanism of filtering individual IP packets using the routers.

Packet filters, called " access control lists" on Internet routers provide the rudimentary form of security.

Filters are configured to allow/discard packets based on attributes such as:

1. Specific source or destination IP addresses
2. Specific protocol types
3. Specific source or destination port numbers
4. TCP flags set/clear in the packet header

Q.15 **a.** What are the advantages of having e-commerce over extranets? **(7)**

Ans:

Extranets are about " joining up" the supply chain-suppliers, resellers, distributors, customers, etc.-enabling business-to business eCommerce and streamlining production and sales process, e.g. through on-line ordering, order tracking and inventory management, and so can dramatically reduce costs in these areas. Business-to-business sales are expected to outstrip business-to-consumer sales as an Internet growth area.

The advantages and benefits to businesses include:

Less Paperwork- with documents and business processes on an Extranet, information and documents can be accessed, processed, downloaded (and if necessary printed out) on demand;

Lower Costs- reduction in need for costly meetings, 'phone calls and travel;

A single interface-the Web browser is the only interface required between you and your business partners, regardless of the computer systems being used;

Easy to use- Web browsers provide an intuitive, point-and-click interface

Up-to-date and timely information-Web-based documents can be updated easily, giving you and your business partners faster access to accurate information.

More efficient customer service- information, such as inventory levels, is easily accessible and available 24 hours a day, 7 days a week… in effect creating self-service customers.

Easy access-because Extranets use Internet technologies, information can be accessed from anywhere in the world, e.g. from a remote office.

b. What are the concerns for growth of e-commerce in India? **(7)**

Ans:

Government as Facilitator for the growth of e-commerce has taken following steps:

- Promotion of competitive telecom/datacom/internet Infrastructure
- Development of suitable legal framework for E-Commerce
- Promoting SMEs as E-commerce Enterprises
- Security for E-commerce
- E-commerce & standards
- Help Indian Industry gain competitive advantage through E-Commerce
- Proactive role in WTO, WIPO and other multilateral organizations for Advantage India

Q.16 **a.** What is the significance of XML in EDI and electronic commerce? **(7)**

Ans:

XML has been described as lightweight SGML

XML shows great promise for its inherent ability to allow a " document" to be marked up in a way that pieces of the document (object) are internally defined and then "nested" within other objects to show related attributes

XML tags also allow you to apply different style sheets, i.e., to create a Web site; create a CD-ROM publication; and create a printed source.

XML is on the horizon and offers new opportunities for managing content.

No doubt, the reusability of objects with XML is very attractive. XML may lead to powerful results by integrating the features of CORBA and HTTP. HTTP allows to build a system which is independent of the data being transferred. XML supports the data exchange format between the systems.
And, so is the promise of using XML as a standard for Electronic Data-Interchange (EDI).

b. Explain the features and utilities available in java, which makes it suitable for developing e-commerce applications. **(7)**

Ans:

Following are the features and utilities available in JAVA which make it suitable for developing e-commerce applications:

1. In a network, the transmission of passive information and active programs are quite common between the server and a PC. Such programs can be very conveniently run on JAVA.

2. Java supports applet. An applet is an application designed to be transmitted over the Internet.
3. Java provides a high level security against risk of virus infection and malicious programs that can gather private information such as credit card number, bank balances etc. The security is assured by 'Fire walls'.
4. It is designed for distributed environment of the Internet.
5. It is simple, robust and portable.

Q.17 a. What are the various phases of consumer merchantile model and also differentiate between prepurchase interaction & post purchase interaction. **(7)**

Ans:

There are three phase of consumer mercantile model as listed below:

1. prepurchase interaction
2. purchase consummation
3. postpurchase interaction

Prepurchase Interaction

The prepurchase interaction for consumers consists of 3 activities:
Product/service search and discovery,
Comparison shopping and product selection, and Negotiation of terms.

Purchase Consummation

This model lists 3 activities in the purchase consummation phase:
Placement of order,
Authorization of payment, and
Receipt of product.

Postpurchase Interaction

Customer service and support: The considerations at this stage can be explained by the following example:
Consider a bundle consisting of a portfolio allocation model, a quadratic programming solution algorithm, and visualization tool to graphically depict percentage allocations into various instruments.
In executing this bundled service, the user notes that upon supplying the inputs, the Expected output (i.e., a pie chart depiction of portfolio allocation) is not returned.

Who should the consumer turn to address this problem? Should the consumer approach the model provider, the solution algorithm provider or the visualization tool provider? Post purchase interaction is done to ensure that the rights of both consumers and providers are protected.

b. What are the limitations of traditional payment instruments? How are these limitations overcome by electronic payment systems. **(7)**

Ans:

The limitations of traditional payment system are that they take a lot of time. These systems require manual work to be done & also requires cash to be paid.

The limitations of this is taken care of by e-payment system. E-payment offers total business-to-business and business-to-consumer E-Commerce Solutions and Services to small, medium and large enterprises and government organizations. Services include Web Design and Hosting, Domain Name Registration, Shopping Cart software and other innovative products for the online world.

Q.18 **a**. What is electronic cash? What are the properties of electronic cash? **(7)**

Ans:

E-cash is a cash which is represented by two models. One is the on-line form of e-cash which allows for the completion of all types of internet transactions.
The other is off-line; essentially a digitially encoded card that could be used for many of the same transactions as cash.
Properties :

1. Monetary Value : Monetary value must be backed by either cash, bank – authorized credit cards or bank certified cashier's cheque.
2. Interoperability : E-cash must be interoperable i.e exchangeable for other e-cash, paper cash, goods or services etc.
3. Retrievability : E-cash must be storable and retrievable.
4. Security : E-cash should not be easy to copy or tamper.

b. What are the types of amart cards used in e-commerce? What are the essential components of an e-banking site? **(7)**

Ans:

Generally there are 2 types of smart cards.
Memory smart cards, which can be viewed as minuscule removable read/ write disks with optional security; and processor cards, which can be viewed as miniature computers with an input and output port.

Java card is a smart card with the potential to set the overall smart card standard, Java Card is comprised of standard classes and APIs that let Java applets run directly on a standard ISO 7816 compliant card. Java Cards enable secure and chip-independent execution of different applications.
E-banking systems rely on a number of common components or processes. The following list includes many of the potential components and processes seen in a typical institution:

- Website design and hosting,
- Firewall configuration and management,

- Intrusion detection system or IDS (network and host-based),
- Network administration,
- Security management,
- Internet banking server,
- E-commerce applications (e.g. bill payment, lending, brokerage),
- Internal network servers
- Core processing system
- Programming support, and
- Automated decision support systems.

These components work together to deliver e-banking services. Each component represents a control point to consider.

Q.19 a. What is EDI (Electronic Data Interchange)? Explain benefits and drawbacks of EDI process. Also explain different EDI components and EDI services? **(7)**

Ans:

EDI: Electronic Data Interchange (EDI) is used by organizations for transactions that occur on regular basis to a pre-determined format. Its one of the E Commerce technologies. EDI consists of direct computer-to-computer transmission of data among various firms.

It is used in number of trade sectors for inter-organization, regular, repeat transactions. These systems require EDI standards, EDI software, an EDI network & trading community.

Advantages of EDI

1. EDI replaces paper transactions with electronic transactions thus it saves times and speeds up transactions .

2. It provides a legal record of business communications

3. Value-added networks (VANs) were required in the past but EDI users are now able to transmit their data encrypted over the Internet at the far lower Internet connection rates via new EDIINT [2] standards for email (AS1), HTTP/HTTPS (AS2), and ftp (AS3).

4. Use of EDI reduces cost. These include the cost of stationery, postage etc.

5. Accurate invoicing can be done using EDI. EDI invoices can be automatically matched against the original order and cleared for payment without any queries which usually arise when paper invoices are matched with orders.

6. Quick response is achieved with EDI. For example if a customer is to be informed that a particular product is not available and if this is one using paper orders it takes lot of time but with EDI a customer can be informed straight away so that he may go for the other option. Therefore, quick response can easily be obtained form the customer using EDI.

Disadvantages

1. The X12 standard is so large and general

2. EDI communications negotiate a technical agreement to define exactly what subset of EDI they will use

3. EDI variants define some optional EDI components as mandatory and others as forbidden, specify additional inter-component restrictions, identify a subset of codes within used code sets that will be accepted and used, may add additional codes, and restrict the transaction sets that will be used.

4. The lack of semantic rigor in the meanings of various components of EDI messages

5. Without being semantically-enabled, EDI messages are unable to be interfaced with Semantic Web Services

6. EDI is too expensive: some companies are only doing business with others who use EDI. If a company wants to do business with three organizations, they have to implement an EDI program. This expense may be very costly for small companies

Different EDI components and services

Three main components including services in EDI System are as follows:

- Application Service: Provides the means of integrating existing or new applications into the EDI System.

- Translation Service: Converts data from internal format standards to an external format and translates data from an external format to an internal format standard.

- Communication Service: Passes documents onto a network via the agreed communication protocol.

b. Define the terms internet, intranet and extranet and explain the role each plays in e-business. **(7)**

Ans:

Internet:

The Internet, an umbrella term covering countless network and services that comprise a super-network, is a global network of computer networks that was initiated in the 1960's by a team of scientists under a US government research contract. The Internet now provides access EC transactions to millions of consumers and businesses.

Intranet:

An Intranet is a type of information system that facilitates communication within the organizations among widely dispersed departments, divisions, and regional locations. Intranets connect people together with Internet technology, using Web Browsers, Web Servers, and Data Warehouses in a single view. With an intranet, access to all information, applications, and data can be made available through the same browser. The objective is to organize each individual's desktop with minimal cost, time and effort to be more productive, cost-efficient, timely and competitive.

Extranet:

Extranet is Extension of an Intranet that makes the latter accessible to outside companies or individuals with or without an intranet. It is also defined as a collaborative Internet connection with other companies and business partners. Parts of the Intranet are made available to the customers or business partners for specific applications. The Extranet is thus an extended Intranet, which isolates business communication from the Internet through secure solutions. Extranets provide the privacy and security of an Intranet while the global reach of the Internet.

Role of Internet, Intranet and extranet in e-business

The following information activities are carried out in any business:

1. Procurement of raw materials
2. Advertising the products
3. Selling the products and providing service after sale.
4. Issuing bills making invoices, receiving and making payments.
5. Transaction with dealers, distributors, customers, banks etc.

The above operations are being performed at different locations. The computer network through Internet has made it possible to communicate with each other and exchange. Information related to specific dealings.

Q.20 a. Discuss any two popular encryption techniques to ensute secured transactions on the net? **(7)**

Ans:

1. Translation table: In this method each chunk of data is used as an offset within a 'translation table' and the resulting 'translated' value from within the table is then written into the output stream. The encryption and decryption programs would each use a table that translates to and from the encrypted data. This method is very simple and fast, the down side is that once the translation table is known, the code is broken.

2. Word/byte rotation and XOR bit masking.
In this method the words or the bytes within a data stream are rotated, using multiple and variable direction and duration of rotation. This is nearly impossible to break. Further, the use of 'XOR mask' in combination with this makes the code breaking process even more difficult.

b. What are the various connectivity options available to Internet Subscribers? Discuss in detail. **(7)**

Ans:

Internet Connectivity Options:

Internet access is perhaps one of the most popular services that Service Providers offer their customers. Customers have flexibility to purchase MPLS VPN services Internet connectivity from separate Service Providers. Customers can alternatively offer Internet connectivity directly from their network may it be from one of their remote sites or the central site. The Internet Service Provider (ISP) does not need to distinguish customer's Internet and VPN traffic, because all traffic traversing through a Service Provider network would be MPLS VPN traffic. Customers who do not purchase Internet connectivity from a Service Provider do need to work out additional variables:

- Routing
- Appropriate location for Internet access within the network
- Network Implementation Translation (NAT) implementation, if the network does not use public addresses
- Security
- Additional management and monitoring

The best way to offload these responsibilities is to purchase services from a Service Provider.

ISPs, Service Providers, and customers often wonder about the best and most highly recommended way to set up Internet connectivity.

There are various possible combinations for using a network infrastructure to implement Internet connectivity, depending on how a Service Provider carries MPLS VPN and Internet traffic. The options at the infrastructure level are:

1. Shared MPLS VPN and Internet Connectivity

2. Partially Shared

3. Full Separation

Q.21 **a**. Describe generic framework for electronic commerce with suitable diagram? **(7)**

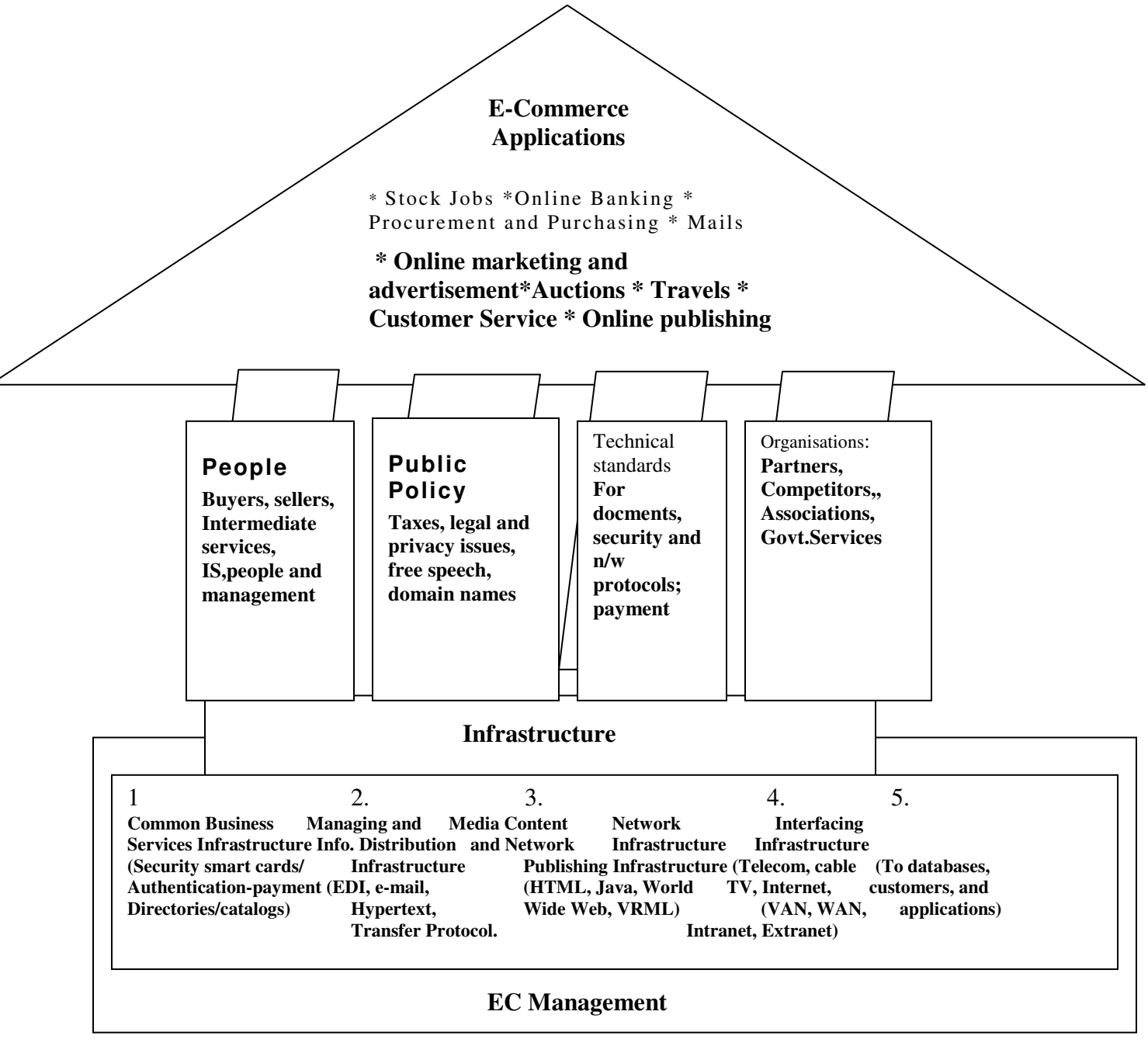
Ans:

Generic Framework for electronic commerce includes the Applications of EC (such as banking, shopping in online stores and malls, buying stocks, finding a job, conducting an auction, and collaborating electronically on research and development projects). To execute these applications, it is necessary to have Supporting Information and Organizational Infrastructure and System, which includes

- Common Business Services Infrastructure (security smart cards/authentication, electronic payment, directories/catalogs)
- Messaging and Information Distribution Infrastructure (EDI, e-mail, Hypertext Transfer Protocol)
- Multimedia Content and Network Publishing Infrastructure (HTML, Java, World Wide Web, VRML)
- Network Infrastructure (Telecom, cable TV, wireless, Internet, VAN, WAN, LAN, Intranet, Extranet) And their implementation is dependent on four major areas as supporting pillars:
- People (Buyers, Sellers, Intermediaries services, IS People, and Management),
- Public policy (Taxes, Legal, Privacy Issues, Free Speech, Domain Names)
- Technical Standards (for Documents, Security and Network Protocols, Payments)
- Organizations (Partners, competitors, Associations, Govt. Services)

The EC management coordinates the applications, Infrastructures, and pillars.

Diagram for Architectural framework for electronic commerce is given below:



b. What do you understand by E-cash? What are the properties of E-cash? What is the basic difference between the transaction made using Smart Card and E-cash? **(7)**

Ans:

E-Cash and its Properties:

Ecash is a cash which is represented by two models. One is the on-line form of e-cash which allows for the completion of all types of internet transactions.
The other is off-line; essentially a digitially encoded card that could be used for many of the same transactions as cash.

Properties :

1. Monetary Value : Monetary value must be backed by either cash, bank – authorized credit cards or bank certified cashier's cheque.
2. Interoperability : E-cash must be interoperable i.e exchangeable for other e-cash, paper cash, goods or services etc.
3. Retrievability : E-cash must be storable and retrievable.
4. Security : E-cash should not be easy to copy or tamper.

Smart Card & E-Cash

E-cash storable smart cards can store and dispense cash electronically, making bills and coins lesser necessary. It transfers funds over phone lines, making it easier to reload your smart cards. You can use this system wherever you see the outlets accepting e-cash. This is basically an electronic wallet that allows person-to-person payments. The telephone or Internet link lets you make this person-to-person payment anywhere in the world. The card can store around five separate currencies at the same time.

Q.22 **a.** Describe Mercantile Process Model from the Merchant's perspective with a suitable diagram. **(7)**

Ans:

This model lists 3 activities in the purchase consummation phase:
Placement of order, Authorization of payment, and Receipt of product.

The interaction for these activities is done in three phases. They are listed below:

(i) Prepurchase Interaction

The prepurchase interaction for consumers consists of 3 activities:
Product/service search and discovery
Comparison-shopping and product selection
Negotiation of terms.

(ii) Purchase Consummation

This model lists three activities in the purchase consummation phase:
Placement of order,
Authorization of payment and
Receipt of product.

(iii) Postpurchase Interaction

Customer service and support:

The considerations at this stage can be explained by the following example:
Consider a bundle consisting of a portfolio allocation model, a quadratic programming solution algorithm, and a visualization tool to graphically depict percentage allocations into various instruments.
In executing this bundled service, the user notes that upon supplying the inputs, the expected output (i.e., a pie chart depiction of portfolio allocation) is not returned.
Who should the consumer turn to address this problem? Should the consumer approach the model provider, the solution algorithm provider or the visualization tool provider?
To ensure that the rights of both consumers and providers are protected.

b. Explain different security protocols used for e-commerce applications. **(7)**

Ans:

The e-commerce systems of today are composed of a number of components including: a commerce server, data transaction protocols, and client software from which transactions originate.

The protocols for e-commerce transactions are : SSL, PCT, SET, S-HTTP, S/MIME, Cybercash, and Digicash among others. Most of these protocols are not interoperable, and consumers must choose one protocol over another. If a merchant is not a subscriber to Cybercash, then a Cybercash consumer will not be able to purchase wares from the merchant. A consumer does not have a browser client that supports S-HTTP, then the consumer will not be able to engage in a secure transaction with a merchant that uses S-HTTP. The market may ultimately decide the winners and losers in standardized protocols, however, the necessity for interoperable, cross-platform components will not lessen. Development of secure components for use in building commerce applications is an important step in the maturation and acceptance process.

Q.23 Differentiate the following:-

i) Traditional Commerce vs Electronics Commerce **(4)**

Ans:

Identity. In traditional commerce customers can easily authenticate the identity of a merchant simply by walking into a bricks-and-mortar store while E-Commerce means

doing business online or selling & buying products and services through web storefronts.

Immediacy. Customers can touch and feel and hold the merchandise.

Value. The item at the center of the commerce transaction -- the product, service, or property that is to be sold/bought.

Discourse. Customers can converse with the merchant face-to-face; unmediated conversation is basic to human communication. People want the feedback available from non-verbal behavior, which forms a large part of our judgment process.

Community. Customers can interact with other customers and gain feedback about the merchant from other customers, as well as by observing the merchant interacting with other customers.

Privacy. In E-Commerce customers can make purchases anonymously with cash; they usually don't have to give their name or address. This is not possible in traditional commerce.

ii) Hypertext vs Hypermedia **(5)**

Ans:

Hypertext is basically the same as regular text - it can be stored, read, searched, or edited - with an important exception: hypertext contains connections within the text to other documents.

Hypermedia documents contain links not only to other pieces of text, but also to other forms of media - sounds, images, and movies. Images themselves can be selected to link to sounds or documents. Here are some
World Wide Web is an open hypermedia system.

iii) E-cheques vs Credit Cards **(5)**

Ans:

E-cheques: E-cheques are used for business dealing in e-commerce. Transactions of these cheques take place on Internet. In this system the electronic cheque is issued by the buyer to the seller. The e-cheques are then deposited by the seller in the bank account. A number of agencies like clearing house, certification authority, buyer's bank and seller's bank participate in the entire process along with the actual seller and the buyer.

Credit Card: This is a normally used transaction card. After purchases are made, the card enters the transactions against the credit card. The card issuer transfers the required amount of money to the seller's account. The bill for such an amount is raised to the cardholder.

Q.24 **a**. Discuss the functioning of various network access equipment. **(7)**

Ans:

The E1 multiplexers MX2000 and MX2411 and E1/T1 MX200 are providing multi interface user access to network PDH or SDH network or a microwave link. They make it possible either to make a point-to-point connection by multiplexing several affluent, or to carry out connections multipoint if the networks cross connect the TS of the links.

The TDM multiplexer multi E1/T1/E3/DS3, QX3440, IX4200-9 and IX4200-28 are termination equipments of PDH network, or they could be used as a node of PDH networks by using the cross-connect of the affluent, broadcasting, protection features. This equipment allows the distribution to subscribers for voice or leased lines and also to manage the installation of very significant networks multipoint. Those will be often used with SDH multiplexers like IX4100, IX7163 that have also the functionalities of mixing PDH and the full ADM STM1 multiplexer HX9100.

b. What do you understand by work flow automation? How do work flow related technologies affect e-business? **(7)**

Ans:

Work Flow Automation:

Organizations often standardize processes across the organization and encourage users to adopt them. Each business unit has a unique set of business processes.

The workflow is a very important part of the software development process.

The workflow automation provides a possibility to plan, manage, control and document the communication process.

There are some standards like WFMC' Reference Model etc.

Work Flow related Technologies affect e-business

Microsoft CRM automates internal business processes by creating workflow to carry out routine tasks that involve daily business operations.

These processes can be designed to ensure that the right information gets to the right people at the right time, and help participants keep track of the steps they have to take to complete their work. Your managers can define, automate, and enforce specific business rules, policies, and procedures.

Microsoft CRM Workflow tools enables one to:

- Define business policies based on established processes.
- Ensure that customer commitments are met.
- Automatically escalate issues to management when required.
- Level workloads across teams and territories.
- Manage key business policies and procedures.

Ensure a consistent service process.

Q.25 **a.** What is e-brokerage? How does electronic brokerage facilitate search and retrieval of information? **(7)**

Ans:

E-brokerage is an investment house that allows you to buy and sell stocks and obtain investment information from its Web site.

E-commerce not only has tremendous potential for growth but also poses unique challenges for both incumbents and new entrants.

Three firm capabilities that are critical for superior firm performance in e-commerce are:-

1. Information technology capability,
2. Strategic flexibility, and
3. Trust-building capability.

E Brokerage facilitates search & retrieval of Information

The success factor of a brokerage is its ability to retain existing clients and to increase their satisfaction through effective coordination and enactment of CRM activities.

Customer relationship management (CRM) is the strategy for optimizing the lifetime value of customers.

It allows companies to gather and access information about customers' buying histories, preferences, complaints, and other data so that they can better anticipate what customers want.

b. What is Supply Chain Management? What are the characteristics of Supply Chain Management in an e-commerce environment? **(7)**

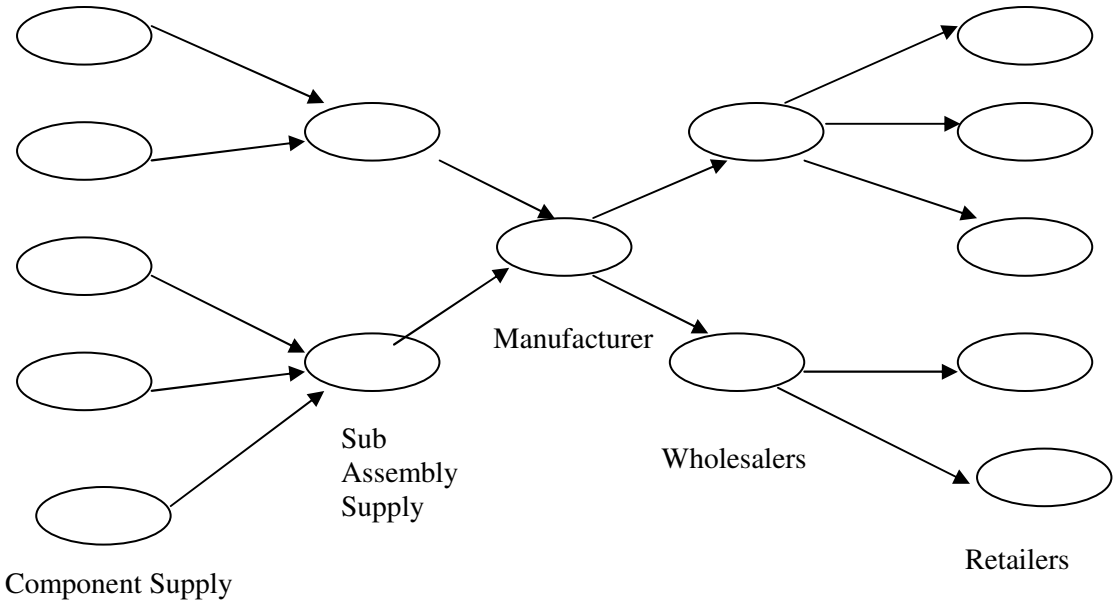
Ans:

Supply Chain Management:

Supply Chain Management involves developing the performance of an organization's supply chain from its suppliers to its customers, by identifying and then selectively improving supply chain processes and organizational infrastructure.

The products sold in shops & purchased for use in organizations are the result of a complex web of relationships between manufacturers, component suppliers, wholesalers, retailers & the logistic infrastructure that links them together.
The web of the trade relationships is referred to as the supply chain or the value chain. Value chains differ between trade sectors

An overall value chain for a manufactured product is shown in a simple form:



Features of Supply Chain Management

Supply Chain Management uses various methods to determine, influence and change how various factors effect buyer behaviour in the supply chain.

Supply Chain Management can help clients to develop brand policies which acknowledge consumer preferences, and supported by appropriate supply contracts maximize the profitability delivered to the business for the available "shelf space".

Supply Chain Management Consultancy's scope is extensive, and typically includes:

1. Sales order processing and customer service
2. Outbound distribution
3. Finished goods warehousing and inventory management
4. Production planning and control
5. Work-in-progress inventory management
6. Raw materials inventory management
7. Inbound distribution
8. Supplier scheduling (or procurement) and
9. Purchasing (or sourcing).

Q.26 **a.** What are the two primary models of Supply Chain Management? Discuss the primary elements of these models. **(7)**

Ans:

Two Primary models of Supply Chain Management

1. Porter's Value Chain Model
2. Supply Chain Model

Primary Elements of these Models are discussed below

They are essentially concerned with Internal Activities of Company. The 3 primary activities of a product process are:

1. Inbound Logistics
2. Operations
3. Outbound Logistics

Also Porter adds 2 further activities, which are

1. Marketing & Sales
2. Service

To support these primary functions there will be a company's infrastructure that performs a number of support activities. These activities are

1. Procurement: Responsible for negotiating quality supplies at an acceptable price with reliable delivery.

2. Technology Development: To ensure that organization's products remain competitive it needs to update its production process, train staff & to manage innovation.

3. Human Resource Management: The recruitment, training & personnel management.

4. Firm Infrastructure: Overall management including planning & accountancy.

b. What does the term convergence mean with respect to E-commerce? Explain three different types of Convergences. **(7)**

Ans:

Convergence with respect to e-commerce

The ability to leverage and integrate the various data sources and processes that make up a corporation's lifeblood and deliver them to the consumer via an integrated web site has given rise to the new world of e-business.

The decreasing cost and increasing speed of wireless devices is driving the move to wireless communication.

Now there is a convergence underway, enabling wireless devices to act as clients in the exploding e-business world.

As more and more businesses seek to build their mission critical business solutions on IP networks, networking providers must examine corporate requirements for electronic commerce using the Internet, intranets, and extranets.

Businesses crave the ease of use of IP technology that promises their companies-to take advantage of the extra benefits, features, enhancements and cost savings that IP technology can provide them like

IP Access
Intranets on Virtual IP Networks
Extranets on Virtual IP Networks

Three different types of convergences are:

1. The convergence of e-commerce and wireless technology
2. The Convergence of E-Commerce and IP Business-Grade Messaging
3. The Convergence of telecommunications and data

Q.27 **a**. What are the server specific middle wares? What can be their role in e-commerce ? Describe the additional features required for an e-commerce server?

(7)

Ans:

Server specific middle wares. Their role in e-commerce. Additional features required by e-commerce server.

Middleware is the term often used to describe the application or business logic present in an application server. Unfortunately, the same term is also used to describe generic services. For example, to communicate, the three tiers may use a messaging system. This is true middleware in the sense that it functions between, or in the middle of the three tiers. When someone uses the term *middleware*, be sure you know what specific meaning is intended.

Commerce Server is media-independent, multi-platform server software that enables the deployment of shopping sites and channels on new media: internet, interactive TV, and mobile.

By the use of infoPlatfrom Commerce Server a shopping service can take advantage of reaching the end user on OpenTV, WAP, SMS and MMS.The product supports functionalities including media-adopted configurable shoping interfaces, personalized promotions, and affiliates integration, besides the mainstream functionalities of a typical e-commerce server such as product and catalogs, shopping carts, ordering and delivery tracking functionality.

The system integrates seamlessly with third-party systems for electronic payment, and delivery tracking with the industry standard protocols such as cXML and ebXML.The system has a full-scale management interface for the operations personal for the e-commerce company.

E-commerce services require dynamic configuration capabilities and seasonal and daily service configuration and content management requirements. The management interface of infoPlatform Commerce Server provides much functionality to easily adapt to the dynamic atmosphere of e-commerce business. The modular software architecture based on J2EE and Oracle enables easy adaptation and customization for specific business rules, and integration to any installed base or third-party system using open standards such as cXML, ebXML, SOAP, HTTP, JDBC, and ODBC.

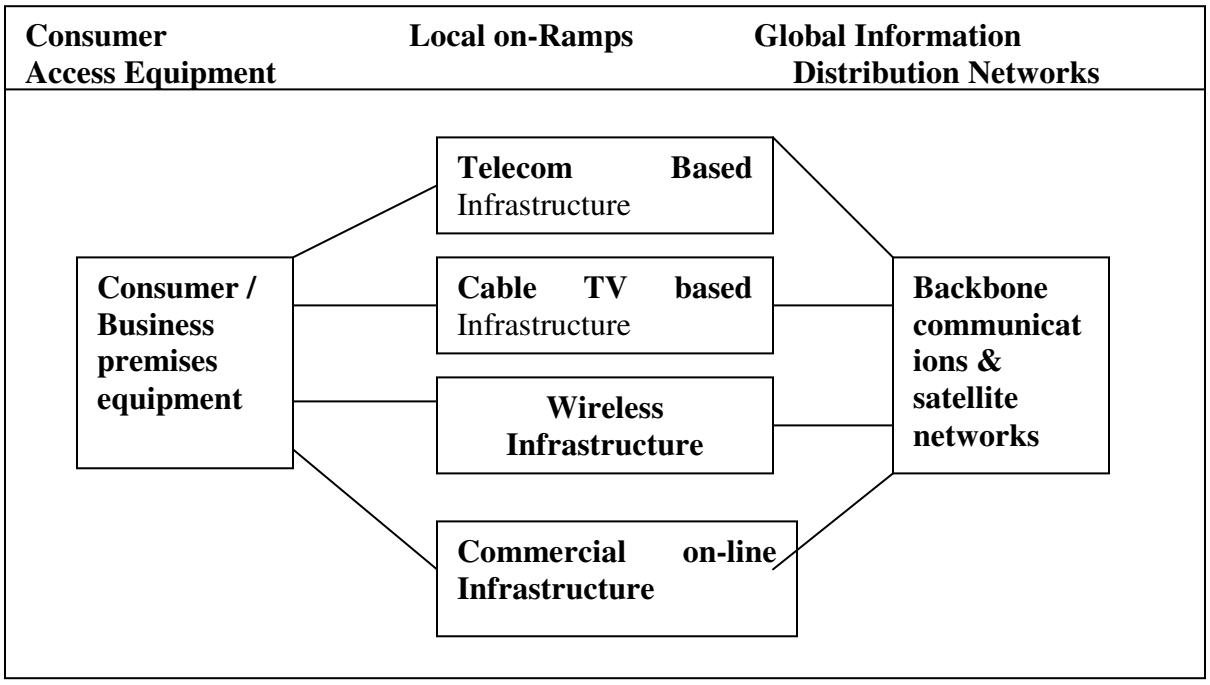
b. What are the components of I-way Infrastructure? Explain with the help of a diagram. Give the block diagram of the basic WWW architecture. Discuss the basic entities in brief.

(7)

Ans:

There are three components of the I-way infrastructure:

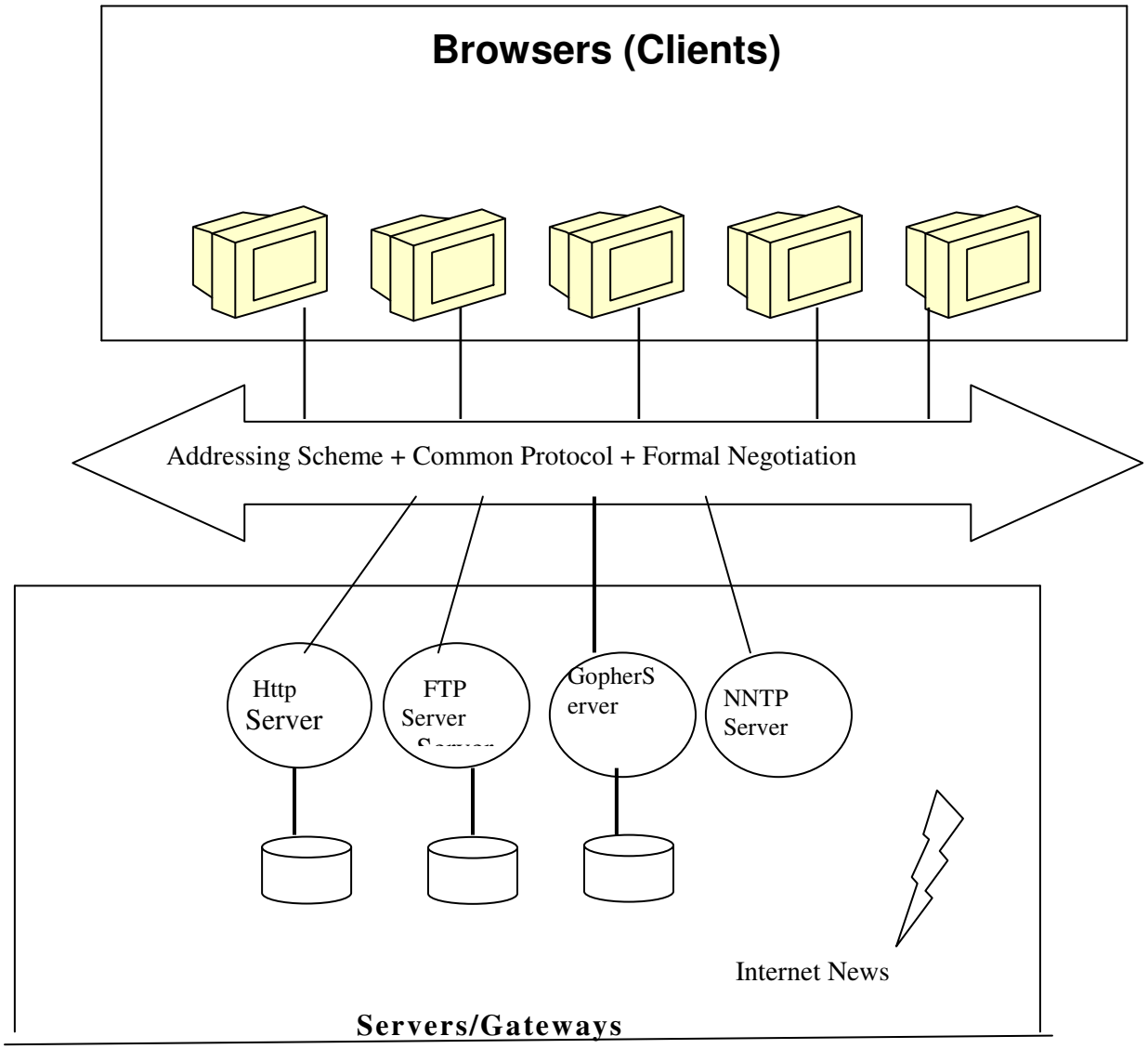
- Consumer access equipment
- Local on-Ramps
- Global information Distribution Network



1. Consumer access equipment represents a critical category, the absence or slow progress of which is holding up other segments of the I-way.
2. Local on-ramps provide linkages between community, businesses, schools and homes to the communications network.
3. Global information distribution networks provide the infrastructure crisscrossing countries and continents.

World Wide Web: It is a complex client/server system in which a web client communicates with the web server. It refers to an interlinked connection of web documents, which can be used by any web server. The documents may be in the form of hypertext, menus, databases etc. Some documents are placed on the web server and the other are available through uniform resource locators. The web documents are marked for formatting and linking with hypertext mark up language (HTML). To deliver web pages, the web servers use Hypertext transfer protocol (HTTP).
The architecture of the WWW, figure below, is the one of clients, such as Netscape, Internet Explorer, or Lynx, "which know how to *present* data but not what its origin is, and servers, which know how to *extract* data", but are ignorant of how it will be presented to the user.

Architecture of the WWW



Q.28 **a**. What is Public Key Cryptography? Explain its advantages and disadvantages. **(7)**

Ans:

Public Key Cryptography:

A cryptographic system that uses two keys-- a *public key* known to everyone and a *private* or *secret key* known only to the recipient of the message.

An important element to the public key system is that the public and private keys are related in such a way that only the public key can be used to encrypt messages and only

the corresponding private key can be used to decrypt them. Moreover, it is virtually impossible to deduce the private key if you know the public key.

They are extremely secure and relatively simple to use.

Advantages & Disadvantages

Increased security and convenience: private keys never need to be transmitted or revealed to anyone.

They can provide a method for digital signatures

Public-key authentication prevents repudiation; each user has sole responsibility for protecting his or her private key. This property of public-key authentication is often called non-repudiation.

A disadvantage of using public-key cryptography for encryption is speed

For encryption, the best solution is to combine public- and secret-key systems.

The public-key system can be used to encrypt a secret key, which is used to encrypt the bulk of a file or message. Such a protocol is called a digital envelope

A successful attack on a certification authority will allow an adversary to impersonate whomever the adversary chooses to by using a public-key certificate from the compromised authority to bind a key of the adversary's choice to the name of another user.

Public-key cryptography is not meant to replace secret-key cryptography, but rather to supplement it, to make it more secure.

b. What are the steps involved in authentication? What is the role of third party and certifying authorities? **(7)**

Ans:

Steps in Authentication

The control over the access of the resources in the repository is exercised in two steps namely Authentication and Authorization.

1. Authentication aims at checking if the user is allowed to connect to the repository server.

2. Authorization aims at checking if the user is allowed to perform the operation he or she is trying to execute.

How basic authentication works

When a particular resource has been protected using basic authentication, Apache sends a 401 Authentication Required header with the response to the request, in order to notify the client that user credentials must be supplied in order for the resource to be returned as requested.

Upon receiving a 401-response header, the client's browser, if it supports basic authentication, will ask the user to supply a username and password to be sent to the server. If you are using a graphical browser, such as Netscape or Internet Explorer, what

you will see is a box, which pops up and gives you a place to type in your username and password, to be sent back to the server. If the username is in the approved list, and if the password supplied is correct, the resource will be returned to the client.

Because the HTTP protocol is stateless, each request will be treated in the same way, even though they are from the same client. That is, every resource, which is requested from the server, will have to supply authentication credentials over again in order to receive the resource.

Fortunately, the browser takes care of the details here, so that you only have to type in your username and password one time per browser session - that is, you might have to type it in again the next time you open up your browser and visit the same web site.

Role of Certifying Authority

A certificate authority is a body, either private or public, that seeks to fill the need for trusted third-party services in E-commerce. A certificate authority accomplishes this by issuing digital certificates that attest to certain facts about the subject of the certificate. VeriSign is one of the pioneering CA's.

IN the context of credit cards the cardholder certificate authority (CCA) issues the certificate to cardholders, the merchant certificate authority (MCA) to merchants who operates e-stores, & the payment gateway certificate authority (PCA) to payment gateway service providers.

Q.29 a. What do you understand by WWW? What is the use of hypertext links in Internet access? Name some popular Internet Browsers. **(7)**

Ans:

WWW: The World Wide is an architectural framework for accessing linked documents spread out over thousands of machines all over the world. It is basically a client server system.

Technically it refers to the Hypertext servers that allow text, graphics & sound files to be mixed together.
Loosely it refers to all type of resources that can be accessed.

Use of Hypertext links in Internet access

From the user’s point of view, the Web consists of a vast, worldwide collection of documents i.e pages. Each page may contain links (pointers) to other related pages, anywhere in the world. Users can follow a link, which then takes them to the page

pointed to. This process can be repeated indefinitely. Pages that point to other pages use Hypertext.

Hypertext is any text that cross-references other textual information with hyperlinks. As all the text in the web is linked through hyperlinks it is of utmost usage for Internet access.

Popular Internet Browsers are:
Internet Explorer, Netscape Navigator and Mosaic

b. How do you achieve workflow automation in e-business environment? **(7)**

Ans:

In order to run smoothly, organizations often standardize processes across the organization and encourage users to adopt them. Unfortunately, each business unit has a unique set of business processes.

There are no any really good stand-alone workflow systems (there are some good work flow systems are part of large ERP System). There are some standards like WFMC' Reference Model etc.

One can use Microsoft CRM to automate internal business processes by creating workflow to carry out routine tasks that involve daily business operations.

Microsoft CRM Workflow tools enable one to:

- Define business policies based on established processes.
- Ensure that customer commitments are met.
- Automatically escalate issues to management when required.
- Level workloads across teams and territories.
- Manage key business policies and procedures.
- Ensure a consistent service process.

Q.30 a. What do you understand by Electronic Funds Transfer? **(6)**

Ans:

Electronic Funds Transfer:

It's an electronic payment method that transfers the money value from one bank account to another in same or different bank.

Today we can also use Internet Based Electronic Funds Transfer (EFT), which implies that the connection between the cyber banks & security protection during the transmission is a must.

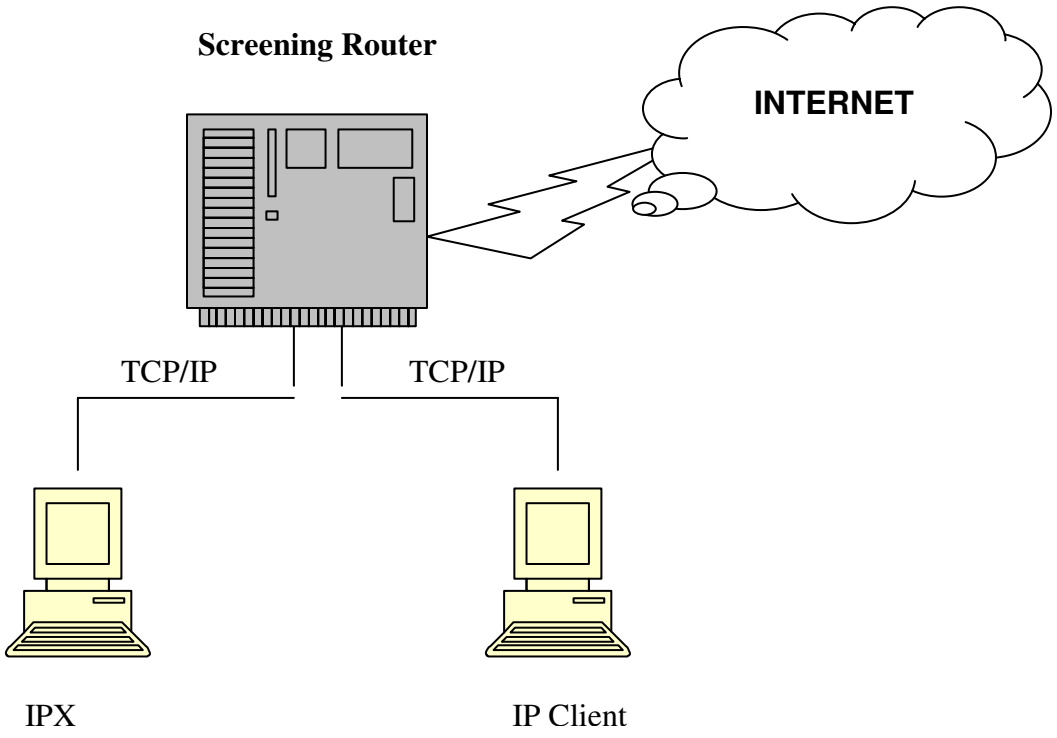
EFT has been in use since the 1970's through automated clearinghouses (ACHs)

b. What is the function of an IP packet screening Router? Explain with the help of a diagram. **(4)**

Ans:

Function of an IP Packet Screening Router:

A screening router is the most basic type of firewall and uses only the packet filtering capability to control and monitor network traffic that passes though the border. Screening routers on a server with packet filtering can block traffic between networks or, for example, traffic to or from specific hosts on an IP port level.Direct communication is usually permitted between multiple hosts on the private network and the Internet. The diagram below shows a basic example of how a screening router works.



Firewall Using Screening Router:

The risk of break-in is large with this type of firewall: each host on the private network is exposed to the Internet and is still a potential break-in point. Unauthorized users can detect and use internal addresses to access information within the firewall. To avoid break-in, screening routers can be set to look at the source address of each incoming IP header instead of the destination address, and drop private addresses that come from the Internet.

c. What are the desirable characteristics of an Electronic Market Place? **(4)**

Ans:

Characteristics of an Electronic Market Place:

a. Its electronic, the business center is not a physical building rather a network-based location where business interactions occur.

b. They are built around publicly accessible networks

c. In this two types of relationships can exist:

- Customer/seller linkage is established at the time of transactions & may be for one transaction only.
- Customer/seller purchase agreement is established whereby seller agrees to deliver services to customers for a defined period of time.

d. When outside communications companies are involved they are typically inline service providers.

e. Customers & sellers independently determine which communication networks they will use in participating in electronic market.

f. No joint guidelines of each party are formulated in advance.

Q.31 a. Discuss password schemes and Biometric systems for implementing client server network security. **(8)**

Ans:

In cyberspace, buyers & sellers cannot see each other. Also in video conferencing, the authenticity of the person dealing with must be verified unless one has dealt with that person before.
Biometric controls provide access procedures that match every valid user identifier (UID).
They also provide authentication method to verify that users requesting access are really the ones who claim to be.
A UID is accomplished in one or more of following ways:

- Provide some password, which only user knows.
- Present something like a smart card or a token which only the user has
- Identify something only the user is, like signature, voice, fingerprint or retinal (eye) scan. It is implemented by biometric controls.

The most common biometrics is the following.

Face geometry (Photo): The computer takes the picture of your face & matches it with a prestored picture.

Fingerprints (Fingerscan): Whenever a user wants access, matching fingerprint against a template containing authorized person's fingerprints.

Hand Geometry: Like fingerprints except the verifier uses a TV like camera to take the picture of the user's hand.

Blood vessel pattern in the retina of a person's eye: A match is done between the pattern of the blood vessels in retina that is being scanned & prestored picture of retina.

Voice (Voice Print): A match between users voice & voice pattern stored on templates.

Signature: matched against the prestored authentication signature.

Keystike Dynamics: Match of person's keyboard pressure & speed against prestored information.

Others: Like Thermography, using a PIN & iris scan.

b. Discuss various threats posed by servers in a client server environment. **(6)**

Ans:

Server Destroyed in an Accident: Leaking pipes, power failures and equipment failures are not uncommon
Destruction of Data Alteration of Data Disclosure of Data Fraud Denial of Service

Users Logged in on Unattended Desktop Computer.

The client/server architecture eliminates the incentive to sign off of mainframe applications to save on connect time charges.

Improperly Protected Server Accounts Used to Gain Access

Poorly protected user accounts, though less powerful, also make tempting targets, since they provide a toehold to gain further privileges

An intruder on a server can disclose sensitive information.

Address Spoofing: Someone spoofs network addresses to gain access to servers, and uses that access to read/alter data or set up future access.

Someone "spoofs" a network address by using their host computer to "impersonate" a trusted computer, thereby improperly gaining special permissions and access that only the trusted computer should have.

Sensitive Data Disclosed Through Packet Sniffing.

Alter data or programs, and leave behind hidden programs which steal passwords from unsuspecting users of the system, and which make future intrusions easier.

Q.32 **a**. Explain the Architectural framework for electronic commerce. **(7)**

Ans:

An application independent framework to classify service interaction relies on four basic dimensions

1. Service topologies
2. Cooperation models,
3. Implementation techniques,
4. Quality of service.

Those most existing implementations rely on a tier-based client-server architecture disregarding the actual demands of the underlying application.

Service Topologies In a service-based architecture there arise dependencies among services resulting from cooperation

Implementation techniques (e.g. remote invocation, HTTP). An event-based cooperation can be implemented using message passing or it can be based on re- mote invocation mechanisms.

A cooperation model describes the way interdependencies are established by cooperative interactions. We identify two basic roles services can play: consumers and providers.

Quality of Service: This is assessed on the basis of customer's satisfaction.

b. What are the risks involved in Electronic Payment Systems? **(7)**

Ans:

From the customer's perspective:

- Stolen payment credentials and passwords
- Dishonest merchants or financial service providers
- Disputes over quality of services or goods

From merchant's perspective:

- Forged or copied payment instruments
- Insufficient funds in customers account, especially with off-line payment systems
- Dishonest or slow financial service providers

From the financial service provider's perspective:

- Stolen customer or service credentials
- Forged or copied payment instruments
- Customers not paying (applies only to credit models)

The risk may be shifted in one direction or the other by using a credit or debit model and by special agreements.

Q.33 a. What are the advantages and disadvantages of a Smart Card? **(6)**

Ans:

Advantages of Smart Card:

1. It provides convenience & support for multiple currencies over borders.
2. Used to store information such as personal identification numbers.
3. Its applications include telephone, transportation & library copies.

Disadvantages of Smart Cards:

1. The value of money can be depleted & recharged.
2. Customers must keep separate e-cash cards, one for each application & can recharge card only at designated locations.
3. In future recharges will be done through one's PC whether it is on internet or your bank network.

b. What are the four types of consumer oriented applications of E-commerce? Discuss them briefly. **(8)**

Ans:

Four types of Consumer Oriented applications in E-Commerce are as follows:

1. B2C (business-to-customer)

   "Electronic commerce" is generally understood mostly as selling goods or services to people ("final consumers"). This is not, so far, the biggest part of online business.

2. B2B (business-to-business)

   The consensus is universal: this is the priority;.

   this is where the money is. It's quite true; so far the bulk of online business is in company-to-company transactions.

3. C2C (customer-to-customer)

   Person-to-person transactions are the oldest form of e-business. They've been there from the beginning, long before there was any widespread use of the Internet. They continue to be all over the place, quite invisible to anyone thinking that all business is on a website.

4. C2B (customer-to-business)

   The most important activity in e-commerce isn't selling. It's buying. Quite often that doesn't mean buying online but checking, comparing, analyzing quality and price before baying in traditional stores or services.

Q.34 **a.** Compare hypertext versus hypermedia. **(5)**

Ans:

Hypertext is basically the same as regular text - it can be stored, read, searched, or edited - with an important exception: hypertext contains connections within the text to other documents.

Hypermedia documents contain links not only to other pieces of text, but also to other forms of media - sounds, images, and movies. Images themselves can be selected to link to sounds or documents. World Wide Web is an open hypermedia system.

b. Explain the components of Information Super Highway Infrastructure. **(5)**

Ans:

The Information Superhighway is more than the Internet. It is a series of components, including the collection of public and private high-speed, interactive, narrow, and broadband networks that exist today and will emerge tomorrow.

- It is the satellite, terrestrial, and wireless technologies that deliver content to homes, businesses, and other public and private institutions.
- It is the information and content that flow over the infrastructure, whether in the form of databases, the written word, a film, a piece of music, a sound recording, a picture, or computer software.
- It is the computers, televisions, telephones, radios, and other products that people will employ to access the infrastructure.
- It is the people who will provide, manage, and generate new information, and those who will help others to do the same.
- And it is the individual Americans who will use and benefit from the Information Superhighway.

The Information Superhighway is a term that encompasses all these components and captures the vision of a nationwide, invisible, seamless, dynamic web of transmission mechanisms, information, appliances, content, and people.

c. Give some examples of malicious data. **(4)**

Ans:

In May 2002, the Norton Anti-Virus software for Windows operating systems detected about 61000 malicious programs. Some of them are named below:

1. The Morris worm released in 1988,
2. The MBDF virus,
3. The Pathogen virus,
4. The Melissa virus, and
5. The Anna worm.
6. ILOVEYOU

Q.35 a. What are the advantages of using XML over HTML? **(7)**

Ans:

The root cause of the problem lies in HTML (Hyper Text Markup Language), the defacto standard for web publication. The major problem with HTML is its 'fixed tagset'. This tagset is mainly for display of the content and HTML provides no tag to address the content precisely.
XML (extensible Markup Language) designed by W3C (World Wide Web Consortium) promises a possible solution to this problem.

The major advantage of XML over HTML is its extensibility i.e., provision of user defined tags and attributes to identify the structural elements of a document. XML also

provides structural complexity to define document structure that can be nested at any level of complexity.
XML also facilitates the transfer of structured data between servers. XML describes data, such as city name, temperature and barometric pressure, while HTML defines tags that describe how the data should be displayed, such as with a bulleted list or a table.

b. What are the essential components of a 3-tier client server/ **(7)**

Ans:

In a three-tier or multi-tier environment, the client implements the presentation logic (the client). The business logic is implemented on an application server(s) and the data resides on database server(s).

The following three component layers thus define a 3-tier architecture:

1. A front-end component, which is responsible for providing portable
2. presentation logic;
3. A back-end component, which provides access to dedicated services, such as a database server.
4. A middle-tier component, which allows users to share and control business logic by isolating it from the actual application;

Q.36 a. What is the difference between intranet and extranet? **(7)**

Ans:

Intranet:

An Intranet is a type of information system that facilitates communication within the organizations among widely dispersed departments, divisions, and regional locations. Intranets connect people together with Internet technology, using Web Browsers, Web Servers, and Data Warehouses in a single view. With an intranet, access to all information, applications, and data can be made available through the same browser. The objective is to organize each individual's desktop with minimal cost, time and effort to be more productive, cost-efficient, timely and competitive.

Extranet:

Extranet is Extension of an Intranet that makes the latter accessible to outside companies or individuals with or without an intranet. It is also defined as a collaborative Internet connection with other companies and business partners. Parts of

the Intranet made available to the customers or business partners for specific applications. The Extranet is thus an extended Intranet, which isolates business communication from the Internet through secure solutions. Extranets provide the privacy and security of an Intranet while the global reach of the Internet.

Following table gives brief overview of the differences among the three types of the network:

Network type	Typical users	Access	Type Of information
Intranet	Authorized employees only	Private and restricted	Specific, corporate, proprietary
Extranet	Authorized groups from collaborating companies	Private and authorized outside partners	Shared in authorized collaborating groups

b. What are the different layers of TCP/IP protocol stack? Discuss their function briefly. **(7)**

Ans:

Layers in the TCP/IP protocol architecture

- Application Layer
- Host-to-Host Transport Layer,
- Network Access Layer,
- Internetwork Layer

Internetwork Layer

The best-known TCP/IP protocol at the internetwork layer is the *Internet Protocol (*IP*),* which provides the basic packet delivery service for all TCP/IP networks. In addition to the physical node addresses used at the network access layer, the IP protocol implements a system of logical host addresses called IP addresses.

Network Access Layer

The network access layer is the lowest layer in the Internet reference model. This layer contains the protocols that the computer uses to

Host-to-Host Transport Layer

The protocol layer just above the inter network layer is the host-to-host transport layer. It is responsible for providing end-to-end data integrity and provides a highly reliable communication service for entities that want to carry out an extended two-way conversation.

Application Layer

The top layer in the Internet reference model is the *application layer.* This layer provides functions for users or their programs, and it is highly specific to the application being performed. It provides the services that user applications use to communicate over the network, and it is the layer in which user-access network processes reside.

Q.37 a. Write short notes on following topics. **(7x2)**

i) Biometric Systems.

Ans:

Biometrics is the science of measuring physical properties of living beings.

Biometric Authentication

(1) Biometric authentication is the automatically recognition of a living being using suitable body characteristics.

(2) By measuring an individual's physical features in an authentication inquiry and comparing this data with stored biometric reference data, the identity of a specific user is determined.

Authentication means identification and verification

The most common biometrics is the following.
Face geometry (Photo): The computer takes the picture of your face & matches it with a presorted picture.
Fingerprints (Fingerscan): Whenever a user wants access, matching fingerprint against a template containing authorized person's fingerprints.
Hand Geometry: Like fingerprints except the verifier uses a TV like camera to take the picture of the user's hand.
Blood vessel pattern in the retina of a person's eye: A match is done between the pattern of the blood vessels in retina that is being scanned and prestored picture of retina.
Voice (Voice Print): A match between users voice and voice pattern stored on templates.
Signature: matched against the prestored authentication signature.
Keystroke Dynamics: Match of person's keyboard pressure and speed against prestored information.
Others: Like thermography, using a PIN & iris scan.

ii) SMTP and FTP.

Ans:

SMTP

Simple Mail Transfer Protocol, a protocol for sending e-mail messages between servers. Most e-mail systems that send mail over the Internet use SMTP to send messages from one server to another; the messages can then be retrieved with an e-mail client using either POP or IMAP.

SMTP is a relatively simple, text-based protocol, where one or more recipients of a message are specified (and in most cases verified to exist) and then the message text is transferred.

SMTP commands are generated by the sender-SMTP and sent to the receiver-SMTP. SMTP replies are sent from the receiver-SMTP to the sender-SMTP in response to the commands. In case a direct connection does not exist between the sender and the final destination, the message may be sent via one or more relay SMTP-servers. The relay SMTP-servers first acts as receivers and then relays the message to the next SMTP. To be able to provide the relay capability the SMTP-server must be supplied with the name of the ultimate destination host as well as the destination mailbox name.

FTP

FTP (File Transfer Protocol) is the protocol used on the Internet for sending files and is generally used for uploading / downloading files (web pages) to and from servers.

To begin an FTP session, you run the FTP client software and request the FTP server that you want to download files from. You can get FTP client software from the Internet.

Two popular examples of FTP clients are: Cute FTP and Leech FTP.

The FTP daemon runs on the FTP server. This daemon handles all FTP transactions. When a FTP client contacts a server, the daemon will ask for an account number (or username) and password. Many FTP sites let anyone log onto them to download files and software. This is called Anonymous FTP.

With anonymous FTP, you often use anonymous for your account number and your e-mail address for your password.

iii) E-brokerage.

Ans:

An e-brokerage is an investment house that allows you to buy and sell stocks and obtain investment information from its Web site.

For Banks the rewards of mastering e-brokerage can be substantial. Such capabilities can pay off in stronger customer relationships, particularly among the more affluent segments.

Many bankers and experts believe that as the various sectors of financial services gradually converge, customers will gravitate to providers who can blend banking and brokerage into one compelling value proposition

iv) Digital Signatures.

Ans:

A digital signature is an electronic rather than a written signature that can be used by someone to authenticate the identity of the sender of a message or of the signer of a document.

It can also be used to ensure that the original content of the message or document that has been conveyed is unchanged.

Additional benefits to the use of a digital signature are that it is easily transportable, cannot be easily repudiated, cannot be imitated by someone else, and can be automatically time-stamped.

A digital signature can be used with any kind of message, whether it is encrypted or not, simply so that the receiver can be sure of the sender's identity and that the message arrived intact.

A digital certificate contains the digital signature of the certificate-issuing authority so that anyone can verify that the certificate is real.

Code: AT-15 **Subject: INTERNET & WEB TECHNOLOGY**
PART-I, VOL-I

TYPICAL QUESTIONS & ANSWERS

OBJECTIVE TYPE QUESTIONS

Each Question carries 2 marks.

Choose the correct or best alternative in the following:

Q.1 A header in CGI script can specify

(A) format of the document. **(B)** new location of the document.
(C) (A) & (B) both. **(D)** start of the document.

Ans: A
A header in CGI script can specify- Format of the document & New location of the document.

Q.2 All exceptions in Java are subclasses of built in class called

(A) Exception **(B)** Error.
(C) Throwable. **(D)** Raise.

Ans: C
All exception in Java are subclasses of built in class called Throwable.

Q.3 In 32bit IP Addressing scheme all 1's represent

(A) this computer. **(B)** directed broadcast.
(C) limited broadcast. **(D)** loop back.

Ans: C
In 32 bit IP Addressing scheme all 1's represent limited broadcast.

Q.4 DMSP stands for

(A) Distributed Mail System Protocol
(B) Distributed Message System Protocol
(C) Distributed Message System Pool
(D) Distributed Mail System Pool

Ans: A
DMSP stands for – Distributed Mail system Protocol.

Q.5 Which Layer is not present in TCP/IP model?

(A) Application Layer **(B)** Internet Layer
(C) Transport Layer **(D)** Presentation Layer

Ans: D
Presentation layer is not present in TCP/IP Model.

Q.6 Let most segment of a name inn DNS represents

(A) Individual Network. **(B)** Individual computer.
(C) Domain name **(D)** Network type.

Ans: B
Left Most segment of a name in DNS represents- Individual computer

Q.7 Address 192.5.48.3 belongs to

(A) class A. **(B)** class B.
(C) class C. **(D)** class D.

Ans: C
Address 192.5.48.3 belongs to class C.

Q.8 Unlike Ipv4, Ipv6 does not include the following field in thee base header

(A) Next Header field. **(B)** Field for Fragmentation information
(B) Flow Label. **(D)** Kind field.

Ans: B
Unlike Ipv4, Ipv6 does not include the Field for Fragmentation information in the base header.

Q.9 The term byte stuffing refers to:

(A) data stuffing used with character oriented hardware.
(B) data stuffing used with bit oriented hardware.
(C) data stuffing used with both (A) & (B)
(D) data stuffing used with byte oriented hardware.

Ans: A
The term byte stuffing refers to data stuffing used with character-oriented hardware.

Q.10 FDDI (Fiber Distributed Data Interconnect) is an example of

(A) token ring. **(B)** token bus
(C) star topology **(D)** multipoint network.

Ans: A
FDDI is an example of token ring.

Q.11 Hardware that calculates CRC(Cyclic Redundancy Check) uses:

(A) Shift register **(B)** Xor unit
(C) Both (A) & (B) **(D)** Instruction register

Ans: B
Hardware that calculates CRC uses shift register and Xor unit.

Q.12 In TCP protocol header "checksum" is of_____

(A) 8 bits **(B)** 16 bits
(C) 32 bits **(D)** 64 bis

Ans: B
In TCP protocol header checksum is of 16 bits.

Q.13 In IP addressing scheme, class used for multicasting is:

(A) Class A **(B)** Class B
(C) Class C **(D)** Class D

Ans: D
In IP addressing scheme, class used for multicasting is class D.

Q.14 CIDR stands for

(A) Classified Internet Domain Routing
(B) Classless Inter Domain Routing
(C) Classless Internet Domain Routing
(D) Classified Inter Domain Routing

Ans: B
CIDR stands for Classless Inter Domain Routing.

Q.15 The total number of class of IP address are

(A) 3. **(B)** 4.
(C) 5. **(D)** 9.

Ans: C
The total number of class of IP addresses are 5.

Q.16 Parent class of all Java classes is

(A) java.lang.system **(B)** java.lang.object
(C) java.lang.class **(D)** java.lang.reflect.object

Ans: B
Parent class of all Java classes is java.lang.object.

Q.17 Exceptions of type error inn JAVA are handled by

(A) User program **(B)** Java run time environment
(C) Operating system kerne **(D)** Interrupt

Ans: B
Exceptions of type error in JAVA are handled by JAVA run time environment.

Q.18 Error detecting method that can detect more errors without increasing additional information in each packet is

(A) checksum **(B)** even parity mechanism
(C) CRC **(D)** odd parity mechanism.

Ans: C
Error detecting method that can detect more errors without increasing additional information in each packet is CRC.

Q.19 A Network uses a star topology if

(A) Computers are arranged in a closed loop.
(B) All computers attach to a central point.
(C) All computers attach to a single long cable.
(D) Computers attach to multiple hierarchical cables.

Ans: B
A Network uses a star topology if all computers attach to a central point.

Q.20 MTU is specified by

(A) IP Datagram size
(B) Hardware technology
(C) TCP Segment size
(D) None of the above.

Ans: B
MTU is specified by hardware technology.

Q.21 Network address prefixed by 1110 is a

(A) Class A address
(B) Multicast address
(C) Class B address
(D) Reserve address.

Ans: B
Network address prefixed by 1110 is a multicast address.

Q.22 FTP does not use

(A) Two transfer mode.
(B) Control connection to remote computer before file can be transferred.
(C) User Datagram Protocol.
(D) Authorization of a user through login and password verification.

Ans: C
FTP does not use User Datagram Protocol.

Q.23 A Header in CGI document can represent

(A) format of the document
(B) location if document used to different URL
(C) both (A) & (B)
(D) None of the above.

Ans: B
A header in CGI document can represent format of the document and the location if document used to different URL.

Q.24 127.0.0.1 is a

(A) limited broadcast address
(B) direct broadcast address
(C) multicast address
(D) loop-back address

Ans: D
127.0.0.1 is a loop-back address.

Q.25 In cyclic redundancy checking CRC is the

(A) divisor **(B)** quotient.
(C) dividend **(D)** remainder.

Ans: D
In cyclic redundancy checking CRC is the remainder.

Q.26 Which one of the following uses the greatest number of layers in the OSI model?

(A) Bridge **(B)** Repeater.
(C) Router. **(D)** Gateway.

Ans: D
Gateway uses the greatest number of layers in the OSI model.

Q.27 Which of the following 802 standard provides for a collision free protocol?

(A) 802.2 **(B)** 802.3
(C) 802.5 **(D)** 802.6

Ans: C
802.5 standards provides for a collision free protocol.

Q.28 The addressing especially used by Transport Layer is

(A) Station address **(B)** Network address
(B) Application port address **(D)** Dialog address

Ans: B
The addressing specially used by transport layer is application port address.

Q.29 Which one of thee following is an error reporting protocol?

(A) ARP **(B)** ICMP
(C) TCP **(D)** UDP

Ans: B
ICMP is an error reporting protocol.

Q.30 Which type of web document is run at the client site

(A) Static
(B) Dynamic
(C) Active
(D) All of the above

Ans: **C**
Active web document is run at client side.

Q.31 The main function of a browser is to

(A) compile HTML
(B) interpret HTML
(C) de-compile HTML
(D) interpret CGI programs

Ans: **B**
The main function of a browser is to interpret HTML.

Q.32 Which of the following is associated with SNMP

(A) SMI
(B) BER
(C) DNS
(D) MIB

Ans: **D**
MIB is associated with SNMP.

Q.33 ATM is an example of

(A) Ring topology
(B) Star topology
(C) Bus topology
(D) None of the above.

Ans: **B** Star topology

Q.34 The first part of the address in electronic mailbox identifies:

(A) User's mailbox
(B) Computer on which mail box resides
(C) Both (A) & (B)
(D) None of the above

Ans: **A** User's mailnbox.

Q.35 Protocol used to monitor and control network devices operates at:

(A) Application layer
(B) Transport layer
(C) Network layer
(D) Data Link layer

Ans: **A** Application layer.

Q.36 DHCP stands for

(A) Dynamic Host Control Protocol
(B) Dynamic Host Configuration Protocol.
(C) Dynamic Host Connection Protocol.
(D) None of the above.

Ans: B Dynamic Host Configuration Protocol.

Q.37 The transport protocol used by TFTP (Trivial File Transfer Protocol) is:

(A) FTP **(B)** UDP
(C) TCP **(D)** IP

Ans: B UDP.

Q.38 The Environment variable SCRIPT_NAME in CGI script specifies:

(A) Domain name of the computer running o server
(B) The path of URL after server name.
(C) Name of the server
(D) None of the above.

Ans: B The path of URL after server name.

Q.39 Application layer (layer 4) in TCP/IP model corresponds to:

(A) Layer 4 and 5 in OSI model **(B)** Layer 5 and 6 in OSI model
(C) Layer 6 and 7 in OSI model **(D)** Layer 1 and 2 in OSI model

Ans: C Layer 6 and 7 in OSI model.

Q.40 UDP (User Diagram Protocol) is

(A) Connectionless **(B)** Message Oriented
(C) Connection oriented **(D)** Both (A) and (B)

Ans: D Both (A) and (B).

Q.41 A network address prefixed by 1000 is:

(A) Class A address **(B)** Class B address
(C) Class C address **(D)** Class D address

Ans: B Class B address.

Q.42 In Java System.out is an object of type

(A) InputStream
(B) PrintStream
(C) OutputStream
(D) None of the above.

Ans: **B** PrintStream.

DESCRIPTIVE TYPE QUESTIONS

Q.1 What are the various parameters inside Applet tag in a HTML file? **(6)**

Ans:

```
<APPLET
    [CODEBASE= codebaseURL]
        CODE=applet file
        [ALT=alternate text]
        [NAME=applet instance name]
         WIDTH=pixels HEIGHT= pixels
        [ALIGN = alignment]
        [VSPACE= pixels][HSPACE = pixels]
    >
    [<PARAM NAME = Attribute name VALUE = Attribute value>]
    ……
</APPLET>
```

CODE BASE
Optional attribute used to specify the base URL of the applet code, which is the Directory that will be searched for applet's executable class file. If the applet Resides in the same directory as HTML file then this attribute is not required.

CODE
This is the requirement attribute used to specify the name of the applet class to be loaded (name of the already compiled. class file).

ALT
Optional attribute used to specify a short text message that should be displayed If browser understands the APPLET tag but can not currently run Java applets.

NAME
Optional attribute used to specify a name for applet instance so that the other applets on the page may refer to this applet.

ALIGN
Optional attribute used for alignment (LEFT, RIGHT, TOP, BOTTOM, MIDDLE)

VSPACE & HSPACE
VSPACE specifies space in pixels above and HSPACE in pixels on each side of applet.

WIDTH & HEIGHT
Required attributes that give the size of display area in pixels.

PARAM NAME & NULL
The PARAM tag allows you to specify applet specific arguments in an HTML page.

Q.2 Write an applet which accepts two integers from the user and displays their sum in the following format. **(8)**

Input a number in each box:

The sum is : 5

Ans:

```
import java .awt.*;
    import java. Applet.*;
    public class sum extends Applet
     {
          TextField text1,text2;
          Public void init()
          {
               text1 = new TextField(8);
               text2 = new TextField(8);
               add(text1);
               add(text2);
               text1.setText ("0");
               text2.setText ("0");
          }
    public void paint (Graphics g)
          {int x = 0, y= 0, z = 0;
           String s1, s2,s;
           g.drawstring("Input a number in each box",10,50);
           s1 = text1.getText();
           x = Integer.parseInt(s1);
           s2 = text2.getText();
           y = Integer.parseInt(s2);
           z = x+y;
```

```
            s = String.valueOf(z);
            g.drawString("The sum is :",10,75);
            g.drawString(s,100,75);
            }
    }
```

Q.3 What are the main differences between OSI and TCP/IP reference models? Explain briefly. **(8)**

Ans:

We will be focusing only on the key differences between the two references models.

Three concepts are central to OSI model: services, interfaces and protocols. OSI model makes the clear distinction between these three concepts.

The TCP/IP model did not originally clearly distinguish between services, interface, and protocol. For example the only real services offered by the Internet layer are SEND IP packet and RECEIVE IP packet.

The OSI reference model was devised before the protocols were invented. This ordering means that model was not biased towards one particular set of protocols, which made it quite general.

With TCP/IP reverse was true: the protocol came first, and the model was really just a description of the existing protocols. So problem was model did not fit for any other protocol stack.

Another difference is in the area of connectionless versus connection-oriented communication. The OSI model supports both connectionless and connection oriented communication in network layer, but only connection oriented in the transport layer. The TCP/IP model has only connection less mode in network layer but supports both the mode in transport layer.

Q.4 Define a socket? How read and write is performed using sockets? **(6)**

Ans:

An application program interface specifies the details of how an application program interacts with protocol software. Socket API is a defacto standard. Once a socket has been established the application can transfer information.

recv() and send() are used to read and write the data.

recv(socket, buffer, length, flags)

The socket is the descriptor of the socket, buffer specifies the address in memory where incoming message should be placed and length specifies the size of the buffer, flags allows the caller to control details.

send(socket, data, length, flags)

Here data is the address of data to be sent and other arguments are same.

Sockets also allows read() and write() to transfer data like send () and recv(). read() and write() have three arguments: a socket descriptor, the location of the buffer in the memory and the length of the memory buffer.

Q.5 How optimization is achieved in DNS? **(7)**

Ans:

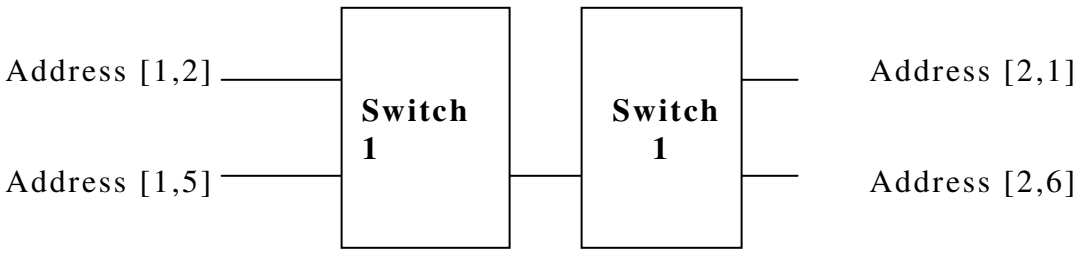
There are two primary optimizations used in DNS: replication and caching. Each root server is replicated; many copies of the server exist around the world. When a new site joins the internet, the site configures its local DNS server with a list of root server. The site server uses whichever root server is most responsive at a given point of time. In DNS caching each server maintains a cache of names. Whenever it looks up a new name, the server places a copy of the binding in its cache. Before contacting another server to request a binding , the server checks its cache, if the cache contains the answer the server uses the cached answer to generate a reply.

Q.6 How physical addressing is performed in WAN? **(7)**

Ans:

WAN networks operate similar to a LAN. Each WAN technology defines the exact frame format a computer uses when sending and receiving data. Each computer connected to a WAN is assigned a physical address. When sending a frame to another computer, the sender must supply the destination's address.

Many WANs use a hierarchical addressing scheme that makes forwarding more efficient. Hierarchical addressing scheme divides an address into multiple parts. The simplest scheme divides address in to two parts; the first part identifies packet switch, and second part identifies computer attached to that packet switch.



Example of Hierarchical addresses in WAN

The above figure shows each address as a pair of decimal integers. A computer connected to port 6 on packet switch 2 is assigned address [2,6].

Q.7 How do you make an image clickable in HTML? Give an example. **(6)**

Ans:
To make an image or text clickable hyperlinks are used, which use the <A> and </A> tags. This tag has various parameters, including HREF(the URL), NMAE(the hyperlink name), and METHODs(access methods).

As an example consider the following HTML fragment:

<A HREF = "http://www.foobar.com"> Foobar Home Page</A>
when a page with this fragment is displayed, following will appear on the screen:

Foobar Home Page

If the user clicks on this, the browser immediately fetches the page whose URL is http://www.foobar.com and displays it. Now we put a image in place of text.
<A HREF = "http://www.foobar.com"> <IMG SRC = "img1.gif" ALT = "Foobar" </A>
when displayed this page shows a picture(img1.gif). clicking on the picture switches to foobar home page just as in previous example.

Q.8 Design a form for a publishing house called foobar that allows the books to be ordered via the Internet. The form should include the customer's name, address, phone no. and Boo's title, author and edition. Payment has to be made in cash on delivery so no credit card information is needed. **(8)**

Ans:

Book Title :

Author :

Edition :

Customer's Name :

Address :

Phone Number :

SUBMIT RESET

```
<html>

<head>
<title>New Page 1</title>
</head>
<body>
<H2 align="center"><b><u>FORM TO PURCHASE BOOK
ONLINE</u></b></H2>
<FORM METHOD ="POST" ACTION ="/cgi-bin/formmail">
<input type="hidden" NAME = "SUBJECT" VALUE = "FORM">
<INPUT TYPE="HIDDEN" NAME = "REDIRECT" VALUE ="FORM">
<p><STRONG>Book Title
:</STRONG>        &
nbsp;      
<INPUT NAME  ="NAME" TYPE = "TEXT" SIZE="50">
</p>
<p><STRONG>Author
:   </STRONG>     &
nbsp;         
   
<INPUT NAME  ="NAME" TYPE = "TEXT" SIZE="50">
</p>
<p><b>Edition
:         &nbs
p;         &nb
sp; 
</b><INPUT NAME  ="NAME1" TYPE = "TEXT" SIZE="11">
</p>
<p><b>Customer's Name :    </b><INPUT NAME
="NAME2" TYPE = "TEXT" SIZE="50">
</p>
<p><b>Address 
:         &nbs
p;        
</b><INPUT NAME  ="NAME3" TYPE = "TEXT" SIZE="50">
</p>
```

```
<p><b>Phone Number 
:        </b><INPUT NAME
="NAME4" TYPE = "TEXT" SIZE="18">
</p>

</form>

<form method="POST" action="_derived/nortbots.htm" webbot-
action="--WEBBOT-SELF--"
onSubmit="location.href='_derived/nortbots.htm';return
false;" WEBBOT-onSubmit>

<p>         &n
bsp;         &
nbsp;         
          
;

<input type="submit" value="SUBMIT"
name="B1">        &
nbsp;
  <input type="reset" value="RESET" name="B2"></p>
</form>
<p> </p>

</body>

</html>
```

Q.9 How non-textual information is contained in a web page? **(7)**

Ans:
Non-textual information such as a graphics image or digitized photo is not inserted directly in a HTML document. Instead the data resides in a separate location, and the document, and the document contains a reference to the data. When the browser encounters such a reference, the browser goes to the specified location, obtains a copy of the image, and inserts the image in the specified document.

Q.10 When web pages containing emails are sent out they are prefixed by MIME Header. Why? **(7)**

Ans:
Initially email consisted messages containing simple text written in English and expressed in ASCII. Now a days on world wide internet messages can be sent in languages with accents like French and German, languages without

alphabet like Chinese and Japanese etc. the basic idea of MIME is to add structure to the message body and define encoding rule for non- ASCII messages.

MIME defines five additional message headers to the RFC 822 format.

Header	Meaning
MIME Version	Identifies the MIME version
Content Description	Readable string telling about message
Content-ID	Unique Identifier
Content transfer encoding	How the body is wrapped for transmission
Content Type	Nature of the message

Q.11 What is trivial file transfer protocol. Explain briefly? **(5)**

Ans:

Trivial File Transfer Protocol (TFTP) is useful for bootstrapping a hardware device that does not have a disk on which to store system software. All the device needs is a network connection and a small amount of read only memory (ROM) into which TFTP, UDP and IP are hardwired. Although TFTP is les powerful than FTP. TFTP does have two advantages. First, TFTP can be used in environments where UDP is available, but TCP is not. Second the code for TFTP requires less memory than the code for FTP.

Q.12 Why Gateways are used during mail transfer? **(5)**

Ans:

Email using SMTP works best when both the sender and the receiver are on the internet and can support TCP connections between sender and receiver. However many machines that are not on the internet still want to send and receive email from internet sites. For example many companies intentionally remove themselves for security reasons.

Another problem occurs when the sender and receiver speaks different protocols so direct communication is impossible.

Both of these problems are solved using application layer email gateways.

Q.13 An SNMP integer whose value is 200 has to be transmitted. Show its representation in ASN.1 syntax. **(4)**

Ans:
An ASN.1 transfer syntax defines how values of ASN.1 types are unambiguously converted to a sequence of bytes for transmission. Every value transmitted consists of up to four fields

a. identifier type
b. the length of data field in bytes
c. the data field
d. the end of content flag, if data length is unknown.

The last one is forbidden by SNMP, so we will assume data length is always known.

Tag	**length**	**value**
0 0 0 0 0 0 1 0	**0 0 0 0 0 0 0 1**	**1 1 0 0 1 0 0 0**

Integer 200

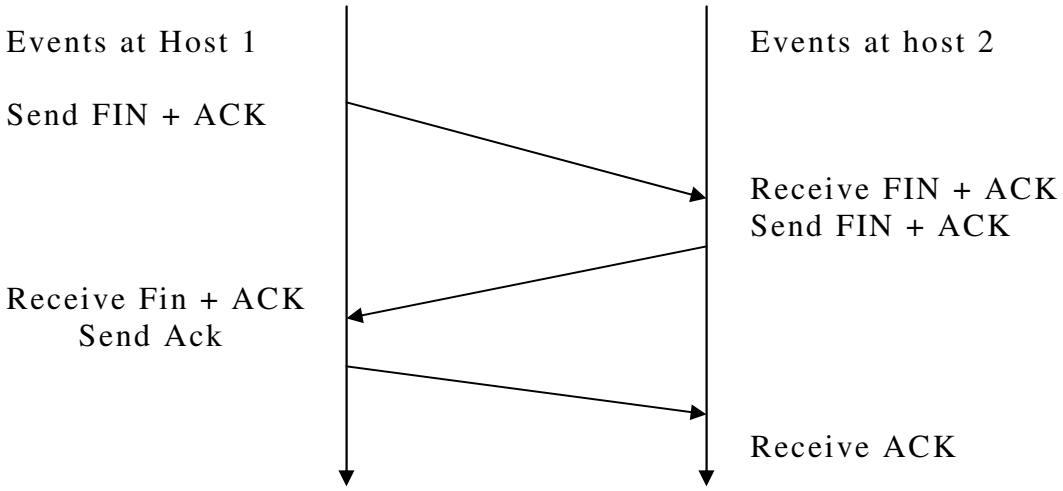
Q.14 Is TCP checksum necessary or could TCP allow IP to checksum the data. **(7)**

Ans:
Yes, TCP Checksum is necessary.
TCP layer is responsible for error detection, error control, retransmission of packets if required, reassembly of packets as well as their fragmentation. Hence for all error control and detection purposes TCP Checksum is essential.

TCP cannot allow IP to checksum data however IP has its own checksum for its header. IP layer is basically responsible for routing of IP datagrams immaterial of whether that packet is intended for TCP services or UDP services. Thus immaterial of what information is contained in data part , IP layer is only responsible for routing of packets and all the issues related to error control, error detection , flow control with regards to Routing only. Hence IP does not have a checksum for data unlike TCP.

Q.15 Explain Three-Way Handshake Mechanism used by TCP to terminate a Session reliably. **(7)**

Ans:
To guarantee that connection are established or terminated reliably, TCP uses 3-way handshake in which three messages are exchanged. TCP uses the term synchronization segment (SYN segment) to describe messages in a 3-way handshake used to create a connection , and the term FIN segment(short for finish) to describe messages in 3-way handshake to close a connection.



Threeway handshake used to close a connection

Q.16 Explain the significance of init() and destroy() methods of an applet? Also explain two ways of invoking an applet. **(7)**

Ans:
The init() method is used for basic initialization inside the applet. It is executed only once during the life time of the applet. This is the first method to be invoked when applet is started.
The destroy() method is used to clear the space from the memory when applet is stopped finally.
There are two ways to invoke an applet
- Executing the applet within a java compatible web browser or
- Using an Applet viewer

To execute an applet in a web browser a short HTML text file is written. Following is the HTML file to execute SimpleApplet:

<applet code = "SimpleApplet" width =200 height =60> </applet>

the width and height specifies the dimensions of the display area used.

To execute SimpleApplet with an applet viewer we will execute HTML file shown above. For example if preceding HTML file is called app.HTML then the following command will execute the SimpleApplet:

C:\<appletviewer app.HTML

Q.17 Why does IPV6 use separate extension headers? Explain. **(7)**

Ans:
The extension headers in Ipv6 are used for economy and extensibility. Partitioning the datagram functionality in to separate headers is economical because it saves space. Also having separate headers in Ipv6 makes it possible to define large set of features without requiring each datagram header to have at least one field for each feature.

Extensibility comes in to existence when new features are required to be added to a protocol. A protocol like Ipv4 needs complete change, the header must be redesigned to accommodate new features. In Ipv6 however, existing protocol header remains unchanged. A new header field can be defined to accommodate new change**.**

Q.18 How address resolution is performed with table lookup? Explain with thee help of a suitable example. **(7)**

Ans:
The table lookup approach to address resolution requires a data structure that contains information about address bindings. The table consists of an array. Each entry in the array contains a pair [P,H] where P is a protocol address and H is the equivalent hardware address. A separate address binding table is used for each physical network. Consequently all IP addresses in a given table have the same prefix. For example the following address binding table corresponds to a network with the class C number 197.15.3.0. therefore, each IP address in the table will begin with 197.15.3 prefix. The chief advantage of this table lookup approach is that a table can store the address bindings for an arbitrary set of computers on a given network.

IP Address	Hardware Address
197.15.3.2	0A:07:4B:12:82:36
197.15.3.3	0A:9C:28:71:32:6D
197.15.3.4	0A:11:C3:68:01:99

An example address binding table

Q.19 Write a CGI program that displays a count of how many times a browser on each computer has contacted the server. **(7)**

Ans:

```
echo Content-type: text/html
echo

N=$QUERY_STRING
Echo "<HTML>"

Case "x$N" in

x)      N = 1
Echo "This is the initial page.<BR><BR>"
;;
   X[0-9]*)  N = 'expr$n + 1'
             Echo "you have displayed this page $N times.<BR><BR>
                ;;

   *)     echo "The URL you used is invalid </HTML>"

          exit 0
          ;;
   esac
   echo "<A HREF=\http://$SERVER_NAME$SCRIPT_NAME?$N\>"
   echo " Click here to refresh the page. </ A></HTML>"
```

Q.20 Write short notes on the following: **(14)**

(i) MIB variables

Ans:

MIB (Management Information Base) variables:

MIB is a set of named items that an SNMP agent understands. To monitor or control a remote computer, a manager must fetch or store values to MIB variables. Because SNMP does not specify a set of MIB variables, the design is flexible. The separation of communication protocol from the definition of the objects permits to define MIB variable as needed. there are MIB variables that corresponds to protocols like UDP, TCP, IP and ARP, as well as MIB variables for network hardware such as Ethernet. In addition to simple variables such as integers that corresponds to counters, a MIB can include a variable that corresponds to a table or an array.

(ii) Circular Dependencies

Ans:

Circular Dependencies**:**

To understand the problem of circular dependencies consider a file server that uses a timeserver to obtain the current time whenever a file is accessed. Circular dependencies can occur if the timeserver also uses the file server. For example suppose a programmer is asked to modify the time server so it keeps a record of each request. If the programmer choose to have the time server become a client of the file server, a cycle can result; the file server becomes a client of the time server which becomes a client of the file server, and so on**.**

(iii) UDP

Ans:

UDP(User Datagram Protocol) :

UDP uses a connectionless communication paradigm. That is, an application using UDP does not need to preestablish a connection before sending data, nor does application need to terminate communication when finished. Furthermore, UDP allows an application to delay an arbitrarily long time between the transmission of two messages. UDP does not use any control messages. Communication consists only of the data messages themselves.

(iv) RPC

Ans:

RPC (Remote Procedure Call) **:**

The facility that was created to help the programmers write client-server software is known as Remote Procedure Call. In RPC instead of giving a programmer explicit communication primitives such as the socket interface, hide communication from the programmer by using a conventional programming language facility. The programming mechanism chosen is a procedure call. The RPC mechanism allows a programmer to place procedures on two or more machines, and automatically generates code that will allow a procedure call to pass from one computer to another.

(v) CGI standard

Ans:

The CGI Standard:

Technology used for building dynamic web documents is known as Common Gateway Interface (CGI). The CGI standard specifies how a server interacts with an application program that implements dynamic documents. The application is called a CGI program.

CGI provides general guidelines and allows a programmer to choose more details. For example, CGI does not specify a particular programming language. Instead the standard permits a programmer to choose an appropriate language for each dynamic document. For example a programmer can use a conventional programming language like C for documents that requires extensive computation, and use a scripting language like perl for documents that require only minor text editing.

(vi) Token Ring.

Ans:

Token Ring:

A token ring is a collection of individual point-to-point links that happen o form a circle. In a token ring a special bit pattern, called the token, circulates around the ring whenever all stations are idle. When a station wants to transmit a frame, it is required to seize the token and remove it fro the ring before transmitting. This action is done by inverting a single bit in the 3-byte token, which instantly changes it into the first 3 bytes of a normal data frame. Because there is only one token one station can transmit at a given instant.

Q.21 Write an applet that sets the background colour to cyan and foreground colour to red and displays a message that illustrates the order in which various applet methods are called when an applet starts up. For example : inside init()...inside start()... **(7)**

Ans:

```
import java.awt. *
 import java.applet. *

public class Method_order Extends Applet
    {
String msg;
//Sets the background and foreground color.public void init( )
{
setBackground(Color.cyan);
          setForeground(Color.red);
msg =" inside init 0- -";
}
//Initialize the string to be displayed.
 public void init( )
    {
msg =" inside startO- - ";
  }

lSets the background and foreground color.
    public void paint(Graphics g )
    {

    msg =" inside paint()-";
g.drawstring(msg, 10 , 30);
}
}
```

Q.22 How exceptions are handled in java? / Explain with the help of suitable example. **(7)**

Ans:

Exception Handling In Java:

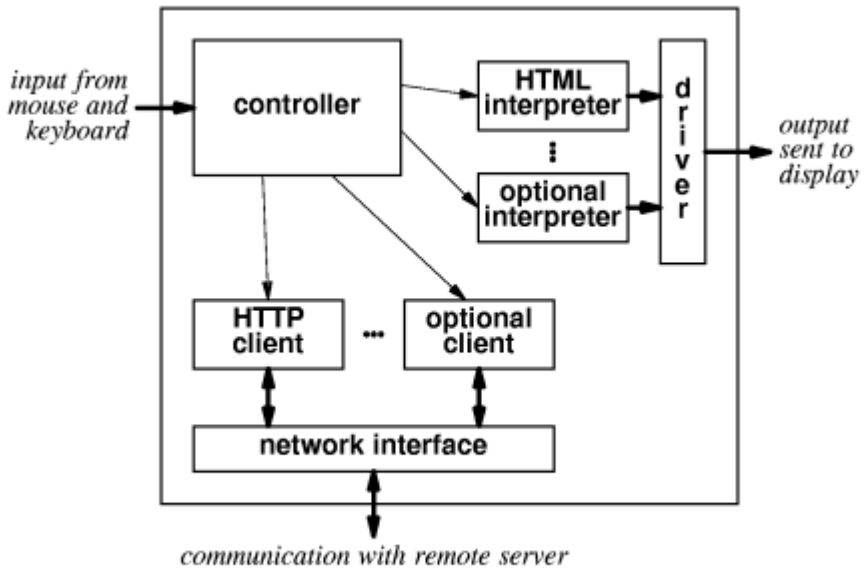
A java exception is an object that describes an exceptional condition that has occurred in a piece of code. When an exceptional condition arises, an object representing that exception is created and thrown in the method that causes the error. That method may choose to handle the exception itself or may pass it on. Either way, at some point exception is caught and processed.

Java exception handling is managed by five keywords: try, catch, throw, throws, and finally. The general form of an exceptional handling block is as follows:

```
try
{
//Block of code to monitor errors.

}
catch(Exception1 e)
{
//Block of code to handle Exception1.

}
catch(Exception2 e)

{
//Block of code to handle Exception2..

}

finally
{

//Block of code to be executed before try block ends.

}
```

Q.23 What are the major components of a web browser? Draw a neat diagram to explain them. **(7)**

Ans:



The Major Component of a Web browser

A browser consists of a set of clients, a set of interpreters, and a controller that manages them. Each browser must contain an HTML interpreter to display the document. Other interpreters are optional. The controller forms the central piece of browser. It interprets both mouse click and key board input, and calls other components to perform operations specified by the user.

Q.24 How aliases are used in DNS? Explain. **(7)**

Ans:
CNAME entries are analogous to a symbolic link in a file system- the entry provides an alias for another DNS entry. Foobar Corporation has two computers named hobbes.foober.com and calvin.foobar.com. further suppose that Foobar decide to run a web server and wants to follow the convention of using the name www for the computer that runs the organization's Web server. Although the organizations could choose to rename one of their computers (e.g. hobbes), a much easier solution exists: the organization can create a CNAME entry for www.foobar.com that points to hobbes. Whenever a resolver sends a request for www.foobar.com , the server returns the address of computer hobbes.

The use of aliases is especially convenient because it permits an organization to change the computer used for a particular service without changing the names or addresses of the computers. For example Foobar Corporation can move its web service from computer

hobbes to computer calvin by moving the server and changing the CNAME record in the DNS server- the two computers retain their original names and IP addresses.

Q.25 What are the three basic types of web documents? Also explain the advantages and disadvantages of each type. **(8)**

Ans:
There are three basic types of web documents:

- static
- Dynamic
- Active

STATIC
A static web document resides in a file that is associated with a web server. The developer of static document determines the contents at. the time the document is written. Because contents do not change, each request for a static document results in exactly the same response.

DYNAMIC

A dynamic web document does not exist in predefined form: Instead a dynamic web document is created by a web server whenever a browser requests the document. When a request arrives, the web server runs an application program-that creates the dynamic document. Because a fresh document is created for each request, the contents of dynamic document can very from one request to another.

ACTIVE

An active document is not fully specified by the server. Instead, an active document consists of a computer program-that understands how to compute and display values. When a browser request an active document, the server returns a copy of the program that the browser must run locally. When it runs active documents can interact with the user and change the display continuously. Thus the contents of an active document are never fixed

ADVANTAGES AND DISADVANTAGES OF EACH DOCUMENT:

The chief advantages of a static document are simplicity, reliability and performance. A browser can display a static document quickly and place a copy in cache on a local disk to speedup the future request for the document.

The chief disadvantage of static document is inflexibility.

The advantage of an active document over a dynamic document lies in its ability to update the information continuously.

The chief disadvantages of active documents arise from the additional costs of creating and running such documents, and from a lack of security. The active document has a potential security risk because the document can export as well as can import the information.

The chief advantage of a dynamic document lies in its ability to report current information. For example a dynamic document can be used to report current stock prices, current weather conditions etc.

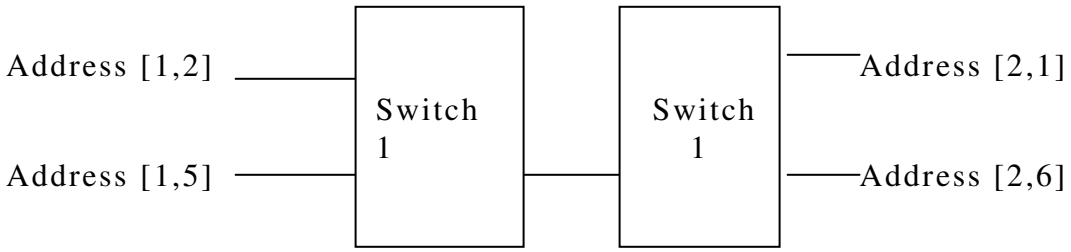
The chief disadvantages of dynamic document approach are increased cost and the inability to display changing information. A dynamic document takes slightly longer to retrieve than a static document because server requires additional time to run the application program that creates the documents.

Q.26 How physical addressing is performed in WAN? **(6)**

Ans:
WAN networks operate similar to a LAN. Each WAN technology defines the exact frame format a computer uses when sending and receiving data. Each computer connected to a WAN is assigned a physical address. When sending a frame to another computer, the sender must supply the destination's address.

Many WANs use a hierarchical addressing scheme that makes forwarding more efficient. Hierarchical addressing scheme divides an address into multiple parts. The simplest scheme divides address in to two parts; the

first part identifies packet switch, and second part identifies computer attached to that packet switch.



Example of Hierarchical addresses in WAN

The figure shows each address as a pair of decimal integers. A computer connected to port 6 on packet switch 2 is assigned address [2,6].

Q.27 Differentiate between http and ftp. **(5)**

Ans:
FTP and HTTP were developed to make Internet transmission better.

FTP is used to exchange files between computer accounts, to transfer files between an account and a desktop computer (upload), or to access software archives on the Internet. It's also commonly used to download programs and other files to your computer from other servers. It transfers files in two different formats ASCII for text files and Binary format for binary files. This allows a user to perform basic file and directory management operations such as deleting, copying, or renaming. Also, there is something

called Anonymous FTP used heavily today by several universities and private organizations. Anonymous FTP is a facility offered by many machines on the Internet. This permits you to log in with the user name 'anonymous' or the user name 'ftp'. When prompted for a password, type your e-mail address -- it's not necessary, but it's a courtesy for those sites that like to know who is making use of their facility. Be courteous. Some sites require a valid e-mail address, others don't.

HTTP is used primarily in today's society as a set of rules for exchanging files (text, graphic images, sound, video, and other multimedia files) on the World Wide Web. It also provides access to other protocols like FTP, SMTP, NNTP, WAIS, Gopher, Telnet, and TN3270. Essential concepts that are part of HTTP include (as its name implies) the idea that files can contain references to other files whose selection will elicit additional

transfer requests. Any web server machine contains, in addition to the HTML and other files it can serve, an HTTP daemon, a program that is designed to wait for HTTP requests and handle them when they arrive. Your Web browser is an HTTP client, sending requests to server machines. When the browser user enters file requests by either "opening" a Web file (typing

in a Uniform Resource Locator) or clicking on a hypertext link, the browser builds an HTTP request and sends it to the Internet Protocol Address indicated by the URL. The HTTP daemon in the destination server machine receives the request and, after any necessary processing, the requested file is returned.

Q.28 Write the HTML code to accomplish the web page: **(3)**

(i) Insert the frame extending 300 pixels across the page from left side.
(ii) Insert scrollable lists that will always display four entries of the list.
(iii) Insert an image onto a page using good.gif as and image and having "welcome" as the ALT text.

Ans:
(i)

```
< FRAMESET COLS = " 300 , *  "  >
          .  .  .
< /FRAMESET  >
```

(ii)

```
< SELECT SIZE = " 4 " >
      . . .
</SELECT>
```

(iii)

```
<IMG SRC = " good.gif "  ALT = " Welcome"
```

Q.29 Write the HTML code for the following table: **(6)**

TEMPERATURE				
CITIES	DELHI	MUMBAI	KOLKATTA	CHENNAI
MAXIMUM	21	35	43	50
MINIMUM	5	14	28	32

Ans:

```
<html>
<head>
<title>New Page 1</title>
</head>
<body>
<div align="center">
  <center>
  <table border="1" width="441" height="106">
    <tr>
      <td width="441" height="19" colspan="5">
        <p align="center"><b>TEMPERATURE</b></td>
    </tr>
    <tr>
      <td width="60" height="23">
        <p align="center">CITIES</td>
      <td width="82" height="23">
        <p align="center">DELHI</td>
      <td width="87" height="23">
        <p align="center">MUMBAI</td>
      <td width="77" height="23">
        <p align="center">KOLKATTA</td>
      <td width="86" height="23">
        <p align="center">CHENNAI</td>
    </tr>
    <tr>
```

```
        <td width="72" height="24">
          <p align="center">MAXIMUM</td>
        <td width="82" height="24">
          <p align="center">21</td>
        <td width="87" height="24">
          <p align="center">35</td>
        <td width="77" height="24">
          <p align="center">43</td>
        <td width="86" height="24">
          <p align="center">50</td>
      </tr>

      <tr>
        <td width="72" height="21">
          <p align="center">MINIMUM</td>
        <td width="82" height="21">
          <p align="center">5</td>
        <td width="87" height="21">
          <p align="center">14</td>
        <td width="77" height="21">
          <p align="center">28</td>
        <td width="86" height="21">
          <p align="center">32</td>
      </tr>
    </table>
    </center>
</div>
</body>
</html>
```

Q.30 Explain dynamic server creation briefly. **(7)**

Ans:

Dynamic Server Creation:

If a server handles one request at a time, all clients must wait while the server fulfills the one request. In contrast, a concurrent server can handle multiple requests simultaneously. When a request arrives, the server assigns the request to a thread of control that can execute concurrently with existing thread. The server program is constructed in two parts: one that accepts request and creates a new thread for the request, and another that consists of the code to handle an individual request. When a concurrent server start executing, only the first part runs. That is the main server thread waits for a request to arrive. When a request arrives, the main thread creates a new service thread to handle the request. The service thread handles one request and then terminates.

Q.31 What is socket inheritance? Explain. **(7)**

Ans:
Socket Inheritance:

In socket inheritance a reference count mechanism is used. When a socket is first created, the system sets the socket's reference count to 1;the socket exists as long as the reference count remains positive. When a program creates an additional thread, the system provides the thread with a list of all the sockets that program owns, and increments the reference count of each by 1. when a thread calls close for a socket, the system decrements the reference count on the socket by 1 and removes the socket from the thread's list.

The main thread of a concurrent server creates the socket that the server uses to accept incoming connections. When a connection request arrives, the system creates a new socket for the new connection. After a service thread finishes, it calls close on the new socket.

Q.32 How does a computer know whether an arriving frame contains an ARP message? Explain. **(7)**

Ans:
The type field in the frame header specifies that the frame contain an ARP message. A sender must assign an appropriate value for the type field before transmitting the frame and a receiver must examine the type field in each incoming frame. For example, the Ethernet standard specifies that the type field in an Ethernet frame carrying an ARP message must contain the hexadecimal value Ox806.

Dest. Address	source address	frame type	data in frame
		806	**complete ARP message**

Illustration of type field in a Ethernet header used to specify the frame contents. A value of 0x806 informs the receiver that the frame contains an ARP message.

Q.33 What is the chief advantage of using virtual packets instead of frames? **(7)**

Ans:
The router cannot transfer a copy of a frame from one type of network to another because the frame formats differ. More importantly, the router cannot simply reformat the frame header because the two networks may use incompatible address format.

To overcome heterogeneity, Internet protocol software defines an inter packet format that is independent of the underlying hardware. This is called virtual packet and can be transferred across the underlying hardware. The underlying hardware does not understand or recognize the Internet packet format, the protocol software creates and handles Internet packets.

Q.34 "A datagram cannot be larger than the MTU of a network over which it is sent." Is the statement true or false? Explain with the help of a suitable example. **(7)**

Ans:
Each hardware technology specifies the maximum amount of data that a frame can carry. This limit is known as maximum transmission unit(MTU). There is no exception to MTU limit, the network hardware is not designed to accept or transfer frame to carry more data than the MTU allows. Thus a datagram must be smaller or equal to network MTU or it cannot be encapsulated for transmission.

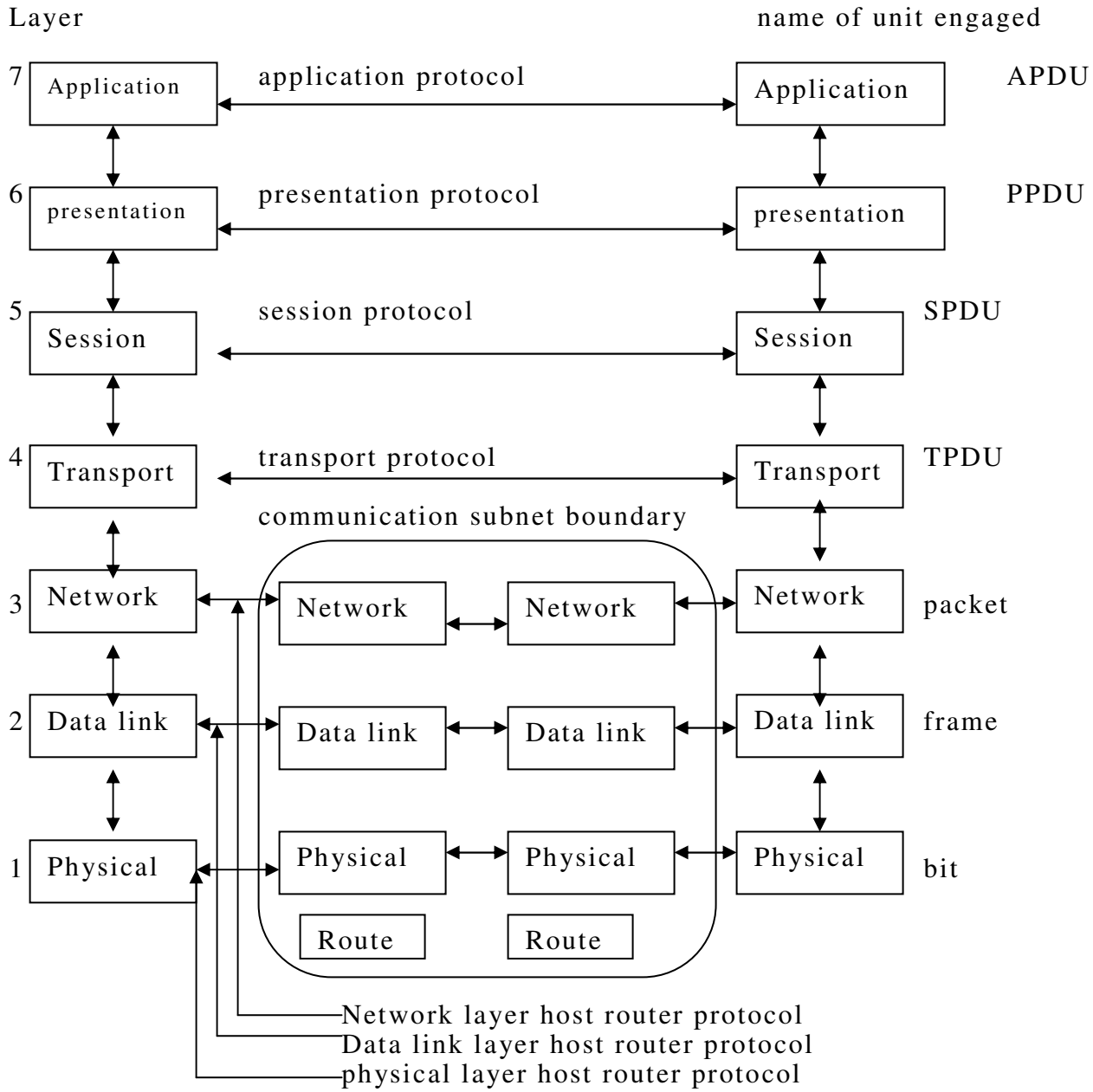
An Example of a router that connects two networks with different MTU values

In the figure host H2 attaches to a network that has an MTU of 1000. Therefore each datagram that H2 transmits must be 1000 octets or less. However, because host HI attaches to a network that has an MTU of 1500 octets, HI can transmit datagrams that contains up to 1500 octets. To solve this problem IP router uses a technique called fragmentation. When a datagram is larger than the MTU of a network over which it is sent, the router divides the

datagram into smaller pieces called fragments and sends each fragment independently. To fragment a datagram for transmission across the network, a router uses the network MTU and datagram header size to calculate maximum amount of data that can be sent in each fragment and number of fragment that will be needed.

Q.35 Draw a neat labeled diagram of the OSI reference model for computer networks showing all the layers and the communication subnet boundary. **(7)**

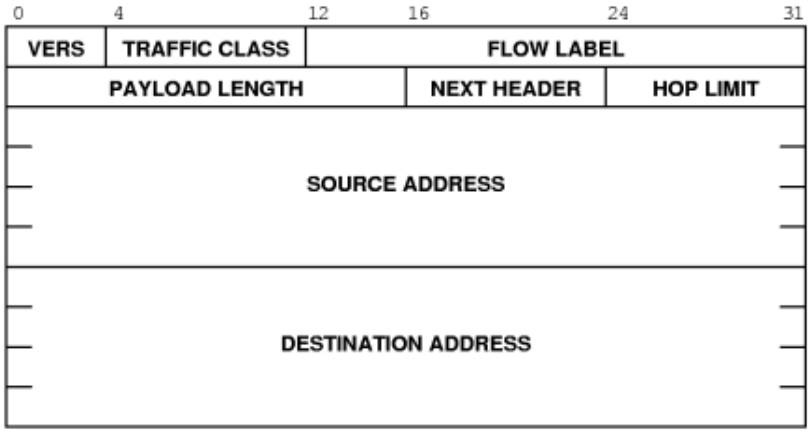
Ans:



The OSI reference Model

Q.36 Explain various fields in IPV6 base header? **(8)**

Ans:
Although IPv6 base header is twice as large as an IPv4 header, it contains less information. Following diagram illustrates the format:



The format of an Ipv6 base header

Most of the space in header is devoted to two fields that identify the sender and recipient. Each address occupies sixteen octets, four times than an IPv4 address.

In addition to source and destination address, the bas header contains six fields. The VERS field identifies the protocol as version 6. the PRIORITY field specifies the routing priority class. The PAYLOAD LENGTH field corresponds to IPv4's datagram length field. The HOP LIMIT corresponds to the IPv4 TIME-TO-LIVE field. IPv6 interprets the HOP LIMIT strictly.. field FLOW LABEL is intended for use with new applications that requires performance guarantees.

Q.37 Differentiate between adaptive and non-adaptive routing. **(6)**

Ans:
Adaptive routing describes the capability of a system, through which routes are characterised by their destination, to alter the path that the route takes through the system in response to a change in conditions. The adaptation is

intended to allow as many routes as possible to remain valid (that is, have destinations that can be reached) in response to the change.

People using a transport system can display adaptive routing. For example, if a local railway station is closed, people can alight from a train at a different station and use another method, such as a bus, to reach their destination.

Systems that do not implement adaptive routing are described as using non-adapting or static routing, where routes through a network are described by fixed paths (statically). A change, such a loss of a node, or loss of a connection between nodes, is not compensated for. This means that anything that wishes to take an affected path will either have to wait for the failure to be repaired before restarting its journey, or will have to fail to reach its destination and give up the journey.

Q.38 How congestion is controlled in TCP? **(7)**

Ans:
One of the most important aspects of TCP is a mechanism for congestion control. In most modern internets, packet loss or extreme long delays are more likely to be caused by congestion than a hardware failure. Interestingly, transport protocols that retransmit can exacerbate the problem of congestion by injecting additional copies of a message.

To avoid such a problem, TCP always uses packet loss as a measure of congestion and responds to congestion by reducing the rate at which it retransmits data.

TCP does not compute an exact transmission rate. Instead, TCP bases transmission on buffers. That is, the receiver advertises a window size and the sender can transmit data to fill the receiver's window before an ACK is received. To control the data rate, TCP imposes a restriction on the window size – by temporarily reducing the window size , the sending TCP effectively reduces the data rate.

TCP congestion control takes over when a message is lost. Instead of retransmit ting enough data to fill the receiver's buffer (the receiver's window size), TCP begins by sending a single message containing data. If the acknowledgement arrives without additional loss, TCP doubles the amount of data being sent and sends two additional messages. If acknowledgement arrive for those two , TCP sends four more and so on.

Q.39 While using FTP what is wildcard expansion in file names? **(7)**

Ans:
To make it easy for users to specify a set of file names, FTP allows a remote computer system to perform traditional file name expansion. The user enters an abbreviation, which FTP expands to produce a valid file name. In abbreviations, a wildcard character stands for zero or more characters. Many computer systems use the asterisk _* as a wildcard. On such systems, the abbreviation li*

Matches all file names that begin with the prefix li. Thus, if a remote computer contains the six files:

Dark light lonely crab link tuft

FTP will expand the abbreviation li* to two names: light and link. File name expansion can be especially useful with commands mget or mput because expansion makes it possible to specify a large set of files without entering each file name explicitly.

Q.40 Write short notes on the following: **(14)**

(i) Multihomed host.

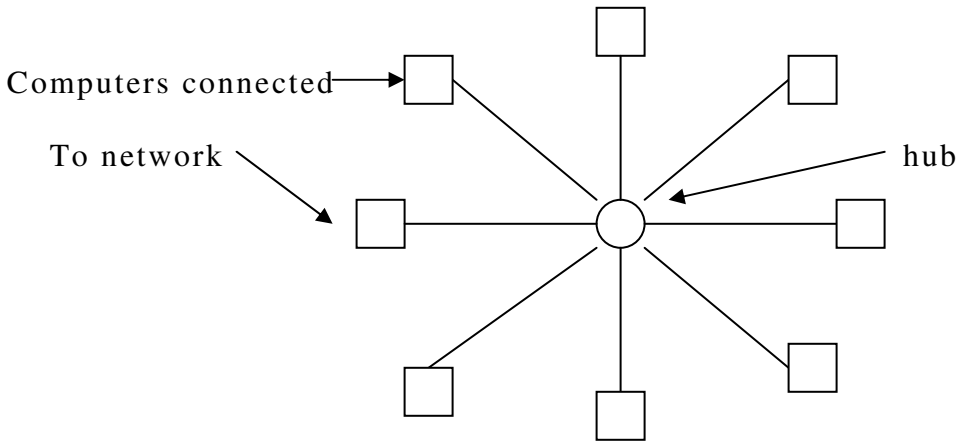
Ans:

A host computer that connects to multiple networks is called multihomed host. Multihoming is sometimes used to increase reliability. If one network fails, the host can still reach the Internet through the second connection. Alternatively multihoming is used to increase performance. Connections to multiple networks can make it possible to send traffic directly and avoid routers, which are sometimes congested. Like a router, a multihomed host has multiple protocol addresses, one for each network connection.

(ii) Star topology.

Ans:

A network uses a star topology if all computers attach to a central point. The following figure illustrates the concept:



Because a star shaped network resembles the spoke of a wheel, center of a star network is often called a hub. In practice, star network seldom have a symmetric shape in which the hub is located at equal distance from all computers. Instead a hub often resides in a location separate from the computers attached to it.

(iii) Remote Procedure Call (RPC).

Ans:

The facility that was created to help the programmers write client server software is known as Remote Procedure Call. In RPC instead of giving a programmer explicit communication primitives such as the socket interface, hide communication from the programmer by using a conventional programming language facility. The programming mechanism chosen is a procedure call. The RPC mechanism allows a programmer to place procedures on two or more machines, and automatically generates code that will allow a procedure call to pass from one computer to another.

(iv) E-mail gateways.

Ans:

Email using SMTP works best when both the sender and the receiver are on the Internet and can support TCP connections between sender and receiver. However, many machines that are not on the internet still want to send and receive emails from internet sites.

Another problem occurs when the sender speaks only RFC 822 and the receiver speaks only X.400 or some vendor specific protocol.
Both of these problems are solved using email gateways. Email gateways are used at application layer.

(v) CIDR.

Ans:

CIDR is a new addressing scheme for the internet which allows for more efficient allocation of IP addresses than old class A, B, and C addressing scheme. Instead of being limited to network identifier (or prefixes) of 8, 16, or 24 bits, CIDR currently uses prefixes any where from 13 to 27 bits. Thus, block of addresses can be assigned to a network as small as 32 hosts or to those with 500,000 hosts. This allows for address assignments that much more closely fit an organization's specific need.

A CIDR address includes the standard 32- bit address and also information on how many bits are used for the network prefix. For example in CIDR address 206.13.01.48/25, the /25 indicates that the first 25 bits are used to identify unique network leaving the remaining bits to identify the specific host.

(vi) HDLC.

Ans:

HDLC - High Level Data Link Control :

Protocol Overall Description:

Layer 2 of the OSI model is the data link layer. One of the most common layer 2 protocols is the HDLC protocol. The basic framing structure of the HDLC protocol is shown below:

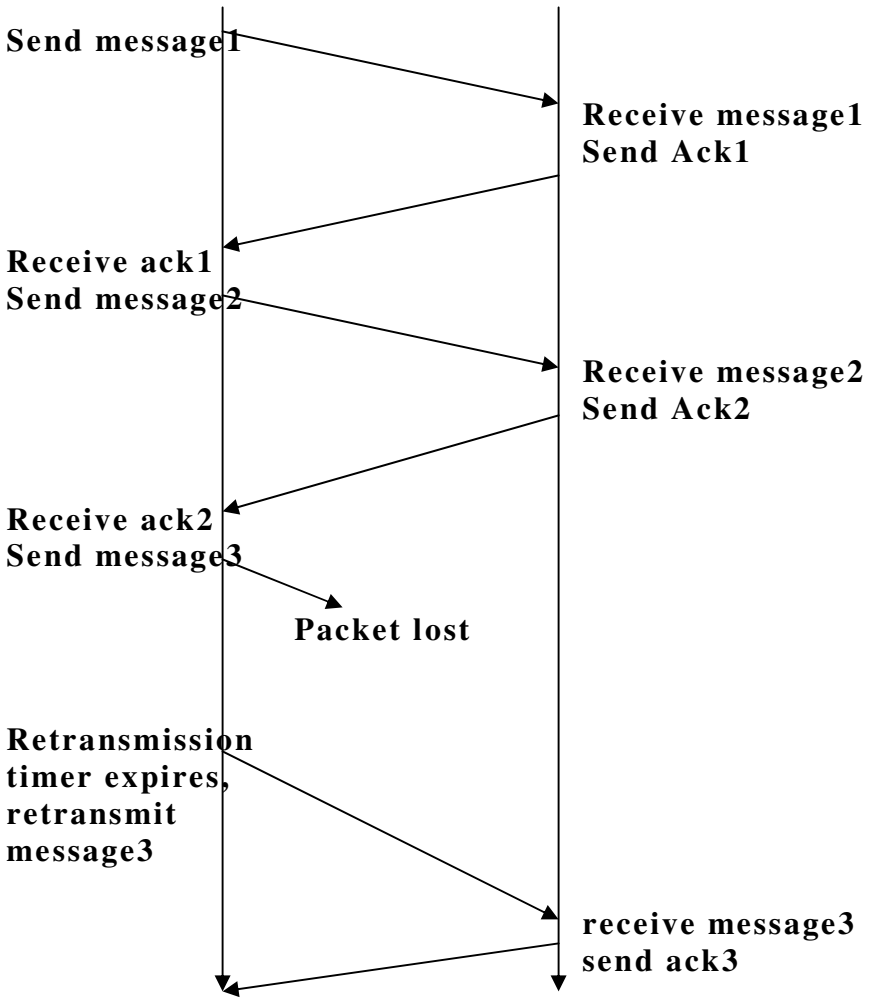
HDLC uses zero insertion/deletion process (commonly known as bit stuffing) to ensure that the bit pattern of the delimiter flag does not occur in the fields between flags. The HDLC frame is synchronous and therefore relies on the physical layer to provide method of clocking and synchronizing the transmission and reception of frames.The HDLC protocol is defined by ISO for use on both point-to-point and multipoint (multidrop) data links. It supports full duplex transparent-mode operation and is now extensively used in both multipoint and computer networks.

HDLC has three operational modes:

1. Normal Response Mode (NRM)
2. Asynchronous Response Mode (ARM)
3. Asynchronous Balanced Mode (ABM)

Q.41 How does TCP achieve reliability? **(4)**

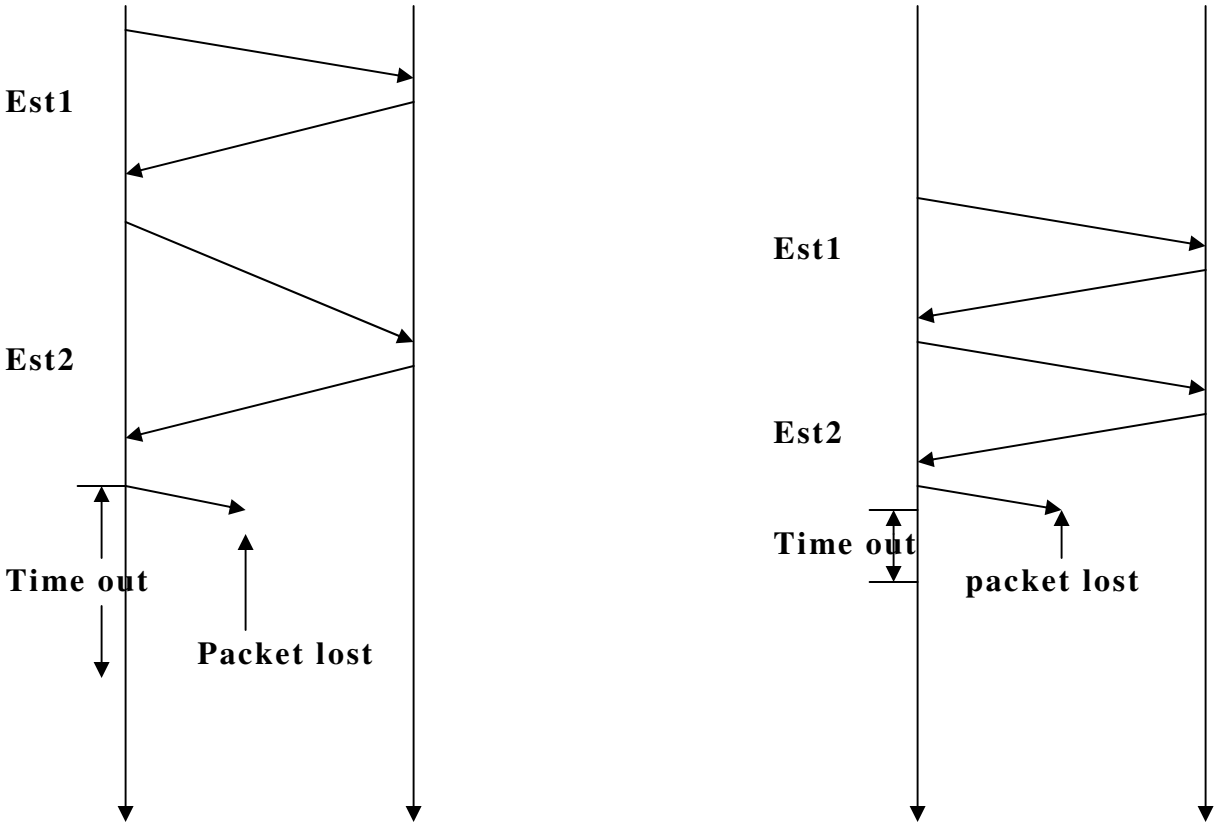
Ans:
One of the most important technologies is retransmission. When TCP stands data the sender compensates for packet loss by implementing a retransmission scheme. Both sides of a communication participate. When TCP receives data, it sends it acknowledgements back to the sender. Whenever it sends data, TCP starts a timer . if the timer expires before an acknowledgement arrives, the sender retransmits the data. The following figure illustrates retransmission.



TCP's retransmission scheme is the key to its success because it handles communication across an arbitrary Internet and allows multiple application programs to communicate concurrently.

Q.42 How adaptive transmission helps TCP to maximize throughput on each connection? **(5)**

Ans:
To understand how adaptive retransmission helps TCP maximize throughput on each connection, consider a case of packet loss on two connections that have different round-trip delay. For example, Figure given below illustrates traffic on two such connections.



Time out and retransmission on two connections that have different round trip delays.

As the figure shows ,TCP sets the retransmission timeout to be slightly longer than the mean round-trip delay. If the delay is large, TCP uses a large retransmission timeout; if the delay is small, TCP uses a small time out. The goal is to wait long enough to determine that a packet was lost, without waiting longer than necessary.

Q.43 Explain that the lost acknowledgement does not necessarily enforce retransmission of the packet. **(5)**

Ans:
To guarantee reliable transfer, protocols use positive acknowledgement with retransmission. When receiver gets the packet an acknowledgement is sent. If an acknowledgement is lost, generally packet is retransmitted. Retransmission can not succeed if a hardware failure has permanently disconnected the network or if receiving computer has crashed. Therefore, protocols retransmitting the messages bound the maximum number of transmissions. When the bound has been reached, the protocol stops retransmission of packet even if acknowledges is not received. So lost acknowledgement does not necessarily enforce retransmission of packet.

Q.44 Does it make sense for two domain servers to contain exactly the same set of names? Why or why not? **(5)**

Ans:
Yes. It is very advantageous for two domain servers containing same set of names. If there is only one server than traffic on one server would be in tolerable, because it would be the only server to receive all the request and handle them appropriately. If there is more than one server containing same set of data then geographically closet server will respond thus reducing the load on one server. Also if one server is down due to some problem then another server containing same set of data can be used to fulfill the incoming requests.

Q.45 Reassembling of IP fragments at the ultimate destination is advantageous. Give reasons. **(5)**

Ans:
Requiring the ultimate destination to reassemble the fragments has two main advantages. First, it reduces the amount of state information in routers. When forwarding a datagram, a router does not need to know whether a datagram is a fragment. Second, it allows routes to change dynamically. If an intermediate router reassembles fragments, all fragments would need to reach the router.

Q.46 What is the maximum number of fragments that can result from a single IP Datagram? Explain. **(4)**

Ans:
To fragment a datagram for transmission across a network, a router uses the network MTU (Maximum Transmission Unit) and the datagram header size to calculate the maximum amount of data that can be sent in each fragment and number of fragment that will be needed. The router than creates the fragments. A datagram can not be larger than the MTU of a network over which it is sent. If a fragment eventually reaches another network that has a smaller MTU then fragment is further divided in to smaller fragments. IP does not distinguish between original fragments and sub fragments. So the maximum number of fragments from a single datagram will depend on size of datagram and MTU of the networks over which it is sent along its path.

Q.47 Does a numeric mailbox identifier have any advantage over a mnemonic identifier? Explain. **(7)**

Ans:
Some software systems allow the system administrator to choose mailbox names, while other systems require a user's mail box identifier by concatenating a user's first name, middle initial and last name, with underscore to separate the three items. For example, the email address for employee john Quiggley Public at Foobar Corporation might be:

John_Q_Public@foober.com

On systems that require a user's login identifier to be used as a mailbox identifier, the resulting e-mail address is not nearly as readable. For example, if login accounts on a computer at nonexistent Corporation consist of two six-digit numbers separated by a period, an individual's e-mail address on that computer might be:

912743.253843@nonexist.com

Obviously mnemonic from makes the mailbox portion of an e-mail address easier to remember and enter correctly.

Q.48 What are the various address Translation schemes? Explain which scheme is used inn Internet? **(7)**

Ans:
Translation from a computer's protocol address to an equivalent hardware address is known as address resolution. Address resolution algorithms can be grouped into three basic categories:

Table lookup-

Bindings or mapping are stored in a table in memory, which the software searches when it needs to resolve an address.

Closed Form Computation-

The protocol address assigned to a computer is chosen carefully so the computer's hardware address can be computed from the protocol address using basic boolean and arithmetic operations.

Message Exchange-

Computer exchange messages across a network to resolve an address. One computer sends a message that requests an address binding (i.e., translation), and another computer sends a reply that contains the requested information.

TCP/IP can use any of the three methods; the method chosen for a particular network depends on the addressing scheme used by underlying hardware.

Generally third scheme is used over internet. The TCP/IP suite contains a standard address resolution protocol (ARP). ARP defines the format of the messages that computers exchange to resolve an address as well as rules for handling ARP messages.

Q.49 What are the three basic types of web documents? Discuss the advantages & disadvantages of each type. **(5)**

Ans:

Static-

A static web document resides in a file that is associated with a Web server. The author of a static document determines the contents at the time the document is written. Because the contents do not change, each request for a static document results in exactly the same response.

Dynamic-

A dynamic web document does not exist in a predefined form. Instead a dynamic document is created by a web server whenever a browser requests the document. When a request arrives the web server runs an application program that creates the dynamic document. The server returns the output of the program as a response to the browser that requested the document.

Because a fresh document is created for each request the contents of dynamic document can very from one request to another.

Active-
An active document is not fully specified by the server. Instead an active document consists of a computer program that understands how to compute and display values. When a browser requests an active document, the server returns a copy of the program that the browser must run locally. When it runs the active document program can interact with the user and change the display continuously. Thus the contents of an active document are never fixed- they can continue to change as long as the user allows the program to run.

ADVANTAGES AND DISADVANTAGES OF EACH DOCUMENT:

The chief advantages of a static document are simplicity, reliability and performance. A browser can display a static document quickly and place a copy in cache on a local disk to speedup the future request for the document.
The chief disadvantage of static document is inflexibility.

The advantage of an active document over a dynamic document lies in its ability to update the information continuously.

The chief disadvantages of active documents arise from the additional costs of creating and running such documents, and from a lack of security. The active document has a potential security risk because the document can export as well as can import the information.

The chief advantage of a dynamic document lies in its ability to report current information. For example a dynamic document can be used to report current stock prices, current weather conditions etc.
The chief disadvantages of dynamic document approach are increased cost and the inability to display changing information. A dynamic document takes slightly longer to retrieve than a static document because server requires additional time to run the application program that creates the documents.

Q.50 What is the advantage of caching in a web browser? **(5)**

Ans:
Like other applications browsers use a cache to improve document access. The browser places a copy of each item it retrieves in a cache on the local disk. When a user selects an item the browser checks the disk cache before retrieving a fresh copy. If the cache contains the item the browser obtains the copy from the cache without using the network.

Keeping items in a cache can improve performance dramatically- a browser can read the item from disk without waiting for network connections. For example, consider a user who connects to the Internet over a dialup telephone line. Although a high-speed modem can transfer data at 28.8 Kbps, the effective rate can be substantially lower it the connection is noisy. At such speeds, retrieving a large item from a local disk cache, in fact, local access can seem instantaneous when compared to internet access.

Q.51 **D**escribe the advantages of JAVA servlets over CGI interface. **(4)**

Ans:

The Advantage of Servlets Over "Traditional" CGI:

Java servlets are more efficient, easier to use, more powerful, more portable, and cheaper than traditional CGI and than many alternative CGI-like technologies. (More importantly, servlet developers get paid more than Perl programmers :-).

- Efficient. With traditional CGI, a new process is started for each HTTP request. If the CGI program does a relatively fast operation, the overhead of starting the process can dominate the execution time. With servlets, the Java Virtual Machine stays up, and each request is handled by a lightweight Java thread, not a heavyweight operating system process..
- Convenient. Hey, you already know Java. Why learn Perl too? Besides the convenience of being able to use a familiar language, servlets have an extensive infrastructure for automatically parsing and decoding HTML form data, reading and setting HTTP headers, handling cookies, tracking sessions, and many other such utilities.
- Powerful. Java servlets let you easily do several things that are difficult or impossible with regular CGI. For one thing, servlets can talk directly to the Web server (regular CGI programs can't). This simplifies operations that need to look up images and other data stored in standard places. Servlets can also share data among each other, making useful things like database connection pools easy to implement. They can also maintain information from request to request, simplifying things like session tracking and caching of previous computations.
- Portable. Servlets are written in Java and follow a well-standardized API. Consequently, servlets written for, say I-Planet Enterprise Server can run virtually unchanged on Apache, Microsoft IIS, or WebStar. Servlets are supported directly or via a plugin on almost every major Web server.

- Inexpensive. There are a number of free or very inexpensive Web servers available that are good for "personal" use or low-volume Web sites. However, with the major exception of Apache, which is free, most commercial-quality Web servers are relatively expensive. Nevertheless, once you have a Web server, no matter the cost of that server, adding servlet support to it (if it doesn't come preconfigured to support servlets) is generally free or cheap.

Q.52 Write short notes: **(14)**

(i) Client Server Model.

Ans:

In the client- server model, communication generally takes the form of a request message from the client to the server asking for some work to be done. The servers than does the work and sends back the reply.

A server application waits passively for contact, while a client application initiates communication actively.

A client and server must select a transport protocol that supports connectionless service or one that supports connection-oriented service. Connectionless service allows an application to send a message to an arbitrary destination at any time; the destination does not need to agree that it will accept the message before transmission occurs. In contrast, connection oriented service requires two application to establish a transport connection before data can be sent.

(ii) POP.

Ans:

POP (Post Office Protocol)

The Post Office Protocol provides remote access to an electronic mail box. The protocol allows a user's mailbox to reside on a computer that runs a mail server, and allows the user to access items in the mailbox from another computer.

This protocol requires an additional server to run on the computer with the mailbox. The additional server uses the POP protocol. A user runs email software that becomes a client of POP server to access the contents of the mailbox.

(iii) Anonymous FTP.

Ans:

Use of a login name and password helps keep file secure from unauthorized access. But sometimes such authorization can also be inconvenient. In particular, requiring each user to have a valid name and password makes it difficult to allow arbitrary access. For

example, suppose a corporation finds a bug in one of the programs it sells. The corporation might create file of changes, and make the file available to any one.

To permit arbitrary users to access a file without a specific login and password anonymous FTP is used, In which access to an FTP server is allowed using login name anonymous and password guest.

(iv) SNMP.

Ans:

SNMP (simple network management protocol)

When SNMP is used the management station sends a request to an agent asking it for information or commanding it to update its state. SNMP defines seven messages that can be sent. The following six messages form an initiator.

Get-request- requests the value of one or more variables

Get-next-request – requests the value of next variable

Get-bulk-request – used for large transfer like tables

Set-request – updates one or more variables

Inform-request – allows the manager to update an agent's variables

SnmpV2-trap – Agent to manager trap report.

(v) UDP.

Ans:

UDP (User Datagram Protocol):

UDP uses a connectionless communication paradigm. That is, an application using UDP does not need to preestablish a connection before sending data, nor does application need to terminate communication when finished. Furthermore, UDP allows an application to delay an arbitrarily long time between the transmissions of two messages. UDP does not use any control messages. Communication consists only of the data messages themselves.

(vi) Direct broadcast & limited broadcast.

Ans:

Broadcast is a way to send a packet to all the stations on a particular network at once. Broadcast systems allow the possibility of addressing a packet to all destinations by using a special code in the address field. When a packet with this code is transmitted, it is received and processed by every machine on the network. This is called direct broadcasting. The directed broadcast address for a network is formed by adding a suffix that consists of all 1 bits to the network prefix.

The limited broadcast refers to a broadcast on a local physical network. Limited broadcast is used during system startup by a computer that dosed not yet know the network number. The IP limited Broadcast address is found by setting all 32 bits of the IP address to a value of 1.

Q.53 Write a CGI program that keeps a list of computers that have contacted the server. If comp1 is contacting first time it will display the message: **"**This is the first contact from comp1" else it will display the message "Computer comp1 has requested this URL previously." **(7)**

Ans:

```
#!/bin/sh
FILE = ipaddress

echo Content-type: text/plain
echo
# see if IP address computer appears in file ipaddress

if grep –s $REMOTE_ADDR  $FILE>/dev/null 2 >&1
then

echo Computer $REMOTE_ADDR has requested this URL previously.
else
# append browser's address to the file
 echo $REMOTE_ADDR >> $FILE
 echo This is the first contact from computer $REMOTE_ADDR.

fi
```

Q.54 Design a HTML form for a company Bookonline that allows you to order the books via internet. Form includes the following information:

(i) Book title, author, edition.

(ii) Customer's name, address, phone number. **(7)**

Ans:

Book Title :

Author :

Edition :

Customer's Name :

Address :

Phone Number :

SUBMIT RESET

```
<html>

<head>
<title>New Page 1</title>
</head>
<body>
<H2 align="center"><b><u>FORM TO PURCHASE BOOK
ONLINE</u></b></H2>
<FORM METHOD ="POST" ACTION ="/cgi-bin/formmail">
<input type="hidden" NAME = "SUBJECT" VALUE = "FORM">
<INPUT TYPE="HIDDEN" NAME = "REDIRECT" VALUE ="FORM">
<p><STRONG>Book Title
:</STRONG>        
       
<INPUT NAME  ="NAME" TYPE = "TEXT" SIZE="50">
</p>
<p><STRONG>Author
:   </STRONG>     
         &nbs
p;   
```

```
<INPUT NAME  ="NAME" TYPE = "TEXT" SIZE="50">
</p>


<p><b>Edition
:         &nb
sp;         &
nbsp; 
</b><INPUT NAME  ="NAME1" TYPE = "TEXT" SIZE="11">
</p>
<p><b>Customer's Name :    </b><INPUT NAME
="NAME2" TYPE = "TEXT" SIZE="50">
</p>
<p><b>Address 
:         &nb
sp;        
</b><INPUT NAME  ="NAME3" TYPE = "TEXT" SIZE="50">
</p>
<p><b>Phone Number 
:        </b><INPUT
NAME  ="NAME4" TYPE = "TEXT" SIZE="18">
</p>
</form>
<form method="POST" action="_derived/nortbots.htm" webbot-
action="--WEBBOT-SELF--"
onSubmit="location.href='_derived/nortbots.htm';return
false;" WEBBOT-onSubmit>
<p>         &
nbsp;         
;         &nb
sp;         &
nbsp;
  <input type="submit" value="SUBMIT"
name="B1">        
 
  <input type="reset" value="RESET" name="B2"></p>
</form>
<p> </p>

</body>

</html>
```

Q.55 How does the Applet update its window when information changes? **(6)**

Ans:

Whenever an applet needs to update the information displayed in its window, it simply calls repaint() method. The repaint() method is defined by AWT(abstract window toolkit). It causes the AWT run-time system to execute a call to applet's update() method, which in its default

implementation calls paint(). Thus for another part of the applet to output to its window, simply stores the output and then calls repaint(). The AWT will then execute a call to paint( ).

Q.56 Write an applet that display the directory holding the HTML file that started the applet and the directory from which, applet class file was loaded. **(8)**

Ans:

```
import java.awt.*;
import java.applet.*;
import java.net.*;

public class Base extends Applet
      {
         public void piant(Graphics g)
               {

                     String msg;
                     URL url = getCodebase();
                     Msg = "Code base:" +url.tostring();
                     g.drawstring(msg,10,20);
                     URL url = getDocumentbase();
                     Msg = "Document base: " +url.tostring();
                     g.drawstring(msg,10,40);
               }

       }
```

Q.57 How many Octets does the smallest possible IPV6 datagram contain? Explain the significance IPV6 over IPV4. **(4)**

Ans:

The maximum size of an Ipv6 datagram is 65575 bytes, including the 0 bytes Ipv6 header. Ipv6 also define a minimum reassembly buffer size: the minimum datagram size that we are guaranteed any implementation must support. The minimum size for Ipv6 datagram is 1500 bytes.

Despite retaining the basic concepts from IPv4, IPv6 changes all the details. IPv6 uses larger addresses and an entirely new datagram format. IPv6 uses a

series of fixed-length headers to handle header information. Thus, unlike IPv4, which places key information in fixed fields of the header and only appends variable-length options for less important information, the IPv6 header is always variable size.

Q.58 Suppose you have to develop an error recovery protocol for a link that is unreliable and delay sensitive, which of the following protocol would you choose? **(6)**

(i) Stop & wait.
(ii) Selective Repeat.
(iii) Go back.
Justify your answer.

Ans:
Selective- repeat and Go -back work well if errors are rare. But if error rate is high than a lot of bandwidth will be wasted in retransmission of frames. So both the above methods are not suitable for a link that is unreliable and delay sensitive.

For this kind of link stop and wait protocol is most suitable protocol. In this protocol sender waits after transmitting each packet. When the receiver is ready for another packet, the receiver sends a control message, usually in form of acknowledgement. Although this protocol prevents overrun, they can cause extremely inefficient use of network capacity.

Q.59 Explain the term middleware inn context of RPC. **(4)**

Ans:
A variety of commercial tools have been developed to help the programmer in constructing client- server software. Such tools are generally called middleware because they provide software that fits between a conventional application program and the network software. Now designers are creating new middleware systems that extend method invocation across computers in the same way that remote procedure call extended procedure call. Such systems are known as distributed object systems.

Q.60 Differentiate between **(6)**

(i) message switching, packet switching and circuit switching

Ans:

Message switching:

Recourse computer sends data to switching office which stores the data in buffer and looks for a free link. If link is available than sends it to another switching office. This process continues until data are delivered to destination computer.

Circuit switching versus Packet switching:

In circuit switching an end-to-end path is to be established before any data can be sent. Once a connection is in its place, data can be sent across the connection. Finally when communication is complete, the connection must be terminated. Circuit switching provides connection-oriented interface.

In principle, circuit switching and packet switching both are used in high-capacity networks. In circuit-switched networks, network resources are static, set in "copper" if you will, from the sender to receiver before the start of the transfer, thus creating a "circuit". The resources remain dedicated to the circuit during the entire transfer and the entire message follows the same path. In packet-switched networks, the message is broken into packets, each of which can take a different route to the destination where the packets are recompiled into the original message.

(ii) Bridges & Gateways.

Ans:

Gateways and Bridges:

A machine which connects a LAN to the Internet is called a *gateway*. The gateway machine is responsible for routing packets which are destined for a domain outside the local domain. These machines are called `routers`.

A *bridge* is a machine which transparently connects two segments of the LAN together. These two segments have the same domain name and behave as if part of the same LAN. In our case, we connect the Cape Town LAN and the Sutherland LAN by a PC running the public domain software called PCBRIDGE. Similarly, a PC at Sutherland, also running PCBRIDGE, connects the Sutherland LAN to the Cape Town LAN. These are soon to be replaced by CISCO bridges.

Q.61 What is the use of urgent pointer in TCP segment? **(4)**

Ans:
To accommodate out of band signaling, TCP allows the sender to specify data as urgent, meaning that the receiving program should be notified of its arrival as quickly as possible, regardless of its position in the stream. For this purpose Urgent pointer field is used. The mechanism used to mark urgent data when transmitting it in a segment consists of URG code bit and the URGENT POINTER field. When the URG bit is set, the urgent pointer specifies the position in the segment where urgent data ends.

Q.62 Why does FTP use two standard ports whereas other protocols, in general use only one port? Justify. **(4)**

Ans:
FTP uses a control connection only to send commands and receive responses. When it transfers a file, FTP does not send the data across the control connection. Instead, the client and server establish a separate data connection for each file transfer, use it to send one file, and then close the connection. If the user requests another transfer, the client and server establish a new data connection. To avoid conflict between control and data connections, FTP uses a different port number for each.

Q.63 Does the use of wire-center have any influence on the performance of a token ring? Explain. **(4)**

Ans:
One problem with a ring network is that if the cable breaks somewhere, the ring dies. This problem can be solved by the use of wire center.
Inside the wire center are bypass relays that are energized by current from the stations. If the ring breaks or a station goes down, loss of drive current will release the relay and bypass the station. The relays can also be operated by software to permit diagnostic program to remove stations one at a time to find faulty station and ring segments. The ring can then continue operation with the bad segment bypassed.

Q.64 Explain the meaning of following socket primitive: **(8)**
BIND, LISTEN, ACCEPT and CONNECT.

Ans:

The bind Primitive

when created, a socket has neither a local address nor a remote address. A server uses the bind procedure to supply a protocol port number at which the server will wait for contact. Bind takes three arguments:

bind(socket, localaddr, addrlen)

Argument socket is the descriptor of a socket that has been created but not previously bound; the call is a request that the socket be assigned a particular protocol port number. Argument localaddr is a structure that specifies the local address to be assigned to socket and argument addrelen is an integer that specifies the length of the address.

The listen Primitive

After specifying a protocol port a server must instruct the operating system to place a socket in passive mode so it can be used to wait for contact from clients. To do so a server calls the listen procedure which takes to arguments:
listen(socket, queuesize)

argument socket is the descriptor of a socket that has been created and bound to a local address and argument queuesize specifies a length for the socket's request queue.

The Accept Primitive

A server that uses connection-oriented transport must call procedure accept to accept the next connection request. If a request as present in the queue, accept returns immediately; if no request have arrived the system blocks the server until a client forms a connection. The accept call has the form:
newsock= accept(socket, caddress, addresslen)

Argument socket is the descriptor of a socket the server has created and bound to a specific protocol port. Argument caddress is the address of a structure of type sockaddr and caddresslen is a pointer to an integer. Accept fills in fields of argument caddress with
the address of the client that formed the connection and sets caddresslen to the length of he address.

The connect Primitive
Clients use procedure connect to establish connection with a specific server. The form is

connect(socket, saddress, saddresslen)

Argument socket is the descriptor of a socket on the client's computer to use for the connection. Argument saddress is a sockaddr structure that specifies the server's address and protocol port number. Argument saddresslen specifies me length of the server's address measured in octets.

Q.65 What are two reasons for using layered protocol? **(2)**

Ans:
Layered protocol means protocols used in each layer are the layer's own business i.e. they don't affect protocol of another layer.

- So each layer can use any protocol as long as it gets the gob done
- They can be replaced easily as the technology changes.

Being able to make such changes is the idea for using layered protocols.

Q.66 Give the format of Ethernet frame and explain the semantics of each field. **(6)**

Ans:
An Ethernet frame begins with a header that contains three fields. The 64-bit preamble, that precedes the frame contains alternating 1's and 0's that allows the receiver's hardware to synchronize with the incoming signal.

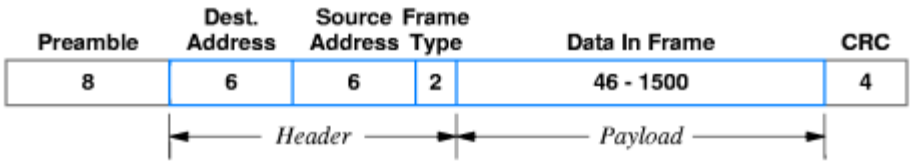


Figure 9.3 Illustration of the frame format used with Ethernet. The number in each field gives the size of the field measured in 8-bit octets.

The First two fields of the header contain the physical addresses. The third field of the header consists of a 16- bit Ethernet frame type.

Q.67 Why is packet switching important? Give at least two reasons. **(4)**

Ans:
Packet switching is important because of the following to reasons:

1. A sender and the receiver need to coordinate transmission to ensure that data arrives correctly. Dividing the data into small blocks helps a sender and receiver determines which block arrive intact and which do not.

2. Second, because communication circuits and the associated modem hardware are expansive, multiple computers often share underlying connections and hardware. To ensure that all computers receive fair, prompt access to a shared communication facility, a network system allows one computer to deny access to others. Using small packets helps ensure fairness.

Q.68 Differentiate between Transport and Session layers of OSI model. **(4)**

Ans:
OSI Model Transport Layer
The transport layer uses the services provided by the network layer, such as best path selection and logical addressing, to provide end-to-end communication between source and destination.

- The transport-layer data stream is a logical connection between the endpoints of a network.
- End-to-end control is provided by sliding windows and reliability in sequencing numbers and acknowledgments.
- The transport layer regulates information flow to ensure end-to-end connectivity between host applications reliably and accurately.
- The TCP/IP protocol of Layer 4 (transport layer) has two protocols. They are TCP and UDP.

The transport layer accepts data from the session layer and segments the data for transport across the network. Generally, the transport layer is responsible for making sure that the data is delivered error-free and in the proper sequence. Flow control generally occurs at the transport layer.

OSI Model Session Layer

The session layer establishes, manages, and terminates communication sessions. Communication sessions consist of service requests and service responses that occur

between applications located in different network devices. These requests and responses are coordinated by protocols implemented at the session layer. The session layer establishes, manages, and terminates sessions between applications

- Functions of the session layer and the different processes that occur as data packets travel through this layer. More specifically, you learned that
- Communication sessions consist of mini-conversations that occur between applications located in different network devices
- Requests and responses are coordinated by protocols implemented at the session layer
- The session layer decides whether to use two-way simultaneous communication or two-way alternate communication by using dialogue control
- The session layer uses dialogue separation to initiate, terminate, and manage communication in an orderly fashion

Q.69 Describe the various characteristics of UDP protocol. **(6)**

Ans:
The characteristics of the UDP are as follows:

- End to end: UDP is a transport protocol that can distinguish among multiple application programs running on a given computer.

- Connectionless:
  The UDP follows a connectionless paradigm.

- Message Oriented:
  An application that uses UDP sends and receives individual messages.

- Best Effort:
  UDP offers application the same best effort delivery semantics as IP.

- Arbitrary interaction:

UDP allows an application to send to many other applications, receive from many other applications, or communicate with exactly one another application.

- Operating system independent:

UDP provides a mean of identifying application programs that does not depend on Identifiers used by the local operating system.

Q.70 Why is fragmentation needed on Internet not on a typical WAN? **(4)**

Ans:
TCP/IP protocol uses the name IP datagram to refer to an Internet packet. The amount of data carried in a datagram is not fixed. The sender chooses an amount of data that is appropriate for a particular purpose. If size of a datagram is larger than network MTU than fragmentation is performed.

When a datagram is larger than the MTU of a network over which it is sent, the router divides the datagram into smaller pieces called fragments.

Fragmentation is needed on Internet as it in connection of different networks using different LAN technologies (such as Ethernet, Token ring, etc.). Each technology has a different frame size, which is Smaller to fit an entire IP datagram.

While a typical Wan is consists of electronic devices called packet switches interconnected by communication lines. The packet switches consist of special purpose hardware. A packet switch is implemented with special purpose computer that is dedicated to the task of providing communication.

Q.71 How does TCP take of wrapping over of sequence numbers? **(4)**

Ans:
To handle out of order deliveries, transport protocols use sequencing. The sending side attaches a sequence number to each packet. The receiving side stores both the sequence number of the last packet received in order as well as a list of additional packets that arrived out of order. When a packet arrives, the receiver examines the sequence number to determine how the packet should be handled.

TCP take care of wrapping over of sequence numbers by leaving a set of consecutive sequence numbers between initial sequence numbers of two TCP connection beginning within some specified time. This duration is known as forbidden region.

Q.72 Write notes o Frame Filtering Techniques. **(5)**

Ans:
Frame filtering:

The most valuable function performed by bridges is frame filtering. A bridge does not forward a frame unless necessary. In particular, if a computer attached to one segment sends a frame to a computer on the same segment, the bridge does not need to forward a copy of the frame to the other segment.

To determine whether to forward a frame, a bridge uses the physical address found in the frame headers. When a frame arrives on a segment, the bridge extracts and checks the destination address. If the bridge knows that the destination computer is attached to a segment over which the frame arrived, the destination will also have received a copy of the frame, so the bridge can discard a frame without forwarding a copy.
Most bridges are called adaptive or learning bridges because they learn the location of a computer automatically. The bridge uses the source address to automatically determine the location of the computer that sent a frame, and uses the destination address to determine whether to forward a frame.

Q.73 What functions connect() and accept() call in Socket interfacing? **(5)**

Ans:
connect() system call:
clients use procedure connect to establish connection with a specific server. The form is

connect(socket, saddress, saaddresslen)

Argument socket is the descriptor of a socket on the client's computer to use for the connection. Argument sockaddress is a sockaddr structure that specifies the server's address and protocol port number. Argument saaddresslen specifies the length of the server's address measured in octets.
The client does not have to bind a local address before calling connect(). connect() internally can call bind() to connect to a local address if not done earlier.

accept() system call:
After a connection oriented server executes the listen() system call, an actual connection from some client process is waited for by having the server execute the accept() system call. The form is:

accept(sockfd, sockaddr * peer, int* addrlen)

This system call returns up to three values: an integer return code that is either an error indication or a new socket descriptor, the address of the client process (peer), and the size of this address(addrlen).
Accept automatically creates a new socket descriptor, assuming the server is a concurrent server.
When a connection request is received, the process forks, with the child process servicing the connection and the parent process waiting for the another connection request.

Q.74 Can SMTP be used as transfer protocol for Web pages? Why? **(4)**

Ans:
SMTP is a simple mail transfer protocol. It uses ASCII text for all communication. SMTP requires reliable delivery- the sender must keep a copy of the message until the receiver has stored a copy in nonvolatile memory.

SMTP can not be used as transfer protocol for web pages as it is not necessarily use hypertext and its header needs information of sender and receiver mail ID which is not required for web pages.

Q.75 Describe the Electronic Data Exchange (EDI) architecture in brief. **(5)**

Ans:
Electronic Data Interchange (EDI) may be most easily understood as the replacement of paper-based purchase orders with electronic equivalents. It is actually much broader in its application than the procurement process, and its impacts are far greater than mere automation.
A more careful definition of EDI is 'the exchange of documents in standardized electronic form, between organizations, in an automated manner, directly from a computer application in one organization to an application in another'**.**

Architecture for EDI

EDI can be compared and contrasted with electronic mail (email). Email enables free-format, textual messages to be electronically transmitted from one person to another. EDI, on the other hand, supports structured business messages (those which are expressed in hard-copy, pre-printed forms or business documents), and transmits them electronically between computer applications, rather than between people.
The essential elements of EDI are:

- the use of an electronic transmission medium (originally a value-added network, but increasingly the open, public Internet) rather than the dispatch of physical storage media such as magnetic tapes and disks;
- the use of **structured, formatted messages based on agreed standards** (such that messages can be translated, interpreted and checked for compliance with an explicit set of rules);

- relatively fast delivery of electronic documents from sender to receiver (generally implying receipt within hours, or even minutes); and
- direct communication between applications (rather than merely between computers).

EDI depends on a moderately sophisticated information technology infrastructure. This must include data processing, data management and networking capabilities, to enable the efficient capture of data into electronic form.

Q.76 How do active web pages work? Describe with a small example. **(5)**

Ans:

Active Web Pages:

An active document is not fully specified by the server. Instead an active document consists of a computer program that understands how to compute and display values. When a browser requests an active document, the server returns a copy of the program that the browser must run locally. When it runs the active document program can interact with the user and change the display continuously. Thus the contents of an active document are never fixed- they can continue to change as long as the user allows the program to run. Following is the example of active document using java applets.

```
import java.applet.*;
import java.awt.*;

public class clickcount extends Applet {
  int count;
  TextField f;

  public void init() {
    count = 0;
    add(new Button("Click Here"));
    f = new TextField("The button has not been clicked at all.");
    f.setEditable(false);
    add(f);
  }

  public boolean action(Event e, Object arg) {
    if (((Button) e.target).getLabel() == "Click Here") {
       count += 1;
       f.setText("The button has been clicked " + count + " times.");
    }
    return true;
  }
}
```

Figure 37.2 An example applet that counts the number of times a user clicks a button.

Q.77 Discuss the life cycle of JSP. **(4)**

Ans:
A JSP page services requests as a servlet. Thus, the life cycle and many of the capabilities of JSP pages (in particular the dynamic aspects) are determined by Java Servlet technology

Servlet Life Cycle

The life cycle of a servlet is controlled by the container in which the servlet has been deployed. When a request is mapped to a servlet, the container performs the following steps.

1. If an instance of the servlet does not exist, the Web container
    a. Loads the servlet class.
    b. Creates an instance of the servlet class.
    c. Initializes the servlet instance by calling the `init` method.

2. Invokes the `service` method, passing a request and response object.

If the container needs to remove the servlet, it finalizes the servlet by calling the servlet's `destroy` method.

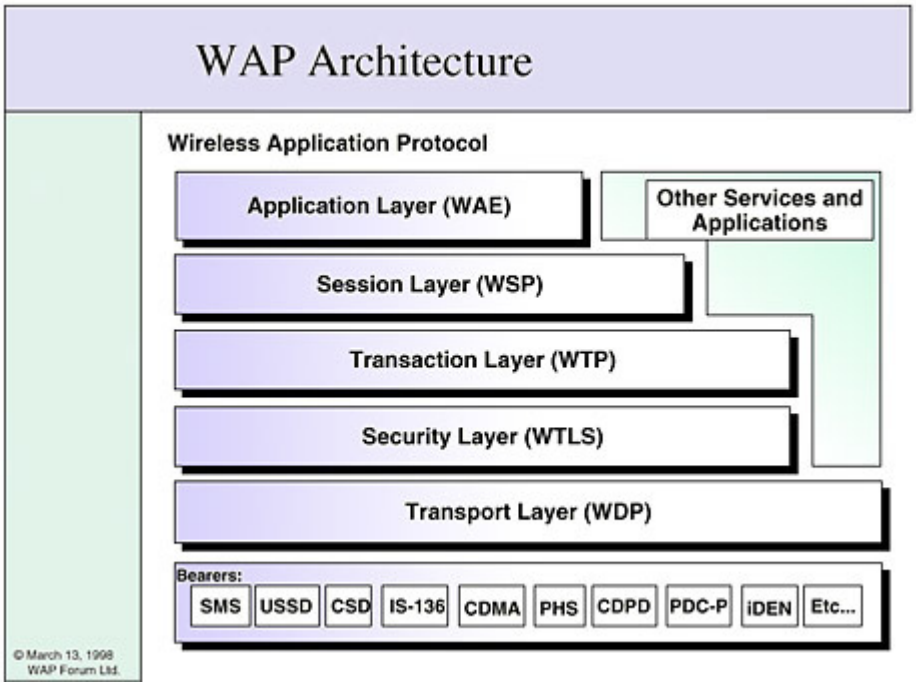
Handling Servlet Life-Cycle Events

You can monitor and react to events in a servlet's life cycle by defining listener objects whose methods get invoked when life cycle events occur. To use these listener objects, you must define the listener class.

Q.78 Discuss the WAP stack inn brief. **(5)**

Ans:

Wap Stack implemented the protocol stack part for the WAP research and development platform. The protocol stack will be exploited in various WAP specific system components such as WAP proxy servers and gateways



Wap Stack: WAP Architecture

WAPSTACK develops software for:

API (interface to applications/Wireless Application Environment)

WAP Forum stack layers (WSP/Session, WTP/Transaction, WTLS/Transport Layer Security, WDP/Datagram, WCMP/Control Message)

Simulation layer (simulates the behavior of an air interface and underlying communication layer protocols of a wireless data channel)

Q.79 Discuss the main tags of WML. **(5)**

Ans:

Tag	Definition
<wml> </wml>	**It defines the beginning and the ending of the page, like <html> </html>.**
<card> </card>	It defines the beginning and the ending of the card. WML pages are organized in decks and cards. A deck is a collection of cards, and cards are the basic unit of a WML page. Like a website consists of a lot of html pages and that html page is the basic unit. The id identifies this card in the deck. It has to be unique for each card in the deck. It is used to referred to this card by other cards. The next example will show you how to use it.
** **	**All text within these tags are organized in paragraph. It is required for all cards to have at least one paragraph.**
<!--	**All text after this are comments. They are not interpreted by the browser.**
<do> </do>	This tag gives the user a general way to perform actions. It has to be defined between the card tags. It is usually placed before the first tag. There are different actions that the user can perform as follows: The accept indicates the type of action. For this one, it is a general positive acknowledge. It's like the user clicking an Ok button.
<go/>	It is the action that will happen when the user performs the action. For this one, it will go the WML page that we specify. Unlike the other tags, there is no closing tag. Instead there is a slash at the end of the tag. The href indicates the destination WML page. It can be URL which will go to another deck, or for this one, another card in the same deck. An # plus the id of another card will identify that card.

<do> </do> **The ontimer indicates the destination (a card or a deck) after the timer expires. Again for a card, an # is required.**

<timer> **This tag let you specify the value for the timer. It is an error to have two timer tags in the same card.**

The value sets how much time for the timer to expire. The unit is one-tenth of a second. For the example above, the timer event will trigger after 15 seconds.

<a> ** **This tag is exactly like the <a> in HTML. It's an anchor, or a link that when the user click it, it will navigate to the next page.

The href is the place where you can set the destination. .

Q.80 Differentiate between WML-Script and Web-based scripting languages. **(4)**

Ans:
WML Script is a lightweight procedural scripting language, optimized for small-memory, low-power CPU wireless devices. It has its roots in the ECMA- Script scripting language, a standardized version of JavaScript based on core features of that language.

WML Script language constructs, syntax, flow control structures, and so on are similar to those of JavaScript because of this inheritance. It provides an optimized and extended subset of JavaScript for the narrowband wireless network-based devices like mobile phones, PDAs, and two-way pagers.

Difference between WML script and other web based languages:

- WML Script is loosely coupled with WML and can be used independently as a stand-alone tool
- The way WML Script is transferred from the WAP gateway to be executed on the wireless client is different from the way JavaScript is transferred over the Web for execution on the client browser. JavaScript is transferred in clear text, while WML Script is compiled by the WAP gateway into byte code before being transmitted to the wireless client. The advantage of this approach is that the byte code is generally much smaller in size than WML Script source code. This enables faster download of WML Script with less bandwidth consumption. It also allows low-memory capacity of the wireless device to store more application data.

- Another key difference is that the WS code isn't embedded with the WML source file but is kept as an independent module.

- WML Script doesn't support global variables; only variables declared inside functions or passed as function parameters are allowed.
- Language support for arrays is not present.
- Support for low-level binary arithmetic operations is also present in the form of bit-wise operators.

The wireless device needs a WML Script virtual machine to interpret this WS byte code. If the device doesn't contain a WS VM, it can ignore the reference to WML Script in a WML file. When the WML document interpreter comes across a reference to WML Script, it asks the WAP gateway for the referred WML Script module (compilation).

Q.81 Two computers using TDM take up turns to send 100-bytes packet over a shared channel that operates at 64000 bits per second. The hardware takes 100 microseconds after one computer stops sending before the other can begin. How long will it take for each computer to send one megabyte data file? **(5)**

Ans:
channel rate is 64000 bits/second or 8000 bytes per second.
Therefore 1000 bytes size packet will take 1000/8000 seconds that is .125 seconds.

One megabyte file will contain 1000000/1000 = 1000 packets of 1000 bytes size each. two system sending one megabyte file each means 2000 packets will be sent

Therefore, 2000 packets will take 2000 X .125 = 250.000 seconds

Hardware take 100 micro second or 0.001 seconds.

Computer send packets turn by turn between every two consecutive packets there will be 0.001 second gap for 200 packets gap is 2000 X
0.001 = 2.0 seconds.

Total time will 250 + 2 = 252 seconds a computer.

Q.82 Compute the number of Ethernet frames formed for a data of 64 KB IP packet. **(5)**

Ans:

Following figure illustrates the format of the Ethernet frame. Maximum data in a frame is 1500 bytes.

Number of frames formed for 64 KB IP packets is:

64*1000/1500 = 43

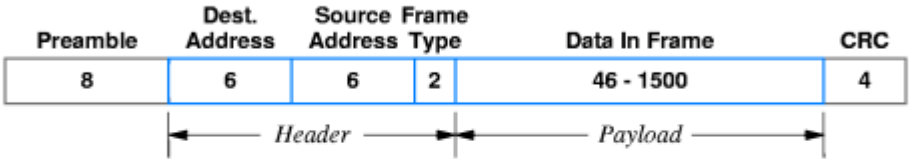


Figure 9.3 Illustration of the frame format used with Ethernet. The number in each field gives the size of the field measured in 8-bit octets.

Q.83 Why does Ethernet specify a minimum frame size. **(4)**

Ans:

Ethernet frame specifies a minimum frame size of 46 bytes. While a data field of zero byte is legal, it causes a problem. When a transceiver detects a collision, it truncates the current frame, which means that stray bits and pieces of frames appear on the cable all the time. To make it easier to distinguish valid frames from garbage, Ethernet specifies that valid frame must be at least 64 bytes long from destination address to checksum. If data portion of frame is less the pad field is used to fill out the frame to the minimum size.

Q.84 A router connects to at most K networks. How many routers R are required to connect to N networks? Derive an equation that gives R in terms of N and K. **(5)**

Ans:
R = 1 if N <= K
otherwise R = (N -1)/ (K-1) for N >K and K >=2

it is K-1 since one network is connected to two routers. and
N- 1 is end router will also connect a network.

Q.85 How many responses does a computer expect to receive when it broadcast an ARP request? Why? **(4)**

Ans:
An ARP request message is placed in a hardware frame and broadcast to all computer on the network. Each computer receives the request and examines the IP address. The computer mentioned in the request sends a response, all other computers process and discard the request without sending a response. So response will be obtained only from the machine for which request is being sent not fro the other machines on the network.

Q.86 Explain the technique used in the asymmetric Key Cryptography. **(5)**

Ans:
Asymmetric or public-key cryptography differs from conventional cryptography in that key material is bound to a single user. The key material is divided into two components:

- a private key, to which only the user has access, and
- a public key, which may be published or distributed on request.

Each key generates a function used to transform text. The private key generates a private transformation function, and the public key generates a public transformation function. The functions are inversely related, i.e., if one function is used to encrypt a message, the other is used to decrypt the message. The order in which the transformation functions are invoked is irrelevant. Note that since the key material is used to generate the transformation functions, the terms *private key* and *public key* not only reference the key values, but also the transformation functions. For example, the phrase, ``*the message is encrypted using the message recipient's public key''*, means the recipient's public key transformation function is invoked using the recipient's public key value and the message as inputs, and a cipher text representation of the message is generated as output.

The advantage of a public-key system is that two users can communicate securely without exchanging secret keys. For example, assume an originator needs to send a message to a

recipient, and secrecy is required for the message. The originator encrypts the message using the recipient's public key. Only the recipient's private key can be used to decrypt the message. This is due to the computational infeasibility of inverting the public key transformation function. In other words, without the recipient's private key, it is computationally infeasible for the interceptor to transform the cipher text into its original plain text.

Q.87 Give the format of ICMP header and explain meaning of each field. **(4)**

Ans:

Internetwork Control Message Protocol (ICMP)

Type	**Code**	**Checksum**
Data		

ICMP Header and Data

ICMP is a error reporting protocol. ICMP uses IP to transport each error message. ICMP header contains an 8-bit integer message TYPE field that identifies the message, an 8-bit CODE field that provides further information about the message type, and a 16-bit Checksum field.

Q.88 How does TCP/IP decide the size of an IP fragment? Explain. **(4)**

Ans:

TCP/IP protocol uses the name IP datagram to refer to an Internet packet. The amount of data carried in a datagram is not fixed. The sender chooses an amount of data that is appropriate for a particular purpose. If size of a datagram is larger than network MTU than fragmentation is performed.

When a datagram is larger than the MTU of a network over whicjh it is sent, the router divides the datagram into smaller pieces called fragments and sends each fragment independently. To fragment a datagram for transmission across the network, a router uses the network MTU and datagram header size to calculate maximum amount of data that can be sent in each fragment and number of fragment that will be needed.

Q.89 Could you directly use TCP over Ethernet without using IP? Justify. **(3)**

Ans:
Each Ethernet station has a unique, burned-in hardware address known as a Medium Access Control (MAC) address.

No matter what higher layer protocols are being used, all addressing in Ethernet must be eventually done at layer 2 as a MAC. For example, if TCP/IP is being used at layers 3 and 4, each computer will be assigned a 32-bit IP address. Before communicating, an Ethernet-attached station must resolve any IP address to a MAC address. This is done using Address Resolution Protocol, or ARP.

The address resolution protocol (ARP) is a protocol used by the Internet Protocol (IP), specifically IPv4, to map IP network address to the hardware addresses used by a data link protocol.So we cannot directly use TCP without IP over Ethernet.

Q.90 What is the largest UDP message that can fit into single Ethernet frame? **(3)**

Ans:
UDP uses IP for delivery. Like ICMP UDP packet is encapsulated in IP datagram. So entire UDP message must fit into IP datagram, and that datagram must fit into single Ethernet frame. Maximum size of a Ethernet frame is 1500 bytes. So largest UDP message that can fit into single Ethernet frame should not be larger than 1500 bytes.

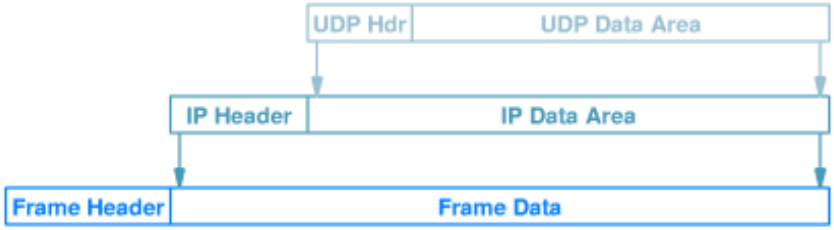
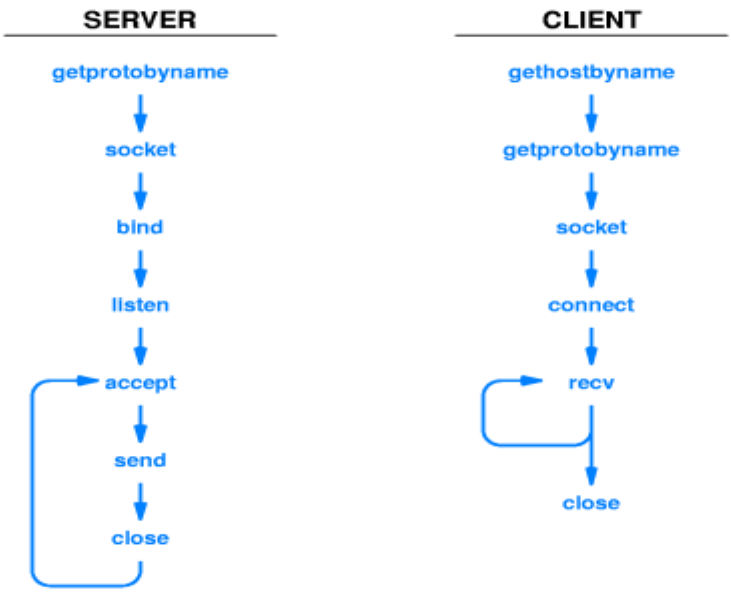


Figure 24.2 The encapsulation of a UDP message in an IP datagram. The entire UDP message, including the header and data areas resides in the data area of the IP datagram.

Q.91 Give the sequence of procedure calls for both server and client for connection-oriented application. **(4)**

Ans:



The Sequence of Procedure calls in connection oriented communication

Q.92 Can both client and server use the same protocol port on the same computer at the same time? Explain. **(4)**

Ans:
Client and server cannot use the same port number on the same computer at the same time. It is because two end points will be same and same port cannot be assigned to two processes. Also when a process request a port then new empty port is assigned.

Single socket bind on a remote machine can accept more than one connection from the other machine.

Q.93 Does it make sense for two domain name servers to contain exactly the same set of names? Why? **(3)**

Ans:
Yes. It is very advantageous for two domain servers containing same set of names. If there is only one server than traffic on one server would be in tolerable, because it would be the only server to receive all the request and handle them appropriately. If there are more than one server containing same set of data then geographically closet server will respond thus reducing the load on one server. Also if one server is down due to some problem then another server containing same set of data can be used to fulfill the incoming requests.

Q.94 If voice is converted to digital form using PCM, how many bits of data will be produced in half a second? **(3)**

Ans:
when voice is converted to digital form using PCM (Pulse Code Modulation) a device called codec (coder decoder) is used producing a 8=bit number. The codec makes 8000 samples per second. Out of 8 bits seven bits are for data and 1 bit is for control.

So data produced per second will be 7*8000=56000 bits per second and 56000/2 = 28000 bits in half a second.

Q.95 Discuss the advantages of Electronics Data Exchange (EDI). **(4)**

Ans:
Advantages of EDI :

EDI's saves unnecessary re-capture of data. This leads to faster transfer of data, far fewer errors, less time wasted on exception-handling, and hence a more stream-lined business process. Benefits can be achieved in such areas as inventory management, transport and distribution, administration and cash management. EDI offers the prospect of easy and cheap communication of structured information throughout the government community, and between government agencies and their suppliers and clients.

EDI can be used to automate existing processes. In addition, the opportunity can be taken to rationalize procedures, and thereby reduce costs, and improve the speed and quality of services. Because EDI necessarily involves business partners, it can be used as a catalyst for gaining efficiencies across organizational boundaries. This strategic potential inherent in EDI is expected to be, in the medium term, even more significant that the short-term cost, speed and quality benefits.

Q.96 Describe architecture of WAP gateway. **(4)**

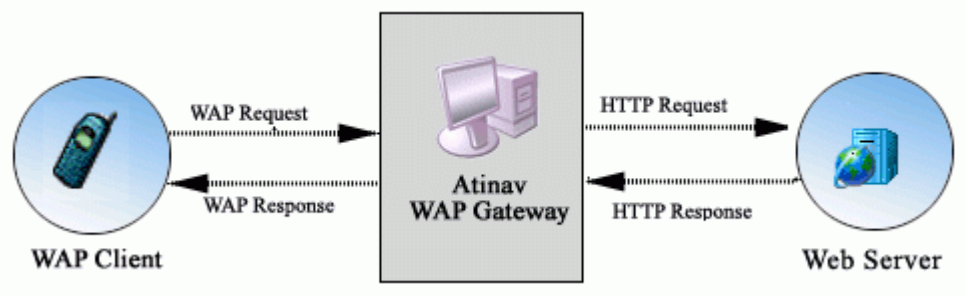
Ans:
WAP GATEWAY :

The WAP Gateway is a very unique product providing semi-automatic redirection of HTML documents to WAP compatible mobile phones.

Wireless Application Protocol (WAP) is a global, open standard that gives mobile users access to Internet services through handheld devices. WAP Gateway, proves to be the perfect answer to the growing demand for wireless mobile services across the world.

The WAP Gateway acts as a bridge between the Internet world and the mobile world and offers services such as end-user authentication, encoding of WML pages, and WML script compiling. WAP uses the underlying web structure to render more efficient communication between content providers and mobile devices. The wireless protocol employs Wireless Markup Language (WML) for application contents instead of Hypertext Markup Language (HTML).

How WAP Gateway works



An important feature of WAP is the support of telephony service integrated with micro browsing of data. aveAccess WAP Gateway acts as a proxy between the wireless network and the Internet while encoding WAP data into byte code so as to conserve bandwidth.

Q.97 Why is XML superior to other forms of data exchange? **(3)**

Ans:
The XML provides universal data format for integrated electronic business solutions. Relational and other database systems cannot meet all the demands of electronic business because they process data independently of its context.

Traditional databases may be well suited for data that fits into rows and columns, but cannot adequately handle rich data such as audio, video, nested data structures or complex documents, which are characteristic of typical Web content. To deal with XML, traditional databases are typically retrofitted with external conversion layers that mimic XML storage by translating it between XML and some other data format. This conversion is error-prone and results in a great deal of overhead, particularly with increasing transaction rates and document complexity.

XML databases, on the other hand, store XML data natively in its structured, hierarchical form. Queries can be resolved much faster because there is no need to map the XML data tree structure to tables. This preserves the hierarchy of the data and increases performance.

Q.98 What are the limitations of mobile devices? **(3)**

Ans:
For mobile devices we need both PC-integrated applications and specialized mobile services rather than re purposed website content.

The latest mobile devices are agonizingly close to being practical, but still lack key usability features required for mainstream use.

Email must be reconceptualized for mobile devices. The old model of "anything sent to this address goes into this mailbox" doesn't work for mobile. We need both better filtering and a way to summarize mail that arrives at the office so you can get what you need on the road without being bogged down by a flood of non-urgent messages.

Information browsing also needs to change. Currently, the best we can hope for are websites that are basically scaled-down and redesigned to eliminate graphics and multi-column layouts. At worst, websites offer no mobile version, so you get crunched images and skinny columns that are almost impossible to read.

To cater to mobile devices, websites and services should offer

- much shorter articles,
- dramatically simplified navigation, and
- highly selective features, retaining only what's needed in a mobile setting.

Q.99 What interprets can a browser contain besides HTML and HTTP? **(4)**

Ans:

Besides an HTTP client and an HTML interpreter, a browser can contain components that enable the browser to perform additional task. For example, many browsers include an FTP client that is used to access the file transfer services. Some browsers also contain e-mail client software that enables the browser to send and receive e-mail messages.

Q.100 Can we specify file transfer in a Web page? Explain with the help of suitable example. **(8)**

Ans:

Yes, file transfer can be specified in a web page. The first field in a URL specifies a protocol. The URL

ftp://ftp.cs.purdue.edu/pub/comer/netbook/client.c

specifies that browser should use anonymous FTP to retrieve the file pub/comer/netbook/client.c from computer ftp.cs.purdue.edu.

A URL that specifies FTP can be easily embedded I HTML consider the following example:

Source code from
<A HREF = "ftp://ftp.cs.purdue.edu/pub/comer/netbook/client.c "> an example client program

</A>
is available online.

The program segment will be displayed as given below:

Source code from an example client program is available online.

If the user select the underlined segment, the browser uses it's FTP client to obtain a copy of the file client.c.

Q.101 Why is Java called Machine Independent? **(4)**

Ans:
When a java program is compiled it is not converted into an executable code. Rather, it is converted into a byte code. Byte code is highly optimized set of instructions designed to be executed by java run time system called

java Virtual Machine (JVM). Translating a program into byte code helps make it much easier to run a program on variety of environments. Only JVM needs to be implemented for each platform. Interpreters for various platforms can interpret same byte code.

Q.102 Write an applet that can interact with both the HTTP client and HTML interpreter in a browser. **(8)**

Ans:

```
import java.applet.*;
import java.net.*;
import java.awt.*;

public class buttons extends Applet {

   public void init() {
      add(new Button{“Yin”});
      add(new Button{“Yang”});
   }

   public Boolean action(Event e, Object arg) {
      if ((( Button) e.target).getLabel() == “Yin”) {
         try {
            getAppletContext ().showDocument(new
            URL(http://www.nonexist.com/yin();
         }
         catch(Exception ex) {
         // note: code to handle the exception goes here //
         }
      }
      else if(((Button) e.target).getLabel() == “Yang”) {
         try {
            getAppletContext ().showDocument(new
            URL(http://www.other.com/yang();
         }
         catch(Exception ex) {
         // note: code to handle the exception goes here //
         }
      }
```

```
    return true;
    }
}
```

Q.103 Write an HTML program segment that contains hypertext links from one document to another. **(8)**

Ans:

HTML allows any item to be placed as a hypertext reference. Thus a single word, a phrase, an entire paragraph, or an image can refer to another

document. The HTML uses tags<A> and </A> for the reference. All items between the two are the part of anchor. Consider the following example:

```
This Article is written by
<A HREF = http://www.ideanews.com>
IDEA NEWS, </A>
One of the popular newspapers worldwide.

Contains an anchor that references the URL
    http://www.ideanews.com
When displayed it produces the following:

This article is written by IDEA NEWS,
One of the popular newspapers worldwide.
```

When underlined text is selected, the document specified by the URL http://www.ideanews.com is opened.

Q.104 Write a CGI program that prints date and time at which it was run. **(6)**

Ans:

```
#!/bin/sh

#
# CGI script that prints the date and time at which it was run
#

# Output the document header followed by a blank line
```

```
echo Content-type: text/plain
echo

# Output the date

echo This document was created on ‘date’
```

Q.105 What is the advantage of dividing an email address into two parts? **(4)**

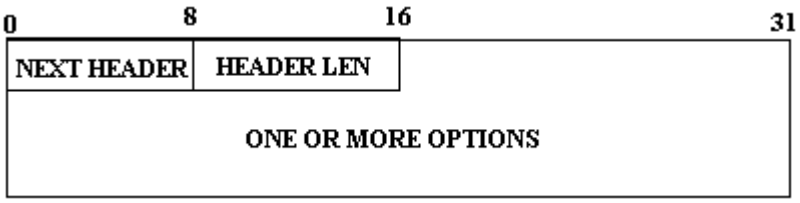
Ans:
The division of an e-mail address into two parts is important because it achieve two goals. First, the division allows each computer system to assign mailbox identifiers independently. Second, the division permits the user on arbitrary computer systems to exchange e-mail messages. E-mail software on the sender’s computer uses the second part to determine which computer to contact, and the e-mail software on the recipient computer uses the first part of the address to select a particular mailbox into which message should be placed.

Q.106 Why can CRC detect more errors than simple Checksum? **(6)**

Ans:
There are two reasons a CRC can detect more errors than a simple Checksum. First, because an input bit is shifted through all three registers, a single bit of the message affects the resulting CRC in dramatic ways. Second, hardware uses feedback in which output from the leftmost shift register affects each exclusive or unit, the effect from a single bit of the message cycle through the shift registers more than one time.

Q.107 As IPV6 contain multiple headers, how does it know where particular header ends and next item begins? **(7)**

Ans:
Some headers types have fixed size. For example a base header has a fixed size of exactly forty octets. To move to the item following base header, IPV6 software simply adds forty to the address of the base header. But some extension headers do not have a fixed size. In such cases, the header must contain sufficient information to allow IPV6 to determine where the header ends. The following figure illustrates the general form of an IPV6 option header.



The Ipv6 option extension header. Because the size of the options header can very fro one datagram to another the HEADER LEN field specifies the exact length.

Q.108 Why is TCP called end-to-end protocol? **(3)**

Ans:

TCP is called an end-to-end protocol because it provides a connection directly from an application on one computer to an application on a remote computer. The applications can request that TCP form a connection, send and receive data, and close the connection. The connections provided by TCP are called virtual connections because they are achieved in software

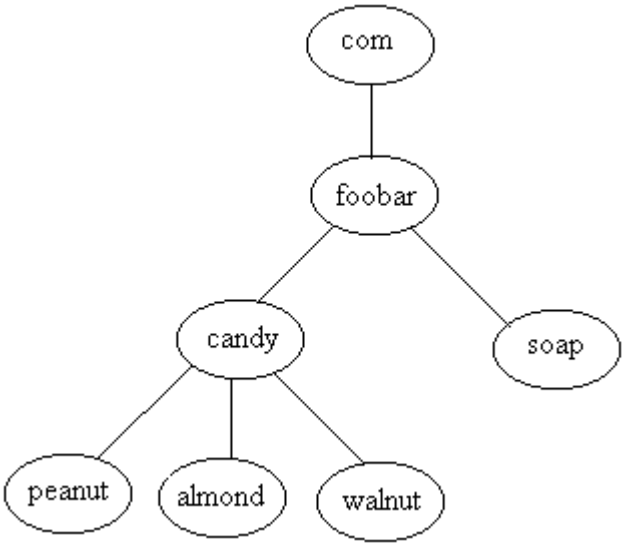
Q.109 IP specified that datagram can arrive in a different order than they were sent. If a fragment from one datagram arrives at a destination before all the segments from a previous datagram arrive, how does the destination know to which datagram the fragments belong? **(6)**

Ans:

To solve the problem of reassembling the fragments that arrive out of order a unique identification number is placed in the IDENTIFICATION field of each outgoing datagram. When a router fragments a datagram, the router copies the identification number into each segment. A receiver uses the identification number and IP source address in an incoming fragment to determine the datagram to which the fragment belongs. In addition the fragment OFFSET field tells a receiver how to order fragments within a given datagram.

Q.110 What is the advantage of using abbreviations in DNS? **(5)**

Ans:
While using abbreviation some part of the name of the computer can be skipped. Because users tend to enter names for local computer more often, abbreviations for local names are very convenient. For example Foobar Corporation might choose to allow users to omit Foobar.com while entering a domain name.



with such an abbreviation in effect, a user could enter the name venus.walnut.candy to refer to computer venus inn the walnut subdivision of the candy division in place of venus.walnut.candy.foobar.com.

Q.111 Describe ARP message format in brief. **(6)**

Ans:

0	8	16	24	31
HARDWARE ADDRESS TYPE		PROTOCOL ADDRESS TYPE		
HADDER LEN	PADDER LEN	OPERATION		
SENDER HADDER (first 4 octets)				
SENDER HADDER (last 2 octets)		SENDER PADDER (first 2 octets)		
SENDER PADDER (last 2 octets)		TARGET HADDER (first 2 octets)		
TARGET HADDER (last 4 octets)				
TARGET pADDER (all 4 octets)				

The Format of an ARP Message

The first two 16-bit fields contain values that specify the type of hardware and protocol addresses being used. For example HARDWARE ADDRESS TYPE contains 1 when ARP is used with Ethernet, and field PROTOCOL ADDRESS TYPE contains 0x0800 when ARP is used with IP. The second pair of fields HADDR LEN and PADDR LEN specify the number of octets in a hardware address and a protocol address. Field OPERATION specifies whether the message is a request or a response.

Each ARP message contains fields for two address bindings. One for the recipient and the other for the intended recipient, which ARP calls the target.

Q.112 What is the advantage or disadvantage of using INADDR_ANY instead of the IP address of the computer running on the server? **(5)**

Ans:
Structure sockaddr_in defines the format TCP/IP uses to represent an address. Structure contains field for both an IP address and a protocol port number. Although a server can choose to fill in the IP address when specifying an IP address, doing so causes problems when a host is multi-homed because it means the server only accepts requests sent to one specific address. To allow a server to operate on a multi-homed host, the socket API includes a special symbolic constant, INADDR_ANY that allows a server to use a specific port at any of the computer's IP address.

Q.113 Is the TCP checksum necessary or could TCP allow IP to checksum thee data? **(5)**

Ans:
Yes, TCP Checksum is necessary.
TCP layer is responsible for error detection, error control, transmission of packets if required, reassembly of packets as well as their fragmentation. Hence for all error control and detection purposes TCP Checksum is essential.

TCP cannot allow IP to checksum data however IP has its own checksum for its header. IP layer is basically responsible for routing of IP datagrams immaterial of whether that packet is intended for TCP services or UDP services. Thus immaterial of what information is contained in data part, IP layer is only responsible for routing of packets and all the issues related to error control, error detection, and flow control with regards to Routing only. Hence IP does not have a checksum for data unlike TCP.

Q.114 What is the chief advantage of CIDR over the original classful addressing scheme? **(6)**

Ans:
CIDR (Classless Inter-Domain Routing) is a new addressing scheme for the Internet, which allows for more efficient allocation of IP address than the old classful scheme.

There are a maximum number of networks and hosts that can be assigned using 32-bit classful addressing scheme. Some addresses are reserved (for broadcasting etc.), and there were a lot of wasted addresses also.

For example if you needed 100 addresses you would be assigned the smallest class addresses (class C), but that still means 154 unused addresses. The overall result was Interest was running out of unassigned addresses.

A related problem was the size of the Internet global routing tables. As the number of networks on the Internet increased, so did the number of route.

Instead of being limited to network identifiers of 8, 16 or 32 bits, CIDR currently uses prefixes any where between 13 to 27. Thus, block of addresses can be assigned to networks as small as 32 hosts or to those with over 5000,000 hosts. A CIDR address includes standard 32-bit IP address and also information on how many bits are used for the network prefix.

In the CIDR address 206.13.01.48/25, the 25 indicate the first 25 bits are used to identify the unique network leaving the remaining bits to identify the specific hosts.

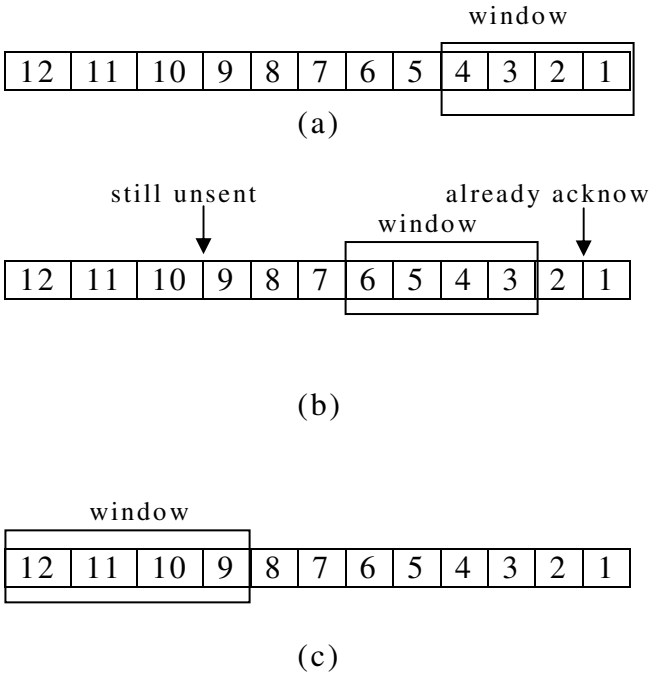
CIDR addressing scheme also enables route-aggregation in which single high-level route entry can represent many lower level routes in the global routing table.

Q.115 Write short notes: **(16)**

(i) Sliding Window Protocol.

Ans:
To obtain high throughput rates, protocols use a flow control technique known as sliding window. The sender and receiver are programmed to use a fixed window size, which is the maximum amount of data that can

be sent before an acknowledge arrives. For example, the sender and receiver might agree on a window size of four packets.



A 4-packet window sliding through outgoing data.
(a) When transmission begins (b) after two packets has been acknowledged, and (c) after eight packets have been acknowledged.

(ii) Digital Signature.

Ans:

This technique is used to authenticate the sender of a message. To sing a message, the sender encrypts the message using a key known only to the sender. The recipient uses the inverse function to decrypt the message. The recipient knows who sent the message because only the sender has the key needed to perform the encryption. To ensure that encrypted messages are not copied and resent later, the original message can contain date and time it was sent.

A public key system can be used to provide a digital signature.
To sign a message, a user encrypts the message using his or her own private key. To verify thee signature, thee recipient looks up thee user's public key, only the user can encrypt a message that can be encoded with the public key.

(iii) BOOTP(Boot Strap Protocol)

Ans:

TCP/IP designer observed that many of the configuration steps could be combined into a single step if a server was able to supply more than one item of configuration information. To provide such a service BOOTP was invented.

To obtain configuration information, protocol software broadcasts a BOOTP request message. A BOOTP server that receives the request looks up several pieces of configuration information for the computer that issued the request into a single BOOTP response message, and returns the reply to the requesting computer.

(iv) Virtual Packets

Ans:

The router cannot transfer a copy of a frame from one type of network to another because the frame formats differ. More importantly, the router cannot simply reformat the frame header because the two networks may use incompatible address format.

To overcome heterogeneity, Internet protocol software defines an inter packet format that is independent of the underlying hardware. This is called virtual packet and can be transferred across the underlying hardware. The underlying hardware does not understand or recognize the Internet packet format, the protocol software creates and handles Internet packet.

Q.116 What is XML DTD (Document Type Definition)? What is the advantage of having a DTD for an XML document? **(6)**

Ans:

DTD is a document that defines legal building blocks of an XML document. It defines the document structure with a list of legal elements. A DTD can be declared inline in your XML document, or as external reference.

Example XML document with External DTD:

```
<?xml version = “1.0”?>
<|DOCTYPE note SYSTEM “note.dtd”>
<note>
<to>TOM</to>
```

```
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend.</body>
</note>
```

note.dtd

```
<?xml version = "1.0"?>
<|ELEMENT note (to, from, heading, body)>
<|ELEMENT to (#PCDATA)>
<|ELEMENT from (#PCDATA)>
<|ELEMENT heading (#PCDATA)>
<|ELEMENT body (#PCDATA)>
```

Why use a DTD?

XML provides an application independent way of sharing data. With a DTD, independent group of people can agree to use a common DTD for interchanging data. An application can use a standard DTD to verify that

data that an application receives from outside world is valid. An application can also use a DTD to verify it's own data.

Q.117 What is WAP? Why WAP gateways are used? **(6)**

Ans:

Wireless Application Protocol (WAP) is a global, open standard that gives mobile users access to Internet services through handled devices. WAP Gateway proves to be the perfect answer to the growing demand for wireless mobile services across the world.

The WAP Gateway is a very unique product providing semi-automatic redirection of HTML documents to WAP compatible mobile phones.

The WAP Gateway acts as a bridge between the Internet world and the mobile world and offers services such as end-user authentication, encoding of WML script compiling. WAP uses the underlying web structure to render more efficient communication between content providers and mobile devices. The wireless protocol employs Wireless Markup Language (WML) for application contents instead of Hypertext Markup Language (HTML).

An important feature of WAP is the support of telephony service integrated with micro browsing of data. AveAccess WAP Gateway acts as a proxy between the wireless network and the Internet while encoding WAP data into byte code so as to conserve bandwidth.

TYPICAL QUESTIONS & ANSWERS

PART – I

OBJECTIVE TYPE QUESTIONS

Each Question carries 2 marks.

Choose the correct or best alternative in the following:

Q.1 If the crystal oscillator is operating at 15 MHz, the PCLK output of 8284 is

(A) 2.5 MHz. **(B)** 5 MHz.
(C) 7.5 MHz. **(D)** 10 MHz.

Ans: (A)

Q.2 In which T-state does the CPU sends the address to memory or I/O and the ALE signal for demultiplexing

(A) T1. **(B)** T2.
(C) T3. **(D)** T4.

Ans, During the first clocking period in a bus cycle, which is called T1, the address of the memory or I/O location is sent out and the control signals ALE, DT/R' and IO/M' are also output. Hence answer is (A).

Q.3 If a $1M \times 1$ DRAM requires 4 ms for a refresh and has 256 rows to be refreshed, no more than _____ of time must pass before another row is refreshed.

(A) 64 ms. **(B)** 4 ns.
(C) 0.5 ns. **(D)** 15.625 μs.

Ans Answer is (B)

Q.4 In a DMA write operation the data is transferred

(A) from I/O to memory. **(B)** from memory to I/O.
(C) from memory to memory. **(D)** from I/O to I/O.

Ans A DMA writes operation transfers data from an I/O device to memory. Hence answer is (A).

Q.5 Which type of JMP instruction assembles if the distance is 0020 h bytes

(A) near. **(B)** far.
(C) short. **(D)** none of the above.

Ans The three byte near jump allows a branch or jump within ± 32K bytes. Hence answer is (A).

Q.6 A certain SRAM has $\overline{CS} = 0$, $\overline{WE} = 0$ and $\overline{OE} = 1$. In which of the following modes this SRAM is operating

(A) Read **(B)** Write

(C) Stand by **(D)** None of the above

Ans For CS'=WE'=0, write operation. Hence answer is (B).

Q.7 Which of the following is true with respect to EEPROM?
- **(A)** contents can be erased byte wise only.
- **(B)** contents of full memory can be erased together.
- **(C)** contents can be erased using ultra violet rays
- **(D)** contents can not be erased

Ans Answer is (C).

Q.8 Pseudo instructions are basically
- **(A)** false instructions.
- **(B)** instructions that are ignored by the microprocessor.
- **(C)** assembler directives.
- **(D)** instructions that are treated like comments.

Ans Pseudo-instructions are commands to the assembler. All pseudo-operations start with a period. Pseudo-instructions are composed of a pseudo-operation which may be followed by one or more expressions. Hence answer is (C).

Q.9 Number of the times the instruction sequence below will loop before coming out of loop is

```
MOV AL, 00h
A1: INC AL
JNZ A1
```

(A) 00 **(B)** 01
(C) 255 **(D)** 256

Ans Answer is (D)

Q.10 What will be the contents of register AL after the following has been executed

```
MOV BL, 8C
MOV AL, 7E
ADD AL, BL
```

(A) 0A and carry flag is set **(B)** 0A and carry flag is reset
(C) 6A and carry flag is set **(D)** 6A and carry flag is reset

Ans, Result is 1,0A. Hence answer is (A).

Q.11 Direction flag is used with
- **(A)** String instructions. **(B)** Stack instructions.
- **(C)** Arithmetic instructions. **(D)** Branch instructions.

Ans The direction flag is used only with the string instructions. Hence answer is (A).

Q.12 Ready pin of a microprocessor is used
- **(A)** to indicate that the microprocessor is ready to receive inputs.
- **(B)** to indicate that the microprocessor is ready to receive outputs.
- **(C)** to introduce wait states.

(D) to provide direct memory access.

Ans This input is controlled to insert wait states into the timing of the microprocessor. Hence answer is (C).

Q.13 These are two ways in which a microprocessor can come out of Halt state.
(A) When hold line is a logical 1.
(B) When interrupt occurs and the interrupt system has been enabled.
(C) When both **(A)** and **(B)** are true.
(D) When either **(A)** or **(B)** are true.

Ans Answer is (A)

Q.14 In the instruction FADD, F stands for
(A) Far. **(B)** Floppy.
(C) Floating. **(D)** File.

Ans Adds two floating point numbers. Hence answer is (C).

Q.15 SD RAM refers to
(A) Synchronous DRAM **(B)** Static DRAM
(C) Semi DRAM **(D)** Second DRAM

Ans, Answer is (A)

Q.16 In case of DVD, the speed is referred in terms of n X (for example 32 X). Here, X refers to
(A) 150 KB/s **(B)** 300 KB/s
(C) 1.38 MB/s **(D)** 2.4 MB/s

Ans Answer is (C).

Q.17 Itanium processor of Intel is a
(A) 32 bit microprocessor. **(B)** 64 bit microprocessor.
(C) 128 bit microprocessor. **(D)** 256 bit microprocessor.

Ans The Itanium is a 64-bit architecture microprocessor. Hence answer is (B).

Q.18 LOCK prefix is used most often
(A) during normal execution. **(B)** during DMA accesses
(C) during interrupt servicing. **(D)** during memory accesses.

Ans LOCK is a prefix which is used to make an instruction of 8086 non-interruptable. Hence answer is (C).

Q.19 The Pentium microprocessor has_____execution units.
(A) 1 **(B)** 2
(C) 3 **(D)** 4

Ans The Pentium microprocessor is organized with three execution units. One executes floating-point instructions, and the other two (U-pipe and V-pipe) execute

integer instructions. Hence answer is (C).

Q.20 EPROM is generally erased by using

(A) Ultraviolet rays **(B)** infrared rays
(C) 12 V electrical pulse **(D)** 24 V electrical pulse

Ans The EPROM is erasable if exposed to high-intensity ultraviolet light for about 20 minutes or less. Hence answer is (A)

Q.21 Signal voltage ranges for a logic high and for a logic low in RS-232C standard are

(A) Low = 0 volt to 1.8 volt, high = 2.0 volt to 5 volt
(B) Low =-15 volt to –3 vol, high = +3 volt to +15 volt
(D) Low = +3 volt to +15 volt, high = -3 volt to -15 volt
(E) Low = 2 volt to 5.0 volt, high = 0 volt to 1.8 volt

Ans Answer is (B)

Q.22 The PCI bus is the important bus found in all the new Pentium systems because

(A) It has plug and play characteristics
(B) It has ability to function with a 64 bit data bus
(C) Any Microprocessor can be interfaced to it with PCI controller or bridge
(D) All of the above

Ans, Answer is (D).

Q.23 Which of the following statement is true?

(A) The group of machine cycle is called a state.
(B) A machine cycle consists of one or more instruction cycle.
(C) An instruction cycle is made up of machine cycles and a machine cycle is made up of number of states.
(D) None of the above

Ans An instruction cycle consists of several machine cycles. Hence Answer is (B).

Q.24 8251 is a

(A) UART
(B) USART
(C) Programmable Interrupt controller
(D) Programmable interval timer/counter

Ans The Intel 8251 is a programmable communication interface. It is USART.

Q.25 8088 microprocessor has

(A) 16 bit data bus **(B)** 4 byte pre-fetch queue
(C) 6 byte pre-fetch queue **(D)** 16 bit address bus

Ans The 8088 is a 16-bit microprocessor with an 8-bit data bus. The 16-bit address bus. Hence answer is (D).

Q.26 By what factor does the 8284A clock generator divide the crystal oscillator's output frequency?

(A) One **(B)** Two
(C) Three **(D)** Four

Ans When F/C' is at logic 0; The oscillator output is steered through to the divide-by-3 counter. Hence answer is (c).

Q.27 The memory data bus width in Pentium is

(A) 16 bit **(B)** 32 bit
(C) 64 bit **(D)** None of these

Ans The Data bus width is 64 bits. Hence answer is (C).

Q.28 When the 82C55 is reset, its I/O ports are all initializes as

(A) output port using mode 0 **(B)** Input port using mode 1
(C) output port using mode 1 **(D)** Input port using mode 0

Ans A RESET input to the 82C55 causes all ports to be set up as simple input ports using mode 0 operations. Hence answer is (D).

Q.29 Which microprocessor pins are used to request and acknowledge a DMA transfer?

(A) reset and ready **(B)** ready and wait
(C) HOLD and HLDA **(D)** None o these

Ans, The HOLD pin is an input that is used request a DMA action and the HLDA pin is an output that that acknowledges the DMA action. Hence answer is (C).

Q.30 Which of the following statement is false?

(A) RTOS performs tasks in predictable amount of time
(B) Windows 98 is RTOS
(C) Interrupts are used to develop RTOS
(D) Kernel is the one of component of any OS

Ans Operating systems, like Windows, defer many tasks and do not guarantee their execution in predictable time. Hence answer is (B).

Q.31 The VESA local bus operates at

(A) 8 MHz **(B)** 33 MHz
(C) 16 MHz **(D)** None of these

Ans The VESA local bus operates at 33 MHz. Hence answer is (B).

Q.32 The first modern computer was called_____.

(A) FLOW-MATIC **(B)** UNIVAC-I
(C) ENIAC **(D)** INTEL

Ans, ENIAC (Electronic Numerical Integrator And Computer) was the first general-purpose electronic computer. It was a Turing-complete, digital computer capable of being reprogrammed to solve a full range of computing problems. ENIAC was

designed to calculate artillery firing tables for the U.S. Army's Ballistic Research Laboratory. Hence answer is (c).

Q.33 Software command CLEAR MASK REGISTER in DMA

(A) Disables all channels.
(B) Enables all channels.
(C) None.
(D) Clears first/last flip-flop within 8237.

Ans Enables all four DMA channels. Hence answer is (B).

Q.34 The first task of DOS operating system after loading into the memory is to use the file called_____.

(A) HIMEM.SYS **(B)** CONFIG.SYS
(C) AUTOEXEC.BAT **(D)** SYSTEM.INI

Ans, The first task of the DOS operating system, after loading into memory, is to use a file called the CONFIG.SYS file. This file specifies various drivers that load into the memory, setting up or configuring the machine for operation under DOS.

Q.35 If the programmable counter timer 8254 is set in mode 1 and is to be used to count six events, the output will remain at logic 0 for _____ number of counts

(A) 5 **(B)** 6
(C) 0 **(D)** All of the above

Ans. OUT continues for the total length of the count. Hence answer is (B).

Q.36 The flash memory is programmed in the system by 12 V programming pulse.

(A) TRUE **(B)** FALSE

Ans The flash memory device requires a 12V programming voltage to erase and write new data. Hence answer is (A).

Q.37 A plug and play (PnP) interface is one that contains a memory that holds configuration information of the system.

(A) TRUE **(B)** FALSE

Ans Answer is (A)

Q.38 The accelerated graphics port (AGP) allows virtually any microprocessor to be interfaced with PCI bus via the use of bridge interface.

(A) TRUE **(B)** FALSE

Ans, this port probably will never be used for any devices other than the video card. Hence answer is (B).

Q.39 A Bus cycle is equal to how many clocking periods

(A) Two (B) Three
(C) Four (D) Six

Ans Typically, the bus-cycle of the 8086 and 8088 processors consist of four clock cycles or pulses. Thus, duration of a bus-cycle is = '4*T'. Hence Answer is (C).

Q.40 The time required to refresh a typical DRAM is

(A) 2 – 4 us (B) 2 – 4 ns
(C) 2 – 4 ms (D) 2 – 4 ps

Ans The capacitor Cs discharges through the internal resistance of the NMOS transistor T1. Typically Cs = 0.2 pF and the internal resistance Rin = 10^{10} ohms, so:
Cs x Rin = 0.2 x 10^{-12} x 10^{10} x 10^{3} ms = 2 ms
So the typical refresh time interval is 2 ms. Hence Answer is (C).

Q.41 The no. of address lines required to address a memory of size 32 K is

(A) 15 lines (B) 16 lines
(C) 18 lines (D) 14 lines

Ans 32K = 32 X 1024 bits = 2^5 X 2^{10} = 2^{15} Hence answer is (A).

Q.42 The no. of wait states required to interface 8279 to 8086 with 8MHz clock are

(A) Two (B) Three
(C) One (D) None

Ans Two wait states used so that device can function with an 8 MHz. Hence answers is (A).

Q.43 NMI input is

(A) Edge sensitive (B) Level sensitive
(C) Both edge and level triggered (D) edge triggered and level sensitive

Ans Non-maskable interrupt (NMI) is an **edge –triggered** input that requests an interrupt on the positive edge (0 to 1 transition).

Q.44 Data rate available for use on USB is

(A) 12 Mbits per second **(B)** 1.5 Mbits per second
(C) Both **(A)** and **(B)** **(D)** No restriction

Ans Data transfer speeds are 12 Mbps for full speed operation and 1.5 Mbps for slow speed operation. Hence answer is (c).

Q.45 In 80186, the timer which connects to the system clock is

(A) timer 0 **(B)** timer 1
(C) timer 2 **(D)** Any one can be connected

Ans. Timer 2 is internal and clocked by the master clock. Hence answer is (c).

Q.46 Conversion of the +1000 decimal number into signed binary word results

(A) 0000 0011 1110 1000 (B) 1111 1100 0001 1000
(C) 1000 0011 1110 1000 (D) 0111 1100 0001 1000

Ans

1000 /2 =>500 →0
500/2=>250→0
250/2=>125→0
125/2=>62→1
62/2=>31→0
31/2=>15→1
15/2=>7→1
7/2=>3→1
3/2=>1→1

16 bit signed number is 1000,0011,1110,1000
Hence Answer is (C).

Q.47 What do the symbols [ ] indicate?

(A) Direct addressing (B) Register Addressing
(C) Indirect addressing (D) None of the above

Ans Answer is (C).

Q.48 SDRAM refers to

(A) static DRAM (B) synchronous DRAM
(C) sequential DRAM (D) semi DRAM

Ans, Answer is (B)

Q.49 Which pins are general purpose I/O pins during mode-2 operation of the 82C55?

(A) PA0 – PA7 (B) PB0-PB7
(C) PC3-PC7 (D) PC0-PC2

Ans In mode 2 Port-A can be programmed to operate as bidirectional port. The mode-2 operation is only for Port-A. Hence Answer is (A)

PART – II

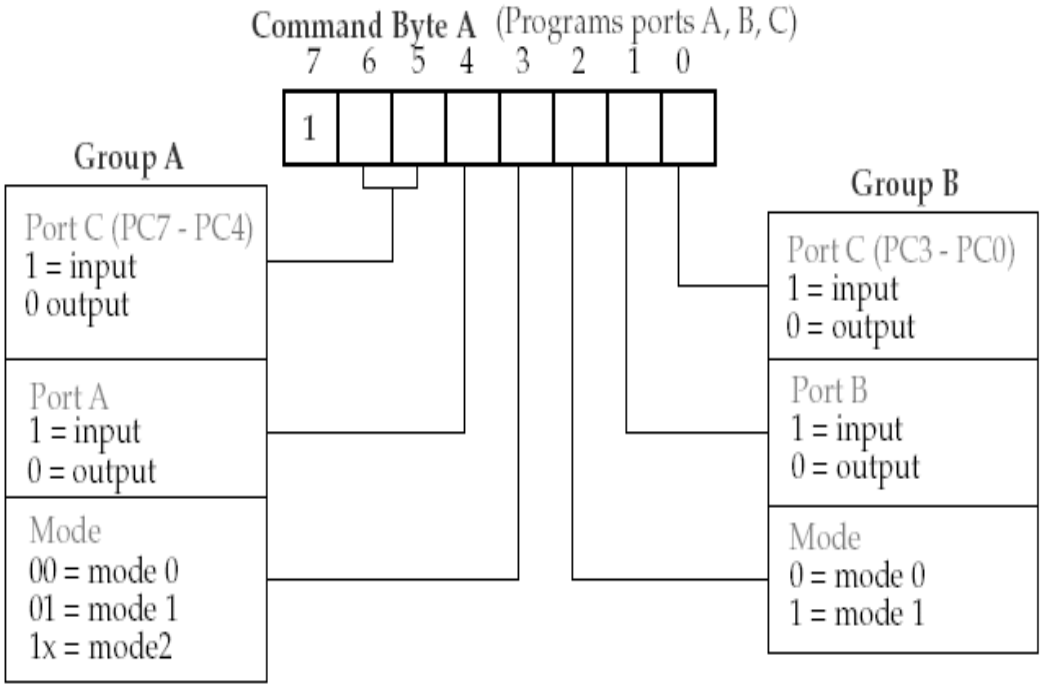
DESCRIPTIVES

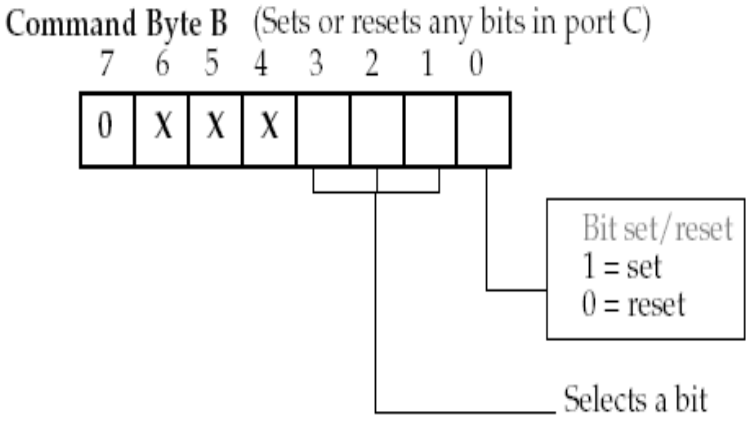
Q.1 Describe the operation performed by the instruction OUT 47 h, AL. **(3)**

Ans : It transfers the content of AL to I/O port 47h. Notice that I/O port number appears as 0047h on the 16 bit address bus and that data from AL appears on the data bus of the microprocessor.

Q.2 How is 8255 (Programmable Peripheral Interface) configured if its control register contains 9B h. **(3)**

Ans

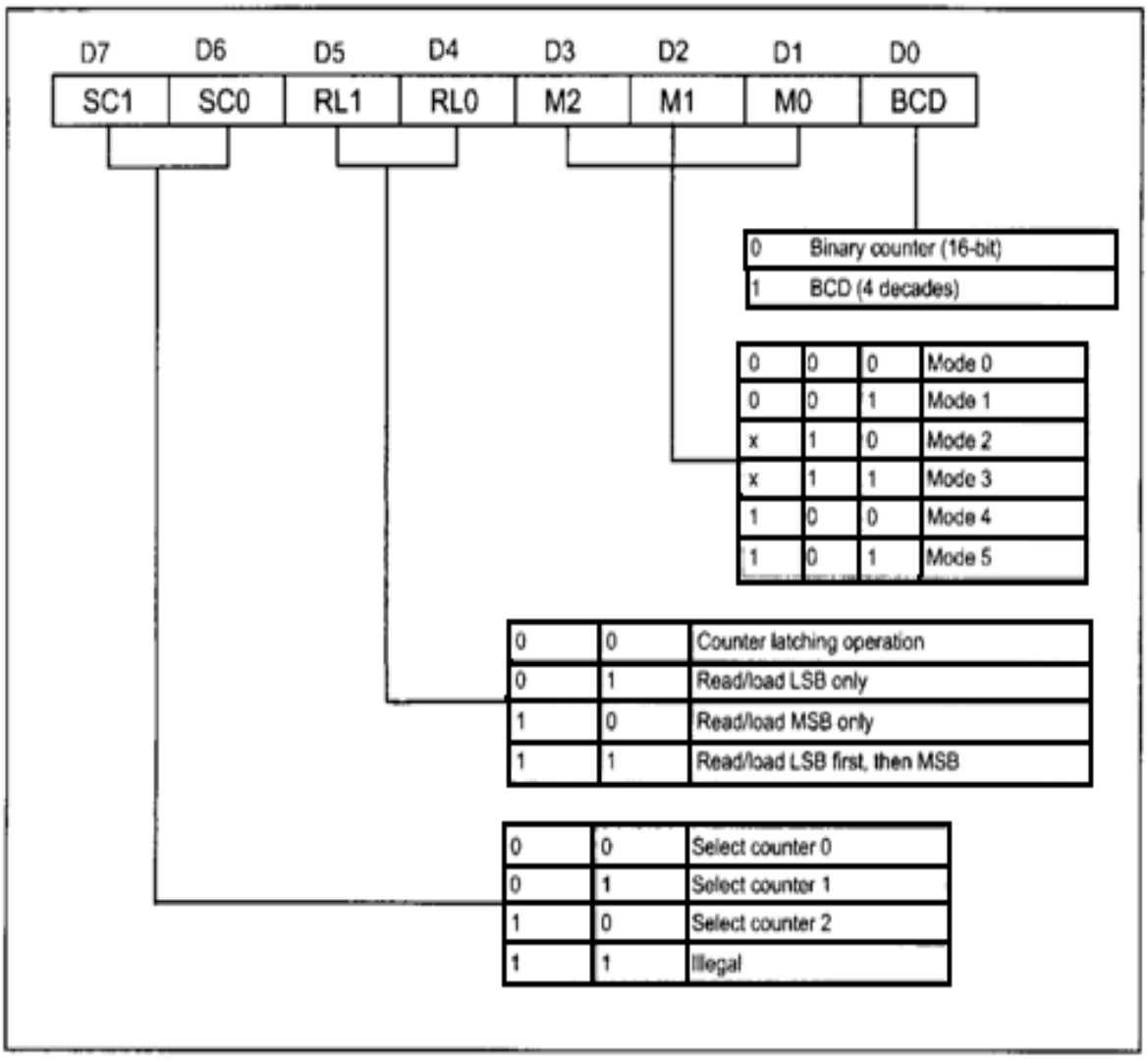




9BH => 1001 1011 =>
b6b5=00-> Mode0
b4=0-> Port A as input.
b3=1-> Port C as input (PC7-PC4)
b2=0-> Mode 0
b1=1-> Port B as input
b0=1-> Port C as input (PC3-PC0).

Q.3 Write a control word for counter 1 of 8253 / 8254 that selects the following options: load least significant byte only, mode 5 of operation and binary counting. Then write an instruction sequence that will load the control word into 8253 / 8254 that is located at address 01000 h of memory address space. Assume that 8253 / 8254 is attached to the I/O bus of the CPU and the address inputs A_0 and A_1 are supplied by A_2 and A_3 respectively. **(5)**

Ans



8253/54 Control Word Format

Based on the above given conditions and assuming counter 0 is used. The control word becomes 0001 1010h.

Identify the port address

- The CS is enabled when A7=1
- The Control Register is selected when A1 and A0 =1
- Assuming unused address lines A6 to A2 are at logic 0,

Then port address will be as follows

Control Register = 83H
Counter 2 = 82H

```
    MVI A,B0H
    OUT 83H
    MVI A, LOWBYTE
    OUT 82H
    MVI A,HIGHBYTE
    OUT 82H
LOOP:MVI A,80H
    OUT 83H
    IN 82H
    MOV D,A
    IN 82H
    ORA D
    JNZ LOOP
    RET
```

Q.4 'Pentium processor has a superscalar architecture'. Explain the meaning of the statement. **(4)**

Ans

The Pentium microprocessor is organized with three execution units. One executes floating-point instructions, and the other two (U-pipe and V-pipe) execute integer instructions. This means that it is possible to execute three instructions simultaneously.

Q.5 Write a short note on RS-232-C. **(8)**

Ans

The RS-232 standard is a collection of connection standards between different pieces of equipment. The EIA RS-232 serial communication standard is a universal standard, originally used to connect teletype terminals to modem devices. In a modern PC the RS-232 interface is referred to as a COM port. The COM port uses a 9-pin D-type connector (Refer Fig (a)) to attach to the RS-232 cable. The RS-232 standard defines a 25-pin D-type connector (Refer Fig (b)) but IBM reduced this connector to a 9-pin device so as to reduce cost and size.

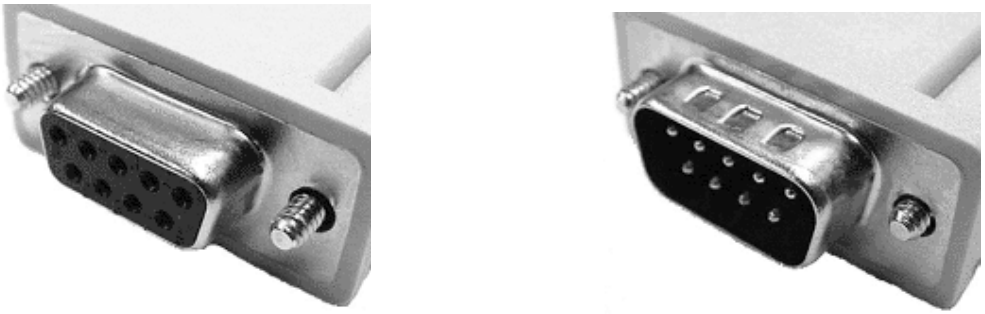


Fig (a) Female & Male "DB-9" Connector

Fig 1(b) Female & Male "DB-25" Connector

Q.6 Explain the terms: simplex, half duplex and full duplex. **(6)**

Ans

Simplex Transmission

Data in a simplex channel is always one way. Simplex channels are not often used because it is not possible to send back error or control signals to the transmit end. An example of a simplex channel in a computer system is the interface between the keyboard and the computer, in that key codes need only be sent one way from the keyboard to the computer system.

Half Duplex Transmission

A half duplex channel can send and receive, but not at the same time. It's like a one-lane bridge where two-way traffic must give way in order to cross. Only one end transmits at a time, the other end receives.

Full Duplex Transmission

Data can travel in both directions simultaneously. There is no need to switch from transmit to receive mode like in half duplex. It's like a two lane bridge on a two-lane highway.

Q.7 How DRAM's are different from SRAM's? Why DRAMs are said to employ address multiplexing? **(4)**

Ans

Dynamic RAM (DRAM) is essentially the same as SRAM, except that it retains data for only 2 or 4 ms on an internal capacitor. After 2 or 4 ms, the contents of the DRAM must be completely rewritten (refreshed) because the capacitors, which store logic 1 or logic 0, lose their charges. The entire content of the memory is refreshed with 256 reads in a 2-to-4 ms interval. Refreshing also occurs during a write, a read or during a special refresh cycle.

Q.8 Explain the operation of 8279. Explain the following terms:
(i) N key Roll over.
(ii) Key board debounce.
(iii) FIFO RAM. **(9)**

Ans
The 8279 is a programmable keyboard and display interfacing component that scans and encodes up to a 64-key keyboard and controls up to a 16-digit numerical display. The keyboard interface has built in first-in first-out (FIFO) buffer that allows it store up to eight keystrokes before the microprocessor must retrieve a character. The display section controls up to 16 numeric displays from an internal 16 X 8 RAM that stores the coded display information.
The keyboard section consists of eight lines that can be connected to eight columns of a keyboard, plus two additional lines as well as to shift and CNTL/STB keys. The key pressed are automatically debounced and the keyboard can operate in two modes two –key lock out or n-key rollover. If two keys in the two –key lock out mode are pressed simultaneously, only first key is recognized. In the N-key roll over mode, simultaneous key are recognized and their codes are stored in the internal buffer.

Q.9 What are the differences between CGA and VGA graphics adapters? **(4)**

Ans
The Color Graphics Adapter (CGA), originally also called the Color/Graphics Adapter or IBM Color/Graphics Monitor Adapter introduced in 1981, was IBM's first color graphics card, and the first color computer display standard for the IBM PC.
The standard IBM CGA graphics card was equipped with 16 kilobytes of video memory, and could be connected either to a NTSC-compatible monitor or TV via an RCA jack, or to a dedicated 4-bit "RBGI" interface CRT monitor, such as the IBM 5153 color display.
The term Video Graphics Array (VGA) refers specifically to the display hardware first introduced with the IBM PS/2 line of computers in 1987, but through its widespread adoption has also come to mean either an analog computer display standard, the 15-pin D-subminiature VGA connector or the 640×480 resolution itself. While this resolution has been superseded in the personal computer market, it is becoming a popular resolution on mobile devices.
VGA was officially superseded by IBM's XGA standard, but in reality it was superseded by numerous slightly different extensions to VGA made by clone manufacturers that came to be known collectively as "Super VGA".

Q.10 What do you mean by A/D conversion? Explain any one of the following A/D techniques:

(i) Successive approximation.
(ii) Parallel / flash converter.

(5)

Ans
The electronic circuit, which translates an analog signal into a digital signal, is known as Analog - to – Digital converter (ADC).

(i) Successive approximation ADC

One method of addressing the digital ramp ADC's shortcomings is the so-called successive approximation ADC. The only change in this design is a very special counter circuit known as a successive-approximation register. Instead of counting up in binary sequence, this register counts by trying all values of bits starting with the most significant bit and finishing at the least-significant bit. Throughout the count process, the register monitors the comparator's output to see if the binary count is less than or greater than the analog signal input, adjusting the bit values accordingly. The way the register counts is identical to the "trial-and-fit" method of decimal-to-binary conversion, whereby different values of bits are tried from MSB to LSB to get a binary number that equals the original decimal number. The advantage to this counting strategy is much faster results: the DAC output converges on the analog signal input in much larger steps than with the 0-to-full count sequence of a regular counter.

Without showing the inner workings of the successive-approximation register (SAR), the circuit looks like this:

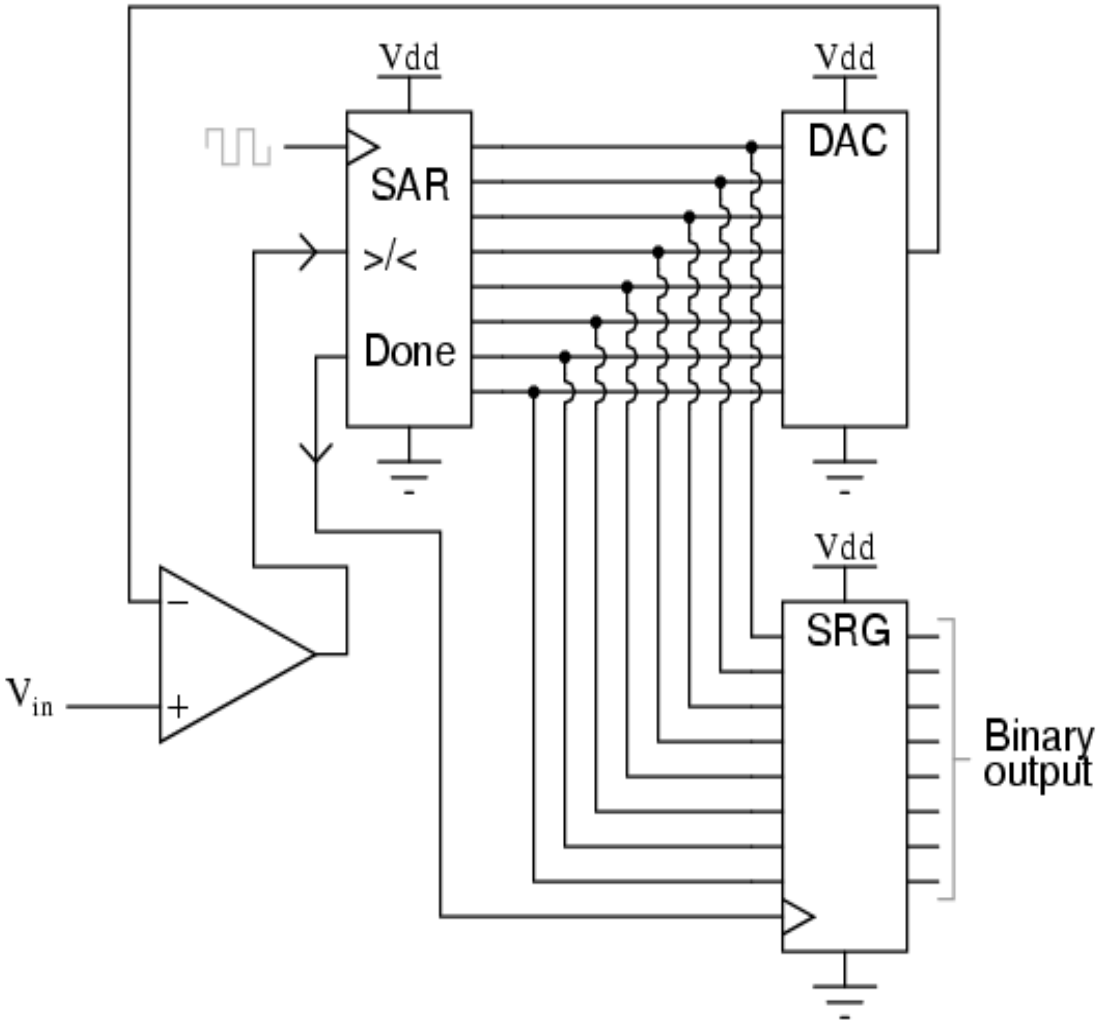


Fig: Successive Approximation ADC Circuit

It should be noted that the SAR is generally capable of outputting the binary number in serial (one bit at a time) format, thus eliminating the need for a shift register. Plotted over time, the operation of a successive-approximation ADC looks like this:

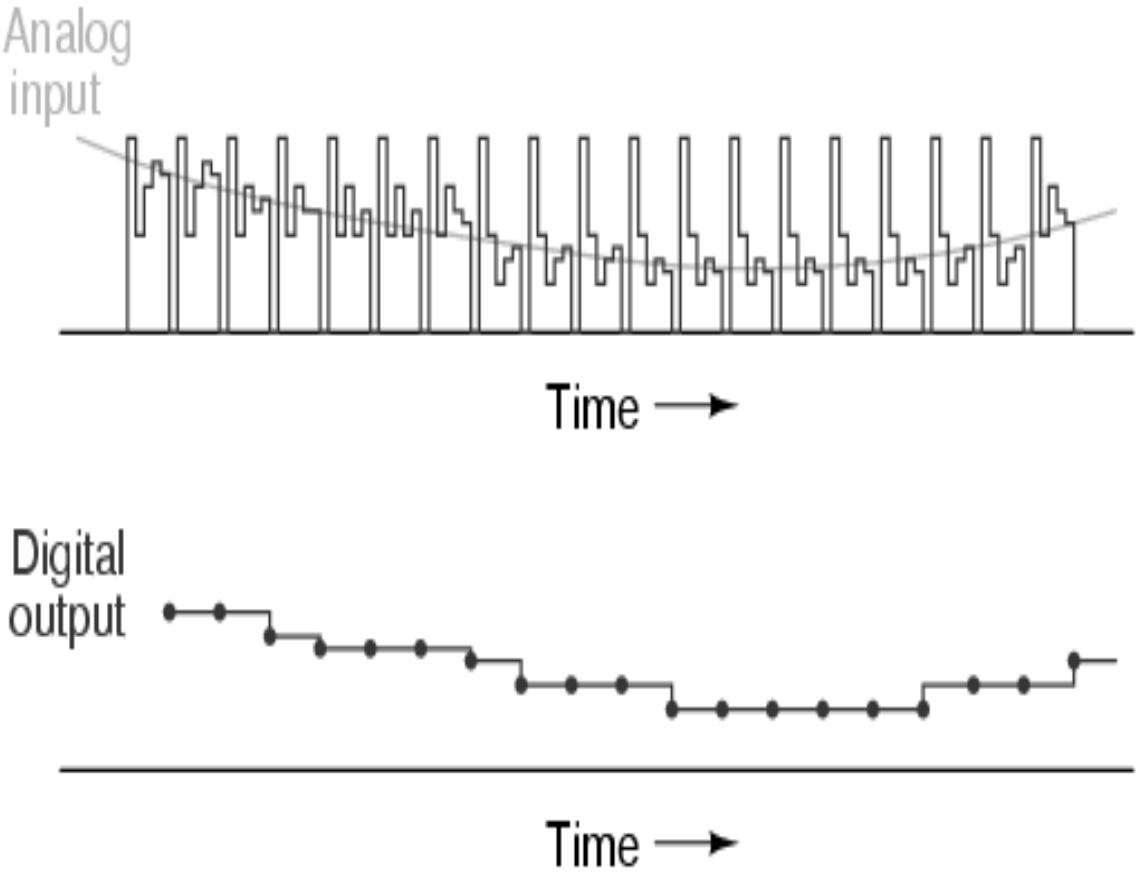


Fig: Successive Approximation ADC Circuit Input and output Waveforms

ii. Parallel / flash converter.

Also called the parallel A/D converter, this circuit is the simplest to understand. It is formed of a series of comparators, each one comparing the input signal to a unique reference voltage. The comparator outputs connect to the inputs of a priority encoder circuit, which then produces a binary output. The following illustration shows a 3-bit flash ADC circuit:

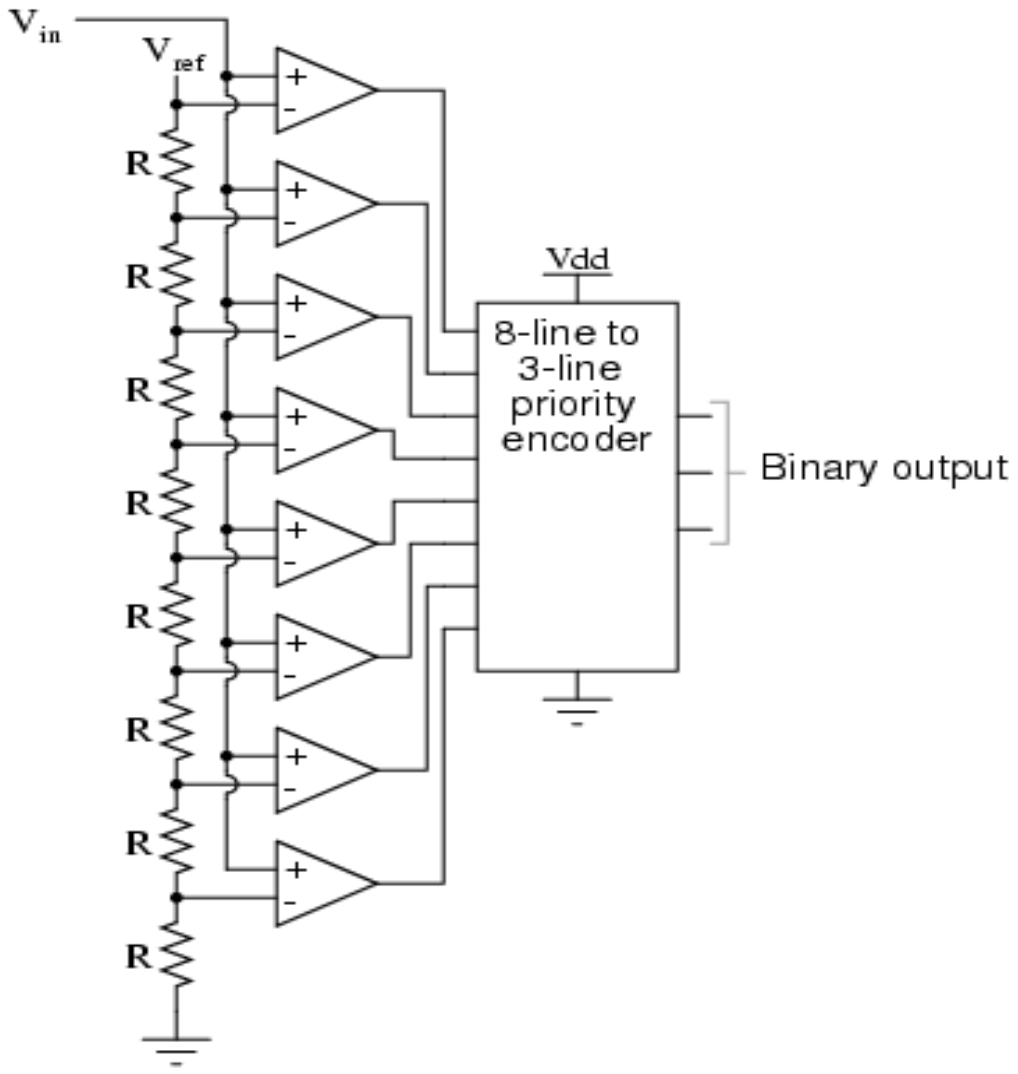


Fig: FLASH ADC Circuit

Vref is a stable reference voltage provided by a precision voltage regulator as part of the converter circuit, not shown in the schematic. As the analog input voltage exceeds the reference voltage at each comparator, the comparator outputs will sequentially saturate to a high state. The priority encoder generates a binary number based on the highest-order active input, ignoring all other active inputs.

Q.11 What do you mean by external and internal data bus? How are these two related in 8088 processor. **(2)**

Ans Internal Data Bus: A bus that operates only within the internal circuitry of the CPU, communicating among the internal caches of memory that are part of the CPU chip's design. This bus is typically rather quick and is independent of the rest of the computer's operations.
External Data Bus: A bus that connects a computer to peripheral devices. The 8088 microprocessor has 16-bit registers, 16-bit internal data bus and 20-bit address bus, which allows the processor address up to 1 MB of memory.

Q.12 What is the difference between XT and AT computer system? **(2)**

Ans
XT ->extended and AT->Advanced Technology
Some differences between the PC and XT include the type of power supply originally included--63 vs 135 watts; the number and spacing of expansion slots--5 vs 8; the PC has a cassette tape interface connector on the back.

Q.13 What are program-invisible registers? **(2)**

Ans the global and local descriptor tables are found in the memory system. In order to access and specify the address of these tables, the program invisible registers used. The program invisible registers are not directly addressed by software so they are given name.
The GDTR (global descriptor table register) and IDTR (interrupt descriptor table register) contain the base addresses of the descriptor table and its limit. The limit of each descriptor table is 16 bits because the maximum table length is 64 Kbytes. When the protected mode operation is desired, the address of the global descriptor table and its limit are loaded into the GDTR.

Q.14 The interrupt vector table is always created in the first 1K area of the memory. Justify the statement. **(2)**

Ans When the CPU receives an interrupt type number from the PIC, it uses this number to look up the corresponding interrupt vector in memory. There are 256 interrupt types. Each interrupt vector occupies 4 bytes. Therefore, a total of 4 x 256 = 1K bytes of memory is reserved at the beginning of the processor memory address space for storing interrupt vectors.

Q.15 What is the purpose of carry (c) flag and zero (z) flag? **(2)**

Ans Carry flag holds the carry after addition or the borrow after subtraction. The carry flag also indicates error conditions, as dictated by some programs and procedures.
The Zero flag shows that the result of an arithmetic or logical operation is zero. If Z=1, the result is zero; if Z=0, the result is not zero.

Q.16 What is 16-bit ISA? Compare it with 8-bit ISA bus. **(6)**

Ans The only difference between the 8 and 16-bit ISA bus is that an additional connector is attached behind the 8-bit connector. 16-bit ISA card contains two edge connectors. One plugs into the original 8-bit connector and other plugs into the 16-bit connector. The added features that are most often used are the additional interrupt request inputs and DMA request signals. Interfaces found for the ISA bus are modems and sound cards.

Q.17 Compare memory mapped I/O with I/O mapped I/O. **(4)**

Ans Memory Mapped I/O Scheme: In this scheme there is only one address space. Address space is defined as all possible addresses that microprocessor

can generate. Some addresses are assigned to memories and some addresses to I/O devices. An I/O device is also treated as a memory location and one address is assigned to it. In this scheme all the data transfer instructions of the microprocessor can be used for both memory as well as I/O device. This scheme is suitable for a small system.
In I/O mapped I/O scheme the addresses assigned to memory locations can also be assigned to I/O devices. Since the same address may be assigned to a memory location or an I/O device, the microprocessor must issue a signal to distinguish whether the address on the address bus is for a memory location or an I/O device.

Q.18 Explain in brief the functions of the clock generator chip, 8284. **(4)**

Ans, **8284 Clock generator:**
The 8284 is an ancillary component to the microprocessors. Without clock generator, many additional circuits are required to generate the clock in an microprocessor based system. A 8284 provides the following basic functions or signals: Clock generation, RESET synchronization, READY synchronization, and a TTL-level peripheral clock signal.

Q.19 Write a brief note on MMX technology. **(4)**

Ans, MMX (Multimedia extensions) technology adds 57 new instructions to the instruction set of the Pentium – 4 microprocessors. The MMX technology also introduces new general purpose instructions. The new MMX instructions are designed for application such as motion video, combined graphics with video, image processing, audio synthesis, speech synthesis and compression, telephony, video conferencing, 2D graphics, and 3D graphics. These new instructions operate in parallel with other operations as the instruction for the arithmetic coprocessor.
The MMX architecture introduces new packed data types. The data types are eight packed, consecutive 8-bit bytes; four packed, consecutive 16-bit words; and two packed, consecutive 32-bit double words.

Q.20 What are the different modes in which 8255 Programmable Peripheral Interface (PPI) can operate? Write the 8086 initialisation routine required to program 8255 for mode 1 with Port A and Port B as output Ports and Port C as an input port. Indicate all the relevant signals. **(6)**

Ans
- 24 I/O lines in 3 8-bit port groups – A, B, C
- A, B can be 8-bit input or output ports
- C can serve as 2 4-bit input or output ports
- 3 modes of operation:
  - Mode 0:A, B, C simple input or output level sensitive ports
  - Mode 1:A, B input or output ports with strobe control in C
  - Mode 2:A is bidirectional with control/handshake in B and C
- A, B can only change 1 byte at a time
- C has individual bit set/reset capability
- Advantage is non-dedicated circuit can change port configuration with software and no “glue logic”

- Ports A, B, and C are used for I/O data.
- The control register is programmed to select the operation mode of the three ports A, B, and C.
- Mode 0 : simple I/O mode
- Any of the ports A, B, CL and CU can be programmed as input or output.
- No control of individual bits (all bits are out or all bits are in)

Mode0:

Mode 0 operation causes the 82C55 to function as a buffered input device or as a latched output device.

In previous example, both ports A and B are programmed as (mode 0) simple latched output ports.

Port A provides the segment data inputs to display and port B provides a means of selecting one display position at a time.

The values for the resistors and the type of transistors used are determined using the current requirements (see text for details).

Textbook has the assembly code fragment demonstrating its use.

Examples of connecting LCD displays and stepper motors are also given.

Mode1:

Port A and/or port B function as latching input devices. External data is stored in the ports until the microprocessor is ready.

Port C used for control or handshaking signals (cannot be used for data).

Signal definitions for Mode 1 Strobed Input

$\overline{\text{STB}}$	The strobe input loads data into the port latch on a 0-to-1 transition
IFB	**Input buffer full** is an output indicating that the input latch contain information
INTR	**Interrupt request** is an output that requests an interrupt
INTE	The **interrupt enable signal** is neither an input nor an output; it is an internal bit programmed via the PC4(port A) or PC2(port B) bits.
PC7,PC6	The port C pins 7 and 6 are general-purpose I/O pins that are available for any purpose.

Mode2:

Only allowed with port A. Bi-directional bused data used for interfacing two computers, GPIB interface etc.

INTR	**Interrupt request** is an output that requests an interrupt
$\overline{\text{OBF}}$	**Output buffer full** is an output indicating that the output buffe contains data for the bi-directional bus
$\overline{\text{ACK}}$	**Acknowledge** is an input that enables tri-state buffers which are otherwise in their high-impedance state
$\overline{\text{STB}}$	The strobe input loads data into the port A latch
IFB	**Input buffer full** is an output indicating that the input latch contains information for the external bi-directional bus
INTE	**Interrupt enable** are internal bits that enable the INTR pin. Bit PC6(INTE1) and PC4(INTE2)
PC2,PC1 and PC0	Theses port C pins are general-purpose I/O pins that are available for any purpose.

Q.21 Explain the operation of IRET instruction. What memory locations contain the vector for an INT 34 instruction? **(4)**

Ans

The Interrupt return (IRET) instruction is used only with software or hardware interrupt service procedures. Whenever an IRET instruction executes, it stores the contents of I and T from the stack. This is important because it preserves the state of the flag bits. If interrupts were enabled before an interrupt service procedure, they automatically re-enabled by the IRET instruction because it restores the flag register.

Interrupt Number 20-FF are stored at an address 80 – 3FFH.

Q.22 Explain the following terms:

i. Branch prediction logic.
ii. Paging.
iii. Assembler.
iv. Microprocessor development system. **(8)**

Ans

(i) Branch prediction logic in Pentium: The Pentium microprocessor uses branch prediction logic to reduce the time required for a branch caused by internal delays. These delays are minimized because when a branch instruction is encountered, the microprocessor begins pre-fetch instruction at the branch address. The instructions are loaded into the instruction cache, so when the branch occurs, the instructions are present and allow the branch to execute in one clocking period. If for any reason the branch prediction logic errors, the branch requires an extra three clocking periods to execute. In most cases, the branch prediction is correct and no delay ensues.

(ii) Paging Unit: The paging mechanism functions with 4K – byte memory pages or with a new extension available to the Pentium with 4M byte-memory pages. In the Pentium, with the new 4M-byte paging feature memory for the page-table reduced to single page table.

(iii) Assembler: An assembler or macro-assembler generally forms a part of the operating system. Which translates a assembly language program into machine language program.

(iv) Microprocessor development system: Computer systems have undergone many changes recently. Machines that once filled large areas have been reduced to small desktop computer systems because of the microprocessor. Although these desktop computers are compact, they possess computing power that was only dreamed of a few years ago.

The blocks of the microprocessor based system are

1. The Memory and I/O System
2. The DOS Operating System
3. The Microprocessor

Q.23 Explain the following instructions:
(i) TEST (ii) NEG (iii) CMP (iv) DAA. **(8)**

Ans

(i) TEST: The TEST instruction performs the AND operation. The difference is that the AND instruction changes the destination operand, while the TEST instruction does not. A TEST only affects the condition of the flag register, which indicates the result of the test.

(ii) NEG: Arithmetic sign inversion or two's complement (NEG). The NEG instruction two's complements a number, which means that the arithmetic sign of a signed number changes from positive to negative or from negative to positive.

(iii)CMP: The comparison instruction (CMP) is a subtraction that changes only the flag bits; the destination operand never changes. A comparison is useful for checking the entire contents of a register or a memory location against another value. A CMP is normally followed by a conditional jump instruction, which tests the condition of the flag bits

(iv)DAA: The DAA instruction follows the ADD or ADC instruction to adjust the result into a BCD result. The DAA instruction functions only with the AL register, this addition must occur eight bits at a time.

Q.24 With respect to serial communication define the following:
(i) baud rate. (ii) asynchronous communication.
(iii) parity. (iv) half duplex. **(4)**

Ans

Half Duplex Transmission: A half duplex channel can send and receive, but not at the same time. It's like a one-lane bridge where two-way traffic must give way in order to cross. Only one end transmits at a time, the other end receives.
Asynchronous means "no synchronization", and thus does not require sending and receiving idle characters. However, the beginning and end of each byte of data must be identified by start and stop bits. The start bit indicates when the data byte is about to begin and the stop bit signals when it ends. The requirement to send these additional two bits causes asynchronous communication to be slightly slower than synchronous however it has the advantage that the processor does not have to deal with the additional idle characters.

The rate of data transfer in serial data communication is denoted in bps. Bits per second (bps) is the rate of transfer of information bits. Baud is the number of signal level changes per second in a line, regardless of the information content of those signals. The baud and bps rates are not necessarily equal. The ratio of BPS to baud depends on the information-coding scheme that you are using. For example, each character in asynchronous RS-232 coding includes a start and stop bit that are not counted as information bits, so the BPS rate is actually less than the baud rate.

Besides the synchronization provided by the use of start and stop bits, an additional bit called a parity bit may optionally be transmitted along with the data. Figure shows the inclusion of an additional parity bit for error control purposes. A parity bit affords a small amount of error checking, to help detect data corruption that might occur during transmission. You can choose even parity, odd parity, mark parity, space parity or none at all. When even or odd parity is being used, the numbers of marks (logical 1 bit) in each data byte are counted, and a single bit is transmitted following the data bits to indicate whether the number of 1 bit just sent is even or odd.

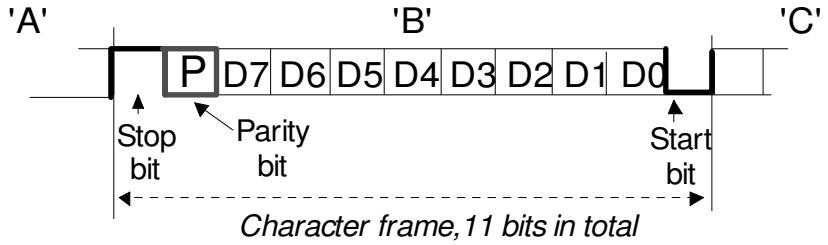


Fig. Framed data including a parity bit

For example, when even parity is chosen, the parity bit is transmitted with a value of 0 if the number of preceding marks is an even number. For the binary value of 0110 0011 the parity bit would be 0. If even parity were in effect and the binary number 1101 0110 were sent, then the parity bit would be 1.

Q.25 What is the importance of RS232-C in serial communication? Name some application where you see its use. **(4)**

Ans RS-232 stands for Recommend Standard number 232 and C is the latest revision of the standard. The serial ports on most computers use a subset of the RS-232C standard. The full RS-232C standard specifies a 25-pin "D" connector of which 22 pins are used. Most of these pins are not needed for normal PC communications, and indeed, most new PCs are equipped with male D type connectors having only 9 pins. In the world of serial communications, there are two different kinds of equipment:

- DTE - Data Terminal Equipment
- DCE - Data Communications Equipment

Q.26 Write short notes on (Any **FOUR**):-

(i) 8259.
(ii) Real time clock.
(iii) Real and protected mode.
(iv) Super scalar architecture.
(v) Comparison between Motorola processors and INTEL processors.

(4 x 4 = 16)

Ans

(i) 8259:

The 8259A adds 8 vectored priority encoded interrupts to the microprocessor. It can be expanded to 64 interrupt requests by using one master 8259A and 8 slave units. CS and WR must be decoded. Other connections are direct to microprocessor.

The pins D7 – D0: the bidirectional data connection, IR7 – IR0: Interrupt request, used to request an interrupt & connect to a slave in a system with multiple 8259A. WR :-Connects to a write strobe signal (lower or upper in a 16 bit system) , RD :- Connects to the IORC signal , INT :- Connects to the INTR pin on the microprocessor from the master and is connected to a IR pin on a slave and INTA :- Connects to the INTA pin on the microprocessor. In a system only the master INTA signal is connected

A0 :- Selects different command words with in the 8259A, CS :- Chip select - enables the 8259A for programming and control, SP/EN :- Slave Program (1 for master, 0 for slave)/Enable Buffer (controls the data bus transceivers in a large microprocessor based system when in buffered mode) and CAS2-CAS0 :- Used as outputs from the master to the slaves in cascaded systems.

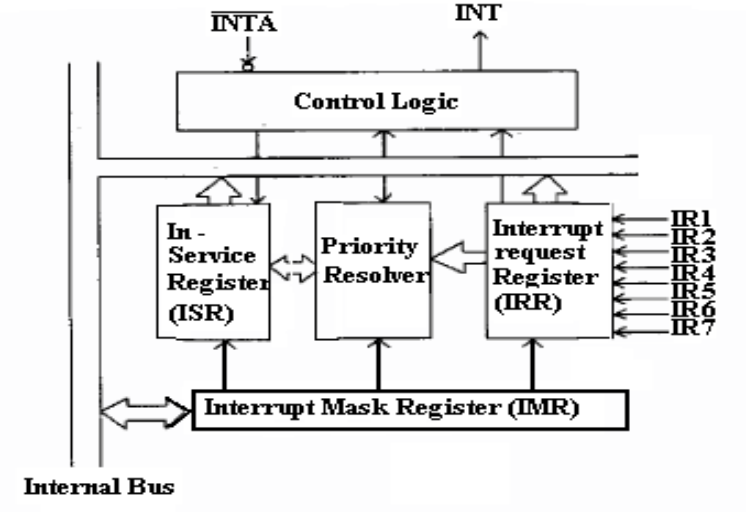


Fig : 8259 Block Diagram

(ii) Real time clock:

A real-time clock keeps the time in real time – that is, in hours and minutes. The software for the real-time clock contains an interrupt service procedure that is called 60 times per second and a procedure that updates the count located in four memory locations.

Assembler directives:

An assembler directive is a statement to give direction to the assembler to perform the task of assembly process. The assembler directives control organization of the program and provide necessary information to the assembler to understand assembly language programs to generate machine codes. They indicate how an operand or section of a program is to be processed by the assembler. An assembler supports directives to define data, to organize segments to control procedures, to define macros etc.

(iii) Real and protected mode:

Operation of Real mode interrupt: When the microprocessor completes executing the current instruction, it determines whether an interrupt is active by checking (1) instruction execution, (2) single –step, (3) NMI, (4) co-processor segment overrun, (5) INTR, and (6) INT instruction in the order presented. If one or more of these interrupt conditions are present, the following sequence of events occurs:

1. The contents of the flag register are pushed onto the stack
2. Both the interrupt (IF) and trap (TF) flags are cleared. This disables the INTR pin and the trap or single-step feature.
3. The contents of the code segment register (CS) are pushed onto the stack.
4. The contents of the instruction pointer (IP) are pushed onto the stack.
5. The interrupt vector contents are fetched, and then placed into both IP and CS so that the next instruction executes at the interrupt service procedure addressed by the vector.

Protected mode interrupt:

In the protected mode, interrupts have exactly the same assignments as in real mode, but the interrupt vector table is different. In place of interrupt vectors, protected mode uses a set of 256 interrupt descriptors that are stored in an interrupt descriptor table (IDT).

(iv). Super scalar architecture:

The Pentium microprocessor is organized with three execution units. One executes floating-point instructions, and the other two (U-pipe and V-pipe) execute integer instructions. This means that it is possible to execute three instructions simultaneously.

(v). Comparison between Motorola processors and INTEL processors:

AMD/Intel processors are really about the same thing. They run the same software and operate in a very similar manner. AMD is often less expensive than Intel, and depending on what you use a computer for one may be somewhat faster than the other.

Motorola has been largely relegated to the "also-ran" category of microprocessor manufactures since Apple computer stopped using them in favor of the IBM Power PC processor (Apple has since switched to Intel).

Motorola had an excellent 32 bit processor design years before Intel. Furthermore, the design of the Motorola 68000 processor line (from a programmer's perspective) was immensely better. The two major features of the 68000 line that made this true were
1) Orthogonality of register access and
2) Number of registers available.
These features made writing code for Motorola CPUs much simpler.

Q.27 What is (i) USB (ii) AGP (iii) XMS (iv) EMS (v) TSR (vi) EDO RAM **(6)**

Ans

(i). **USB:** The USB (UNIVERSAL SERIAL BUS) is intended to connect peripheral devices such as keyboards, a mouse, modems, and sound cards to the microprocessor through a serial data path and a twisted pair of wires. The main idea is to reduce system cost by reducing the number of wires. Another advantage is that the sound system can have a separate power supply from the PC, which means less noise. The data transfer rates through the USB are 12 Mbps at present.

(ii). **AGP:** The latest addition to many computer systems is the inclusion of the **accelerated graphics port** (AGP). The AGP operates at the bus clock frequency of the microprocessor. It is designed so that a transfer between the video card and the system memory can progress at a maximum speed. The AGP can transfer data at a maximum rate of 528M bytes per second. This port probably will never be used for any devices other than the video card.

(iii). **XMS:**The memory system is divided into three main parts. TPA **(transient program area),** system area, and XMS (**extended memory system**). The type of microprocessor in your computer determines whether an extended memory system exists.

(iv). **EMS:**The area at location C8000H-DFFFFFH is often open or free. This area is used for the **expanded memory system** in a PC or XT system, or for the upper memory system in an AT system. The EMS allows a 64K-byte page frame of memory to be used by application programs.

(v). **TSR:**The TPA also holds TSR (terminate and stay resident) programs that remain in memory in an active state until activated by a hot-key sequence or another event such as an interrupt.

(vi). **EDO RAM:**A slight modification to the structure of the DRAM changes the device into an EDO (extended data output) DRAM device. In the EDO memory, any memory access, including a refresh, stores the 256 bits selected by RAS' into latches. These latches hold the next 256 bits of information, so in most programs, which are sequentially executed, that data are available without any wait states.

Q.28 What are program invisible registers? Explain the purpose of the GDTR. If the microprocessor sends linear address 00200000H to the paging mechanism, which paging directory entry and which page table entry is accessed? **(3)**

Ans, the global and local descriptor tables are found in the memory system. In order to access and specify the address of these tables, the program invisible registers used. The program invisible registers are not directly addressed by software so they are given name.
The GDTR (global descriptor table register) and IDTR (interrupt descriptor table register) contain the base addresses of the descriptor table and its limit. The limit of each descriptor table is 16 bits because the maximum table length is 64 Kbytes. When the protected mode operation is desired, the address of the global descriptor table and its limit are loaded into the GDTR.
For linear address 00000000H – 003FFFFFH, the first entry of the page directory is accessed. Each page directory entry represents or repages a 4-Mbyte section of the memory system. The contents of the page directory select a page table that is indexed by the next 10 bits of the linear address. This means that address 00000000H – 00000FFFH selects page directory entry 0 and page table entry 0.

Q.29 Discuss the salient features of a parallel programmable interface, 8255. **(4)**

Ans

- 24 I/O lines in 3 8-bit port groups – A, B, C
- A, B can be 8-bit input or output ports
- C can serve as 2 4-bit input or output ports
- 3 modes of operation:
  - Mode 0:A, B, C simple input or output level sensitive ports
  - Mode 1:A, B input or output ports with strobe control in C
  - Mode 2:A is bidirectional with control/handshake in B and C
- A, B can only change 1 byte at a time
- C has individual bit set/reset capability
- Advantage is non-dedicated circuit can change port configuration with software and no "glue logic"
- Ports A, B, and C are used for I/O data.
- The control register is programmed to select the operation mode of the three ports A, B, and C.
- Mode 0 : simple I/O mode
- Any of the ports A, B, CL and CU can be programmed as input or output.
- No control of individual bits (all bits are out or all bits are in)
- Mode 1: Ports A and B can be used as input or output ports with handshaking.
- Mode 2 : Port A can be used as bidirectional I/O port with handshaking

Q.30 What do you understand by assembler directives? What do the following assembler directives do?
(i) ASSUME
(ii) SEGMENT
(iii) DB
(iv) PUBLIC **(8)**

Ans

(i) ASSUME: This directive will be used to map the segment register names with memory addresses.
The Syntax is as follows:

ASSUME SS: Stackseg, DS : Dataseg, CS:Codeseg
The ASSUME will tell the assembler to use the SS register with the address of the stack segment whose name is stackseg.

(ii) SEGMENT: This directive defines to the assembler the start of a segment with name segment-name. The segment name should be unique and follows the rules of the assembler
The Syntax is as follows:
Segment Name SEGMENT {Operand (Optional)} ; Comment
.
.
.

Segment Name ENDS.

(iii) DB (Define Byte): The DB directive defines a byte-type variable (i.e. a variable which occupies one byte of memory space). In a given directive statement, there may be single initial value or multiple values of the defined variable. If there is one initial value, one byte of memory space is reserved. If there are multiple values, one byte of memory space is reserved for each value. The general format is:
Name of Variable DB Initial value or values.

(iv) The **PUBLIC** and EXTRN directives are very important to modular programming. PUBLIC used to declare that labels of code, data, or entire segments are available to other program modules. EXTRN (external) declares that labels are external to modules. Without these statements, modules could not be linked together to create a program by using modular programming techniques. They might link, but one module would not be able to communicate to another.

The PUBLIC directive is placed in the opcode field of an assembly language statement to define a label as public, so that the label can be used by other modules.

Q.31 Discuss the role of a bus arbiter in a multiprocessor configuration. **(4)**

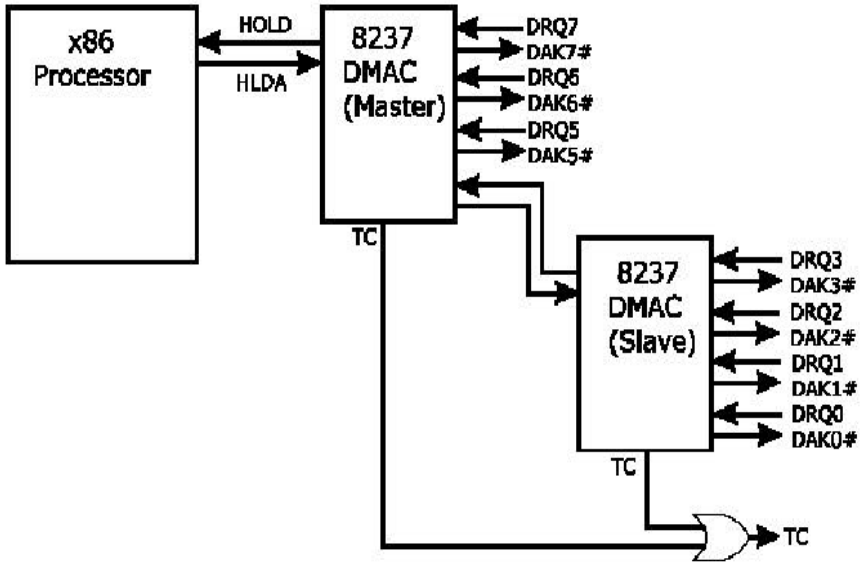
Ans, Bus arbiter: Which functions to resolve priority between bus masters and allows only one device at a time to access the shared bus. The 8289 bus arbiter controls the interface of a bus master to a shared bus. This is designed to function with the 8086/8088 microprocessors. Each bus master or microprocessor requires an arbiter for the interface to the shared bus, which Intel calls the MULTIBUS and IBM calls the MICRO CHANNEL.

The shared bus used only to pass information from one microprocessor to another; otherwise, the bus master function in their own local bus modes by using their own local programs, memory, and I/O space. Microprocessors connected in this kind of system are often called parallel or distributed processors because they can execute software and perform tasks in parallel.

Q.32 Show how a typical DMA controller can be interfaced to an 8086/8085 based maximum mode system. **(8)**
Ans, **For 8088 in maximum mode:**

The RQ/GT1 and RQ/GT0 pins are used to issue DMA request and receive acknowledge signals. Sequence of events of a typical DMA process

1) Peripheral asserts one of the request pins, e.g. RQ/GT1 or RQ/GT0 (RQ/GT0 has higher priority)
2) 8088 completes its current bus cycle and enters into a HOLD state
3) 8088 grants the right of bus control by asserting a grant signal via the same pin as the request signal.
4) DMA operation starts
5) Upon completion of the DMA operation, the peripheral asserts the request/grant pin again to relinquish bus control.



Q.33 What is a co-processor? What is its use in a typical microprocessor based system. **(8)**

Ans 8087 NDP (numerical data processor) is also known as math co-processor which is used in parallel with the main processor for number crunching applications, which would otherwise require complex programming. It is also faster than 8086/8088 processor in performing mathematical computation. It has its own specialized instruction sets to handle mathematical programs.

It is a processor which works in parallel with the main processor. It has its own set of specialized instructions. The number crunching part of the program is executed by 8087. Instruction for 8087 are written in the main program interspersed with the 8086 instructions. All the 8087 instruction codes have 11011 as the most significant bits of their first code byte.

Q.34 What is segmentation? What are its advantages? How is segmentation implemented in typical microprocessors? **(8)**

Ans

Segment memory addressing divides the memory into many segments. Each of these segments can be considered as a linear memory space. Each of these segment is addressed by a segment register.

However since the segment register is 16 bit wide and the memory needs 20 bits for an address the 8086 appends four bits segment register to obtain the segment address. Therefore, to address the segment 10000H by , say the SS register, the SS must contain 1000H.

The first advantage that memory segmentation has is that only 16 bit registers are required both to store segment base address as well as offset address. This makes the internal circuitry easier to built as it removes the requirement for 20 bits register in case the linear addressing method is used. The second advantage is relocatability.

Q.35 What is a PCI bus? Discuss its features and usage. **(6)**

Ans

Peripheral Component Interconnect (PCI): This bus was developed by Intel and introduced in 1993. It is geared specifically to fifth- and sixth-generation systems, although the latest generation 486 motherboards use PCI as well.

PCI bus has plug – and – play characteristics and the ability to function with a 64-bit data bus. A PCI interface contains series of registers, located in a small memory device on the PCI interface that contains information about the board.

Q.36 How is EISA bus different from ISA bus? **(4)**

Ans

The Extended Industry Standard Architecture (EISA) is a 32 bit modification to the ISA bus. As computers became larger and had wider data buses, a new bus was needed that would transfer 32-bit data. The clocking speed limited up to 8MHz. The most common application for the EISA bus is a disk controller or as a video graphics adapter. These applications benefit from the wider data bus width because the data transfer rate for these devices are high.

Q.37 Differentiate between synchronous and asynchronous types of serial communication. **(6)**

Ans Serial data communication uses two basic types, synchronous and asynchronous. With synchronous communications, the two devices initially synchronize themselves to each other, and then continually send characters to stay in sync. Even when data is not really being sent, a constant flow of bits allows each device to know where the other is at any given time. That is, each character that is sent is either actual data or an idle character.

Asynchronous means "no synchronization", and thus does not require sending and receiving idle characters. However, the beginning and end of each byte of data must be identified by start and stop bits. The start bit indicates when the data byte is about to begin and the stop bit signals when it ends. The requirement to send these additional two bits causes asynchronous communication to be slightly slower than synchronous however it has the advantage that the processor does not have to deal with the additional idle characters.

Q.38 Draw and explain the block diagram of programmable interrupt controller 8259. **(8)**

Ans The 8259A adds 8 vectored priority encoded interrupts to the microprocessor. It can be expanded to 64 interrupt requests by using one master 8259A and 8 slave units. CS and WR must be decoded. Other connections are direct to microprocessor.

The pins D7 – D0: the bidirectional data connection, IR7 – IR0: Interrupt request, used to request an interrupt & connect to a slave in a system with multiple 8259A. WR :-Connects to a write strobe signal (lower or upper in a 16 bit system) , RD :- Connects to the IORC signal , INT :- Connects to the INTR pin on the microprocessor from the master and is connected to a IR pin on a slave and INTA :- Connects to the INTA pin on the microprocessor. In a system only the master INTA signal is connected

A0 :- Selects different command words with in the 8259A, CS :- Chip select - enables the 8259A for programming and control, SP/EN :- Slave Program (1 for master, 0 for slave)/Enable Buffer (controls the data bus transceivers in a large microprocessor based system when in buffered mode) and CAS2-CAS0 :- Used as outputs from the master to the slaves in cascaded systems.

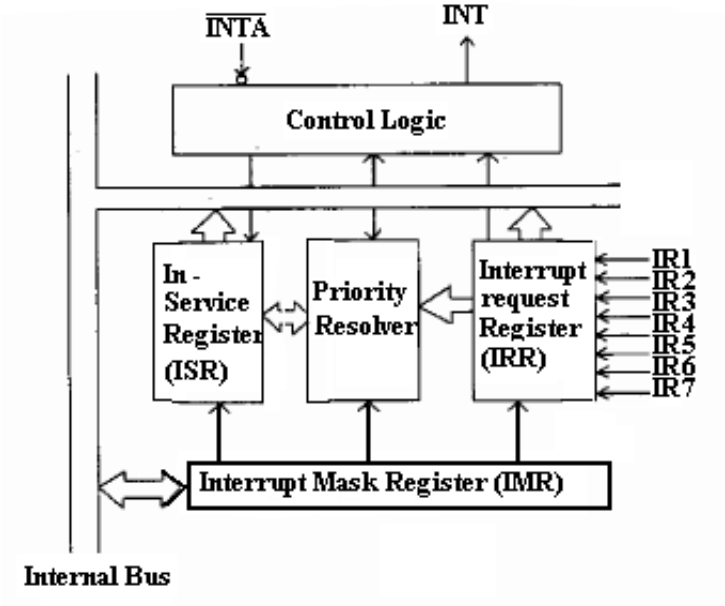


Fig : 8259 Block Diagram

Q.39 Discuss the various types of memory devices that you are familiar with. **(8)**

Ans

All of the memory used as main store in a modern computer is implemented as semiconductors fabricated on wafers of silicon. Semiconductor memory is fast and easy to use. To fulfil the needs of modern computer systems it is becoming increasingly dense (more bits per chip) and cheap.

A semiconductor memory chip consists of a large number of cells organized into an array, and the logic necessary to access any array in the cell easily. Semi-conductor memory may be classed according to the mechanism used by each cell to store data. The simplest type of memory is called static memory. In static

memory each cell uses a flip-flop made from four or six transistors. The data in each cell is remembered until the power is switched off. Static memory is easy to use and reliable, but is relatively bulky, slow and expensive. Most computer systems therefore use dynamic memory as their main store. Dynamic memory uses just a single transistor per cell, and is therefore denser, faster and cheaper. Unfortunately each cell gradually forgets the data stored in it, and so extra circuitry must be used to continually refresh the cells.

Memory, with regard to computers, most commonly refers to semiconductor devices whose contents can be accessed (i.e., read and written to) at extremely high speeds. The main characteristics of semiconductor memory are based on capacity, organization and access time. In microprocessor-based systems semiconductor memories are used as primary storage for code and data.

In contrasts with storage, which (1) retains programs and data regardless of whether they are currently in use or not, (2) retains programs and data after the power supply has been disconnected, (3) has much slower access speeds and (4) has a much larger capacity (and a much lower cost). Examples of storage devices are hard disk drives (HDD), floppy disks, optical disks (e.g., CDROMS and DVDs) and magnetic tape.

The term memory as used in a computer context originally referred to the magnetic core memory devices that were used beginning in the 1950s. It was subsequently applied to the semiconductor memory devices that replaced core memories in the 1970s.

Computer memory today consists mainly of dynamic random access memory (DRAM) chips that have been built into multi-chip modules that are, in turn, plugged into slots on the motherboard (the main circuit board on personal computers and workstations). This DRAM is commonly referred to as RAM (random access memory), and it constitutes the main memory of a computer.

The random in random access memory refers to the fact that any location in such memory can be addressed directly at any time. This contrasts with sequential access media, such as magnetic tape, which must be read partially in sequence regardless of the desired content.
There are three basic kinds of memory used in microprocessor systems - commonly called ROM, RAM, and hybrid. ROM and RAM are - "Read Only Memory" and "Random Access Memory". The program may be stored in ROM or RAM - the program does not normally change while it executes - while data is stored in the registers and RAM. Of course, if you turn off the chip and turn it on again, you have lost all the contents of the registers, and RAM.
In a typical computer, as much as possible is in RAM, to give the maximum possible flexibility; you have basic programmes allowing you to interact with discs, keyboards and the display in ROM, and load in as much of the software as possible when you run the programs.

Q.40 Write explanatory notes on Microprocessor development system. **(16)**

Ans,

Microprocessor development system:

Computer systems have undergone many changes recently. Machines that once filled large areas have been reduced to small desktop computer systems because of

the microprocessor. Although these desktop computers are compact, they possess computing power that was only dreamed of a few years ago.
The blocks of the microprocessor based system are

1. The Memory and I/O System
2. The DOS Operating System
3. The Microprocessor

Q.41 Discuss DOS function call and BIOS function call with one example of each. **(5)**

Ans
DOS function call:
In order to use DOS function calls, always place the function number into register AH and load all other pertinent information into registers, as described in the entry data table (Refer Text1-page no 809). Once this is accomplished, follow with an INT 21H to execute the DOS function.
Example: MOV AH, 6
MOV DL, 'A'
INT 21H.
Example shows how to display an ASCII A on the CRT screen at the current cursor position with a DOS function call.
BIO stands for Basic Input Output System. It is a set of programs to provide most basic low-level services such as services keyboard, disks, serial port, printer, display, and bootstrap. BIOS programs are stored in a ROM. When power is switched on ROM-BIOS takes the control of a computer. First of all, ROM-BIOS programs for power-on-self test are executed. These tests check that whether the computer is in proper working order after this test, the process of loading the operating system into main memory is called booting. ROM-BIOS contains a program called bootstrap loader, this directs CPU to read from the disk a specific program called boot and to load it into main memory.
BIOS function calls are found stored in the system and video BIOS ROMs. These BIOS ROM function directly control the I/O devices, with or without DOS loaded into a system.
INT10H: This is a BIOS interrupt is often called the video services interrupt because it directly controls the video display in a system. The INT10H instruction uses a register AH to select video services provided by this interrupt. The video BIOS ROM is located on the video board and varies from one video card to another.
INT11H: This function used to determine the type of equipment installed in the system.
INT12H: The memory size is returned by the INT 12 H instructions.
INT13H: This call controls the diskettes and also fixed or hard disk drives attached to the system.
INT14H: This call controls the serial COM ports attached to the computer.

Q.42 Differentiate between real and protected modes of an Intel microprocessor. Discuss protected mode memory addressing in brief. **(7)**

Ans Operation of Real mode interrupt: When the microprocessor completes executing the current instruction, it determines whether an interrupt is active by checking (1) instruction execution, (2) single –step, (3) NMI, (4) co-processor segment overrun, (5) INTR, and (6) INT instruction in the order presented. If one or

more of these interrupt conditions are present, the following sequence of events occurs:

1. The contents of the flag register are pushed onto the stack
2. Both the interrupt (IF) and trap (TF) flags are cleared. This disables the INTR pin and the trap or single-step feature.
3. The contents of the code segment register (CS) are pushed onto the stack.
4. The contents of the instruction pointer (IP) are pushed onto the stack.
5. The interrupt vector contents are fetched, and then placed into both IP and CS so that the next instruction executes at the interrupt service procedure addressed by the vector.

Protected mode interrupt:

In the protected mode, interrupts have exactly the same assignments as in real mode, but the interrupt vector table is different. In place of interrupt vectors, protected mode uses a set of 256 interrupt descriptors that are stored in an interrupt descriptor table (IDT).

Q.43 What do you mean by the term procedure? What is the difference between near call and far call? **(4)**

Ans

PROC: The PROC and ENDP directives indicate the start and end of a procedure. These directives force structure because the procedure is clearly defined. The PROC directive indicates the start of a procedure, must also be followed with a NEAR or FAR. A NEAR procedure is one that resides in the same code segment as the program. A FAR procedure may reside at any location in the memory system.

Q.44 Design an address decoding logic using a 3:8 decoder (74138) to interface a total of 64k memory locations in the address range from F0000 to FFFFF. Divide 64k memory locations in eight blocks of 8 k locations each and generate eight chip select signals. **(8)**

Ans

Text1-page 350

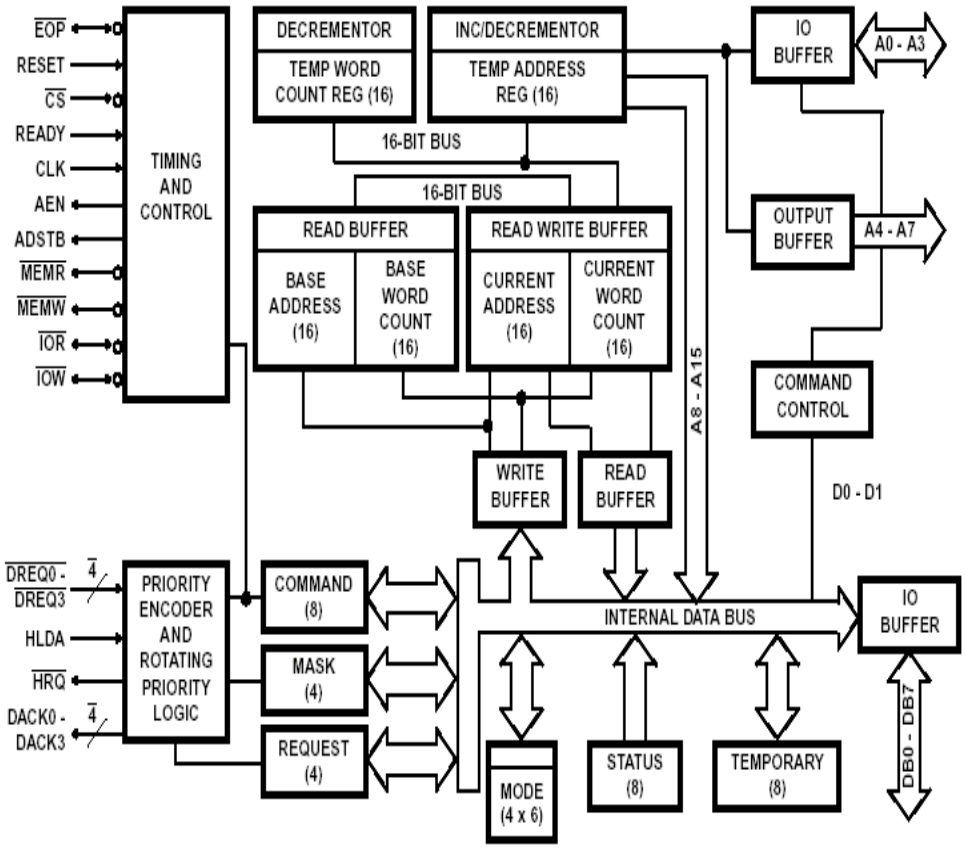
Q.45 Draw and explain the block diagram of DMA controller. Also explain the various modes in which DMAC works. **(8)**

Ans

Direct memory access (DMA) is a process in which an external device takes over the control of system bus from the CPU.DMA is for **high-speed data transfer** from/to mass storage peripherals, e.g. hard disk drive, magnetic tape, CD-ROM, and sometimes video controllers. For example, a hard disk may boasts a transfer rate of 5 M bytes per second, i.e.1 byte transmission every 200 ns. To make such data transfer via the CPU is both undesirable and unnecessary.

The basic idea of **DMA** is to transfer blocks of data directly between memory and peripherals. The data don't go through the microprocessor but the data bus is occupied. "Normal" transfer of one data byte takes up to 29 clock cycles. The DMA transfer requires only 5 clock cycles.

Block Diagram



The modes of operation include demand mode, single mode, block mode, and cascade mode. Demand mode transfers data until an external EOP is input or until the DREQ input becomes inactive. Single mode releases the HOLD after each byte of data transferred. Block mode automatically transfers the number of bytes indicated by the count register for the channel. Cascade mode is used when more than one 8237 is present in a system.

Q.46 What is DRAM? What do you understand by DRAM refreshing? With the help of a block diagram, show how DRAM can be interfaced to a microprocessor. **(6)**

Ans Dynamic RAM (DRAM) is essentially the same as SRAM, except that it retains data for only 2 or 4 ms on an internal capacitor. After 2 or 4 ms, the contents of the DRAM must be completely rewritten (refreshed) because the capacitors, which store logic 1 or logic 0, lose their charges. The entire content of the memory is refreshed with 256 reads in a 2-to-4 ms interval. Refreshing also occurs during a write, a read or during a special refresh cycle.
Text 1 – Fig (page 342).

Q.47 Discuss mode –2 (bi-directional mode) of 8255 (Programmable Peripheral Interface). **(6)**

Ans

Only allowed with port A. Bi-directional bused data used for interfacing two computers, GPIB interface etc.

INTR	**Interrupt request** is an output that requests an interrupt
$\overline{\textbf{OBF}}$	**Output buffer full** is an output indicating that the output buffe contains data for the bi-directional bus
$\overline{\textbf{ACK}}$	**Acknowledge** is an input that enables tri-state buffers which are otherwise in their high-impedance state
$\overline{\textbf{STB}}$	The strobe input loads data into the port A latch
IFB	**Input buffer full** is an output indicating that the input latch contains information for the external bi-directional bus
INTE	**Interrupt enable** are internal bits that enable the INTR pin. Bit PC6(INTE1) and PC4(INTE2)
PC2,PC1 and PC0	Theses port C pins are general-purpose I/O pins that are available for any purpose.

Q.48 Discuss the following: **(ANY THREE) (12)**

(i) Some features of Pentium series of microprocessors.
(ii) Virtual memory.
(iii) MMX Technology.
(iv) Graphics adapters.

Ans

(i). **Some features of Pentium series of microprocessors:**

The Pentium is a 32-bit superscalar, CISC microprocessor. The term superscalar is used for the processor which contains more than one pipeline to execute more than one instruction simultaneously in parallel.

The main features of Pentium are, it has two ALU's, one floating-point unit, two 8 KB cache, pre-fetch buffers, a branch target buffer. Two ALU's means that there are two pipelines. Each ALU contains five functional units. The two pipelines are integer pipelines. They are named U and V pipeline.

When Pentium was introduced, its operating frequency was 60 MHz. gradually; the operating frequency was raised to 233 MHz. The Pentium uses 0.6 micron Bi-CMOS process technology. It uses power management feature.

The memory management is improved by adding paging unit and a new system memory-management mode.

Paging Unit: The paging mechanism functions with 4K – byte memory pages or with a new extension available to the Pentium with 4M byte-memory pages. In the Pentium, with the new 4M-byte paging feature memory for the page-table reduced to single page table.

Memory – management mode: The system memory-management mode (SMM) is on the same level as protected mode, real mode, and virtual mode, but it is provided to function as a manager. The SMM is not intended to be used as an application or a system level feature. It is intended for high-level system functions such as power management.

(ii).Virtual memory:

The term virtual memory refers to something which appears to be present but actually it is not. The virtual memory technique allows users to use more memory for a program than the real memory of a computer. A programmer can write a program which requires more memory space than the capacity of the main memory. Such a program is executed by virtual memory technique. The program is stored in the secondary memory. The memory management unit (MMU) transfers the currently needed part of the program from the secondary memory to the main memory for execution. This part of the program is executed by the processor. After execution this part of the program is sent back to the secondary memory together with the immediate results. Thereafter, the CPU takes another part of the program for execution. Thus the main memory always keeps only the currently needed part of the program. This type of 'to and fro' movement instructions and data between the main memory and secondary memory is called swapping. Thus a program requiring more memory space than the capacity of the main memory can executed using a swapping technique. This concept is known a virtual memory technique.

(iii). MMX Technology:

MMX (Multimedia extensions) technology adds 57 new instructions to the instruction set of the Pentium – 4 microprocessors. The MMX technology also introduces new general purpose instructions. The new MMX instructions are designed for application such as motion video, combined graphics with video, image processing, audio synthesis, speech synthesis and compression, telephony, video conferencing, 2D graphics, and 3D graphics. These new instructions operate in parallel with other operations as the instruction for the arithmetic coprocessor.

The MMX architecture introduces new packed data types. The data types are eight packed, consecutive 8-bit bytes; four packed, consecutive 16-bit words; and two packed, consecutive 32-bit double words.

(iv) Graphics adapters:

Video card converts digital output from the computer into an analog video signal and sends the signal through a cable to the monitor also called a graphics card.

- The number of colours a video card displays is determined by its bit depth
- The video card's bit depth, also called the color depth, is the number of bits it uses to store information about each pixel
- i.e. 8-bit video card uses 8 bits to store information about each pixel; this video card can display 256 colors (2x2x2x2x2x2x2x2)
- i.e. 24-bit video card uses 24 bits to store information about each pixel and can display 16.7 million colors
- The greater the number of bits, the better the resulting image

Video Electronics Standards Association (VESA)**,** which consists of video card and monitor manufacturers, develops video stands to define the resolution, number of colors, and other display properties.

a. Monochrome Display Adapter (MDA)
b. Hercules Graphics Card
c. Colour Graphics Adapter (CGA)
d. Enhanced Graphics Adapter (EGA)
e. Video Graphics Adapter (VGA)
f. Super VGA (SVGA) and Other Standards Beyond VGA

Q.49 Write explanatory notes on **(ANY FOUR)**
(i) Paging
(ii) 8284 Clock generator
(iii) Assembler Directives
(iv) Hard disk drive controller **(16)**

Ans

(i) Paging**:**

The memory paging mechanism located within the 80386 and above allows any physical memory location to be assigned to any linear address. The linear address is defined as the address generated by a program. With the memory paging unit, the linear address is invisibly translated into any physical address, which allows an application written to function at a specific address to be located through the paging mechanism. It also allows memory to be placed into areas where no memory exists.

(ii) **8284 Clock generator:**

The 8284 is an ancillary component to the microprocessors. Without clock generator, many additional circuits are required to generate the clock in an microprocessor based system. A 8284 provides the following basic functions or signals: Clock generation, RESET synchronization, READY synchronization, and a TTL-level peripheral clock signal.

(iii) **Assembler Directives:**

An assembler directive is a statement to give direction to the assembler to perform the task of assembly process. The assembler directives control organization of the program and provide necessary information to the assembler to understand assembly language programs to generate machine codes. They indicate how an operand or section of a program is to be processed by the assembler. An assembler supports directives to define data, to organize segments to control procedures, to define macros etc.

(iv) **Hard disk drive controller:** This converts instructions from software running on the computer to the electrical signals required to operate the hard disk. The function of a disk controller is disk drive selection, track and sector selection, head loading, to parallel and parallel to serial conversion of data, error checking etc. The data recorded on a magnetic disk is the combination of clock and data. Therefore, data read must be separated from the clock information. The data processed by a CPU or stored in the main memory is in the byte form. The bytes to be recorded on a magnetic disk must be converted into serial format.

Q.50 What do you mean by Macro? Discuss merits and demerits of Macro over procedures . **(6)**

Ans MACRO: A sequence of instructions to which a name is assigned is called macro. Macros and subroutines are similar. Macros are used for short sequence of instructions whereas subroutines for longer ones. Macros executes faster than subroutines.

The MACRO directive informs assembler the beginning of a macro This is used with ENDM directive to enclose a macro. The general format of the MACRO directive is :

Macro Name MACRO ARG1, ARG2 , …..,ARG N.

The difference is that a procedure is accesses via a CALL instruction, while a macro and all the instructions defined in the macro, are inserted in the program at the point of usage. Creating macro is very similar to creating a new op-code that can be used in the program.

Q.51 Draw and discuss power failure detection circuit interrupt NMI. **(6)**

Ans

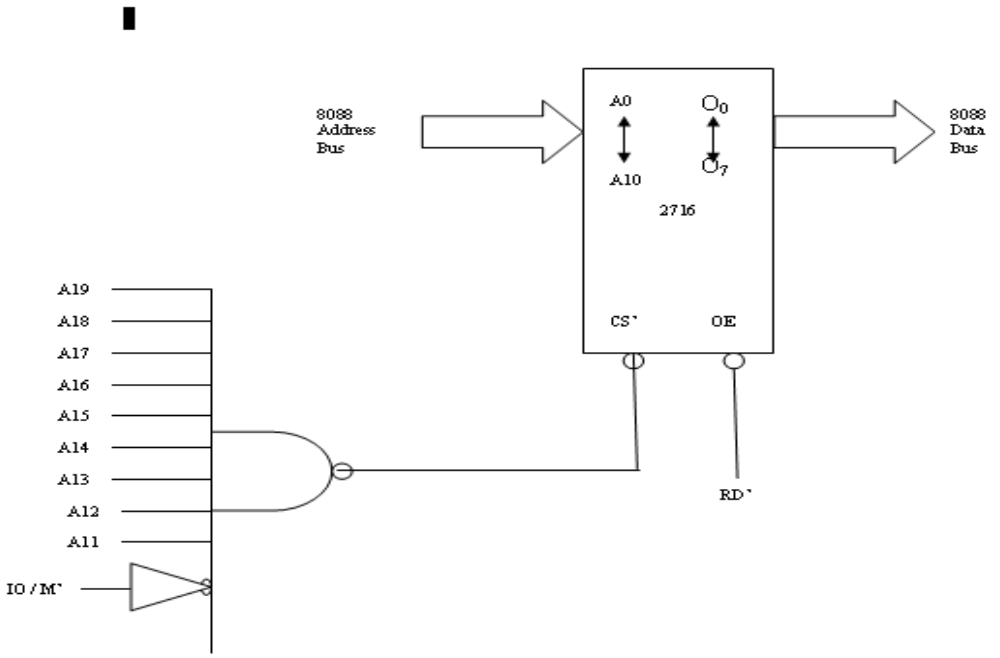
The non-maskable interrupt (NMI) is an edge-triggered input that requests an interrupt on the positive-edge. After a positive edge, the NMI pin must remain logic 1 until it is recognized by the microprocessor.

The NMI input is often used for parity errors and other major system faults, such as power failures. Power failure is easily detected by monitoring the AC power line and causing an NMI interrupt whenever AC power drops out. In response to this type of interrupt, the microprocessor stores all of the internal register in a battery backed-up-memory or an EEPROM. The below fig shows a power failure detection circuit that provides a logic 1 to the NMI input when ever AC power is interrupted.

Q.52 Interfaced 2k X 8 (i.e 2716) EPROM using multiple input NAND gate decoder for memory locations FF800H-FFFFFH. **(4)**

Ans

Simple NAND gate Decoder: When the 2k x 8 EPROM is used, address connection A10 – A0 of the 8088 are connected to address inputs A10-A0 of the EPROM. The remaining nine address pins (A19-A11) are connected to the inputs of a NAND gate decoder. The decoder selects the EPROM from one of the many 2Kbyte sections of the entire 1Mbyte address range of the 8088 microprocessor.



A simple NAND gate decoder used to select a 2716 EPROM

In this circuit, a single NAND gate decodes the memory address. The output of the NAND gate is logic 0 whenever the 8088 address pins attached to its inputs (A19-A11) are all logic 1s. The active low , logic 0 output of the NAND gate decoder is connected to the CE' input pin that selects (enables) the EPROM.

Q.53 Explain the functions of the following:

(i) Debugger (ii) Assembler
(iii) Linker **(6)**

Ans

(i) Debugger: It is a program which allows user to test and debug programs. All computers including microprocessor kits provide debugging facility. To detect errors a program can be tested in single steps. Each step of the program is executed and tested. The debugger allows the user to examine the contents of registers and memory locations after each step of execution. This also provides facility to insert breakpoint in the programs.

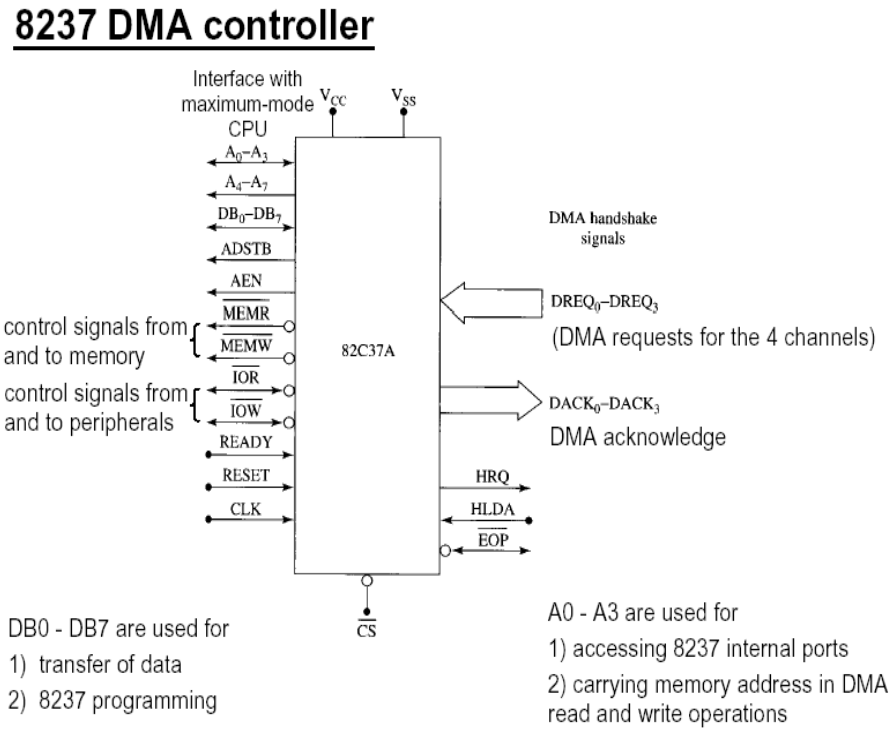
(ii) Assembler: An assembler or macro-assembler generally forms a part of the operating system. Which translates a assembly language program into machine language program.

(iii) Linker: A large program is divided in smaller programs known as modules. A linker is a program which links smaller programs together to form a large program. While developing a program subroutines, which are stored in library file, are frequently used in the program. The linker also links these subroutines with the main program.

Q.54 Discuss DMA definition and operation in brief **(4)**

Ans

Direct memory access (DMA) is a process in which an external device takes over the control of system bus from the CPU.DMA is for **high-speed data transfer** from/to mass storage peripherals, e.g. harddisk drive, magnetic tape, CD-ROM, and sometimes video controllers. For example, a hard disk may boasts a transfer rate of 5 M bytes per second, i.e.1 byte transmission every 200 ns. To make such data transfer via the CPU is
both undesirable and unnecessary.
The basic idea of **DMA** is to transfer blocks of data directly between memory and peripherals. The data don't go through the microprocessor but the data bus is occupied. "Normal" transfer of one data byte takes up to 29 clock cycles. The DMA transfer requires only 5 clock cycles.
Nowadays, DMA can transfer data as fast as 60 M byte per second. The transfer rate is limited by the speed of memory and peripheral devices.



Q.55 Write an assembly language program to find average of 'n' integers. **(6)**

Ans

```
MOV  AX, 0000  ; Initial sum 0000
MOV  BX, 0000
MOV SI , 0201H
MOV CX, [SI]
BACK: INC SI
INC SI
 ADD AX, [SI]
 JAE  GO
```

```
        INC BX
GO :    LOOP BACK
        MOV [0401], AX
        MOV [0403], BX
        INT 3
```

Q.56 Explain following instructions in 8086 family with example and their effect on flag.
(i) CWD (ii) IDIV (iii) AAS (iv) SAR
(v) LOOP (vi) SAHF (vii) BOUND (viii) IMUL **(12)**

Ans

(i) CWD (Convert signed word to signed double word): CWD instruction extends the sign bit of a word in AX register to all the bits of the DX register. It is used before a signed word in AX is to be divided by another signed word using IDIV instruction. No flags are affected.

(ii) IDIV : This instruction is used to divide a 16-bit signed number by an 8-bit signed number or 32 bit signed number by a 16-bit signed number. The 32 bit dividend is placed in DX and AX registers. The 16 bit divisor is placed in a specified 16-bit register or memory locations. No flags are affected.

(iii) AAS: (ASCII adjust after subtraction) It is used to adjust the AX register after a subtraction operation.

(iv) SAR: (Shift each bit of operand right by specified number of bits), this instruction shifts each bit of the operand which is contained in an 8-bit or 16-bit register or memory locations, right by the specified number of bits. The LSB of the operand is shifted into carry flag. The MSB which is a sign bit for the sign operand is retained in MSB position.
Flags affected are: OF, SF, ZF, PF and CF.

(v) LOOP: (Jump to specified label until CX = 0) this is used to repeat a sequence of instructions for the specified number of times. The number of times the specified sequence is to be repeated is stored in CX register. No flags are affected.

(vi) SAHF: (Store AH register into flag register) It is an instruction used to store the data in the AH register into the lower eight bits of the flag register.

(vii) BOUND: The BOUND instruction, which has two operands, compares a register with two words of memory data.

(viii) IMUL: This is an instruction for multiplication of two signed numbers. The result is a signed numbers. The OF (Over flow) and CF (Carry flag) are get affected.

Q.57 Explain keyboard interfacing to 8088 through 8279. **(8)**

Ans

The 8279 is a programmable keyboard and display interfacing component that scans and encodes up to a 64-key keyboard and controls up to a 16-digit numerical display. The keyboard interface has built in first-in first-out (FIFO) buffer that allows it store up to eight keystrokes before the microprocessor must retrieve a character. The display section controls up to 16 numeric displays from an internal 16 X 8 RAM that stores the coded display information.

The keyboard section consists of eight lines that can be connected to eight columns of a keyboard, plus two additional lines as well as to shift and CNTL/STB keys. The key pressed are automatically debounced and the keyboard can operate in two modes two –key lock out or n-key rollover. If two keys in the two –key lock out mode are pressed simultaneously, only first key is recognized. In the N-key roll over mode, simultaneous key are recognized and their codes are stored in the internal buffer.

Control Word: 000DDMMM - Mode set is command with an op-code of 000 and two fields programmed to select the mode of operation for the 8279. The DD field selects the mode of operation for the display and the MMM field selects the mode of operation for the keyboard.

D7	D6	D5	Function	Purpose
0	0	0	Mode Set	Selects the number of display positions, left or right entry, and type of keyboard scan.
0	0	1	Clock	Programs the internal clock and sets the scan and de-bounce times
0	1	0	Read FIFO	Selects the type of FIFO read and the address of the read
0	1	1	Read display	Selects the type of display read and address of the read
1	0	0	Write display	Selects the type of write and address of the write
1	0	1	Display write inhibit	Allows half-bytes to be blanked
1	1	0	Clear	Clears the display of FIFO
1	1	1	End interrupt	Clears the IRQ signal to the microprocessor

The 8279 control word summary

Q.58 Discuss the operation of a real mode interrupt and protected mode interrupt. **(6)**

Ans Operation of Real mode interrupt: When the microprocessor completes executing the current instruction, it determines whether an interrupt is active by checking (1) instruction execution, (2) single –step, (3) NMI, (4) co-processor segment overrun, (5) INTR, and (6) INT instruction in the order presented. If one or more of these interrupt conditions are present, the following sequence of events occurs:

1. The contents of the flag register are pushed onto the stack
2. Both the interrupt (IF) and trap (TF) flags are cleared. This disables the INTR pin and the trap or single-step feature.
3. The contents of the code segment register (CS) are pushed onto the stack.
4. The contents of the instruction pointer (IP) are pushed onto the stack.
5. The interrupt vector contents are fetched, and then placed into both IP and CS so that the next instruction executes at the interrupt service procedure addressed by the vector.

Protected mode interrupt:

In the protected mode, interrupts have exactly the same assignments as in real mode, but the interrupt vector table is different. In place of interrupt vectors, protected mode uses a set of 256 interrupt descriptors that are stored in an interrupt descriptor table (IDT).

Q.59 Write an assembly language program to find one's complement and two's complement of an 8-bit number **(4)**

Ans

One's complement of an 8-bit number

```
LDA 2501H
CMA
STA 2502H
HLT.
```

Two's complement of an 8-bit number

```
LDA 2501H
CMA
INR A
STA 2502H
HLT.
```

Q.60 Discuss the following terms: (Any six)

(i) Branch prediction logic in Pentium
(ii) Cache structure in Pentium
(iii) Threaded system
(iv) Super scalar architecture
(v) Real time operating system
(vi) D/A conversion **(12)**

Ans,

(i) Branch prediction logic in Pentium: The Pentium microprocessor uses branch prediction logic to reduce the time required for a branch caused by internal delays. These delays are minimized because when a branch instruction is encountered, the microprocessor begins pre-fetch instruction at the branch address. The instructions are loaded into the instruction cache, so when the branch occurs, the instructions are present and allow the branch to execute in one clocking period. If for any reason the branch prediction logic errors, the branch requires an extra three clocking periods to execute. In most cases, the branch prediction is correct and no delay ensues.

(ii) Cache structure in Pentium: The cache in the Pentium has been changed from the one found in the 80486 microprocessor. The Pentium contains two 8K-byte cache

memories instead of one as in the 80486. There is an 8K-byte data cache and an 8K-byte instruction cache. The instruction cache stores only instructions, while the data cache stores data used by instructions.

(iii) Threaded system: At times we need to implement an operating system that can process multiple threads. Multiple threads are handled by the kernel using a real-time clock interrupt. One method for scheduling processes in a small RTOS is to use a time slice to switch between various processes. The basic time slice can be any duration and is some what dependent on the execution speed of the microprocessor. Each time slice is activated by a timer interrupt. The interrupt service procedure must look to a queue to see whether a task is available to execute, and if it is, it must start execution of the new task. If no new task is present, it must continue executing an old task or enter an idle state and wait for a new task to be queued. The queue is circular and may contain any number of tasks for the system up to some finite limit.

(iv) Super scalar architecture: The Pentium microprocessor is organized with three execution units. One executes floating-point instructions, and the other two (U-pipe and V-pipe) execute integer instructions. This means that it is possible to execute three instructions simultaneously.

(v) Real time operating system (RTOS): The RTOS is an operating system used in embedded applications that performs tasks in a predictable amount of time. RTOS much like any other operating system in that it contains the same basic sections.
There are three components to all operating systems: (1) initialization, (2) the kernel, (3) data and procedures. The initialization section is used to program all hardware components in the system, load drivers specific to a system, and program the contents of the microprocessor's registers. The kernel performs the basic system task, provides system calls or functions, and comprises the embedded system. The data and procedure section holds all procedures and any static data used by the operating system.

(vi) D/A conversion: Digital-to-analog and analog-to-digital conversions are two very important aspects of digital data processing. Digital-to-analog involves conversion of digital data into equivalent analog data. For example, the output of a digital system might be converted to analog form using a D/A converter for driving a servomotor, which drives the cursor arm of a plotter or a pen recorder. It clearly shows in this example DAC emulating decoding device action.

Q.61 Explain explanatory notes on (Any four)

(i) Comparison of RS232C and RS422A standards

(ii) 8259 programmable interrupt controller

(iii) A/D conversion **(16)**

Ans

(i) Comparison of RS232C and RS422A standards:

RS232C	RS422A
1. Standard defined for asynchronous communications, where there is specified timing between data bits and no fixed timing between the characters that the bits form. 2. This standard defined 25 signal lines and 50ft is the maximum guaranteed distance. 3. This standard defines a serial system with just a single wire for each direction. 4. Signal levels are : -25 to -3V and +3 to +25V.	1. Date Rate : 10 Mbits/s 2. Driving ability upto 4000ft and 10 receivers 3. It is Differential standard i.e – Each signal is represented by a pair of wires and voltage difference across these wires is what is sensed at the receiver. This minimizes the effect of ground noise or the voltage drop along the signal leads. 4. Signal levels are : -2 to -6V and +2 to +6V

(ii) 8259 programmable interrupt controller:

The 8259A adds 8 vectored priority encoded interrupts to the microprocessor. It can be expanded to 64 interrupt requests by using one master 8259A and 8 slave units. CS and WR must be decoded. Other connections are direct to microprocessor.

The pins D7 – D0: the bidirectional data connection, IR7 – IR0: Interrupt request, used to request an interrupt & connect to a slave in a system with multiple 8259A. WR :-Connects to a write strobe signal (lower or upper in a 16 bit system) , RD :- Connects to the IORC signal , INT :- Connects to the INTR pin on the microprocessor from the master and is connected to a IR pin on a slave and INTA :- Connects to the INTA pin on the microprocessor. In a system only the master INTA signal is connected

A0 :- Selects different command words with in the 8259A, CS :- Chip select - enables the 8259A for programming and control, SP/EN :- Slave Program (1 for master, 0 for slave)/Enable Buffer (controls the data bus transceivers in a large microprocessor based system when in buffered mode) and CAS2-CAS0 :- Used as outputs from the master to the slaves in cascaded systems.

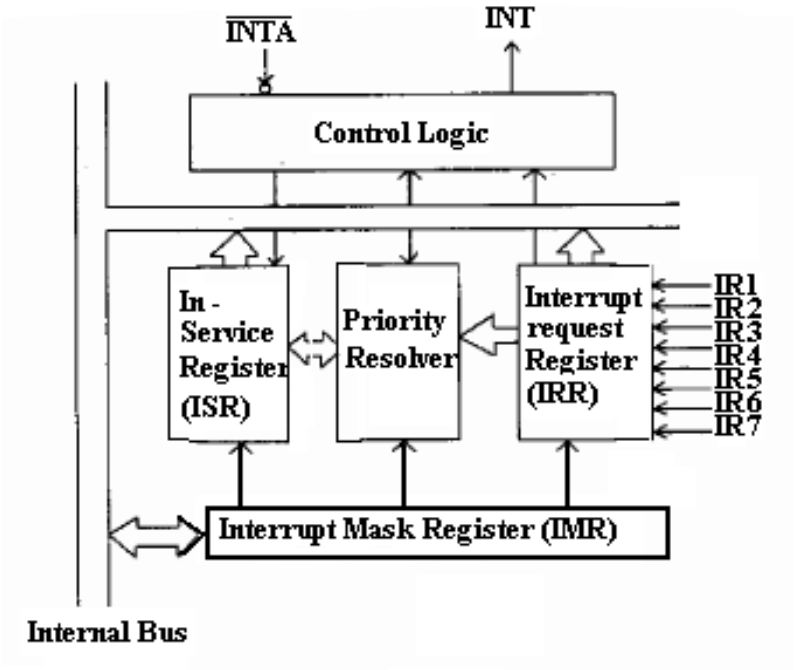


Fig : 8259 Block Diagram

(iii) A/D conversion:

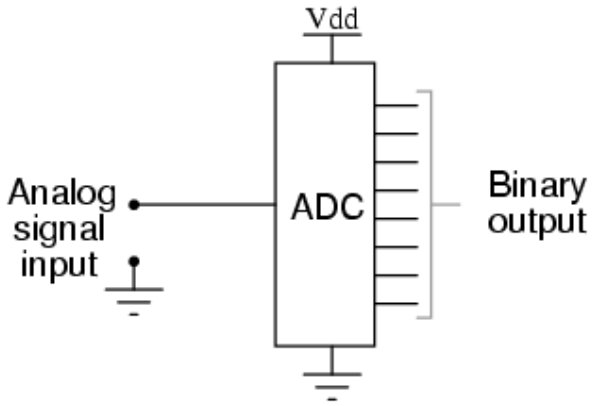


Fig : Block Diagram Representation of ADC Operation

The digital inventory are working a revolution in the field of technology, microcontrollers, microprocessors are used more effectively than those of analog circuitry. But the output of any sensors, which deals with physical equality like temperature, humidity, pressure, viscosity, velocity, which are, used most of the data acquisition flat forms are in the form of analog signals or continuous signals. Microcontrollers and microprocessors are do nothing with these signals. Because they require the signal in the form of binary numbers. So we should convert these analog signals into digital format. The following popular methods are used for Analog to Digital conversion.

1. Flash ADC
2. Digital Ramp ADC

3. Successive Approximation ADC
4. Tracking ADC
5. Slope (Integration) ADC.

Q.62 Explain with proper diagram all the six modes of operation of programmable interval timer 8254. **(8)**

Ans

Mode 0 - Interrupt on terminal count
Mode 1 - Programmable one-shot
Mode 2 - Rate Generator
Mode 3 - Square wave rate generator
Mode 4 - Software triggered strobe
Mode 5 - Hardware trigger strobe

Mode 0: The output in this mode is initially low, and will remain low for the duration of the count if GATE = 1.
Width of low pulse = N×T
Where N is the the clock count loaded into counter, and T is the clock period of the CLK input.
When the terminal count is reached, the output will go high and remain high until a new control word or new count number is loaded. In this mode, if GATE input becomes low at the middle of the count, the count will stop and the output will be low. The count resumes when the gate becomes high again. This in effect adds to the total time the output is low.

Mode 1: This mode is also called hardware triggerable one-shot.The triggering must be done through the GATE input by sending a 0-to-1 pulse to it. The following two steps must be performed:

- Load the count registers.
- A 0-to-1 pulse must be sent to the GATE input to trigger the counter.

Contrast this with mode 0, in which the counter produces the output immediately after the counter is loaded as long as GATE = 1.In mode 1 after sending the 0-to-1 pulse to GATE, OUT becomes low and stays low for a duration of N×T, then becomes high and stays high until the gate is triggered again.

Mode 2: This mode is also called divide-by-N counter. In this mode, if GATE = 1, OUT will be high for the N×T clock period, goes low for only one clock pulse, then the count is reloaded automatically, and the process continues indefinitely.

Mode 3: In this mode if GATE = 1, OUT is a square wave where the high pulse is equal to the low pulse if N is an even number. In this case the high part and low part of the pulse have the same duration and are equal to (N/2)×T (50% duty cycle). If N is an odd number, the high pulse is one clock pulse longer. This mode is widely used as a frequency divider and audio-tone generator.

Mode 4: In this mode if GATE = 1, the output will go high upon loading the count. It will stay high for the duration of N×T. After the count reaches zero (terminal

count), it becomes low for one clock pulse, then goes high again and stays high until a new command word or new count is loaded. To repeat the strobe, the count must be reloaded again. Mode 4 is similar to mode 2, except that the counter is not reloaded automatically. In this mode, the count starts the moment the count is written into the counter.

Mode 5: This mode is similar to mode 4 except that the trigger must be done with the GATE input. In this mode after the count is loaded, we must send a low-to-high pulse to the gate to start the counter.

Q.63 What is a macro? Discuss different conditional constructs/statements used while programming a macro. **(4)**

Ans MACRO: A sequence of instructions to which a name is assigned is called macro. Macros and subroutines are similar. Macros are used for short sequence of instructions whereas subroutines for longer ones. Macros executes faster than subroutines.

The MACRO directive informs assembler the beginning of a macro This is used with ENDM directive to enclose a macro. The general format of the MACRO directive is :

Macro Name MACRO ARG1, ARG2 , …..,ARG N.

Conditional assembly language statements are available for use in the assembly process and in macro sequences. The conditional statements create instructions that control the flow of the program and are variations of the IF-THEN, IF-THEN-ELSE, DO-WHILE, and REPEAT-UNTIL constructs used in high-level language programming languages.

; assembled portion with WIDT = TRUE and LENGT=TRUE;

```
        IF WIDT
WIDE    DB 72
        ELSE
        ENDIF
        IF LENGT
LONG    DB -1
        ELSE
        ENDIF
```

Q.64 A 450 ns EPROM won't work directly with a 5MHz 8088.Why? Explain. **(2)**

Ans When the 8088 is operated with a 5 MHz clock, it allows 460 ns for the memory to access data. Because of the decoder's added time delay 12ns, it is impossible for this memory to function within 460 ns.

Q.65 What is an interrupt? Discuss all the five software interrupt instructions. **(6)**

Ans An interrupt is either a hardware-generated CALL or software-generated CALL.

The INTEL family microprocessor has software interrupts INT, INT0, INT3 ,BOUND and IRET. Out of these five interrupts INT and INT3 are very similar, BOUND and INT0 are conditional, and IRET is special interrupt return instruction.

The BOUND instruction, which has two operands, compares a register with two words of memory data.
INT0 instruction checks the overflow flag (OF). If OF=1, the INT0 instruction calls the procedure whose address is stored in interrupt vector type number 4. If OF=0, then the INT0 instruction performs no operation and next sequential instruction in the program executes.
INT n instruction calls the interrupt service procedure that begins at the address represented in vector number n. For example, an INT 80H or INT 128 calls the interrupt service procedure whose address is stored in vector type 80H (000200H – 000203H). To determine the vector address, just multiply the vector number (n) by 4, which gives the beginning address of the 4-byte long interrupt vector. For example, an INT 5 = 4 x 5 = 20 (14H). The vector for INT5 begins at address 000014H and continues to 000017H. The only exception is the INT3 instruction, a 1-byte instruction.
The IRET instruction is a special return instruction used to return for both software and hardware interrupts. The IRET instruction is much like a RET, because it retrieves the return address from the stack.

Q.66 Discuss programmable keyboard and display interface -8279 control word summary. **(8)**

Ans,
The 8279 is a programmable keyboard and display interfacing component that scans up to 64-key keyboard and controls up to a 16-digit numerical display. This interface has a built-in FIFO (First-In-First-Out) buffer that allows it to store up to eight keystrokes before the microprocessor must retrieve a character. The display section controls up to 16 numeric displays from an internal 16 x 8 RAM that stores the coded display information.
Control Word: 000DDMMM - Mode set is command with an op-code of 000 and two fields programmed to select the mode of operation for the 8279. The DD field selects the mode of operation for the display and the MMM field selects the mode of operation for the keyboard.

D7	D6	D5	Function	Purpose
0	0	0	Mode Set	Selects the number of display positions, left or right entry, and type of keyboard scan.
0	0	1	Clock	Programs the internal clock and sets the scan and de-bounce times
0	1	0	Read FIFO	Selects the type of FIFO read and the address of the read
0	1	1	Read display	Selects the type of display read and address of the read
1	0	0	Write display	Selects the type of write and address of the write

1	0	1	Display write inhibit	Allows half-bytes to be blanked
1	1	0	Clear	Clears the display of FIFO
1	1	1	End interrupt	Clears the IRQ signal to the microprocessor

The 8279 control word summary

Q.67 State the importance of PUBLIC, EXTRN directives in modular programming. **(4)**

Ans The PUBLIC and EXTRN directives are very important to modular programming. PUBLIC used to declare that labels of code, data, or entire segments are available to other program modules. EXTRN (external) declares that labels are external to modules. Without these statements, modules could not be linked together to create a program by using modular programming techniques. They might link, but one module would not be able to communicate to another.

The **PUBLIC** directive is placed in the op-code field of an assembly language statement to define a label as public, so that the label can be used by other modules.

The **EXTRN** statement appears in both data and code segments to define labels as external to the segment. If data are defined as external, their sizes must be defined as BYTE, WORD or DWORD.

Q.68 What is the main difference between 16 bit and 32 bit versions of C/C++ while using in line assembler. **(4)**

Ans The 32-bit applications are written using Microsoft Visual C/C++ for windows and the 16-bit applications are written using Microsoft C/C++ for DOS. The main difference is that Visual C/C++ for windows is more common today, but does not easily call DOS functions such as INT21H.

Q.69 Explain how memory management is improved in Pentium processors? **(4)**

Ans The memory management is improved by adding paging unit and a new system memory-management mode.

Paging Unit: The paging mechanism functions with 4K – byte memory pages or with a new extension available to the Pentium with 4M byte-memory pages. In the Pentium, with the new 4M-byte paging feature memory for the page-table reduced to single page table.

Memory – management mode: The system memory-management mode (SMM) is on the same level as protected mode, real mode, and virtual mode, but it is provided to function as a manager. The SMM is not intended to be used as an application or a system level feature. It is intended for high-level system functions such as power management and security, which most Pentiums use during operation.

Q.70 Mention how do the following instructions differ in their functionality- **(4)**

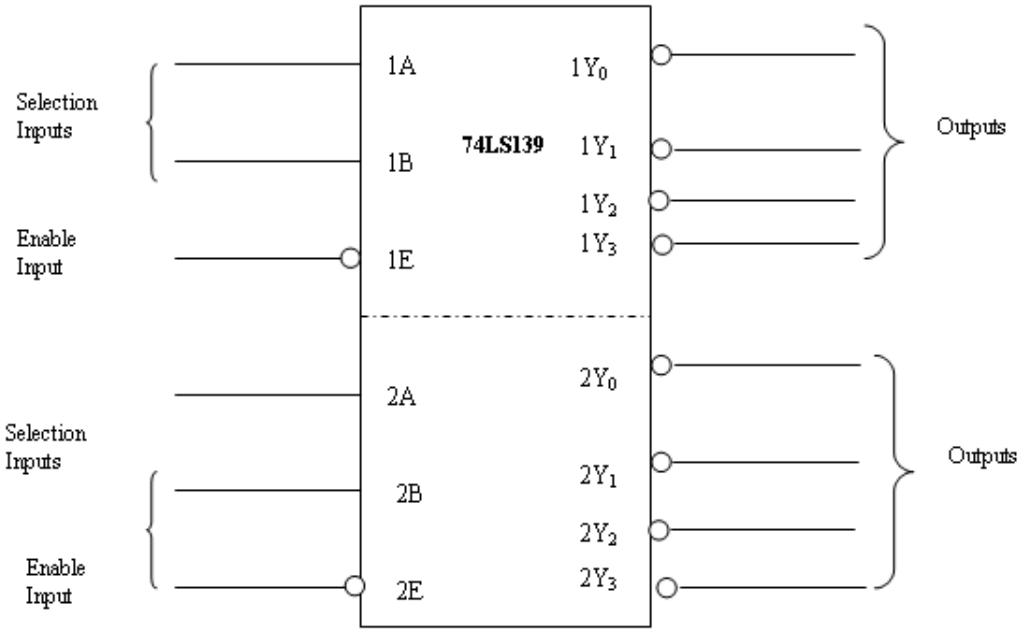
(i) NEG & NOT (ii) DIV & IDIV

(iii) AND & TEST (iv) CMP & SUB

Ans, NOT: Logical inversion or the one's complement and NEG: arithmetic sign inversion or the two's complement.
DIV: Unsigned numbers division and IDIV: Signed number division.
AND: Performs the AND operation and changes the destination operand. TEST: Test instruction performs the AND operation and it wont changes destination operand but it only affects the condition of the flag register.
SUB: Performs the subtraction operation and changes the destination operand.
CMP: Comparison instruction is a subtraction that changes only the flag bits; the destination operand never changes.

Q.71 Why memory decoding is required? Describe 74LS139 memory decoder **(4)**

Ans In order to attach a memory device to the microprocessor, it is necessary to decode the address sent from the microprocessor. Decoding makes the memory function at a unique section or partition of the memory map. Without an address decoder, only one memory device can be connected to a microprocessor, which would make it virtually useless.
The 74LS139 is a dual 2-to-4 line decoder. It contains two separate 2-to-4 line decoders – each with its own address, enable, and output connections.



The Pin-out of the 74LS139

Q.72 Explain data addressing modes (with examples) available in microprocessors.**(8)**

Ans, **Direct Mode:**

- Instruction includes memory access.
- CPU accesses that location in memory.

Example:

LDAC 5

Reads the data from memory location 5, and stores the data in the CPU's accumulator.

Indirect Mode:

- Address specified in instruction contains address where the operand resides.

Example:

LDAC @5 or LDAC (5)

Retrieves contents of location 5, uses it to access memory address.

Register Direct and Register Indirect Modes

- Does not specify a memory address. Instead specifies a register.

Example:

LDAC R

Where R is a register containing the value 5.The instruction copies the value 5 from register and into the CPU's accumulator.

Immediate Mode

- The operand specified in this mode is the actual data it self.

Example:

LDAC #5

Moves the value 5 into the accumulator.

Implicit Mode

- Does not exactly specify an operand. Instruction implicitly specifies the operand because it always applies to a specific register.

Example:

CLAC

Clears the accumulator, and sets value to zero. No operands needed.

Relative Mode

- Operand supplied is an offset, not the actual address. Added to the contents of the CPU's program counter register to generate the required address.

Example:

LDAC $5 is located at memory location 10, and it takes up two blocks of memory. Thus the value retrieved for this instruction will be 12 + 5, and will be stored in the accumulator

Index Mode and Base Address Mode

- Address supplied by the instruction is added to the contents of an index register.
- Base address mode is similar except, the index register is replaced by a base address register.

Example:

LDAC 5(X) where X = 10

Reads data from location (5 + 10) = 15 and stores it in the accumulator.

Q.73 What is the use of these assembler directives-?

(i) .MODEL (ii) PROC **(2)**

Ans

MACRO: A sequence of instructions to which a name is assigned is called macro. Macros and subroutines are similar. Macros are used for short sequence of instructions whereas subroutines for longer ones. Macros executes faster than subroutines.

The MACRO directive informs assembler the beginning of a macro This is used with ENDM directive to enclose a macro. The general format of the MACRO directive is:

Macro Name MACRO ARG1, ARG2 , …..,ARG N.

PROC: The PROC and ENDP directives indicate the start and end of a procedure. These directives force structure because the procedure is clearly defined. The PROC directive indicates the start of a procedure, must also be followed with a NEAR or FAR. A NEAR procedure is one that resides in the same code segment as the program. A FAR procedure may reside at any location in the memory system.

Q.74 (i) Convert binary number in two's compliment form 0100 1000
(ii) Convert hexadecimal BCH to decimal **(2)**

Ans.
01001000 => 10111000
BCH =>1011 1100 => 188.

Q.75 What is TPA (transient program area)? Draw the memory map of TPA in a personal computer and explain different areas. **(6)**

Ans
The memory system is divided into three main parts : TPA, System are and XMS (extended memory system).
The TPA holds the DOS operating system and other programs that control the computer system. The TPA also stores any currently active or inactive DOS application programs. The length of the TPA is 640K bytes.

9FFFF	*MSDOS Program*
9FFF0	*Free TPA*
. . .	. . .
08E30	*COMMAND.COM*
08490	*Device drivers such as MOUSE.SYS*
02530	*MSDOS Program*
01160	*IO.SYS Program*
00700	*DOS Communication area*
00500	*BIOS Communication area*
00400	*Interrupt Vectors*
00000	

The memory map of the TPA in a Personal Computer

Q.76 What is memory paging? Explain how it is used for memory addressing. **(6)**

Ans

The memory paging mechanism located within the 80386 and above allows any physical memory location to be assigned to any linear address. The linear address is defined as the address generated by a program. With the memory paging unit, the linear address is invisibly translated into any physical address, which allows an application written to function at a specific address to be located through the paging mechanism. It also allows memory to be placed into areas where no memory exists.

Q.77 Describe in detail the software interrupts available in INTEL family. How interrupts are executed in real and protected mode. **(8)**

Ans

The INTEL family microprocessor has software interrupts INT, INT0, INT3 ,BOUND and IRET. Out of these five interrupts INT and INT3 are very similar, BOUND and INT0 are conditional, and IRET is special interrupt return instruction.

The BOUND instruction, which has two operands, compares a register with two words of memory data.

INT0 instruction checks the overflow flag (OF). If OF=1, the INT0 instruction calls the procedure whose address is stored in interrupt vector type number 4. If OF=0, then the INT0 instruction performs no operation and next sequential instruction in the program executes.

INT n instruction calls the interrupt service procedure that begins at the address represented in vector number n. For example, an INT 80H or INT 128 calls the interrupt service procedure whose address is stored in vector type 80H (000200H – 000203H). To determine the vector address, just multiply the vector number (n) by 4, which gives the beginning address of the 4-byte long interrupt vector. For example, an INT 5 = 4 x 5 = 20 (14H). The vector for INT5 begins at address 000014H and continues to 000017H. The only exception is the INT3 instruction, a 1-byte instruction.

The IRET instruction is a special return instruction used to return for both software and hardware interrupts. The IRET instruction is much like a RET, because it retrieves the return address from the stack.

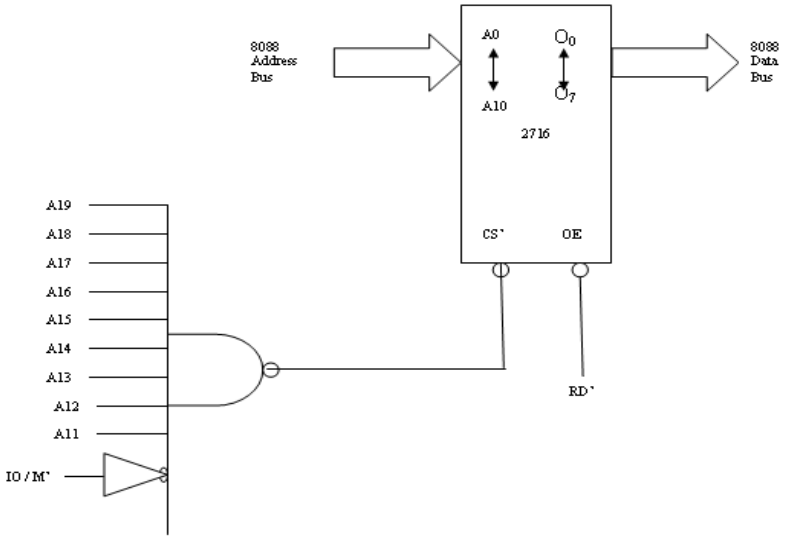
Q.78 Explain the necessity of decoding when memory device is attached to a microprocessor? With neat diagram indicate how a simple NAND gate decoder is used to select a 2716 EPROM memory component for memory locations FF800H-FFFFFH. **(5)**

Ans

In order to attach a memory device to the microprocessor, it is necessary to decode the address sent from the microprocessor. Decoding makes the memory function at a unique section or partition of the memory map. Without an address decoder, only one memory device can be connected to a microprocessor, which would make it virtually useless.

Simple NAND gate Decoder: When the 2k x 8 EPROM is used, address connection A10 – A0 of the 8088 are connected to address inputs A10-A0 of the EPROM. The remaining nine address pins (A19-A11) are connected to the inputs of a NAND gate

decoder. The decoder selects the EPROM from one of the many 2Kbyte sections of the entire 1Mbyte address range of the 8088 microprocessor.



A simple NAND gate decoder used to select a 2716 EPROM

In this circuit, a single NAND gate decodes the memory address. The output of the NAND gate is logic 0 whenever the 8088 address pins attached to its inputs (A19-A11) are all logic 1s. The active low , logic 0 output of the NAND gate decoder is connected to the CE' input pin that selects (enables) the EPROM.

Q.79 Write a Program in assembly language to find the largest of n numbers stored in the memory. **(8)**

Ans

```
MOV AX, 0000
MOV SI, 0200
MOV CX, [SI]
BACK : INC SI
INC SI
CMP AX, [SI]
JAE  GO
MOV AX, [SI]
GO: LOOP BACK
MOV [0251], AX
INT 3.
```

Q.80 Define the following **(3)**
(i) Isolated I/O (ii) memory mapped I/O
(iii) Hand shaking

Ans

There are two schemes for the allocation of addresses to memories and input / output devices.

(i). Memory Mapped I/O Scheme: In this scheme there is only one address space. Address space is defined as all possible addresses that microprocessor can generate. Some addresses are assigned to memories and some addresses to I/O devices. An I/O device is also treated as a memory location and one address is assigned to it. In this scheme all the data transfer instructions of the microprocessor can be used for both memory as well as I/O device. This scheme is suitable for a small system.

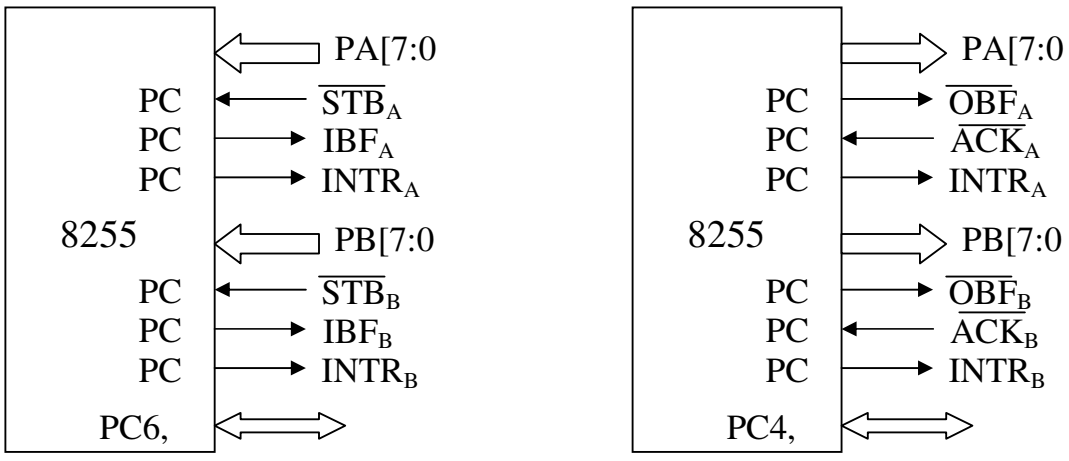
(ii). In I/O mapped I/O scheme the addresses assigned to memory locations can also be assigned to I/O devices. Since the same address may be assigned to a memory location or an I/O device, the microprocessor must issue a signal to distinguish whether the address on the address bus is for a memory location or an I/O device.

(iii). Hand shaking: In an ASYNCHRONOUS data transfer is not based on predetermined timing pattern. This technique of data transfer is used when the speed of an I/O device does not match the speed of the microprocessor, and the timing characteristic of I/O device is not predictable. In this technique the status of the I/O device i.e. whether the device is ready or not, is checked by the microprocessor before the data are transferred. The microprocessor initiates the I/O device to get ready and then continuously checks the status of the I/O device till the I/O device becomes ready to transfer data. When I/O device becomes ready, the microprocessor sends instructions to transfer data. This mode of data transfer is also called **handshaking** mode of data transfer. The microprocessor issues an initiating signal to the I/O device to get ready. When I/O device becomes ready it sends signals to the processor to indicate that it is ready. Such signals are called **handshake signals.**

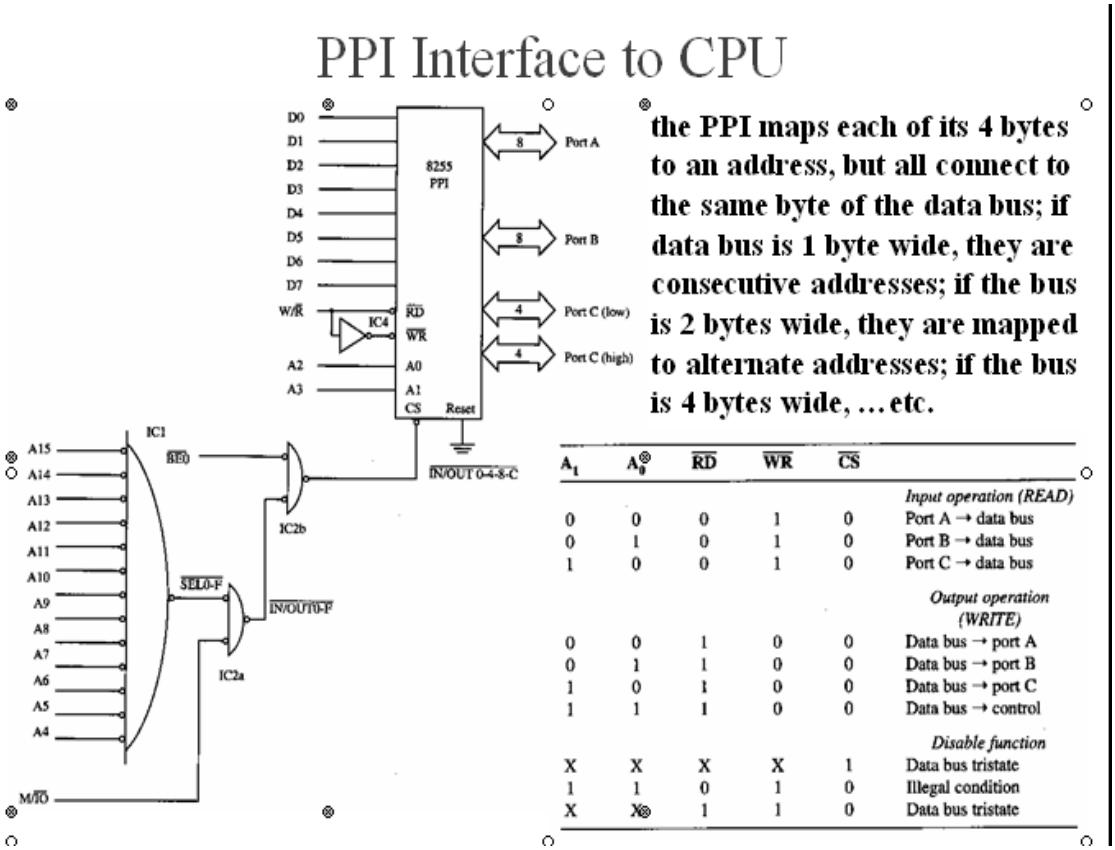
Q.81 Explain in detail the operation of 8255 in mode1 taking suitable example. **(8)**

Ans

In mode1, Ports A and B are programmed as input or output ports and Port C is used for handshaking.



Example:



A_1	A_0	RD	WR	CS	
					Input operation (READ)
0	0	0	1	0	Port A → data bus
0	1	0	1	0	Port B → data bus
1	0	0	1	0	Port C → data bus
					Output operation (WRITE)
0	0	1	0	0	Data bus → port A
0	1	1	0	0	Data bus → port B
1	0	1	0	0	Data bus → port C
1	1	1	0	0	Data bus → control
					Disable function
X	X	X	X	1	Data bus tristate
1	1	0	1	0	Illegal condition
X	X	1	1	0	Data bus tristate

Port A and/or port B function as latching input devices. External data is stored in the ports until the microprocessor is ready.

Port C used for control or handshaking signals (cannot be used for data).

Signal definitions for Mode 1 Strobed Input

$\overline{STB}$ The strobe input loads data into the port latch on a 0-to-1 transition

IFB **Input buffer full** is an output indicating that the input latch contain information

INTR **Interrupt request** is an output that requests an interrupt

INTE The **interrupt enable signal** is neither an input nor an output; it is an internal bit programmed via the PC4(port A) or PC2(port B) bits.

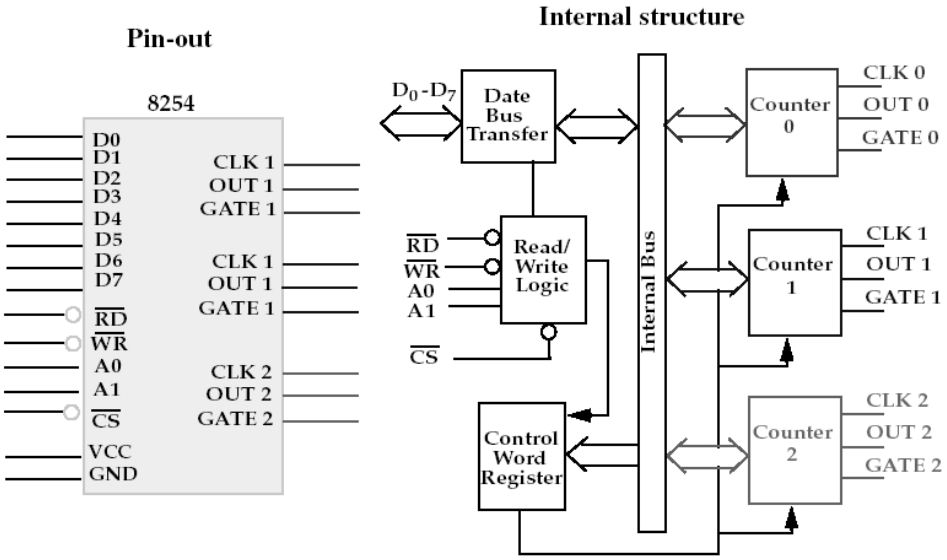
PC7,PC6 The port C pins 7 and 6 are general-purpose I/O pins that are available for any purpose.

Q.82 What is the function of 8254 Programmable Interval Timer? Discuss any one of its applications in detail. **(8)**

Ans

- 8253/54 Timer Description and Initialization
- PTI (programmable Interval Timer/Counter)
- 8253 and 8254 have exactly the same pin-out.
- 8254 is a superset of the 8253.
- It Generates accurate time delays
- It can be used for applications such as a real-time clock, an event counter, a digital one shot, a square wave generator and a complex waveform generator

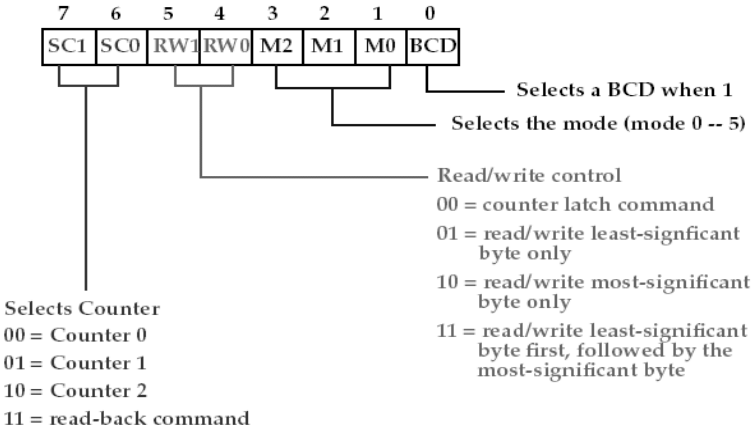
8254 Functional Description:



8254 Programming:

Each counter is individually programmed by writing a control word, followed by the initial count.

The control word allows the programmer to select the counter, model of operation, binary or BCD count and type of operation (read/write).

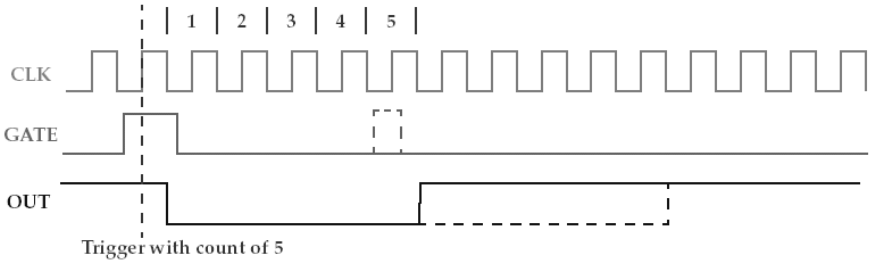


8254 Mode1 Operation:

Mode 1 causes the counter to function as a retriggerable monostable multivibrator (one-shot)

The G input triggers the counter the counter so that it develops a pulse at the OUT connection that becomes a logic 0 for the duration of the count.

if (G) occurs within the duration of the output pulse, the counter is reloaded with the count and the OUT continues for the total length of the count.



Q.83 Discuss the control words (ICWS) of IC8259. **(5)**

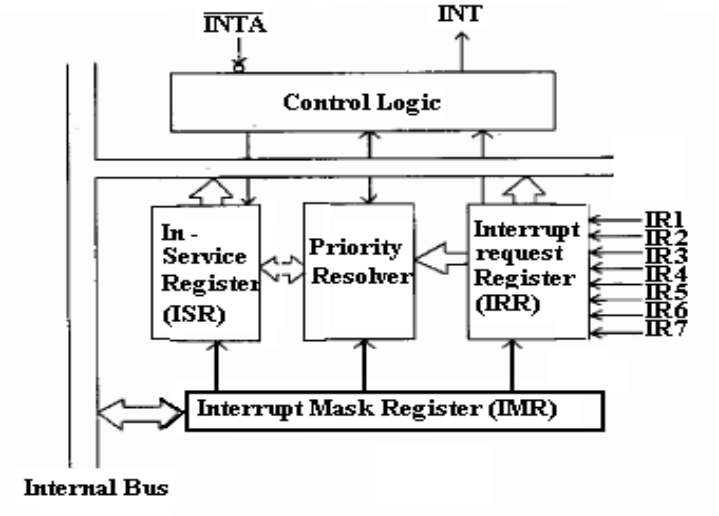
Ans

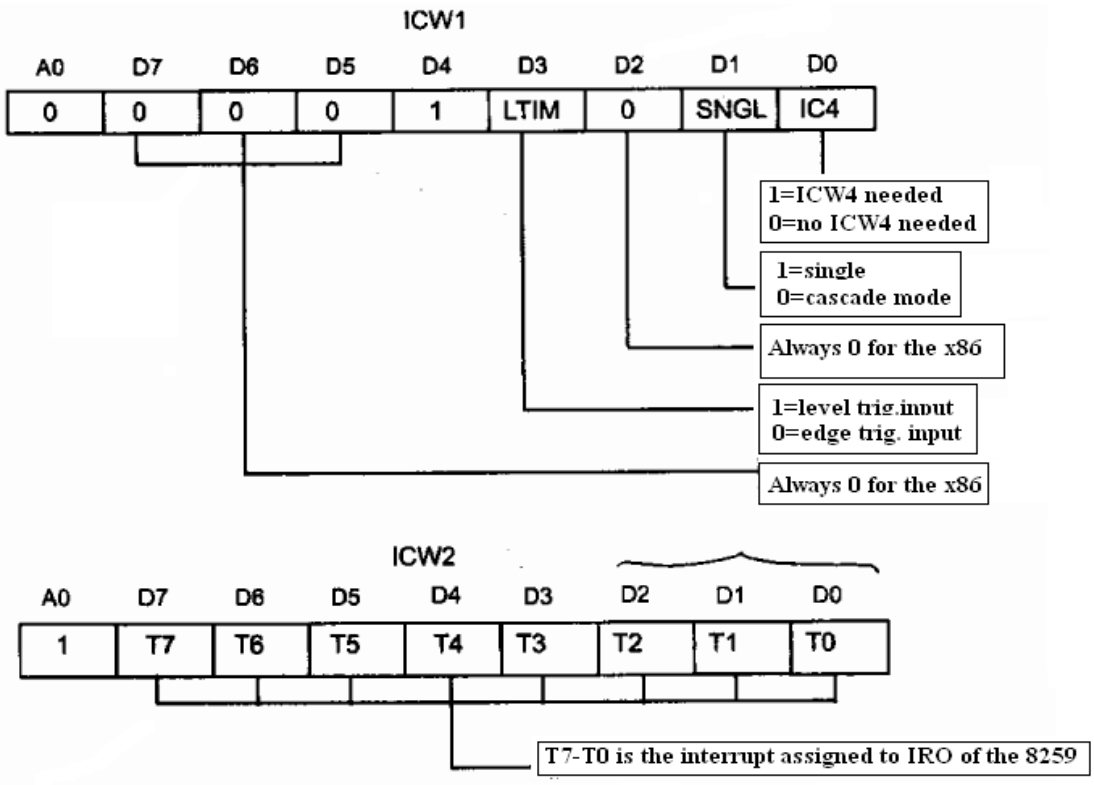
The Programmable interrupt controller is used when several I/O devices transfer data using interrupt and they are connected to the same interrupt line of the microprocessor.

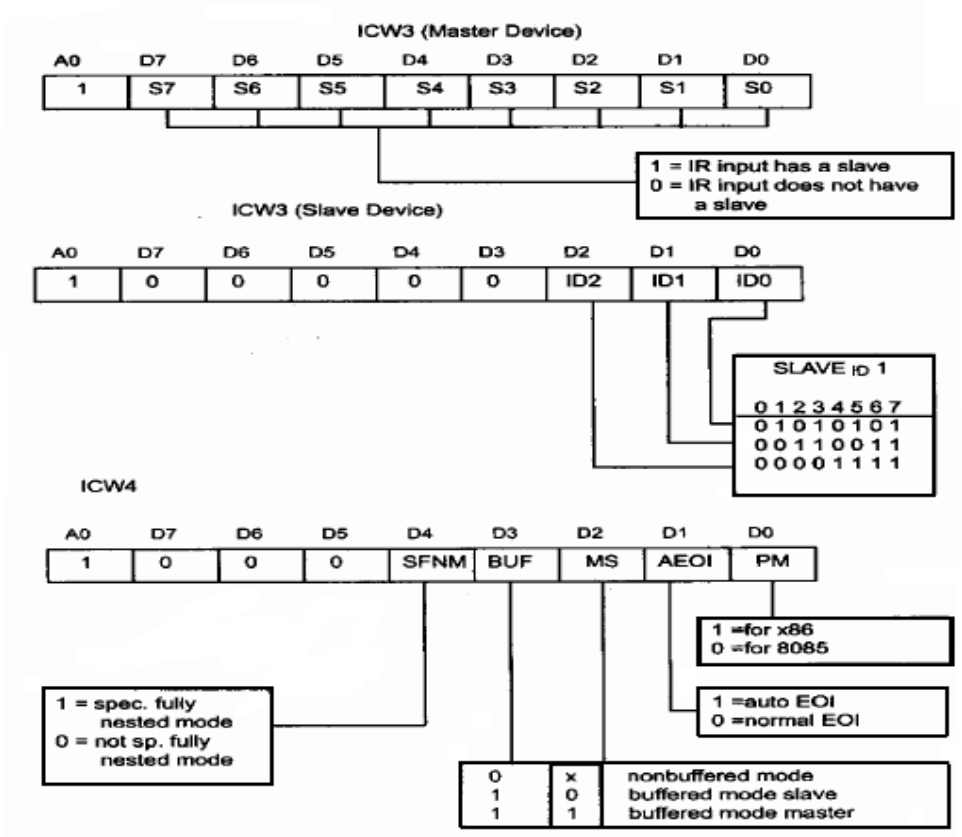
The Intel 8259 is a single chip programmable interrupt controller. It is compatible with 8086, 8088 and 8085 microprocessor. It is a 28 –pin DIP I.C package and uses N-MOS technology.

8259 Control Word Initialization

#CS	A0	Initialization
0	0	ICW1
0	1	ICW2, ICW3, ICW4
1	X	Not addressed




ICW1
A0 D7 D6 D5 D4 D3 D2 D1 D0
0 0 0 0 1 LTIM 0 SNGL IC4
1=ICW4 needed
0=no ICW4 needed
1=single
0=cascade mode
Always 0 for the x86
1=level trig.input
0=edge trig. input
Always 0 for the x86
ICW2
A0 D7 D6 D5 D4 D3 D2 D1 D0
1 T7 T6 T5 T4 T3 T2 T1 T0
T7-T0 is the interrupt assigned to IRO of the 8259


ICW3 (Master Device)
A0 D7 D6 D5 D4 D3 D2 D1 D0
1 S7 S6 S5 S4 S3 S2 S1 S0
1 = IR input has a slave
0 = IR input does not have a slave
ICW3 (Slave Device)
A0 D7 D6 D5 D4 D3 D2 D1 D0
1 0 0 0 0 0 ID2 ID1 ID0
SLAVE ID 1
0 1 2 3 4 5 6 7
0 1 0 1 0 1 0 1
0 0 1 1 0 0 1 1
0 0 0 0 1 1 1 1
ICW4
A0 D7 D6 D5 D4 D3 D2 D1 D0
1 0 0 0 SFNM BUF MS AEOI PM
1 =for x86
0 =for 8085
1 =auto EOI
0 =normal EOI
1 = spec. fully nested mode
0 = not sp. fully nested mode
0 x nonbuffered mode
1 0 buffered mode slave
1 1 buffered mode master

Q.84 Write short note on "ANY FOUR" of the following **(16)**

(i) ISA BUS
(ii). Graphic Adapter and MONITOR
(iii). DMA controller
(iv). Protected mode addressing

Ans

(i) The ISA or Industry Standard Architecture, bus has been around since the very start of the IBM-compatible personal computer system. In fact, any card from the very first personal computer will plug into and function in any of the modern Pentium 4-based computers. This is all made possible by the ISA bus interface found in all these machines, which is still compatible with the early personal computers.

ISA bus has evolved from its original 8-bit standard to the 16-bit standard found in most systems today. The ISA bus connector contains the entire de-multiplexed address bus (A19-A0) for the 1M byte 8088 system, the 8-bit data bus (D7-D0), and the four control signals MEMR', MEMW', IOR' and IOW' for controlling I/O and any memory that might be placed on the printed circuit card. ISA Card only operates at 8 MHz rate.

(ii) Video card converts digital output from the computer into an analog video signal and sends the signal through a cable to the monitor also called a graphics card.

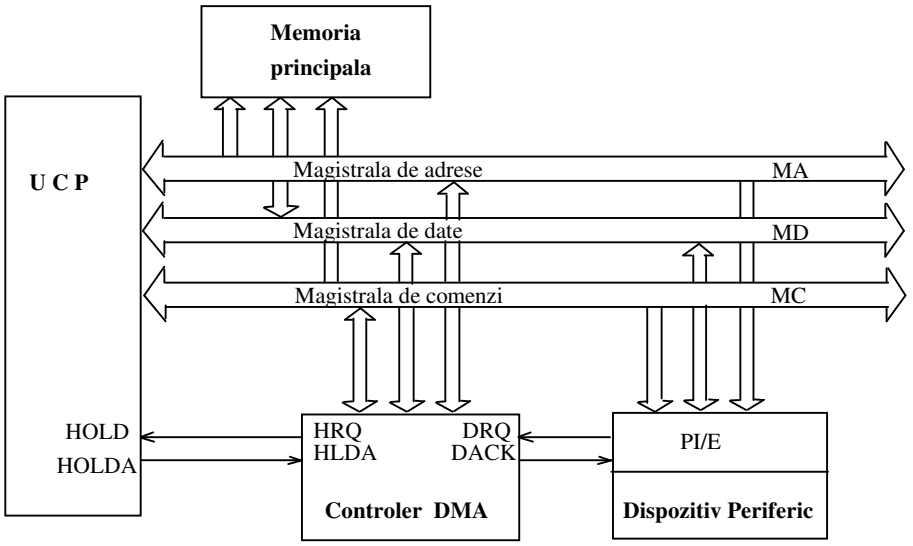
- The number of colors a video card displays is determined by its bit depth
- The video card's bit depth, also called the color depth, is the number of bits it uses to store information about each pixel
- i.e. 8-bit video card uses 8 bits to store information about each pixel; this video card can display 256 colors (2x2x2x2x2x2x2x2)
- i.e. 24-bit video card uses 24 bits to store information about each pixel and can display 16.7 million colors
- The greater the number of bits, the better the resulting image

Video Electronics Standards Association (VESA)**,** which consists of video card and monitor manufacturers, develops video stands to define the resolution, number of colors, and other display properties.

g. Monochrome Display Adapter (MDA)
h. Hercules Graphics Card
i. Color Graphics Adapter (CGA)
j. Enhanced Graphics Adapter (EGA)
k. Video Graphics Adapter (VGA)
l. Super VGA (SVGA) and Other Standards Beyond VGA

(iii) A DMA controller interfaces with several peripherals that may request DMA. The controller decides the priority of simultaneous DMA requests communicates with the peripheral and the CPU, and provides memory addresses for data transfer. DMA controller commonly used with 8088 is the 8237 programmable device.
The 8237 is in fact a special-purpose microprocessor. Normally it appears as part of the system controller chip-sets. The 8237 is a 4-channel device. Each

channel is dedicated to a specific peripheral device and capable of addressing 64 K bytes section of memory.



(iv). This addressing allows access to data and programs located above the first 1M byte of memory, as well as within the first 1M byte of memory. Addressing this extended section of the memory system requires a change to the segment plus an offset addressing scheme used with real mode memory addressing. When data and programs are addressed in extended memory, the offset address is still used to access information located within the memory segment.

The segment register contains a selector that selects a descriptor from a descriptor table. The descriptor describes the memory segment's location, length, and access rights.

Q.85 Compare RS232C and RS422A standards.

Ans

RS232C	RS422A
1. Standard defined for asynchronous communications, where there is specified timing between data bits and no fixed timing between the characters that the bits form. 2. This standard defined 25 signal lines and 50ft is the maximum guaranteed distance. 3. This standard defines a serial system with just a single wire for each direction. 4. Signal levels are : -25 to -3V and +3 to +25V.	1. Date Rate : 10 Mbits/s 2. Driving ability upto 4000ft and 10 receivers 3. It is Differential standard i.e – Each signal is represented by a pair of wires and voltage difference across these wires is what is sensed at the receiver. This minimizes the effect of ground noise or the voltage drop along the signal leads. 4. Signal levels are : -2 to -6V and +2 to +6V

Q.86 Discuss the feature of Pentium in brief.

Ans

The Pentium is a 32-bit superscalar, CISC microprocessor. The term superscalar is used for the processor which contains more than one pipeline to execute more than one instruction simultaneously in parallel.

The main features of Pentium are, it has two ALU's, one floating-point unit, two 8 KB cache, pre-fetch buffers, a branch target buffer. Two ALU's means that there are two pipelines. Each ALU contains five functional units. The two pipelines are integer pipelines. They are named U and V pipeline.

When Pentium was introduced, its operating frequency was 60 MHz. gradually; the operating frequency was raised to 233 MHz. The Pentium uses 0.6 micron Bi-CMOS process technology. It uses power management feature.

Q.87 Discuss the following assembler directives with example

i. DWORD
ii. OFFSET
iii. SEGMENT
iv. MACRO
v. ASSUME
vi. ENDP

Ans

(i). **DWORD:** It defines word type variable. The defined variable may have one or more initial values in the directive statement. If there is one value, two bytes of memory space are reserved. The general format is

Name of variable DW Initial value or values.

(ii). **OFFSET:** It is an operator to determine the offset (displacement) of a variable or procedure with respect to the base of the segment which contains the named variable or procedure. The operator can be used to load a register with the offset of a variable.

The operator can be used as follows :

```
MOV SI, OFFSET ARRAY
```

(iii). **SEGMENT :** This directive defines to the assembler the start of a segment with name segment-name. The segment name should be unique and follows the rules of the assembler

The Syntax is as follows:

```
Segment Name   SEGMENT      {Operand (Optional)} ; Comment
.
.
.

Segment Name    ENDS.
```

(iv). **MACRO:** A sequence of instructions to which a name is assigned is called macro. Macros and subroutines are similar. Macros are used for short sequence of instructions whereas subroutines for longer ones. Macros executes faster than subroutines.

The MACRO directive informs assembler the beginning of a macro This is used with ENDM directive to enclose a macro. The general format of the MACRO directive is :

Macro Name MACRO ARG1, ARG2 , …..,ARG N.

(v). **ASSUME:** This directive will be used to map the segment register names with memory addresses.

The Syntax is as follows:

ASSUME SS: Stackseg, DS : Dataseg, CS:Codeseg

The ASSUME will tell the assembler to use the SS register with the address of the stack segment whose name is stackseg.

(vi). **ENDP:** (End Procedure) It informs assembler the end of a procedure. In assembly language programming, subroutines are called procedures. A procedure may be an independent program module to give certain result or the required value to the calling program. This directive is used together with PROC directive to enclose procedure. The general format of ENDP directive is:

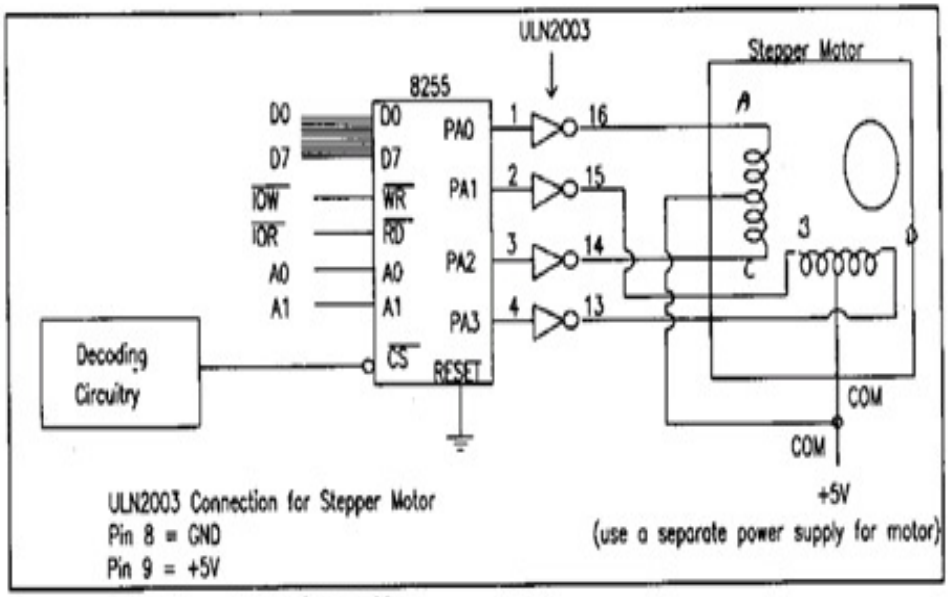
Procedure Name ENDP

Q.88 Discuss Stepper motor interfaced to the 82C55.

Ans

A stepper motor rotates in steps in response to digital pulse input. The shaft of the motor rotates in equal increments when a train of input pulses is applied. To control direction, numbers of steps to appropriate pulses are applied to the stator windings of the motor.

12 V supply is used to energize the poles. Pulses sent by the microprocessor switch on rated voltage to the windings of the desired poles. A delay subroutine is incorporated in the program. After energizing one set of pole windings some delay is provided, then the power supply is switched onto the other set of pole windings. This delay governs the speed of motor.



8255 Connection to Stepper Motor

Q.89 Discuss the EISA bus and need of PCI bus.

Ans

The Extended Industry Standard Architecture (EISA) is a 32 bit modification to the ISA bus. As computers became larger and had wider data buses, a new bus was needed that would transfer 32-bit data. The clocking speed limited up to 8MHz. The most common application for the EISA bus is a disk controller or as a video graphics adapter. These applications benefit from the wider data bus width because the data transfer rate for these devices are high.

Peripheral Component Interconnect (PCI): This bus was developed by Intel and introduced in 1993. It is geared specifically to fifth- and sixth-generation systems, although the latest generation 486 motherboards use PCI as well.

PCI bus has plug – and – play characteristics and the ability to function with a 64-bit data bus. A PCI interface contains series of registers, located in a small memory device on the PCI interface, that contains information about the board.

Q.90 Explain cascading of multiple PIC 8259.

Ans

The 8259A adds 8 vectored priority encoded interrupts to the microprocessor. It can be expanded to 64 interrupt requests by using one master 8259A and 8 slave units. CS and WR must be decoded. Other connections are direct to microprocessor.

The pins D7 – D0: the bidirectional data connection, IR7 – IR0: Interrupt request, used to request an interrupt & connect to a slave in a system with multiple 8259A. WR :-Connects to a write strobe signal (lower or upper in a 16 bit system) , RD :- Connects to the IORC signal , INT :- Connects to the INTR pin on the microprocessor from the master and is connected to a IR pin on a slave and INTA :- Connects to the INTA pin on the microprocessor. In a system only the master INTA signal is connected

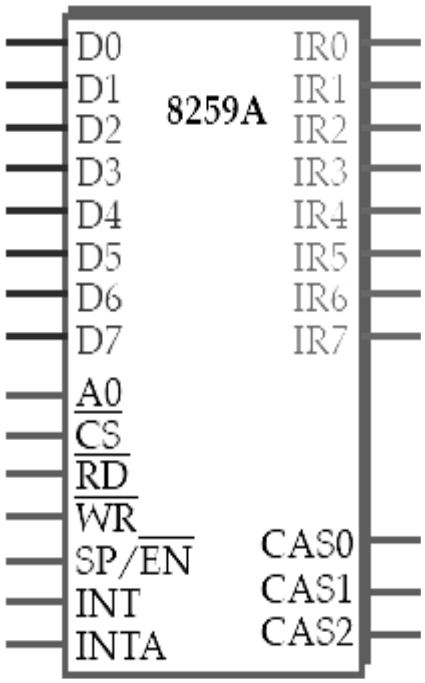


Fig: 8259 Pin Diagram

A0:- Selects different command words with in the 8259A, CS :- Chip select - enables the 8259A for programming and control, SP/EN :- Slave Program (1 for master, 0 for slave)/Enable Buffer (controls the data bus transceivers in a large microprocessor based system when in buffered mode) and CAS2-CAS0 :- Used as outputs from the master to the slaves in cascaded systems.

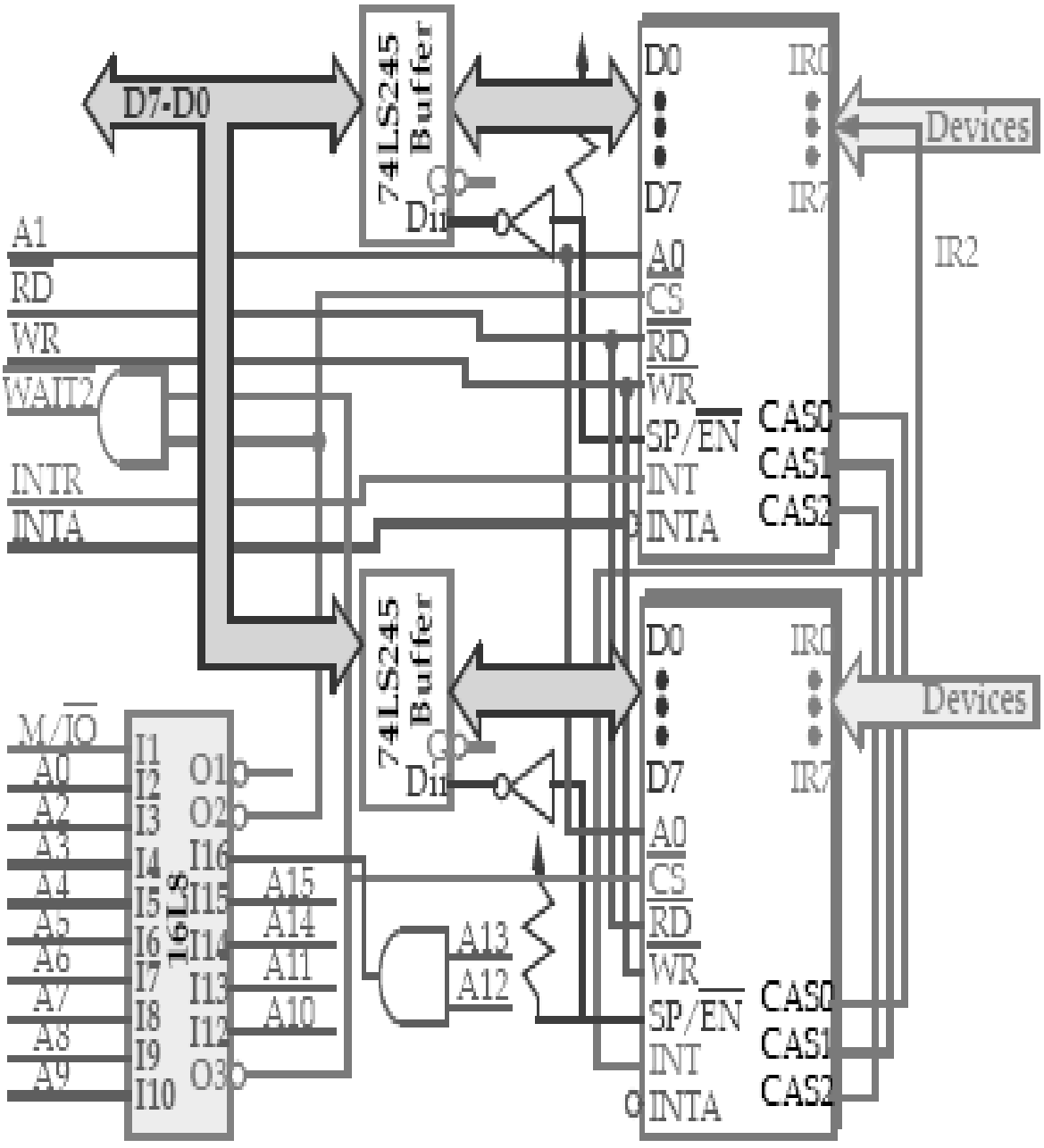


Fig: cascading multiple 8259

Q.91 Discuss need of Pipelining and Caches.

Ans

The simplest technique for improving performance through hardware parallelism is pipelining. Pipelining consists of breaking up the operations to be performed into simpler independent operations, sort of like the breaking up the operations of assembling a car in an assembly line. A typical processor pipeline consists of 4 pipeline stages (1) Instruction fetch, (2) Instruction decode (3) Instruction execute and (4) Register file write-back/memory access. In practice however, real architectures have many more physical pipeline stages, with multiple physical stages corresponding to one of the above stages. For example the execute stage might occupy 4 physical pipeline stages

The primary advantages of pipelining are
- Parallelism
- Smaller cycles time

Caches are the other big thing done in the last 2 decades to improve performance. Keep things locally if they are going to be used soon. From a physics point of view, an access to memory which is almost exclusively off chip will mean signals have to travel that much further in one cycle. In practice, since we do not want to make the system as slow as its slowest component (memory), and the cycle time is not determined by the memory access time, and rather memory takes several cycles to complete.

Q.92 Explain in brief steps to develop a Microprocessor based computer system

Ans

The design of a microcomputer system must begin with the **CPU module.** This module establishes the basic system timing, provide an orderly means of starting up the processor, and provide access to the system buses.

The second step is **adding Memory,** it is essential to the stored program computer. From this unit that the CPU fetches instructions directing it in some task. But within a particular computer system there may be several types of memories each with its own hierarchy.

The third step is adding **input / output**: This is also known has user interface. There are basically two hardware techniques for getting data into and out of a computer. The first is the parallel interface and is the most natural for microprocessor. The second technique is the serial interface.

Code: AC11 **Subject: OBJECT ORIENTED PROGRAMMING**
PART - I, VOL – I

TYPICAL QUESTIONS & ANSWERS

OBJECTIVE TYPE QUESTIONS

Each Question carries 2 marks.

Choose correct or the best alternative in the following:

Q.1 The address of a variable temp of type float is

(A) *temp **(B)** &temp
(C) float& temp **(D)** float temp&

Ans: B

Q.2 What is the output of the following code

```
char symbol[3]={'a','b','c'};
for (int index=0; index<3; index++)
cout << symbol [index];
```

(A) a b c **(B)** "abc"
(C) abc **(D)** 'abc'

Ans: C

Q.3 The process of building new classes from existing one is called _____.

(A) Polymorphism **(B)** Structure
(C) Inheritance **(D)** Cascading

Ans: C

Q.4 If a class C is derived from class B, which is derived from class A, all through public inheritance, then a class C member function can access

(A) protected and public data only in C and B.
(B) protected and public data only in C.
(C) private data in A and B.
(D) protected data in A and B.

Ans: D

Q.5 If the variable count exceeds 100, a single statement that prints "Too many" is

(A) if (count<100) cout << "Too many";
(B) if (count>100) cout >> "Too many";
(C) if (count>100) cout << "Too many";
(D) None of these.

Ans: C

Q.6 Usually a pure virtual function
- **(A)** has complete function body.
- **(B)** will never be called.
- **(C)** will be called only to delete an object.
- **(D)** is defined only in derived class.

Ans: D

Q.7 To perform stream I/O with disk files in C++, you should
- **(A)** open and close files as in procedural languages.
- **(B)** use classes derived from ios.
- **(C)** use C language library functions to read and write data.
- **(D)** include the IOSTREAM.H header file.

Ans: B

Q.8 Overloading the function operator
- **(A)** requires a class with an overloaded operator.
- **(B)** requires a class with an overloaded [ ] operator.
- **(C)** allows you to create objects that act syntactically like functions.
- **(D)** usually make use of a constructor that takes arguments.

Ans: A

Q.9 In C++, the range of signed integer type variable is _____

(A) 0 to 2^{16} **(B)** -2^{15} to $2^{15}-1$

(C) -2^{7} to $2^{7}-1$ **(D)** 0 to 2^{8}

Ans: B

Q.10 If $x = 5, y = 2$ then $x \wedge y$ equals_____.

(where $\wedge$ is a bitwise XOR operator)

(A) 00000111 **(B)** 10000010

(C) 10100000 **(D)** 11001000

Ans: A

Q.11 If an array is declared as

int a[4] = {3, 0, 1, 2}, then values assigned to a[0] & a[4] will be _____

(A) 3, 2 **(B)** 0, 2

(C) 3, 0 **(D)** 0, 4

Ans: C

Q.12 Mechanism of deriving a class from another derived class is known as____

(A) Polymorphism **(B)** Single Inheritance
(C) Multilevel Inheritance **(D)** Message Passing

Ans: C

Q.13 RunTime Polymorphism is achieved by _____

(A) friend function **(B)** virtual function
(C) operator overloading **(D)** function overloading

Ans: B

Q.14 A function call mechanism that passes arguments to a function by passing a copy of the values of the arguments is _____

(A) call by name **(B)** call by value
(C) call by reference **(D)** call by value result

Ans: B

Q.15 In C++, dynamic memory allocation is accomplished with the operator ____

(A) new **(B)** this
(C) malloc( ) **(D)** delete

Ans: A

Q.16 If we create a file by 'ifstream', then the default mode of the file is _____

(A) ios :: out **(B)** ios :: in
(C) ios :: app **(D)** ios :: binary

Ans: B

Q.17 A variable defined within a block is visible

(A) from the point of definition onward in the program.
(B) from the point of definition onward in the function.
(C) from the point of definition onward in the block.
(D) throughout the function.

Ans: C

Q.18 The break statement causes an exit

(A) from the innermost loop only. **(B)** only from the innermost switch.
(C) from all loops & switches. **(D)** from the innermost loop or switch.

Ans: D

Q.19 Which of the following cannot be legitimately passed to a function

(A) A constant. **(B)** A variable.
(C) A structure. **(D)** A header file.

Ans: D

Q.20 A property which is not true for classes is that they
(A) are removed from memory when not in use.
(B) permit data to be hidden from other classes.
(C) bring together all aspects of an entity in one place.
(D) Can closely model objects in the real world.

Ans: C

Q.21 You can read input that consists of multiple lines of text using
(A) the normal cout << combination.
(B) the cin.get( ) function with one argument.
(C) the cin.get( ) function with two arguments.
(D) the cin.get( ) function with three arguments.

Ans: C

Q.22 The keyword *friend* does not appear in
(A) the class allowing access to another class.
(B) the class desiring access to another class.
(C) the private section of a class.
(D) the public section of a class.

Ans: C

Q.23 The process of building new classes from existing one is called
(A) Structure. **(B)** Inheritance.
(C) Polymorphism. **(D)** Template.

Ans: B

Q.24 If you wanted to sort many large objects or structures, it would be most efficient to

(A) place them in an array & sort the array.
(B) place pointers to them in an array & sort the array.
(C) place them in a linked list and sort the linked list.
(D) place references to them in an array and sort the array.

Ans: C

Q.25 Which statement gets affected when i++ is changed to ++i?
(A) i = 20; i++;
(B) for (i = 0; i<20; i++) { }

(C) a = i++;
(D) while (i++ = 20) cout <<i;

Ans: A

Q.26 A friend function to a class, C cannot access
(A) private data members and member functions.
(B) public data members and member functions.
(C) protected data members and member functions.
(D) the data members of the derived class of C.

Ans: D

Q.27 The operator that cannot be overloaded is
(A) ++ **(B)** ::
(C) ( ) **(D)** ~

Ans: B

Q.28 A struct is the same as a class except that
(A) there are no member functions.
(B) all members are *public*.
(C) cannot be used in inheritance hierarchy.
(D) it does have a *this* pointer.

Ans: C

Q.29 Pure virtual functions
(A) have to be redefined in the inherited class.
(B) cannot have *public* access specification.
(C) are mandatory for a virtual class.
(D) None of the above.

Ans: A

Q.30 Additional information sent when an exception is thrown may be placed in
(A) the throw keyword.
(B) the function that caused the error.
(C) the catch block.
(D) an object of the exception class.

Ans: C

Q.31 Use of virtual functions implies
(A) overloading. **(B)** overriding.
(C) static binding. **(D)** dynamic binding.

Ans: D

Q.32 *this* pointer

(A) implicitly points to an object.
(B) can be explicitly used in a class.
(C) can be used to return an object.
(D) All of the above.

Ans: D

Q.33 Within a *switch* statement

(A) *Continue* can be used but *Break* cannot be used
(B) *Continue* cannot be used but *Break* can be used
(C) Both *Continue* and *Break* can be used
(D) Neither *Continue* nor *Break* can be used

Ans:B

Q.34 Data members which are *static*

(A) cannot be assigned a value
(B) can only be used in *static* functions
(C) cannot be defined in a *Union*
(D) can be accessed outside the class

Ans:B

Q.35 Which of the following is false for cin?

(A) It represents standard input.
(B) It is an object of istream class.
(C) It is a class of which stream is an object.
(D) Using cin the data can be read from user's terminal.

Ans:C

Q.36 It is possible to declare as a *friend*

(A) a member function **(B)** a global function
(C) a class **(D)** all of the above

Ans:D

Q.37 In multiple inheritance

(A) the base classes must have only default constructors
(B) cannot have virtual functions
(C) can include virtual classes
(D) None of the above.

Ans:C

Q.38 Declaration of a pointer reserves memory space
(A) for the object.
(B) for the pointer.
(C) both for the object and the pointer.
(D) none of these.

Ans:B

Q.39 for (; ;)
(A) means the test which is done using some expression is always true
(B) is not valid
(C) will loop forever
(D) should be written as for( )

Ans:C

Q.40 The operator << when overloaded in a class
(A) must be a member function **(B)** must be a non member function
(C) can be both **(A)** & **(B)** above **(D)** cannot be overloaded

Ans:C

Q.41 A *virtual* class is the same as
(A) an abstract class **(B)** a class with a virtual function
(C) a base class **(D)** none of the above.

Ans:D

Q.42 Identify the operator that is NOT used with pointers
(A) -> **(B)** &
(C) * **(D)** >>

Ans:D

Q.43 Consider the following statements

```
char *ptr;
ptr = "hello";
cout << *ptr;
```

What will be printed?
(A) first letter **(B)** entire string
(C) it is a syntax error **(D)** last letter

Ans:A

Q.44 In which case is it mandatory to provide a destructor in a class?
(A) Almost in every class

(B) Class for which two or more than two objects will be created
(C) Class for which copy constructor is defined
(D) Class whose objects will be created dynamically

Ans:D

Q.45 The members of a class, by default, are
(A) public **(B)** protected
(C) private **(D)** mandatory to specify

Ans:C

Q.46 Given a class named ***Book,*** which of the following is not a valid constructor?
(A) Book ( ) { } **(B)** Book (Book b) { }
(C) Book (Book &b) { } **(D)** Book (char* author, char* title) { }

Ans:B

Q47 Which of the statements is true in a protected derivation of a derived class from a base class?
(A) Private members of the base class become protected members of the derived class
(B) Protected members of the base class become public members of the derived class
(C) Public members of the base class become protected members of the derived class
(D) Protected derivation does not affect private and protected members of the derived class.

Ans:C

Q48 Which of the following statements is NOT valid about operator overloading?
(A) Only existing operators can be overloaded.
(B) Overloaded operator must have at least one operand of its class type.
(C) The overloaded operators follow the syntax rules of the original operator.
(D) none of the above.

Ans:D

Q.49 Exception handling is targeted at
(A) Run-time error **(B)** Compile time error
(C) Logical error **(D)** All of the above.

Ans:A

Q.50 A pointer to the base class can hold address of
(A) only base class object
(B) only derived class object
(C) base class object as well as derived class object
(D) None of the above

Ans:C

Q.51 Function templates can accept
- **(A)** any type of parameters
- **(B)** only one parameter
- **(C)** only parameters of the basic type
- **(D)** only parameters of the derived type

Ans:C

Q.52 How many constructors can a class have?
- **(A)** 0
- **(B)** 1
- **(C)** 2
- **(D)** any number

Ans:D

Q.53 The new operator
- **(A)** returns a pointer to the variable
- **(B)** creates a variable called new
- **(C)** obtains memory for a new variable
- **(D)** tells how much memory is available

Ans:C

Q.54 Consider the following statements:

```
int x = 22,y=15;
x = (x>y) ? (x+y) : (x-y);
```

What will be the value of x after executing these statements?
- **(A)** 22
- **(B)** 37
- **(C)** 7
- **(D)** Error. Cannot be executed

Ans:B

Q.55 An exception is caused by
- **(A)** a hardware problem
- **(B)** a problem in the operating system
- **(C)** a syntax error
- **(D)** a run-time error

Ans:D

Q.56 A template class
- **(A)** is designed to be stored in different containers
- **(B)** works with different data types
- **(C)** generates objects which must be identical
- **(D)** generates classes with different numbers of member functions.

Ans:B

Q.57 Which of the following is the valid class declaration header for the derived class **d** with base classes **b1** and **b2**?

(A) class **d** : public **b1**, public **b2** **(B)** class **d** : class **b1**, class **b2**
(C) class **d** : public **b1**, **b2** **(D)** class **d** : **b1**, **b2**

Ans:A

Q.58 A library function exit() causes an exit from

(A) the loop in which it occurs **(B)** the block in which it occurs
(C) the function in which it occurs **(D)** the program in which it occurs

Ans:D

Q.59 RunTime polymorphism is achieved by _____

(A) friend function **(B)** virtual function
(C) operator overloading **(D)** function overloading

Ans:B

Q.60 Declaration of a pointer reserves memory space

(A) for the object.
(B) for the pointer.
(C) both for the object and the pointer.
(D) none of these.

Ans:B

Q.61 An array element is accessed using

(A) a FIFO approach **(B)** an index number
(C) the operator **(D)** a member name

Ans:B

Q.62 If there is a pointer p to object of a base class and it contains the address of an object of a derived class and both classes contain a virtual member function abc(), then the statement p->abc(); will cause the version of abc() in the _____class to be executed.

(A) Base Class **(B)** Derived class
(C) Produces compile time error **(D)** produces runtime error

Ans:B

Q.63 A pure virtual function is a virtual function that

(A) has no body **(B)** returns nothing
(C) is used in base class **(D)** both **(A)** and **(C)**

Ans:D

Q.64 A static function

(A) should be called when an object is destroyed.

(B) is closely connected with and individual object of a class.
(C) can be called using the class name and function name.
(D) is used when a dummy object must be created.

Ans:C

Q.65 We can output text to an object of class *ostream* using the insertion operator<< because
(A) the *ostream* class is a stream
(B) the insertion operator works with all classes.
(C) we are actually outputting to cout.
(D) the insertion operator is overloaded in *ostream.*

Ans:D

Q.66 The statement f1.write((char*)&obj1, sizeof(obj1));
(A) writes the member function of obj1 to f1.
(B) Writes the data in obj1 to f1.
(C) Writes the member function and the data of obj1 to f1.
(D) Writes the address of obj1 to f1.

Ans:B

Q.67 To convert from a user defined class to a basic type, you would most likely use.
(A) A built-in conversion function.
(B) A one-argument constructor.
(C) A conversion function that's a member of the class.
(D) An overloaded '=' operator.

Ans:C

Q.68 Which of the following is not the characteristic of constructor.
(A) They should be declared in the public section.
(B) They do not have return type.
(C) They can not be inherited.
(D) They can be virtual.

Ans:D

Q.69 Name the header file to be included for the use of built in function isalnum()
(A) string.h **(B)** process.h
(C) ctype.h **(D)** dos.h

Ans:C

Q.70 What is the output of given code fragment?

```
int f=1, i=2;
while(++i<5)
```

```
f*=i;
cout<<f;
```

(A) 12 **(B)** 24
(C) 6 **(D)** 3

Ans:A

Q.71 A class defined within another class is:

(A) Nested class **(B)** Inheritance
(C) Containership **(D)** Encapsulation

Ans:A

Q.72 What will be the values of x, m and n after the execution of the following statements?

```
int x, m, n;
m = 10;
n = 15;
x = ++m  +  n++;
```

(A) x=25, m=10, n=15 **(B)** x=26, m=11, n=16
(C) x=27, m=11, n=16 **(D)** x=27, m=10, n=15

Ans:B

Q.73 Which of the following will produce a value 10 if x = 9.7?

(A) floor(x) **(B)** abs(x)
(C) log(x) **(D)** ceil(x)

Ans:D

Q74 The major goal of inheritance in c++ is:

(A) To facilitate the conversion of data types.
(B) To help modular programming.
(C) To extend the capabilities of a class.
(D) To hide the details of base class.

Ans:C

Q.75 Consider the following class definitions:

```
class a
{
};
class b: protected a
{
};
```

What happens when we try to compile this class?

(A) Will not compile because class body of a is not defined.
(B) Will not compile because class body of b is not defined.

(C) Will not compile because class a is not public inherited.
(D) Will compile successfully.

Ans:D

Q.76 Which of the following expressions is illegal?

(A) $(10|6)$. **(B)** (false && true)

(C) bool (x) = (bool)10; **(D)** float y = 12.67;

Ans:C

Q.77 The actual source code for implementing a template function is created when

(A) The declaration of function appears.
(B) The function is invoked.
(C) The definition of the function appears.
(D) None of the above.

Ans:B

Q.78 An exception is caused by

(A) a runtime error.
(B) a syntax error.
(C) a problem in the operating system.
(D)a hardware problem.

Ans:A

Q.79 Which of the following statements are true in c++?

(A) Classes can not have data as public members.
(B) Structures can not have functions as members.
(C) Class members are public by default.
(D) None of these.

Ans:B

Q.80 What would be the output of the following program?

```
int main()
{
int x,y=10,z=10;
x = (y = =z);
cout<<x;
return 0;
}
```

(A) 1 **(B)** 0

(C) 10 **(D)** Error

Ans:A

Q.81 What is the error in the following code?

```
class t
{
virtual void print();
}
```

(A) No error.
(B) Function print() should be declared as static.
(C) Function print() should be defined.
(D) Class t should contain data members.

Ans:A

Q.82 What will be the output of following program?

```
#include<iostream.h>
void main()
{
float x;
x=(float)9/2;
cout<<x;
}
```

(A) 4.5 **(B)** 4.0
(C) 4 **(D)** 5

Ans:A

Q.83 A white space is :
(A) blank space **(B)** new line
(C) tab **(D)** all of the above

Ans:D

Q.84 The following can be declared as friend in a class
(A) an object **(B)** a class
(C) a public data member **(D)** a private data member

Ans:B

Q.85 What would be the output of the following?

```
#include<iostream.h>
void main()
{
char *ptr="abcd"
char ch;
ch = ++*ptr++;
```

```
cout<<ch;
}
```

(A) a **(B)** b
(C) c **(D)** d

Ans:B

Q.86 A copy constructor takes
(A) no argument **(B)** one argument
(C) two arguments **(D)** arbitrary no. of arguments

Ans:B

Q87 Overloading a postfix increment operator by means of a member function takes
(A) no argument **(B)** one argument
(C) two arguments **(D)** three arguments

Ans:A

Q88 Which of the following ways are legal to access a class data member using **this** pointer?
(A) this.x **(B)** *this.x
(C) *(this.x) **(D)** (*this).x

Ans:D

Q.89 If we store the address of a derived class object into a variable whose type is a pointer to the base class, then the object, when accessed using this pointer.
(A) continues to act like a derived class object.
(B) Continues to act like a derived class object if virtual functions are called.
(C) Acts like a base class object.
(D) Acts like a base class, if virtual functions are called.

Ans:B

Q.90 Which of the following declarations are illegal?
(A) void *ptr; **(B)** char *str = “hello”;
(C) char str = “hello”; **(D)** const *int p1;

Ans:C

Q.91 What will be the result of the expression 13 & 25?
(A) 38 **(B)** 25
(C) 9 **(D)** 12

Ans:C

Q.92 Which of the following operator can be overloaded through friend function?
(A) ->
(B) =
(C) ( )
(D) *

Ans:D

Q.93 To access the public function fbase() in the base class, a statement in a derived class function fder() uses the statement.fbase();

(A) fbase();

(B) fder();
(C) base::fbase();
(D) der::fder();

Ans:A

Q.94 If a base class destructor is not virtual, then
(A) It can not have a function body.
(B) It can not be called.
(C) It can not be called when accessed from pointer.
(D) Destructor in derived class can not be called when accessed through a pointer to the base class.

Ans:D

Q.95 Maximum number of template arguments in a function template is
(A) one
(B) two
(C) three
(D) many

Ans:D

Q 96 In access control in a protected derivation, visibility modes will change as follows:

(A) private, public and protected become protected
(B) only public becomes protected.
(C) public and protected become protected.
(D) only private becomes protected.

Ans:C

Q 97 Which of the following statement is valid?
(A) We can create new C++ operators.
(B) We can change the precedence of the C++ operators.
(C) We can change the associativity of the C++ operators.
(D) We can not change operator templates.

Ans:D

Q.98 What will be the output of the following program?

```
#include<iostream.h>
void main()
{
 float x=5,y=2;
int result;
result=x % y;
cout<<result;
}
```

(A) 1 **(B)** 1.0
(C) Error message **(D)** 2.5

Ans:C

Q.99 Which can be passed as an argument to a function?
(A) constant **(B)** expression
(C) another function **(D)** all of the above.

Ans:A

Q.100 Member functions, when defined within the class specification:
(A) are always inline.
(B) are not inline.
(C) are inline by default, unless they are too big or too complicated.
(D) are not inline by default.

Ans:A

Code: AC-11 **Subject:OBJECT ORIENTED PROGRAMMING**
PART – II, VOL – I

DESCRIPTIVES

Q.1 Explain the following. **(16)**

(i) Conversion from Class to Basic Type.
(ii) File Pointers.
(iii) Function Prototyping.
(iv) Overload resolution.

Ans:(i) **Conversion from Class to Basic Type**: We know that conversion from a basic to class type can be easily done with the help of constructors. The conversion from class to basic type is indeed not that easy as it cannot be done using constructors. for this conversion we need to define an overloaded casting operator. This is usually referred to as a conversion function. The syntax for this is as follows:
The above function shall convert a class type data to typename.
A conversion function must follow the following 3 rules:
a. It cannot have a return type
b. It has to be declared inside a class.
c. It cannot have any arguments.
(ii)**File pointers**: we need to have file pointers viz. input pointer and output pointer. File pointers are required in order to navigate through the file while reading or writing. There are certain default actions of the input and the output pointer. When we open a file in read only mode, the input pointer is by default set at the beginning. When we open a file in write only mode, the existing contents are deleted and the file pointer is attached in the beginning.
C++ also provides us with the facility to control the file pointer by ourselves. For this, the following functions are supported by stream classes: seekg(), seekp(), tellg(), tellp().
(iii) **Function prototyping** is used to describe the details of the function. It tells the compiler about the number of arguments, the type of arguments, and the type of the return values. It some what provides the compiler with a template that is used when declaring and defining a function. When a function is invoked, the compiler carries out the matching process in which it matches the declaration of the function with the arguments passed and the return type. In C++, function prototype was made compulsory but ANSI C makes it optional. The syntax is as follows:
type function-name(argument-list);
example int func(int a, int b, int c);
(iv)**Overload resolution**: When we overload a function, the function prototypes are matched by the compiler. The best match is found using the following steps.
a. The compiler first matches that prototype which has the exactly same types of actual arguments.
b. If an exact match is not found by the first resolution, the integral promotions to the actual arguments are used like char to int, float to double.
c. When both the above methods fail, the built in conversions to the actual arguments are used like long square (long n).
d. If all the above steps do not help the compiler find a unique match, then the compiler uses the user defined convergence, in combination with integral promotions and built in conversions.

Q.2 Write a C++ program that prints **(6)**

```
1    1.0000e+00    1.000
2    5.0000e-01    1.500
3    3.3333e-01    1.833
```

Ans:
```
#include<iostream.h>
#include<iomanip.h>
void main()
{
 float x,y,z;
 x=1.0000e+00;
 y=5.0000e-01;
 z=3.3333e-01;
 cout.setf(ios::showpoint);
 cout.precision(3);
 cout.setf(ios::fixed,ios::floatfield);
 cout<<x<<'\n';
 cout<<y<<'\n';
 cout<<z<<'\n';
}
```

Q.3 What are friend functions? Explain there characteristics with a suitable example. **(6)**

Ans: We generally have a belief that a non member function cannot access the private data member of the class. But this is made possible using a friend function. A friend function is that which does not belong to the class still has complete access to the private data members of the class. We simply need to declare the function inside the class using the keyword "friend". The syntax for declaring the friend function is as follows:

```
class demo
{
public:
 friend void func (void);
}
```

The characteristics of a friend function are as follows.

- It can be declared both in the private and the public part of the class without altering the meaning.
- It is not called using the object of the class.
- To access the data members of the class, it needs to use the object name, the dot operator and the data member's name. example obj.xyz
- It is invoked like a normal function.
- It does not belong to the class

An example program is as follows:

```
#include<iostream.h>
using namespace std;
class one;
```

```
class two
{
 int a;
   public:
 void setvalue(int n){a=n;}
 friend void max(two,one);
};
class one
{
 int b;
   public:
 void setvalue(int n){b=n;}
 friend void max(two,one);
};

void max(two s,one t)
{
 if(s.a>=t.b)
            cout<<s.a;
 else
            cout<<t.b;
}

int main()
{
 one obj1;
 obj1.setvalue(5);
 two obj2;
 obj2.setvalue(10);
 max(obj2,obj1);
 return 0;
          }
```

Q.4 write a program to overload the operator '+' for complex numbers. **(7)**

Ans:

To perform complex number addition using operator overloading.

```
# include <iostream.h>
# include <conio.h>

class complex                    {
```

```
        float r, i;
    public:
    complex()
    {
            r=i=0;
    }
    void getdata()
    {
        cout<<"R.P";
        cin>>r;
        cout<<"I.P";
        cin>>i;
    }
    void outdata (char*msg)
    {
        cout<<endl<,msg;
        cout<<"("<<r;
        cout<<"+j" <<i<<")";
    }
    Complex operator+(Complex);
};
Complex complex::Operator+(Complex(2))
{
    complex temp;
    temp.r=r+c2.r;
    temp.i=I=c2.i;
    return(temp);
}
void main()
{
    clrscr();
    complex c1, c2, c3;
    cout<<"Enter 2 complex no: "<<endl;
    cl.getdta();
    c2.getdata();
    c3=c1+c2;
    c3.outdata ("The result is :");
    getch();
}
```

OUTPUT

Enter 2 complex no: R.P: 2 I.P: 2 R.p: 2 I.P:2

The result is: 4+j4

RESULT

Thus the complex number addition has been done using operator overloading

Q.5 Write a complete C++ program to do the following :

(i) 'Student' is a base class, having two data members: entryno and name;

entryno is integer and name of 20 characters long. The value of entryno is 1 for Science student and 2 for Arts student, otherwise it is an error.

(ii) 'Science' and 'Arts' are two derived classes, having respectively data items marks for Physics, Chemistry, Mathematics and marks for English, History, Economics.

(iii) Read appropriate data from the screen for 3 science and 2 arts students.

(iv) Display entryno, name, marks for science students first and then for arts students. **(14)**

Ans:

```
 #include<iostream.h>
class student
{
protected:
 int entryno;
 char name[20];
public:
 void getdata()
 {
            cout<<"enter name of the student"<<endl;
      cin>>name;
 }
 void display()
 {

 cout<<"Name of the student is"<<name<<endl;
 }

};
class science:public student
{

  int pcm[3];
public:
 void getdata()
 {
                        student::getdata();
                        cout<<"Enter marks for Physics,Chemistry and Mathematics"<<endl;
              for(int j=0;j<3;j++)
              {
                        cin>>pcm[j];
              }

 }
  void display()
```

```
 {
   entryno=1;
          cout<<"entry no for Science student is"<<entryno<<endl;
                student::display();
                cout<<"Marks in Physics,Chemistry and Mathematics are"<<endl;
            for(int j=0;j<3;j++)
            {
                cout<<pcm[j]<<endl;;
            }

 }
};
class arts:public student
{

 int ehe[3];
public:
 void getdata()
 {
                student::getdata();
                cout<<"Enter marks for English,History and Economics"<<endl;
            for(int j=0;j<3;j++)
            {
                cin>>ehe[j];
            }

 }
 void display()
 {
   entryno=2;
 cout<<"entry no for Arts student is"<<entryno<<endl;;
                student::display();
                cout<<"Marks in English,History and Economics are"<<endl;
            for(int j=0;j<3;j++)
  {
                cout<<ehe[j]<<endl;;
            }

 }
};
void main()
{
 science s1[3];
 arts a1[3];
 int i,j,k,l;
 cout<<"Entry for Science students"<<endl;
 for(i=0;i<3;i++)
```

```
 {
 s1[i].getdata();
 }
 cout<<"Details of three Science students are"<<endl;
 for(j=0;j<3;j++)
 {
 s1[j].display();
 }

cout<<"Entry for Arts students"<<endl;
 for(k=0;k<3;k++)
 {
 a1[k].getdata();
 }
 cout<<"Details of three Arts students are"<<endl;
 for(l=0;l<3;l++)
 {
 a1[l].display();
 }

}
```

Q.6 An electricity board charges the following rates to domestic users to discourage large consumption of energy :

For the first 100 units – 50 P per unit
Beyond 300 units – 60 P per unit

If the total cost is more than Rs.250.00 then an additional surcharge of 15% is added on the difference. Define a class Electricity in which the function Bill computes the cost. Define a derived class More_Electricity and override Bill to add the surcharge. **(8)**

Ans:

```
#include<iostream.h>
class electricity
{
protected:
 float unit;
 float cost;
public:
 void bill()
 {
        cout<<"\n enter the no. of units"<<endl;
        cin>>unit;
        if(unit<=100)
        {
                cost=0.50*unit;
```

```
                cout<<"cost up to 100 unit is Rs."<<cost<<endl;
        }
        else
        {
                if(unit>300)
                {
                        cost=0.60*unit;
                        cout<<"Beyond 300 units is Rs"<<cost;
                }
        }
 }
};
class more_electricity:public electricity
{
 float surcharge,diff,total_cost;
public:
 void bill()
 {
        electricity::bill();
        if(cost>250.00)
        {
                diff=cost-250;
                surcharge=diff*0.15;
                total_cost=cost+surcharge;
                cout<<" Bill amount with surcharge is Rs"<<total_cost;
        }
        else
        {
                cout<<"Bill amout is Rs."<<cost;
        }}};
void main()
{
 more_electricity me;
 me.bill();
}
```

Q.7 Write a program to open a file in C++ “Hello.dat” and write
“This is only a test”
“Nothing can go wrong”
“All things are fine…”
into the file. Read the file and display the contents. **(6)**

Ans:

```
#include<iostream.h>
#include<fstream.h>
void main()
```

```
{
 ofstream fout;
 fout.open("Hello.dat");
 fout<<"This is only a test\n";
 fout<<"Nothing can go wrong\n";
 fout<<"All things are fine....\n";
        fout.close();
        const int N=100;
        char line[N];
 ifstream fin;
 fin.open("Hello.dat");
 cout<<"Contents of the Hello.dat file\n";
 while(fin)
 {
        fin.getline(line,N);
        cout<<line;
 }
 fin.close();}
```

Q.8 Develop a program in C++ to create a database of the following items of the derived class. Name of the patient, sex, age, ward number, bed number, nature of illness, date of admission. Design a base class consisting of data members : name of the patient, sex and age ; and another base class consisting of the data members : bed number and nature of the illness. The derived class consists of the data member ,date of admission.
Program should carry out the following methods
(i) Add a new entry.
(ii) List the complete record. **(10)**

Ans:.

```
#include<conio.h>
#include<iostream.h>
class A{
 public:
 char name[20];
 char sex[10];
 int age;
 void get_data();
 void disp_data();
};
void A::get_data(){
 cout<<"enter d name:";
 cin>>name;
cout<<"enter d age:";
 cin>>age;
 cout<<"enter d sex:";
 cin>>sex;}
void A::disp_data(){
 cout<<"name:"<<name<<'\n';
```

```
 cout<<"age:"<<age<<'\n';
 cout<<"sex:"<<sex<<'\n';}
class B{
 public:
 int bed_number;
 char nature_illness[40];
 void get_data();
void disp_data();
};
void B::get_data(){
 cout<<"enter d bed_number:";
 cin>>bed_number;
 cout<<"enter d nature_illness:";
 cin>>nature_illness;
}
void B::disp_data(){
 cout<<"bed_number:"<<bed_number<<'\n';
 cout<<"nature_illness:"<<nature_illness<<'\n';
}
class C:public A,public B{
 int date_admission;
 public:
void get_data();
 void disp_data();
 void record();
};
void C::get_data(){
        A::get_data();
        B::get_data();
        cout<<endl<<"Enter Data of Admission:-> ";
        cin>>date_admission;
}
void C::disp_data(){
        A::disp_data();
        B::disp_data();
        cout<<endl<<"Date of Admission\t"<<date_admission;
}
void main(){
 clrscr(); C c1;
 cout<<endl<<"Adding a new record to database\n";
 getch();
 c1.get_data();
 cout<<endl<<"Displaying the added record to database\n";
 c1.disp_data();
 getch();
}
```

Q.9 How many times is the copy constructor called in the following code? **(4)**

```
Apple func(Apple u)
{
Apple w=v;
Return w;
}
void main()
{
Apple x;
Apple y = func (x);
Apple z = func (y);
}
```

Ans: 2 times

Q.10 Write a program code which throws an exception of type char* and another of type int. Write a try ---- catch block which can catch both the exception. **(5)**

Ans:

```
#include <iostream>
    using namespace std;
 int main()
{
 double Operand1, Operand2, Result;

 // Request two numbers from the user
 cout << "This program allows you to perform a division of two numbers\n";
 cout << "To proceed, enter two numbers: ";
 try {
            cout << "First Number: ";
            cin >> Operand1;
            cout << "Second Number: ";
            cin >> Operand2;

// Find out if the denominator is 0
            if( Operand2 == 0 )
                        throw "Division by zero not allowed";

// Perform a division and display the result
            Result = Operand1 / Operand2;
```

```
            cout << "\n" << Operand1 << " / " << Operand2 << " = " << Result << "\n\n";
 }
 catch(const char* Str) // Catch an exception
 {
            // Display a string message accordingly
            cout << "\nBad Operator: " << Str;
 }

 return 0;
#include <iostream.h>

int main()
{
 double Operand1, Operand2, Result;
 const char Operator = '/';

 // Request two numbers from the user
 cout << "This program allows you to perform a division of two numbers\n";
 cout << "To proceed, enter two numbers\n";

 try {
            cout << "First Number: ";
            cin >> Operand1;
            cout << "Second Number: ";
            cin >> Operand2;
               // Find out if the denominator is 0
               if( Operand2 == 0 )
                        throw 0;
            // Perform a division and display the result
            Result = Operand1 / Operand2;
            cout << "\n" << Operand1 << " / " << Operand2 << " = " << Result << "\n\n";
 }
 catch(const int n) // Catch an exception
 {
            // Display a string message accordingly
            cout << "\nBad Operator: Division by " << n << " not allowed\n\n";
 }

 return 0;
}
```

Q.11 Create a class Time that has separate int member data for hours, minutes and seconds. One constructor should initialize this data to zero and another constructor initialize it to fixed values. Write member function to display time in 12 hour as well as 24 hour format. The final member function should add two objects of class Time.
A main() program should create three objects of class time, of which two are initialized to specific values and third object initialized to zero. Then it should add the two

initialized values together, leaving the result in the third. Finally it should display the value of all three objects with appropriate headings. **(14)**

Ans:

```
#include<iostream.h>
#include<string.h>

class time24
{

 int hours,minutes,seconds;
public:
 time24()
   {
        hours=minutes=seconds=0;
 }
 time24(int h,int m,int s)
 {
        hours=h;
        minutes=m;
        seconds=s;
 }
  void display()
 {
        if(hours<10)
                cout<<'0';
        cout<<hours<<":";
        if(minutes<10)
                cout<<'0';
        cout<<minutes<<":";

        if(seconds<10)
                cout<<'0';
        cout<<seconds;
 }
};
class time12
{
bool pm;
 int hour,minute;

public:
 time12()
   {
        pm=true;
```

```
            hour=minute=0;
 }
 time12(bool ap,int h,int m)
 {
            pm=ap;
            hour=h;
            minute=m;
 }
 time12(time 24);
 void display()
 {
            cout<<hour<<":";
            if(minute<10)
                        cout<<'0';
            cout<<minute<<" ";

 char *am_pm=pm ? "p.m." : "a.m.";
 cout<<am_pm;
 }
};

time12::time12(time24 t24)
{
 int hrs24=hours;
 bool pm=hours<12 ? false:true;
            int min=seconds<30 ? minutes:minutes+1;
            if(min==60)
            {
                        min=0;
                        ++hrs24;
                        if(hrs24==12 || hrs24==24)
                                    pm=(pm==true)? false:true;
            }
            int hrs12=(hrs24<13) ? hrs24 : hrs24-12;
            if(hrs12==0)
            {
                        hrs12=12;
                        pm=false;
            }
            return time12(pm,hrs12,min);
 }

int main()
{
 int h1,m1,s1;
```

```
while(true)
{
        cout<<"enter 24-hour time:\n";
        cout<<"Hours(0-23):";
        cin>>h1;
if(h1>23)
        return(1);
cout<<"Minutes:";
cin>>m1;
cout<<"Seconds:";
cin>>s1;

time24 t24(h1,m1,s1);
cout<<"you entered:";
t24.display();
cout<<endl;
time12 t12=t24;
cout<<"\n12-hour time:";
t12.display();
cout<<endl;
}
return 0;
}
```

Q.12 In the following output, the following flag constants are used with the stream member function setf. What effect does each have

(i) ios::fixed
(ii) ios::scientific
(iii) ios::showpoint
(iv) ios::showpos
(v) ios::right
(vi) ios::left **(6)**

Ans: A two argument version of setf() is used to set flag which is a member of a group of flags. The second argument specifies the group and is a bitwise OR of all the flags in the group. The setfO, a member function of the ios class, can provide answers to these and other formatting questions. The setfO *(setf* stands for set flags) function can be us follows:

cout.setf(argl,arg2)

The *argl* is one of the formatting *flags* defined in the class ios. The formatting flag Spi the format action required for the output. Another ios constant, *arg2,* known as bi specifies the group to which the formatting flag belongs.

There are three bit and each has a group of format flags which are mutually exclusive

Examples:

cout.setf(ios::1eft, ios::adjustfied);
cout.setf(ios::scientific, ios::floatfield);
Note that the first argument should be one of the group members of the second argument.
The Setf() can be used with the flag ios::showpoint as a single argument to achieve this form of output. For example
Cout.setf(ios::showpoint);
Would cause cout to display trailing zeros and trailing decimal point.
Similarly, a plus sign can be printed a positive number using the following statement:
Cout.setf(ios::showpos);set to selection

selection	ios::ingroup	Meaning
ios::left	ios::adjustfield	left alignment
ios::right	ios::adjustfield	right alignment
ios::fixed	ios::floatfield	fixed form of floating point
ios::scientific	ios::floatfield	force exponential format
ios::showpoint //		Always show a point
ios::showpos //		If positive still show sign

Q.13 Define a class Distance with feet and inch and with a print function to print the distance. Write a non-member function max which returns the larger of two distance objects, which are arguments. Write a main program that accepts two distance objects from the user, compare them and display the larger. **(8)**

Ans:

```
#include<iostream.h>
class distance
{
 int feet;
 float inches;
public:
 distance()
 {
        feet=0;
        inches=0.0;
 }

 void getdist()
 {
        cout<<"Enter feet:"<<endl;
        cin>>feet;
        cout<<"Enter inches:"<<endl;
        cin>>inches;
 }
 void showdist()
 {
        cout<<feet<<endl<<inches<<endl;
 }
 friend void maxdist(distance d1, distance d2)
 {
 d1.feet=d1.feet+ d1.inches/12;
```

```
 d2.feet=d2.feet+ d2.inches/12;
 if(d1.feet>d2.feet)
 {
            cout<<"first distance is greater";
 }
 else
 {
 cout<<"second distance is greater";
 }
 }
};
void main()
{
 distance d3,d4;
 d3.getdist();
 d3.showdist();
 d4.getdist();
   d4.showdist();
 maxdist(d3,d4);
}
```

Q.14 a) Define the class Person which has name (char*) and age (int). Define the following constructors

a. default
b. with name as argument (assume age is 18)
c. with name and age as argument

Also define the copy constructor. **(7)**

Ans:

```
#include<iostream.h>
#include<string.h>
class person
{
char *name;
int age;
public:
person()
{
 name=NULL;
//strcpy(name,NULL);
age=0;
}
person(char *n)
{
 strcpy(name,n);
age=18;
}
person(char *n1,int a)
```

```
{
 strcpy(name,n1);
age=a;
}
person(person &p)
{
strcpy(name,p.name);
age=p.age;
}
void disp()
{
cout<<"name is"<<name;
cout<<"age is"<<age;
}
};
void main()
{
person p1("ram"),p2("sita",20),p3=p2;
cout<<"hello";
p1.disp();
p2.disp();
p3.disp();
}
```

Q.14 b). Using the class above, define two subclasses Student and Professor. Student subclass displays the name and CGPA (grade points in float) and Professor subclass displays the name and number of publications (int). Write a main program using polymorphism to display the data of one student and one professor. **(7)**

Ans:

```
 #include<iostream.h>
class person
{
protected:
 char name[40];
public:
 void getname()
 {
 cout<<"Enter name:";
 cin>>name;
 }
 void disp()
 {
 cout<<"Name is:"<<name<<endl;
 }
 virtual void getdata()=0;
 virtual bool outstanding()=0;
};
```

```
class student:public person
{
float gpa;
public:
 void getdata()
 {
            person::getname();
            cout<<"Enter student's GPA:";
            cin>>gpa;
 }
 bool outstanding()
 {
            return(gpa>3.5) ? true:false;
 }
};
class professor:public person
{
 int numpubs;
public:
 void getdata()
 {
            person::getname();
            cout<<"enter number of professor's publication:";
            cin>>numpubs;
 }
   bool outstanding()
 {
            return(numpubs>100) ? true:false;
 }
};
void main()
{
 person *perptr[10];
 int n=0;
 char choice;
 do
 {
            cout<<" Enter student or professor(s/p):";
            cin>>choice;
            if(choice=='s')
              perptr[n]=new student;
                        else
                                    perptr[n]=new professor;
                        perptr[n++]->getdata();
                        cout<<"enter another(y/n)?";
                        cin>>choice;
 }while (choice=='y');
 for(int j=0;j<n;j++)
 {
            perptr[j]->disp();
            if(perptr[j]->outstanding())
```

```
                cout<<"this person is outstanding\n";
    }
}
```

Q.15 Write a global function which uses the above operator and prints the number and the number of times it appears (frequency) .

Ans:

```
void frequency_calc()
{
        bag b;
        for(int j=0;j<50;j++)
        {
                int temp;
                cin>>temp;
                b[j]=temp;
        }
         for(int i=0;i<50;i++)
        {
                 int k=0;
                  for(j=j+1;j<50;j++)
                 {
                        if (b[i]==b[j])
                           k++;
                  }
                cout<<b[i]<<"occurred"<<k<<"times";
        }
}
```

Q.16 How are template functions overloaded? Explain with a suitable example. **(8)**

Ans: A template function can be overloaded using a template functions. It can also be overloaded using ordinary functions of its name. The matching of the overloaded function is done in the following manner.

a. The ordinary function that has an exact match is called.
b. A template function that could be created with an exact match is called.
c. The normal matching process for overloaded ordinary functions is followed and the one that matches is called.

If no match is found, an error message is generated. Automatic conversion facility is not available for the arguments on the template functions. The following example illustrates an overloaded template function:

```
#include<iostream.h>
#include<string.h>
using namespace std;
template<class T>
void display(T x)
{
 cout<<"Template display: "<<x<<"\n";
}
void display(int x)
{
 cout<<"Explicit display: "<<x<<"\n";
}

int main()
{
 display(20);
 display(3.6);
 display('A');
 return 0;
}
```

Q.17 Give a function template SEARCH that can be used to search an array of elements of any type and returns the index of the element, if found. Give both the function prototype and the function definition for the template. Assume a class template Array <T> is available. **(5)**

Ans:

```
#include<iostream.h>
template<class atype>

//function returns index number of item, or -1 if not found
int find(atype *array,atype value,int size)
{

 for(int j=0;j<size;j++)
            if(array[j]==value)
                        return j;
            return -1;
```

```
}
char chrarr[]={1,3,5,9,11,13};
char ch=5;
int intarr[]={1,3,5,9,11,13};
int in=6;
long lonarr[]={1L,3L,5L,9L,11L,13L};
long lo=11L;
double dubarr[]={1.0,3.0,5.0,9.0,11.0,13.0};
double db=4.0;
int main()
{
 cout<<"\n 5 in chrarr: index="<<find(chrarr,ch,6);
 cout<<"\n 6 in intarr: index="<<find(intarr,in,6);
 cout<<"\n 11 in lonarr: index="<<find(lonarr,lo,6);
   cout<<"\n 4 in dubarr: index="<<find(dubarr,db,6);

cout<<endl;
return 0;

}
```

Q.18 Answer briefly :

(i) What happens in a while loop if the control condition is false initially ?

(ii) What is wrong with the following loop

```
while (n <= 100)
Sum += n*n;
```

(iii) What is wrong with the following program

```
int main( )
        {
        const double PI;
        int n;
        PI = 3.14159265358979;
        n = 22;
        }
```

(iv) When a multidimensional array is passed to a function, why does C++ require all but the first dimension to be specified in parameter list?

(v) Why can't ** be overloaded as an exponentiation operator?

(vi) How many constructors can a class have, and how should they be distinguished.

(2x6=12)

Ans:

i) If the condition in while loop initially false the loop does not enter into the statements inside the loop and terminates initially.

ii) In the given statement of while loop n is not initialized. Although there is no initialization expression, the loop variable n must be initialized before the loop begins.

iii) The given program code is wrong as const declared PI must be initialized at the time of declaration. Whereas initialization of int n is correct the correct code is
Int main()
{Const double PI=3.1459265358979;
Int n=22;}

iv) Multidimensional array is an array of arrays the function first take the argument as an array of parameter first. It does not need to know how many values in parameter there are, but it does need to know how big each parameter element is, so it can calculate where particular element is. So we must tell it the size of each element but not how many there are.

v) Only existing operator symbols may be overloaded. New symbols, such as ** for exponentiation, cannot be defined.

vi) A class can have Default constructor, constructor with an argument, constructor with two argument and copy constructor.

Q.19 What are the various ways of handling exceptions? Which one is the best? Explain. **(6)**

Ans: Exception handling is a mechanism that finds errors at run time. It detects an error and reports so that an appropriate action can be taken. It adopts the following ways:

a. Hit the exception
b. Throw the exception
c. Catch the exception
d. Handle the exception

The three blocks that is used in exception handling are try throw and catch. A further advancement that is made is of multiple catch handlers.

A program may have more than one condition to throw an exception. For this we can use more than one catch block in a program. An example is shown below illustrate the concept:

```
#include<iostream.h>
using namespace std;
void test(int x){
 try        {
            if(x==1) throw x;
            else
                        if(x==0) throw 'x';
            else
                        if(x==-1) throw 1.0;
            cout<<"End of try-block\n";
 }
 catch(char c)      {
            cout<<"Caught a character \n";
 }
 catch(int m)       {
            cout<<"Caught an integer\n";
 }
 catch(double d)  {
            cout<<"Caught a double\n";
 }
 cout<<""End of try-catch system\n\n;}
```

```
int main(){
 cout<<"Testing multiple catches\n";
 cout<<"x==1\n";
 test(1);
 cout<<"x==0\n";
 test(0);
 cout<<"x==-1\n";
 test(-1);
 cout<<"x==2\n";
 test(2);  return 0;}
```

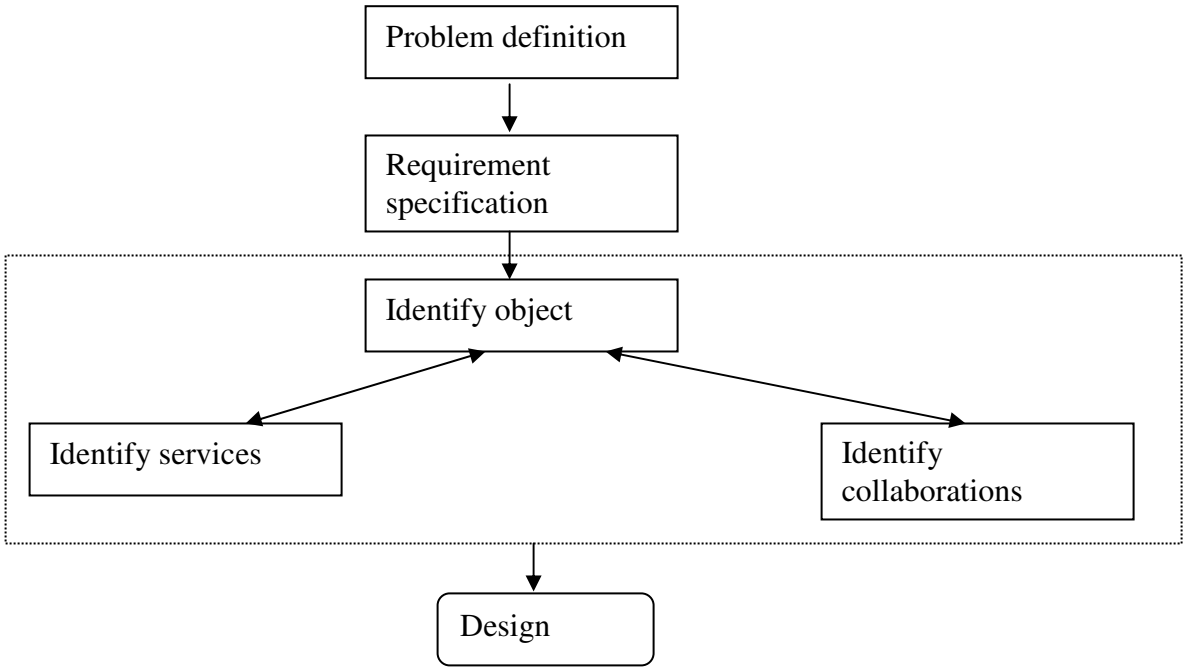
Q.20 Describe, in brief, the steps involved in object oriented analysis. **(6)**

Ans: The steps involved in object oriented analysis.

Object-oriented analysis provides us with a simple, yet powerful, mechanism for identifying objects, the building block of the software to be developed. The analysis is basically concerned with the decomposition of a problem into its component parts and establishing a logical model to describe the system functions.

The object-oriented analysis (OOA) approach consists of the following steps:

1. Understanding the problem.
2. Drawing the specifications of requirements of the user and the software.
3. Identifying the objects and their attributes.
4. Identifying the services that each object is expected to provide (interface).
5. Establishing inter-connections (collaborations) between the objects in terms of services required and services rendered.



Problem Understanding

The first step in the analysis process is to understand the problem of the user. The problem statement should be refined and redefined in terms of computer system engineering that could suggest a computer-based solution. The problem statement should be stated, as far as possible, in a single, grammatically correct sentence. This will enable the software engineers to have a highly focused attention on the solution of the problem. The problem statement provides the basis for drawing the requirements specification of both the user and the software.

Requirements Specification

Once the problem is clearly defined, the next step is to understand what the proposed. System is required to do. It is important at this stage to generate a list of user requirement A clear understanding should exist between the user and the developer of what is require Based on the user requirements, the specifications for the software should be drawn. Developer should state clearly

What outputs are required?
What processes are involved to produce these outputs?
What inputs are necessary?
What resources are required?

These specifications often serve as a reference to test the final product for its performance the intended tasks.

Identification of Objects

Objects can often be identified in terms of the real-world objects as well as the abstra~

objects. Therefore, the best place to look for objects is the application itself. The application may be analyzed by using one of the following two approaches:

1.Data flow diagrams (DFD)
2.Textual analysis (TA)

Identification of Services

Once the objects in the solution space have been identified, the next step is to identify set services that each object should offer. Services are identified by examining all the verbs are verb phrases in the problem description statement. *Doing verbs* and *compare verbs* usually give rise to services (which we call as functions C++). *Being verbs* indicate the existence of the classification structure while *having* verbs' rise to the composition structures.

Establishing interconnection

This step identifies the services that objects provide and receive. We may use an information flow diagram(IED) or an entity-relationship (ER) diagram to enlist this information.

Q.21 Write a program in C++ which calculates the factorial of a given number. **(6)**

Ans: Program to calculate factorial of a given number

```
#include <iostream.h>

        long factorial (long a)
{
 if (a > 1)
  return (a * factorial (a-1));
 else
  return (1);
```

```
}

int main ()
{
 long ln;
 cout << "Type a number: ";
 cin >> ln;
 cout << "!" << ln << " = " << factorial (ln);
 return 0;
}
```

Q.22 Define a class Queue with the following data members

(i) An array of numbers that the Queue can hold

(ii) Indices 'front' and 'rear' to specify the positions in Queue.
Initially set front = rear = -1
and the following member functions

(i) Delete – Remove front element.

(ii) Add – Add an element at rear. **(8)**

Ans:

```
#include<iostream.h>
#include<conio.h>
#include<process.h>

struct Node{
int data;
Node *next;
}*rear,*front;
void Create();
void AddItem();
void DeleteItem();
void Display();
void main(){
int ch=0;
Create();
clrscr();
while(ch!=4){
cout<<"\n1. Add  item";
cout<<"\n2. Delete item";
cout<<"\n3. Display";
cout<<"\n4. Quit";
cout<<"\n Enter choice";
cin>>ch;
switch(ch){
case 1:
AddItem();
break;
case 2:
```

```
DeleteItem();
break;
case 3:
Display();
break;
}
}
}
void Create(){
front=rear=NULL;
}
void AddItem(){
Node *p=new Node;
cout<<"\n Enter the data:";
cin>>p->data;
p->next=NULL;
if(front==NULL){
 front=rear=p;
 }
else{
 rear->next=p;
 rear=p;
    }
}
void DeleteItem(){
if(front==NULL){
 cout<<"\nQueue is empty";
 return;
 }
if(front==rear){
 cout<<endl<<front->data<<" is deleted";
 delete front;
 front=rear=NULL;
 }
else{
 Node *p=front;
 cout<<endl<<front->data<<" is deleted";
 front=front->next;
 delete p;
 }
}
void Display(){
Node *p=front;
while(p!=NULL){
cout<<"\n"<<p->data;
p=p->next;
}}
```

Q.23 What are the two methods of opening a file? Explain with examples. What is the difference between the two methods? **(10)**

Ans: For opening a file, we first create a file stream which we can link to the file name. we can define a file stream using 3 classes namely ifstream, ofstream and fstream that are all contained in the header file fstream. Which class to use shall depend upon the mode in which we want to open the file viz, read or write. There are 2 modes to open a file.

- Using the constructor function of the class
- Using the member function "open()" of the class.

When using the first method that is constructor function of the class, we initialize the file stream object by file name. for example the following statement opens a file named "results" for input.

ifstream infile ("results");

When using the second method of function "open()", multiple files can be opened that use the same stream object for example when we wish to process a number of files in sequential manner, we shall create a single stream object and use it to open each file in turn. The syntax is as follows:

file-stream-class stream-object;

stream-object.open("filename");

The basic difference between the two methods is that the constructor function of the class is used when we wish to open just one file in the stream. The open function is used when we wish to open multiple files using one stream.

Q.24 What is containership? How does it differ from inheritance? **(4)**

Ans: Inheritance, one of the major features of OOP, is to derive certain properties of the base class in the derived class. A derived class can make use of all the public data members of its base classes.

Containership is different from Inheritance. When one class is defined inside another class, it is known as containership or nesting. We can thus imply that one object can be a collection of many different objects. Both the concepts i.e. inheritance and containership have a vast difference. Creating an object that contains many objects is different from creating an independent object. First the member objects are created in the order in which their constructors occur in the body. Then the constructor of the main body is executed. An initialization list of the nested class is provided in the constructor.

Example:

```
class alpha{....};
class beta{....};

class gamma
{
 alpha a;
 beta b;
 public:
```

```
gamma(arglist): a(arglist1), b(arglist2)     {
        //constructor body
}};
```

Q.25. How do the properties of following two derived classes differ?

(i) class X : public A{//..}

(ii) class Y : private A{//..} **(5)**

Ans: The properties of the following two derived class differ:-

(i) class X:public A{//..}

In this class A is publicly derived from class X. the keyword public specifies that objects of derived class are able to access public member function of the base class.

i) class Y:private A{//..}

In this class A is privately derived from class Y. the keyword private specifies that objects of derived class cannot access public member function of the base class. Since object can never access private members of a class.

Q.26 Identify and remove error from the following, if any :-

```
array_ptr = new int [10] [10] [10];
array_ptr = new int [20] [ ] [ ];
```

(2)

Ans: There is an error we cannot assign more than one block sizes to one variable.

Correct array_ptr new int[10];

Q.27 What is the task of the following code :-

```
cout << setw (4) << XYZ << endl
```

(2)

Ans: The task of the given code is to display the value of XYZ after setting field width 4 (3) if XYZ=3; set causes the number or string that follows it in the stream to be printed within a field n characters wide.

Q.28 What is dynamic initialization of objects? Why is it needed? How is it accomplished in C++? Illustrate. **(8)**

Ans: Dynamic initialization of objects means to provide the initial values to an object at run time.

Using dynamic initialization of objects we can provide different initialization formats of data for an object at run time. We can use various data types like int, float or char depending upon our need for the same object created. This provides the flexibility of using different format of data at run time depending upon the situation.

It can be accomplished using the concept of constructor overloading. For example:

```
#include<iostream.h>
class loan
{ long principle;
  int year;
```

```
 float rate;
 public:
 loan(){}
 loan(long p,int y,float r=0.15);
 loan(long p,int y,int r);
};
```

This class uses three overloaded constructors. The parameter values to these constructors are provided at run time. One can input in any of the form to create object of this class.

Q.29 State, with reason, whether the following statement is true or false
$!(4<5)\|\|(6>7)$ is equal to false **(2)**

Ans: The following statement is true as condition !(4<5)||(6>7) is equal to false.

Q.30 The following macro is invoked as $f(x+3)$. What is the output when x = 5
$\#\text{define}\, f(x)\ \ x * x - x$ **(3)**

Ans: The output is 21. when the macro #define f(x) x*x-3 as f(x+3) when x=5

Q.31 List at least three C++ operators which cannot be overloaded. **(3)**

Ans: The operators ::, .*, . and ?: cannot be overloaded.

Q.32 Write a template function that swaps the values of two arguments passed to it. In main( ), use the function with integers and characters. **(4)**

Ans:

```
#include<iostream.h>
using namespace std;
template <class T>
void swap(T &a,T &b)
{
 T temp=a;
 a=b;
 b=temp;
}
void func(int x,int y,char w,char z)
{
 cout<<"x and y before swap: "<<x<<y<<"\n";
 swap(x,y)
 cout<<"x and y after swap: "<<x<<y<<"\n";

 cout<<"w and z before swap: "<<w<<z<<"\n";
 swap(w,z)
 cout<<"w and z after swap: "<<w<<z<<"\n";
```

```
}
int main()
{
 fun(10,20,'s','S');
 return 0;
}
```

Q.33 What is multiple inheritance? Discuss the syntax and rules of multiple inheritance in C++. How can you pass parameters to the constructors of base classes in multiple inheritance? Explain with suitable example. **(12)**

Ans: C++ provides us with a very advantageous feature of multiple inheritance in which a derived class can inherit the properties of more than one base class. The syntax for a derived class having multiple base classes is as follows:

```
Class D: public visibility base1, public visibility base2
{
   Body of D;
}
```

Visibility may be either 'public' or 'private'.
In case of multiple inheritance, the base classes are constructed in the order in which they appear in the declaration of the derived class. However the constructors for virtual base classes are invoked before any non-virtual base classes. If there are multiple virtual base classes, they are invoked in the order in which they are declared. An example program illustrates the statement.

```
#include<iostream.h>
using namespace std;
class M
{
    protected:
            int m;
    public:
            M(int x)
            {
                    m=x;
                    cout<< "M initialized\n";
            }
};
class N
{
    protected:
            int n;
    public:
            N(int y)
            {
```

```
                    n=y;
                    cout<< "N initialized\n";
                }
};
class P: public N, public M
{       int p,r;
    public:
      P(int a,int b,int c, int d):M(a),N(b)
          {
             p=c;
        r=d;
           cout<< "P initialized\n";
    }

};
void main()
{
    P p(10,20,30,40);

}
```

Output of the program will be:
N initialized
M initialized
P initialized

Q.34 Write a program in C++ to sort a list of given numbers in nondecreasing order. **(7)**

Ans:

```
#include<iostream.h>
void main()
{           int *num,i,j,temp,n,flag;
 num=new int[10];
 cout<<"how many number to sort(limit)"<<endl;
 cin>>n;
 cout<<"enter numbers to sort in nondecreasing order"<<endl;
 for(i=0;i<n;i++)
 {
           cin>>num[i];
 }
 for(i=0;i<n-1;i++)
 {
           flag=1;
           for(j=0;j<(n-1-i);j++)
           {
```

```
                    if(num[j]>num[j+1])
                    {
                            flag=0;
                    temp=num[j];
                    num[j]=num[j+1];
                    num[j+1]=temp;
                    }
            }
            if(flag)
                    break;
 }
            for(i=0;i<n;i++)
 {
            cout<<num[i]<<endl;
 }
}
```

Q.35 What do you mean by static class members? Explain the characteristics of static class members with suitable examples. **(8)**

Ans: Static class member variables are used commonly by the entire class. It stores values. No different copies of a static variable are made for each object. It is shared by all the objects. It is just like the C static variables.

It has the following characteristics:

- On the creation of the first object of the class a static variable is always initialized by zero.
- All the objects share the single copy of a static variable.
- The scope is only within the class but its lifetime is through out the program.

An example is given below:

```
#include<iostream.h>
using namespace std;
class num
{
 static int c;
 int n;
    public:
 void getd(int x)
 {
            n =x;
            c++;
 }
 void getc(void)
 {
            cout<<"count: "<<c<<"\n";
 }
};
int num :: c;
```

```
int main()
{
 num o1,o2,o3;
 o1.getc();
 o2.getc();
 o3.getc();

 o1.getd(1);
 o2.getd(2);
 o3.getd(3);
 cout<<"After reading data"<<"\n";
 o1.getc();
 o2.getc();
 o3.getc();
 return 0;
 }
```

Q.36 Create a class Patient that stores the patient name (a string) and the disease (a string) of the patient. From this class derive two classes : In_patient which has a data member roomrent (type float) and Out_patient which has a data member OPD_charges (float). Each of these three classes should have a nondefault constructor and a putdata() function to display its data. Write a main() program to test In_ patient and Out_patient classes by creating instances of them and then displaying the data with putdata(). **(10)**

Ans:

```
#include<iostream.h>
class patient
{
protected:
 char *name,*disease;
public:
patient(char *n,char *d)
{
 name=n;
 disease=d;
}
 void putdata()
 {
  cout<<"patient's name is"<<name<<endl;
  cout<<"Disease is"<<disease<<endl;
 }
};
class in_patient:public patient
{
 float roomrent;
public:
 in_patient(char *n,char *d,float rr):patient(n,d)
```

```
 {
            roomrent=rr;
 }
 void putdata()
 {
            patient::putdata();
            cout<<"Room rent is"<<roomrent;
            cout<<endl;
 }
};
class out_patient:public patient
{
 float opd_charges;
public:
 out_patient(char *n,char *d,float opd):patient(n,d)
 {
            opd_charges=opd;
 }

            void putdata()
 {

      patient::putdata();
            cout<<"OPD charges is"<<opd_charges;
 }
};
void main()
{
 char *nm,*dis;
 float rr1,op1;
 nm=new char;
 dis=new char;
    cout<<"Enter patient's name"<<endl;
  cin>>nm;
  cout<<"Enter disease"<<endl;
  cin>>dis;
  cout<<"Enter room rent"<<endl;
            cin>>rr1;
            cout<<"Enter opd charges"<<endl;
            cin>>op1;
 in_patient ip(nm,dis,rr1);
 out_patient op(nm,dis,op1);
 cout<<"Details of In_patient"<<endl;
 ip.putdata();
 cout<<"Details of Out_patient"<<endl;
 op.putdata();
}
```

Q.37 Consider the following specifications: **(8)**

Structure name	data	type	size
Name	First	array of characters	40
	Mid	array of characters	40
	Last	array of characters	60
Phone	Area	array of characters	4
	Exch	array of characters	4
	Numb	array of characters	6
Class name	data	type	
N_rec	Name	Name	
	Phone	Phone	

(i) Declare structures in C++ for Name and Phone.

(ii) Declare a class N_rec with the following members.

1. Define the constructor (outside the class N_rec) that gathers the information from the user for Name and Phone.
2. Define the display_rec(outside the class N_rec) that shows the current values of the data members.

Ans:

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
struct Name
{
 char First[40], Mid[40], Last[60];
};

struct Phone
{
 char Area[4], Exch[4], Numb[6];
};
class N_Rec
{
 struct Name name;
 struct Phone phone;
 public:
        N_Rec();
        Display_Rec();

};

N_Rec :: N_Rec()
{
 strcpy(name.First,"Shailesh");
```

```
 strcpy(name.Mid," ");
 strcpy(name.Last,"Saurabh");
 strcpy(phone.Area,"011");
 strcpy(phone.Exch,"64");
 strcpy(phone.Numb,"608284");
}

N_Rec :: Display_Rec()
{
 cout<<name.First<<" " <<name.Mid<<" "<<name.Last;
 cout<<endl<<phone.Area<<" "<<phone.Exch<<" "<<phone.Numb;

};

main()
{
 clrscr();
 N_Rec obj1;
 obj1.Display_Rec();
 getch();
}
```

Q.38 When do we use multi catch handlers? Explain with an example. **(8)**

Ans: Multiple Catch handlers:

Just like we use conditions in a switch statement, we can use multi catch statements with one try block when a program has to throw an exception based on more than one condition.

The multi handlers are searched in sequence of which they occur. The first match that is found is executed.If no match is found then the program terminates.

An example program is given below:

```
#include<iostream.h>
using namespace std;
void test(int x)
{
 try
 {
        if(x==1) throw x;
        else
                if(x==0) throw 'x';
        else
                if(x==-1) throw 1.0;
        cout<<"End of try-block\n";
 }
 catch(char c)
 {
        cout<<"Caught a character \n";
 }
```

```
 catch(int m)       {
            cout<<"Caught an integer\n";
 }
 catch(double d)  {
            cout<<"Caught a double\n";
 }
 cout<<""End of try-catch system\n\n;
}
int main()
{
 cout<<"Testing multiple catches\n";
 cout<<"x==1\n";
 test(1);
 cout<<"x==0\n";
 test(0);
 cout<<"x==-1\n";
 test(-1);
 cout<<"x==2\n";
 test(2);
 return 0;}
```

The output:

```
Testing Multiple Catches
x==1
Caught an integer
End of try-catch system
x==0
Caught a character
End of try-catch system
x==-1
Caught a double
End of try-catch system
x==2
End of try-block
End of try-catch system
```

Q.39 Find errors in the following code: **(6)**

```
#include<iostream.h>
class A { int a1;
        public: int a2;
        protected : int  a3;    };
class B :public A
        { public:
        void  func( )
        { int b1, b2, b3;
              b1 = a1;
```

```
        b2 = a2;
        b3 = a3; } };
class C : A
        { public:
                void f( )
                { int c1, c2, c3;
                        c1 = a1;
                        c2 = a2;
                        c3 = a3;   }   };
int main( )
{ int p, q, r, i, j, k;
        B   O1;
        C   O2;
        p = O1.a1;
        q = O1.a2;
        r = O1.a3;
        i = O2.a1;
        j = O2.a2;
        k = O2.a3;
        }
```

Ans:

In the given code:

- Class B cannot have access over the private variable a1 of class A as a private variable is never inherited. Thus b1=a1 cannot be used.
- Class C also cannot have access over the private member a1 of class A as C inherits class B which does not contain a1 as its inherited member. Thus c1=a1 cannot be used.
- Inside main() function, object O1 of class B cannot access a1. Thus the statement O1.a1 is not valid.
- Similarly object O2 of class C also cannot access a1. Thus statement O2.a1 is also not valid.

Inside main() function, the protected member a3 of class B is not accessible.So the statements "r = O1.a3; and k = O2.a3; " is invalid

Q.40 Write short notes on any **TWO** :-

(i) Procedure oriented vs object oriented programming.
(ii) Object oriented design.
(iii) User defined data types in C++. **(2 x 7)**

Ans:

(i) Procedure oriented vs object oriented programming

Procedure-oriented programming basically consists of writing a list of instructions (or actions) for the computer to follow, and organizing these instructions into groups known as *functions* . We normally use a *flowchart* to organize these actions and represent the flow of control from one action to another. While we concentrate on the development of functions, very little

attention is given to the data that are being used by various functions. What happens to the data? How are they affected by the functions that work on them? In a multi-function program, many important data items are placed as *global* so that they may be accessed by all the functions. Each function may have its own *local data.* Global data are more vulnerable to an inadvertent change by a function. In a large program it is very difficult to identify what data is used by which function. In case we need to revise an external data structure, we also need to revise all functions that access the data. This provides an opportunity for bugs to creep in.
Another serious drawback with the procedural approach is that it does not model real world problems very well. This is because functions are action-oriented and do not really correspond to the elements of the problem.

Some characteristics exhibited by procedure-oriented programming are:

- Emphasis is on doing things (algorithms).
- Large programs are divided into smaller programs known as functions. Most of the functions share global data.
- Data move openly around the system from function to function.
- Functions transform data from one form to another.
- Employs *top-down* approach in program design.

Object-Oriented programming

The major motivating factor in the invention of object-oriented approach is to remove some of the flaws encountered in the procedural approach. OOP treats data as a critical element in the program development and does not allow it to flow freely around the system. It ties data more closely to the functions that operate on it, and protects it from accidental modification from outside functions. OOP allows decomposition of a problem into a number of entities called *objects* and then builds data and functions around these objects. The data of an object can be accessed only by the functions associated with that object. However, functions of one object can access the functions of other objects.

Some of the striking features of object-oriented programming are:

- Emphasis is on data rather than procedure.
- Programs are divided into what are known as objects.
- Data structures are designed such that they characterize the objects.
- Functions that operate on the data of an object are tied together in the data structure.
- Data is hidden and cannot be accessed by external functions. .
- Objects may communicate with each other through functions.
- New data and functions can be easily added whenever necessary.
- Follows bottom-up approach in program design.

Object-oriented programming is the most recent concept among programming paradigms and still means different things to different people. It is therefore important to have a working definition of object-oriented programming before we proceed further. We define "object-oriented programming as an approach that provides a way of modularizing programs by creating partitioned memory area for both data and functions that can be used as templates for creating copies of such modules on demand." Thus, an object is considered to be a partitioned area of computer memory that stores data and set of operations that can access that data. Since

the memory partitions are independent, the objects can be used in a variety of different programs without modifications.

ii) Object oriented design

Design is concerned with the mapping of objects in the problem space into objects in the solution space, and creating an overall structure (architectural model) and computational models of the system. This stage normally uses the bottom-up approach to build the structure of the system and the top-down functional decomposition approach to design :the class member functions that provide services. It is particularly important to construct structured hierarchies, to identify abstract classes, and to simplify the inter-object communications. Reusability of classes from the previous designs, classification of the objects into subsystems and determination of appropriate protocols are some of the 'considerations of the design stage. The object oriented design (OOD) approach may involve the following steps:

1. Review of objects created in the analysis phase.
2. Specification of class dependencies.
3. Organization of class hierarchies.
4. Design of classes.
5. Design of member functions.
6. Design of driver program.

Review of Problem Space Objects

The main objective of this review exercise is to refine the objects in terms of their attributes and operations and to identify other objects that are solution specific. Some guidelines that might help the review process are:

1. If only one object is necessary for a service (or operation), then it operates only on that object.
2. If two or more objects are required for an operation to occur, then it is necessary to identify which object's private part should be known to the operation.
3. If an operation requires knowledge of more than one type of objects, then the operation is not functionally cohesive and should be rejected.

Class Dependencies

It is important to identify appropriate classes to represent the objects in the solution space and establish their relationships. The major relationships that are important in the context of design are:

1. Inheritance relationship
2. Containment relationship.
3. Use relationships

Organization of Class Hierarchies

This involves identification of common attributes and functions among a group of related classes and then combining them to form new class. The new class serves as the super class and the other as a subordinate classes.

Design of Class

Given below are some guidelines which should be considered while designing a class:

1. The public interface of a class should have only functions of the class.
2. An object of one class should not send a message directly to a member of another class.
3. A function should be declared public only when it is required to be used by the objects of the class.
4. Each function either accesses or modifies some data of the class it represents.

5. A class should be dependent on as few (other) classes as possible.
6. Interactions between two classes must be explicit.
7. Each subordinate class should be designed as a specialization of the base class with the sole purpose of adding additional features.
8. The top class of a structure should represent the abstract model of the target concept.

Design of Member Functions

The member functions define the operations that are performed on the object's data.

These functions behave like any other C function and therefore we can use the top-down functional decomposition technique to design them.

Design of the Driver Program

The driver program is mainly responsible for:

1. Receiving data values from the user
2 Creating objects from the class definitions
3. Arranging communication between the objects as a sequence of messages for invoking the member functions, and
4. Displaying output results in the form required by the user.

(iii) User defined data types in C++

Structure and Classes

Data types such as struct and class is user defined data type in c++

example

```
struct{
   char title[50];
  char author[50];
} book;
class student{
char title[50];
  char author[50];
public:
void get()}
student s1;
```

Enumerated data type

The enum keyword automatically enumerates a list of words by assigning them values 0,1,2 and so on enum colors_t {black, blue, green, cyan, red, purple, yellow, white};

Q.41 When does ambiguity arise in multiple inheritance? How can one resolve it? Develop a program in C++ to create a derived class having following items: name, age, rollno, marks, empcode and designation. Design a base class student having data members as rollno, marks and the another base class is employee having data members as empcode and designation. These both base classes are inherited from a single base class person with data members name and age. The program should carry out the required input( ) and output( ) member functions for all. **(12)**

Ans:

A class can be derived such that it inherits the properties of two or more base classes. This is called multiple inheritance.

Ambiguity occurs when the base classes have functions with same name. for example if two base classes have the function get_data().then which function will be used by the derived class? This problem can be solved be defining a named instance with the derived class, using the class resolution operator with the function name.

Program:

```
#include<conio.h>
#include<iostream.h>
class person{
 public:
 char name[20];
 int age;
 void get_details();
 void disp_details();
};
 void person::get_details() {
 cout<<"enter name:";
 cin>>name;
 cout<<"enter age:";
 cin>>age;
 }
 void person::disp_details() {
 cout<<"name:"<<name;
 cout<<'\n';
 cout<<"age:"<<age;
 cout<<'\n';
 }
 class student: public person {
 public:
 int roll_no;
 int marks;
 void get_details(int a,int b) {
  roll_no = a;
  marks = b;
 }
 void disp_data()
 {
  cout<<"roll_no:"<<roll_no;
  cout<<'\n';
  cout<<"marks:"<<marks; }
 };
 class employee:public person {
 public:
 int emp_code;
 char desig[20];
 void get_details(int a,char *d) {
  emp_code = a;
  strcpy(design, d); }
```

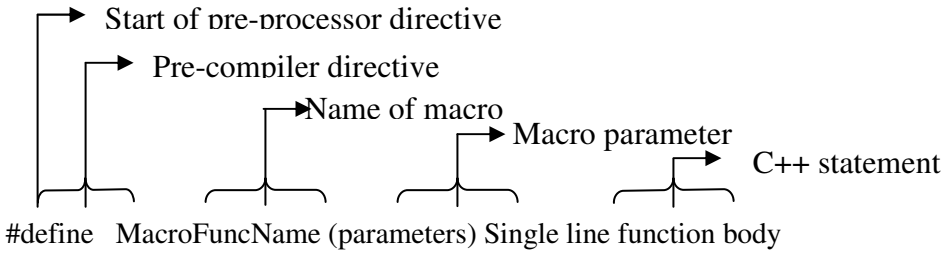
```
void disp_data() {
  cout<<"emp_code:"<<emp_code;
  cout<<'\n';
  cout<<"designation:"<<desig; }
 };
void main(){
 int r,m,c;char d[20];
 cout<<"\n Enter roll_no. and marks :";
 cin>>r,m
 cout<<"\n Enter employee code:";
 cin>>c;
 cout<<"\n Enter designation: ";
 getline(d);
 student ob1;
 employee ob2;
 ob1.get_details(r,m);
 ob2.get_details(c,d);
 ob1.disp_data();
 ob2.disp_data();getch();
}
```

Q.42 Write two advantages of using include compiler directive. **(4)**

Ans: The include compiler directive is a preprocessor directive. Means, it is a command to the preprocessor. A preprocessor is a program that performs the preliminary processing with the purpose of expanding macro code templates to produce a C++ code. By including this header file, the function that we are using is replaced by the entire function definition. This is very helpful in case of small modules which are to be called several times in a program. It reduces the overhead cost of calling the function each time.

Q.43 Explain the difference between macro & function. **(5)**

Ans:Difference between macro and function macro The preprocessor will replace all the macro function used in the program by their function body before the compilation. The distinguishing feature of macro function are that there will be no explicit function call during execution, since the body is substituted at the point of macro call during compilation. Thereby the runtime overhead for the function linking or context-switch time is reduced. The directive # define indicates the start of a macro function as shown in the figure.



Examples:
#define inc(a) a+1
#define add(a, b) (a+b)

Function:A function is a set of program statements that can be processed independer.tly. A function can be invoked which behaves as though its code is inserted at the point of the function call. The communication between a *caller* (calling function) and *callee* (called function) takes place through parameters. The functions can be designed, developed, and implemented independently by different programmers. The independent functions can be grouped to form a software library. Functions are independent because variable names and labels defined within its body are local to it.
The use of functions offers flexibility in the design, development, and implementation of the program to solve complex problems. The advantages of functions includes the following:

- Modular programming
- Reduction in the amount of work and development time
- Program and function debugging is easier
- Division of work is simplitied due to the use of divide-and-conquer principle .
- Reduction in size of the program due to code reusability
- Functions can be accessed repeatedly without redevelopment, which in turn promotes reuse of code .
- *Library of functions* can be implemented by combining well designed, tested and proven function.

Function Components
Every function has the following elements associated with it:
1. Function declaration or prototype.
2. Function parameters (formal parameters)
3. Combination of function declaration and its definition.
4. Function definition (function declaration and a function body).
5. return statement.
6. Function call.

A function can be executed using *a function call* in the program. The various components associated with the function are shown in Figure

```
Void func(int a, int b);    Function declaration
Prototype
                    parameter
……
void func(int a, int b) {   Function definition
……                                Body
}
func(x,y);                  Function Call
```

Q.44 Write a C++ program to find the sum of digits of a number reducing it to one digit. **(10)**

Ans:
```
#include<iostream.h>
void main(){
 int num,n,s,s1,limit;s=s1=0;
 cout<<"enter limit of no. of digits you want to enter";
```

```
cin>>limit;
cout<<"Enter (limit)digit number";
cin>>num;
for(int i=0;i<limit;i++)
{
        n=num%10;
                s=s+n;
        num=num/10;
}
 cout<<"sum of digits is"<<s;
 if(s>1)
 {
  while(s>0)
{
   n=s%10;
   s1=s1+n;
   s=s/10;
  }
cout<<"reduced to one digit is"<<s1;
  }
}
```

Q.45 Explain the use of break and continue statements in switch case statements. **(4)**

Ans: The switch Statement

The **switch** and **case** statements help control complex conditional and branching operations. The **switch** statement transfers control to a statement within its body.

Syntax

selection-statement :
switch (*expression* **)** *statement*
labeled-statement :
case *constant-expression* **:** *statement*
default : *statement*

Control passes to the statement whose **case** *constant-expression* matches the value of **switch (** *expression* **)**. The **switch** statement can include any number of **case** instances, but no two case constants within the same **switch** statement can have the same value. Execution of the statement body begins at the selected statement and proceeds until the end of the body or until a **break** statement transfers control out of the body.

Use of the **switch** statement usually looks something like this:

switch (*expression*)
{
declarations
.
. .
case *constant-expression* **:**

statements executed if the expression equals the

```
   value of this constant-expression
   .
   .
   .    break;
  default :
   statements executed if expression does not equal
   any case constant-expression
}
```

We can use the **break** statement to end processing of a particular case within the **switch** statement and to branch to the end of the **switch** statement. Without **break**, the program continues to the next case, executing the statements until a **break** or the end of the statement is reached. In some situations, this continuation may be desirable.

The **default** statement is executed if no **case** *constant-expression* is equal to the value of **switch** (*expression*). If the **default** statement is omitted, and no **case** match is found, none of the statements in the **switch** body are executed. There can be at most one **default** statement. The **default** statement need not come at the end; it can appear anywhere in the body of the **switch** statement. In fact it is often more efficient if it appears at the beginning of the **switch** statement. A **case** or **default** label can only appear inside a **switch** statement.

The type of **switch** *expression* and **case** *constant-expression* must be integral. The value of each **case** *constant-expression* must be unique within the statement body.

The **case** and **default** labels of the **switch** statement body are significant only in the initial test that determines where execution starts in the statement body. Switch statements can be nested. Any static variables are initialized before executing into any **switch** statements.

The following examples illustrate **switch** statements:

```
switch( c )
{
   case 'A':
      capa++;
   case 'a':
      lettera++;
   default :
      total++;
}
```

All three statements of the **switch** body in this example are executed if c is equal to 'A' since a **break** statement does not appear before the following case. Execution control is transferred to the first statement (capa++;) and continues in order through the rest of the body. If c is equal to 'a', lettera and total are incremented. Only total is incremented if c is not equal to 'A' or 'a'.

```
switch( i )
{
   case -1:
      n++;
```

```
        break;
    case 0 :
        z++;
        break;
    case 1 :
        p++;
        break;
}
```

In this example, a **break** statement follows each statement of the **switch** body. The **break** statement forces an exit from the statement body after one statement is executed. If i is equal to 1, only n is incremented. The **break** following the statement n++; causes execution control to pass out of the statement body, bypassing the remaining statements. Similarly, if i is equal to 0, only z is incremented; if i is equal to 1, only p is incremented. The final **break** statement is not strictly necessary, since control passes out of the body at the end of the compound statement, but it is included for consistency.

Q.46 What is Object Oriented Programming (OOP)? What are the advantages of Object Oriented Programming? **(6)**

Ans:

Ans: Object Oriented Programming (OOP) is an approach that provides a way of modulating programs by creating partitioned memory area for both data and functions that can be used as templates for creating copies of such modules on demand.

The **advantages** of OOP are as follows:

- Function and data both are tied together in a single unit.
- Data is not accessible by external functions as it is hidden.
- Objects may communicate with each other only through functions.
- It is easy to add new data and functions whenever required.
- It can model real world problems very well.

Q.47 Differentiate between recursion and iteration. **(4)**

Ans: Difference between recursion and iteration

Recursion and iteration are two very commonly used, powerful methods of solving complex problems, directly harnessing the power of the computer to calculate things very quickly. Both methods rely on breaking up the complex problems into smaller, simpler steps that can be solved easily, but the two methods are subtlety different. Iteration, perhaps, is the simpler of the two. **In iteration, a problem is converted into a train of steps that are finished one at a time, one after another.** For instance, if you want to add up all the whole numbers less than 5, you would start with 1 (in the 1st step), then (in step 2) add 2, then (step 3) add 3, and so on. In

each step, you add another number (which is the same number as the number of the step you are on). This is called "iterating through the problem." The only part that really changes from step to step is the number of the step, since you can figure out all the other information (like the number you need to add) from that step number. This is the key to iteration: using the step number to find all of your other information. The classic example of iteration in languages like BASIC or C++, of course, is the *for* loop.

If iteration is a bunch of steps leading to a solution, **recursion is like piling all of those steps on top of each other and then quashing them all into the solution.** Recursion is like holding a mirror up to another mirror: in each image, there is another, smaller image that's basically the same.

Example: **recursion**

```
int factorial(int number)
 {
        if (number < 0)
        {
                cout << "\nError - negative argument to factorial\n";
        exit(1);
        }
        else if (number == 0)

                return 1;

        else
                return (number * factorial(number - 1));
 }
```

Iteration :

```
int factorial(int number)
    {
        int product = 1;

        if (number < 0)
        {
                cout << "\nError - negative argument to factorial\n";
                exit(1);
        }
        else if (number == 0)
                return 1;
        else
        {
                for ( ; number > 0 ; number--)
```

```
                product *= number;
                return product;
        }

 }
```

So, the difference between iteration and recursion is that **with iteration, each step clearly leads onto the next, like stepping stones across a river**, while **in recursion, each step replicates itself at a smaller scale, so that all of them combined together eventually solve the problem.** These two basic methods are very important to understand fully, since they appear in almost every computer algorithm ever made.

Q.48 What are the different storage classes in C++. **(5)**

Ans: STORAGE CLASSES

C++ provides 4 storage class specifiers:
AUTO, REGISTER, EXTERN, and STATIC
An identifier's storage class specifier helps determine its storage class, scope, and linkage.
Storage class - determines the period during which that identifier exists in memory
Scope - determines if the identifier can be referenced in a program
Linkage - determines for a mulitple-source-file program whether an identifier is known only in the current source file or in any source file with proper declarations.

Automatic storage class

The **auto** and **register** keywords are used to declare variables of the automatic storage class. Such variables are created when the block in which they are declared is entered, they exist while the block is active, and they are destroyed when the block is exited.
Example:
auto float x, y;
declares that **float** variables **x** and **y** are local variables of automatic storage class, they exist only in the body of the function in which the definition appears.
register int counter = 1;
declares that the integer variable **counter** be placed in one of the computer's register and be initialized to 1.
Either write **auto** or **register** but not both to an identifier.

Static Storage Class

The keywords **extern** and **static** are used to declare identifiers for variables and functions of the static storage class. Such variables exist from the point at which the program begins execution.

There are two types of identifiers with static storage class: external identifiers (such as global variables and function names) and local variables declared with the storage class specifier **static**. Global variables and function names default to storage class specifier **extern**. Global variables are created by placing variable declarations outside any function definition. They retain their values throughout the execution of the program.

Q.49 Write a program to overload the unary minus operator using friend function. **(8)**

Ans:

```
#inlcude<iostream.h>
using namespace std;
class space{
 int x;
 int y;
 int z;
   public:
 void getdata(int a,int b.int c);
 void display(void);
 friend void operator-(space &s);
};
void space :: getdata(int a,int b,int c){
 x=a;
 y=b;
 z=c;
}
void space :: display(void){
 cout<<x<<" ";
 cout<<y<<" ";
 cout<<z<<"\n";
}
void  operator-( space &s){
 s.x=-s.x;
 s.y=-s.y;
 s.z=-s.z;
}
int main(){
 space s;
 s.getdata(10,-20,30);
 cout<<"S : ";
 s.display();
 -s;
 cout<<"S :";
 s.display();
 return 0;
}
```

Q.50 Write a program in which a class has three data members: name, roll no, marks of 5 subjects and a member function Assign() to assign the streams on the basis of table given below: **(9)**

Avg. Marks	**Stream**
90% or more	Computers
80% - 89%	Electronics
75% - 79%	Mechanical
70% - 74%	Electrical

Ans:

```
#include<iostream.h>
class student
{
 char name[10];
 int rollno,marks[5],sum,avg;
public:
 void assign()
 {
   cout<<"enter name of the student";
 cin>>name;
 cout<<"enter rollno";
 cin>>rollno;
 cout<<"enter marks of five subjects";
 for(int i=0;i<5;i++)
 {
          cin>>marks[i];
 }
 sum=avg=0;
 for(int j=0;j<5;j++)
 {
          sum=sum+marks[j];
 }
 //cout<<sum;
 avg=sum/5;
 cout<<"Average marks is"<<avg;
 cout<<endl;
 if(avg>=90)
 {
          cout<<"stream is Computer"<<endl;
 }
          else
          {
                  if(avg>=80 && avg<=89)
                  {
                          cout<<"stream is Electronics";
```

```
        }
                if(avg>=75 && avg<=79)
                {
                        cout<<"stream is Mechanical";
                }
                if(avg>=70 && avg<=74)
                {
                        cout<<"stream is Electrical";
                }
        }
 }
 };

 void main()
 {
        student s1;
        s1.assign();
 }
```

Q.51 Write a program to convert the polar co-ordinates into rectangular coordinates. (hint: polar co-ordinates(radius, angle) and rectangular co-ordinates(x,y) where x = r*cos(angle) and y = r*sin(angle)). **(8)**

Ans:

```
#include<iostream.h>
#include<math.h>
using namespace std;
class coord
{
 float radius,angle,x,y;
 public:
    polar(float r, float a);
    void rectangular(float x, float y);
    void display();
}
void coord :: polar(float r,float a)
{
 cout<<"\n Enter the radius and angle : ";
 cin>>radius>>angle;

}
void coord :: rectangular(float x,float y)
{
 x=radius*cos(angle);
 y=radius*sin(angle);
}
```

```
void display()
{
 cout<<"\n The rectangular coordinates are : "<<x<<" and "<<y;
}
void main()
{
 coord obj;
 obj.polar();
 obj.rectangular();
 obj.display();
}
```

Q.52 Is the following code correct? Justify your answer

```
int intvar = 333;
int * intptr;
cout << *intptr;
```

(3)

Ans: The following code is not correct as it will display nothing the given code is

```
 int intvar=333;
 int *intptr;
cout<<*intptr;
```

since the pointer variable is not assigned any address so it will display nothing. But if we add one more line that is intptr=&intvar; then it will display 333.

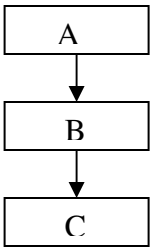
Q.53 What is multilevel inheritance? How is it different from multiple inheritance? **(5)**

Ans: Multilevel Inheritance

We can built hierarchies that contain as many layers of inheritance as we like. For example, given three classes called **A**, **B** and **C**, **C** can be derived from **B**, which can be derived from **A**. When this type of situation occurs, each derived class inherits all of the traits found in all of its base classes. In this case, **C** inherits all aspects of **B** and **A**.

```
class A{…….};   //Base class
class B: public A{…..}; // B derived from A
class C:public B{……}; // C derived from B
```

This process can be extended to any numbers of levels



Base

Intermediate Base Class

Derived class

Multiple Inheritance.
When the number of base classes Inherited exceeds one then the type of inheritance is known as **Multiple Inheritance**.
For example in C++, The syntax:

```
Class D: public A, public B
{
 //code
}
```

Indicates that **D** inherits the attributes of the two classes **A** and **B**.

Though Multiple Inheritance sounds exciting and very useful it has its own drawbacks, owing to which most OOP languages deny support for this feature. These drawbacks include complexities involved when inheriting from many classes and the difficulty in managing such codes.

Q.54 Write a C++ program to prepare the mark sheet of an university examination with the following items from the keyboard :

Name of the student
Roll no.
Subject name
Subject code
Internal marks
External marks
Design a base class consisting of data members Name of the student and Roll no. The derived class consists of the data members Subject name, Subject code, Internal marks and External marks. **(9)**

Ans:

```
#include<iostream.h>
class student
{
 char name[10];
 int rollno;
public:
 void get()
 {
        cout<<"enter student's name"<<endl;
        cin>>name;
        cout<<"enter student's rollno"<<endl;
        cin>>rollno;
 }
 void disp()
 {
```

```
 cout<<"Name of the student is : "<<name<<endl;
            cout<<"Rollno is : "<<rollno<<endl;
 }
};

class marks:public student
{
 char subjectname[15];
 int subcode,intmarks,extmarks;

public:
 void get()
 {
            student::get();
            cout<<"enter subject name";
            cin>>subjectname;
            cout<<"enter subject code";
            cin>>subcode;
            cout<<"enter Internal marks";
            cin>>intmarks;
            cout<<"enter External marks";
            cin>>extmarks;
 }
 void disp()
 {
            student::disp();
            cout<<"Subject name is : "<<subjectname<<endl;
            cout<<"Subject code is : "<<subcode<<endl;
            cout<<"Internal marks : "<<intmarks<<endl;
            cout<<"External marks : "<<extmarks<<endl;
 }
};

void main()
{
 marks m;
 m.get();
 m.disp();
}
```

Q.55 Explain the following:
(i) Non public constructors
(ii) Inline function
(iii) Virtual functions
(iv) Types of Inheritance. **(4x4=16)**

Ans:

i.) **Non-public constructors:** If a constructor is protected or private, it is called non-public constructor. We cannot create objects of a class that have non-public constructors. Only member function and friend function can create objects in this case.

ii.) **Inline functions**: A function which is expanded in line is called an inline function. Whenever we write there are certain costs associated with it such as jumping to the function, saving registers, pushing arguments into the stack and returning to the calling function. To remove this overhead we write an inline function. It is a sort of request from the compiler. If we declare and define a function within the class definition then it is automatically inline.

Advantage of inline functions:

- They have no overhead associated with function cal or return mechanism

Disadvantage of inline functions:

- If the function made inline is too large and called regularly our program grows larger.

There are certain restrictions in which the compiler may apply like:- some compilers will not inline a function if it contains a static variable or if the function contains a loop, a switch , etc..

iii.) **Virtual function:** virtual functions are important because they support polymorphism at run time. A virtual function is a member function that is declared within the base class but redefined inside the derived class. To create a virtual function we precede the function declaration with the keyword virtual. Thus we can assume the model of "one interface multiple method". The virtual function within the base class defines the form of interface to that function. It happens when a virtual function is called through a pointer.

The following is an example that illustrates the use of a virtual function:

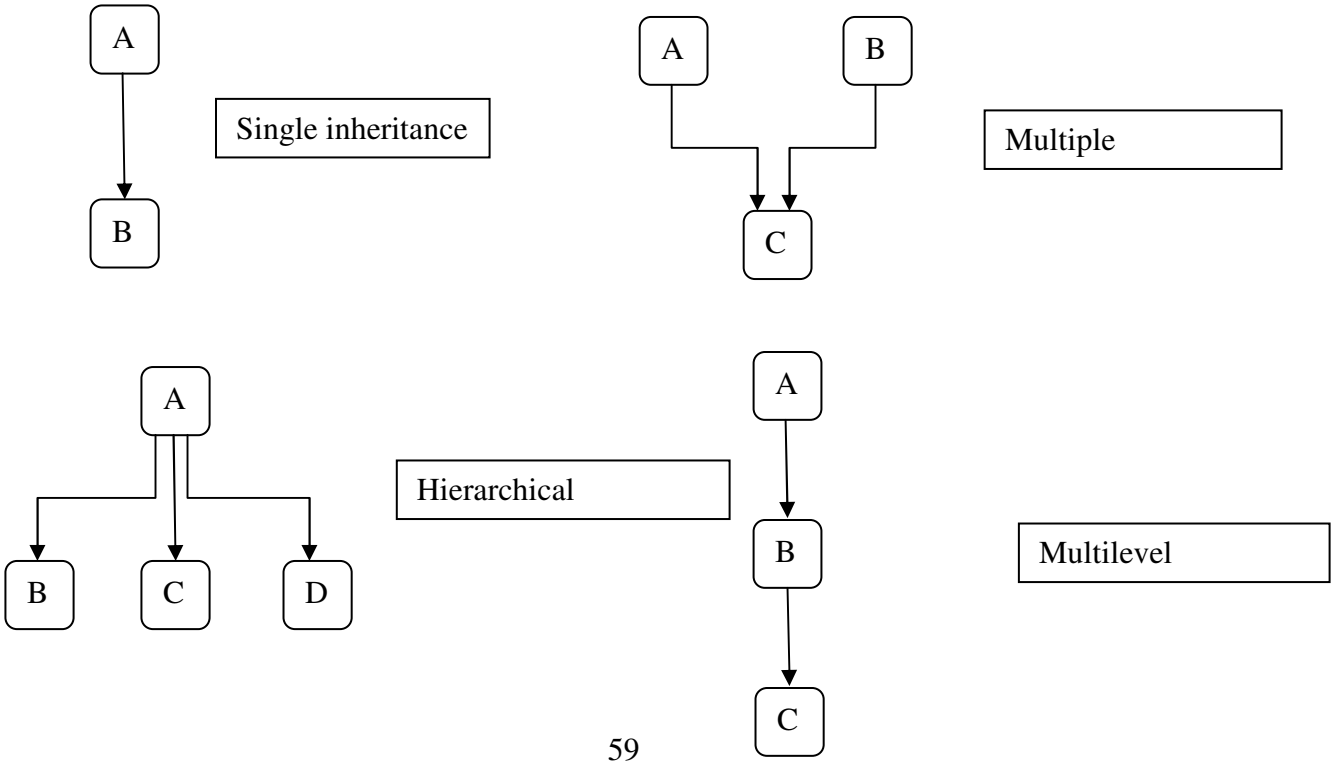
```
#include<iostream.h>
using namespace std;
class base
{
```

```
  public:
 int i;
 base(int x)
 {
            i=x;
 }
 virtual void func()
 {
            cout<<"\n using base version of function";
            cout<<i<<endl;
 }
};
class derived: public base
{
  public:
 derived(int x):base(x){}
 void func()
 {
            cout<<"\n sing derived version";
            cout<<i*i<<endl;
 }
};
class derived1: public base
{
  public:
 derived1(int x): base(x){}
 void func()
 {
            cout<<"\n using derived1 version";
            cout<<i+i<<endl;
 }
```

```
};
int main()
{
 base *p;
 base ob(10);
 derived d_ob(10);
 derived1 d_ob1(10);
 p=&ob;
 p->func();
 p=&d_ob;
 p->func();
 p=&d_ob1;
 p->func();
 return 0;
}
```

iv.) **Inheritance:** the mechanism of deriving a new class from an old one is called inheritance. The old class is called the base class and the new one is known as the derived class or sub class.
Types of inheritance

a. A derived class with only one base class is called single inheritance
b. A derived class with several base classes is called multiple inheritance.
c. When one base class is inherited by more than one class is called hierarchical inheritance.
d. When one derived class inherits another derived class, it is called multilevel inheritance.



Q.56 Write a program that shows the use of read() and write() to handle file I/O involving objects. **(8)**

Ans:

```
#include<iostream.h>
#include<fstream.h>
#include<iomanip.h>
class INVENTORY
{
 char name[10];
 int code;
 float cost;
    public:
 void readdata(void);
 void writedata(void);
};
void INVENTORY :: readdata(void)
{
 cout<<"\n Enter name:";cin>>name;
 cout<<"\n Enter code:";cin>>code;
 cout<<"\n Enter cost:";cin>>cost;
}
void INVENTORY :: writedata(void)
{
 cout<<setiosflags(ios::left)
     <<setw(10)<<name
     <<setiosflags(ios::right)
     <<setw(10<<code)
     <<setprecision(2)
     <<setw(10)<<cost<<endl;
}
```

```
int main()
{
 INVENTORY item[3];
 fstream file;
 file.open("Stock.dat",ios::in|ios::out);
 cout<<"\n Enter details for three  items \n";
 for(int i=0;i<3;i++)
 {
        item[i].readdata();
        file.write((char *) & item[i],sizeof(item[i]));
 }
 file.seekg(0);
 cout<<"\nOutput\n";
 for(i=0;i<3;i++)
 {
        file.readdata((char *) &item[i],sizeof(item[i]));
        item[i].writedata();
 }
 file.close();
 return 0;
}
```

Q.57 Define a function area() to compute the area of objects of different classes – triangle, rectangle, square. Invoke these in the main program. Comment on the binding (static or dynamic) that takes place in your program. **(10)**

Ans:

```
#include <iostream.h>    //  For input/output
class triangle
{
float base, height, areat;
public:
 void area()
 {
  cout << "Enter the base of the triangle: ";
  cin  >> base;
```

```
  cout << endl;   // Go to a new line
   cout << "Enter the height of the triangle: ";
   cin  >> height;
   cout << endl;  //  Go to a new line

   areat = 0.5 * base * height;

    cout << "\n\n\n\n\n\n\n\n\n\n";    //  What is this line doing?

   cout << base   << " is the base length of the triangle. \n\n";
   cout << height << " is the height of the triangle. \n\n";
   cout << areat   << " is the area of the triangle. \n\n";
  }
  };
class rectangle
{
float breadth, length, arear;
public:
  void area()
  {
   cout << "Enter the breadth of the rectangle: ";
   cin  >> breadth;
   cout << endl;   // Go to a new line
   cout << "Enter the length of the rectangle: ";
   cin  >> length;
   cout << endl;  //  Go to a new line

   arear = breadth * length;

    cout << "\n\n\n\n\n\n\n\n\n\n";    //  What is this line doing?

   cout << breadth   << " is the length of the rectangle. \n\n";
   cout << length << " is the height of the rectangle. \n\n";
   cout << arear   << " is the area of the rectangle. \n\n";
  }
};

class square
{
float side,areas;
public:
  void area()
  {

   cout << "Enter the height of the square: ";
   cin  >> side;
```

```
  cout << endl;  //  Go to a new line

  areas = side * side;

   cout << "\n\n\n\n\n\n\n\n\n\n";    //  What is this line doing?

     cout << side << " is the side of the square:. \n\n";
  cout << areas   << " is the area of the square:. \n\n";
 }
};

main()
{
triangle t;
rectangle r;
square s;
t.area();
r.area();
s.area();
  return(0);
}
```

Q.58 What are the basic differences between manipulators and ios member functions in implementation? Give examples. **(8)**

Ans:

The basic difference between manipulators and ios member functions in implementation is as follows:

The ios member function returns the previous format state which can be of use later on but the manipulator does not return the previous format state. Whenever we need to save the old format states, we must use the ios member functions rather then the manipulators. Example: an example program illustrating the formatting of the output values using both manipulators and ios functions is given below.

```
#include<iostream.h>
#include<iomanip.h>
using namespace std;
int main()
{
 cout.setf(ios::showpoint);
 cout<<setw(5)<<"\n"
    <<setw(15)<<"Inverse_of_n"
    <<setw(15)<<"Sum_of_terms\n\n";
```

```
        double term,sum=0;
        for(int n=1;n<=10;n++)
        {
                term=1.0/float(n);
                sum+=term;
                cout<<setw(5)<<n
                   <<setw(14)<<setprecision(4)
                   <<setiosflags(ios::scientific)<<term
                   <<setw(13)<<resetiosflags(ios::scientific)
                   <<sum<<endl;
        }
        return 0;
}
```

Q.59 Class Y has been derived from class X. The class Y does not contain any data members of its own. Does the class Y require constructor. If yes, why? **(4)**

Ans: Yes, it is mandatory for class Y to have a constructor. It is the responsibility of the derived class constructor to invoke the base class constructor by passing required arguments to it.Unless the constructor of class Y is invoked the data members of class X that have been inherited by class Y will not be given any values. Although class Y does not have any data member of its own, yet to use the data members inherited from X, Y needs a constructor.

Y(type1 x, type2 y,…):

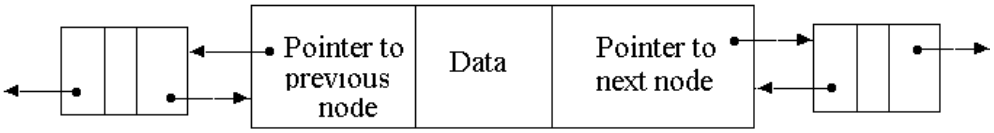
Q.60 Write a program using function template to find the cube of a given integer, float and double number. **(6)**

Ans:

```
//Using a function template

#include <iostream.h>

template < class T >

T cube( T value1)
{
```

```
  return value1*value1*value1;

  }
int main()
{
 int int1;

 cout <<"Input integer value: ";
 cin  >> int1;
 cout << "The cube of integer value is: "<< cube( int1);
 double double1;
 cout << "\nInput double value; ";
 cin  >> double1;
 cout << "The cube of double value is: "<< cube( double1);
 float float1;
 cout     << "\nInput float value";
 cin      >> float1;
 cout     << "The cube of float value is: "<< cube(float1);
cout<< endl;
 return 0;
}
```

Q.61 Write a C++ program to insert and delete a node from the double linked list. The list must be arranged in ascending order. Structure of node is **(14)**



Ans:

```
#include<iostream.h>
#include<conio.h>
#include<process.h>
struct Node{
int data;
Node *next;
}*start;
void Create();
void AddItem();
void DeleteItem();
```

```
void Display();

main(){
int ch=0;
Create();
clrscr();
while(ch!=4){

cout<<"\n1. Add  item";
cout<<"\n2. Delete item";
cout<<"\n3. Display";
cout<<"\n4. Quit";
cout<<"\n Enter choice";
cin>>ch;
switch(ch){
case 1:
AddItem();
break;
case 2:
DeleteItem();
break;
case 3:
Display();
break;
}
}
}

void Create(){
Node *h=new Node;
h->data=9999;
h->next=NULL;
start=h;
}

void AddItem(){
Node *p=new Node;
Node *loc, *old;
cout<<"Enter the Item :";
cin>>p->data;
old=start;
loc=start->next;
while(loc!=NULL){
 if(loc->data>p->data){
            p->next=old->next;
            old->next=p;
```

```
            return;
            }else{
                    old=loc;
                    loc=loc->next;
                    }
 }//End while\
//Insert at end
p->next=NULL;
old->next=p;
}

void DeleteItem(){
 Node *p;
 Node *loc,*old;
 int item;
 if(start->next==NULL){
 cout<<"\nList is empty";
 return;
 }
 cout<<"\nEnter the item to delete :";
 cin>>item;
 old=start;
 loc=start->next;
 while(loc!=NULL){
 if(loc->data==item){
            old->next=loc->next;
            cout<<"\n"<<loc->data<<" is deleted";
            delete loc;
            return;
            }else{
            old=loc;
            loc=loc->next;
            }
 }
 cout<<"\n Item not found";
}

void Display(){
Node *p=start;
while(p!=NULL){
cout<<"\n"<<p->data;
p=p->next;
}
}
```

Q.62 How can a common friend function to two different classes be declared? **(4)**

Ans:

Friend functions are not considered class members; they are normal external functions that are given special access privileges. Friends are not in the classs scope, and they are not called using the member-selection operators (**.** and **>**) unless they are members of another class.

Any data which is declared **private** inside a class is not accessible from outside the class. A function which is not a member or an external class can never access such private data. But there may be some cases, where a programmer will need need access to the private data from non-member functions and external classes. C++ offers some exceptions in such cases.

A class can allow non-member functions and other classes to access its own private data, by making them as **friends**. This part of C++ tutorial essentially gives two important points.

Once a non-member function is declared as a friend, it can access the private data of the class similarly when a class is declared as a friend, the friend class can have access to the private data of the class which made this a friend.

Q.63 Write a C++ program that copies the contents of a text file (i.e. any CPP file) to another file. Invoke the program with two command line arguments-the source file and the destination file – like this. **(10)**

C> ccopy scfile.cpp dstfile.cpp

Ans:

```
#include<iostream.h>
#include<fstream.h>
#include<process.h>
void main(int argc,char *argv[])
{
 if(argc!=3)
 {                  cout<<"\n argument is not passed";
          exit(-1);
 }
 char ch;
 ifstream inf;
 ofstream otf;
 inf.open(argv[1]);
 otf.open(argv[2]);
 if(!inf)
 {                  cout<<"\n can't open"<<argv[1];
          exit(-1);
 }
 while(inf)
 {                  inf.get(ch);
          otf<<ch;
}}
```

Q.64 Explain any three drawbacks of procedure oriented languages. Give an example each in C language to highlight the drawback. **(6)**

Ans: Drawbacks of procedure oriented languages.

Previously, programs were written in a procedural fashion i.e. the statements were written in the form of a batch. But as the requirements grew, it was seen that the programs were getting larger and larger and it became difficult to debug. So functions were introduced to reduce the size of the programs and improve readability in them. Still that was not enough.

One of the major problems with the "Procedural Paradigm" was that data was treated as a stepson and functions were given more priority. Where as, it is the other way. In this procedure the original data could easily get corrupted, as it was accessible to all the functions, even to those which do not have any right to access them. Before OOP, the programmer was restricted to use the predefined data types such as integer, float and character. If any program required handling of the x-y coordinates of some point then it is quite a headache for the programmer. Where as, in OOP this can be handled very easily as the programmer can define his own data types and the corresponding functions.

Q.65 Write a C++ program which reads in a line of text and counts the number of occurrences of the word 'the' in it. It outputs the line of text on one line, 'The number of times *the* appears is' and the count on the next line using a single cout statement. **(8)**

Ans:

```
#include <iostream.h>
#include <conio.h>

void main()
{
clrscr();
int countch=0;
int countwd=1;
cout << "Enter your sentence in lowercase: " << endl;
char ch='a';
while(ch!='\r')
{
ch=getche();
if(ch==' ')
countwd++;
else
countch++;
}
cout << "\n Words = " << countwd << endl;
cout << "Characters = " << countch-1 << endl;
getch();
}
```

This program takes in a sentence as a screen input from the user.

Q.66 Explain the difference between constructor and copy constructor in C++. Which of these is invoked in the following statement
Date D1 (D2); where D2 is also an object of class Date. **(3)**

Ans: *Constructors* are special `member functions' of which exactly one is called automatically at creation time of an object. Which one depends on the form of the variable declaration. Symmetrically, there exists a destructor, which is automatically called at destruction time of an object. Three types of constructors are distinguished: default constructor, copy constructor, and all others (user defined). A constructor has the same name as the class and it has no return type. The parameter list for the default and the copy constructor are fixed. The user-defined constructors can have arbitrary parameter lists following C++ overloading rules.

```
class A {
   int i; // private
public:
   A();            // default constructor
   A( const A& a);  // copy constructor
   A( int n);       // user defined
   ~A();           // destructor
};
int main() {
   A a1;  // calls the default constructor
   A a2 = a1;  // calls the copy constructor (not the assignment operator)
   A a3(a1);   // calls the copy constructor (usual constructor call syntax)
   A a4(1);    // calls the user defined constructor
}  // automatic destructor calls for a4, a3, a2, a1 at the end of the block
```

The compiler generates a missing default constructor, copy constructor, or destructor automatically. The default implementation of the default constructor calls the default constructors of all class member variables. The default constructor is **not** automatically generated if other constructors are explicitly declared in the class (except an explicit declared copy constructor). The default copy constructor calls the copy constructor of all class member variables, which performs a bitwise copy for built-in data types. The default destructor does nothing. All default implementations can be explicitly inhibited by declaring the respective constructor/destructor in the private part of the class.
Constructors initialize member variables. A new syntax, which resembles constructor calls, allows to call the respective constructors of the member variables (instead of assigning new values). The constructor call syntax extends also to built-in types. The order of the initializations should follow the order of the declarations, multiple initializations are separated by comma.

```
class A {
   int i; // private
public:
   A() : i(0) {}              // default constructor
A( const A& a) : i(a.i) {}  // copy constructor, equal to the compiler default
   A( int n) : i(n) {}        // user defined
```

```
   ~A() {}                    // destructor, equal to the compiler default
};
```

Usually, only the default constructor (if the semantics are reasonable), and some user defined constructors are defined for a class. As soon as the class manages some external resources, e.g., dynamically allocated memory, the following four implementations have to work together to avoid resource allocation errors: default constructor, copy constructor, assignment operator, and destructor. Note that the compiler would create a default implementation for the assignment operator if it is not defined explicitly

Q.67 Define a class Rectangle which has a length and a breadth. Define the constructors and the destructor and member functions to get the length and the breadth. Write a global function which creates an instance of the class Rectangle and computes the area using the member functions. **(8)**

Ans:

```
#include <iostream.h>
class CRectangle {
   int width, height;
  public:
    CRectangle (int,int);
    ~CRectangle ();
    friend int area(CRectangle);
};

CRectangle::CRectangle (int a, int b) {

  width = a;
  height = b;
}
CRectangle::~CRectangle () {
// delete width;
  //delete height;
}
int area(CRectangle r)
{
 return r.height * r.width;
}
int main () {
 int h,w;
 cout<<"enter length and breadth of rectangle";
 cin>>h>>w;
  CRectangle rect (h,w);
  cout << "area: " <<area(rect)<< endl;
  return 0;
}
```

Q.68 Why is destructor function required in class? What are the special characteristics of destructors? Can a destructor accept arguments? **(8)**

Ans: Destructor: A destructor is used to destroy the objects that have been created by a constructor. It has the same name as that of the class but is preceded by a tilde. For example, ~integer () { }

Characteristics:

- A destructor is invoked implicitly by the compiler when we exit from the program or block. It releases memory space for future use.
- A destructor never accepts any argument nor does it return any value.
- It is not possible to take the address of a destructor.
- Destructors can not be inherited.
- They cannot be overloaded

For example, the destructor for the matrix class will defined as follows:

```
matrix :: ~matrix()
{
 for(int i=0;i<d1;i++)
 delete p[i];
 delete p;
}
```

Q.69 Define a class Deposit which has a principal, a rate of interest which is fixed for all deposits and a period in terms of years. Write member functions to
(i) alter(float) which can change the rate of interest.
(ii) interest() which computes the interest and returns it.
(iii) print() which prints data as shown below. Note that the width of each column is 20. **(10)**

Principal	Year	Interest
1100.00	1	100.00

Ans:

```
#include<iostream.h>

class deposit
```

```
{
 float p,r,t,i;
public:
float interest()
{
cout << "Enter the principal, rate & time : " << endl;
cin>>p>>r>>t;
i=(p*r*t)/100;
return i;
}
 void print()
 {
cout << "Principal = Rs" << p << endl;
cout << "Rate = " << r << "%" << endl;
cout << "Time = " << t << " years" << endl;
cout << "Simple Interest = Rs" << i << endl;
}
 };
void main()
{
 deposit d;
 d.interest();
 d.print();
}
```

Q.70 Differentiate and give examples to bring out the difference between

(i) private and public inheritance.
(ii) instantiation and specialization of a template class.
(iii) static and dynamic binding.
(i) a class and a struct. **(3.5 x 4)**

Ans:

i) private and public inheritance

In private inheritence the public members of the base class become private members of the derived class. Therefore the object of derived class cannot have access to the public member function of base class.

In public inheritance the derived class inherit all the public members of the base class and retains their visibility. Thus the public member of the base class is also the public member of the derived class.

```
Class a
{
int x;
public:
```

```
void disp()
{
}
};
class y : private a
{
int c;
public:
void disp()
{
}
};
so when d.disp() // disp() is private;
```

ii) instantiation and specialization of template class

A template specialization allows a template to make specific implementations when the pattern is of a determined type. For example, suppose that our class template **pair** included a function to return the result of the module operation between the objects contained in it, but we only want it to work when the contained type is **int**. For the rest of the types we want this function to return **0**. This can be done the following way:

```
// Template specialization
#include <iostream.h>
template <class T>
class pair {
   T value1, value2;
  public:
   pair (T first, T second)
     {value1=first; value2=second;}
   T module () {return 0;}
};
template <>
class pair <int> {
   int value1, value2;
  public:
   pair (int first, int second)
     {value1=first; value2=second;}
   int module ();
};
template <>
int pair<int>::module() {
  return value1%value2;
}
int main () {
  pair <int> myints (100,75);
  pair <float> myfloats (100.0,75.0);
  cout << myints.module() << '\n';
```

```
  cout << myfloats.module() << '\n';
  return 0;
}
```

the process of creating a specific classs from a class template is called instantiation. The compiler will perform the error analysis only when an instantiation takes place.

```
// class templates
#include <iostream.h>
template <class T>
class pair {
   T value1, value2;
  public:
    pair (T first, T second)
      {value1=first; value2=second;}
    T getmax ();
};
template <class T>
T pair<T>::getmax ()
{
  T retval;
  retval = value1>value2? value1 : value2;
  return retval;
}
int main () {
  pair <int> myobject (100, 75);
  cout << myobject.getmax();
  return 0;
}
```

iii) Static and Dynamic binding

The information is known to the compiler at the compile time and, therefore compiler is able to select the appropriate function for a particular call at the compile time itself. This is called early binding or static binding.

The linking of function with a class much later after the compilation, this process is termed as late binding or dynamic binding because the selection of the appropriate function is done dynamically at run time. This requires the use of pointers to objects.

iv)a class and a struct

A class is a logical method to organize data and functions in the same structure. They are declared using keyword class, whose functionality is similar to that of the C keyword struct, but with the possibility of including functions as members, instead of only data.

Its form is:

```
class class_name {
  permission_label_1:
    member1;
  permission_label_2:
    member2;
```

... } object_name;

where ***class_name*** is a name for the class (user defined *type*) and the optional field ***object_name*** is one, or several, valid object identifiers. The body of the declaration can contain ***members***, that can be either data or function declarations, and optionally ***permission labels***, that can be any of these three keywords:

private:, **public:** or **protected:**

In struct all members declared are public by default.

Struct student{ char name[10]; int age;}stud;

Both struct and class is used as user defined data type.

Q.71 Explain the order in which constructors are invoked when there is multiple inheritance. **(4)**

Ans: The order in which constructor is invoked when there is multiple inheritance is order in which the class is derived that is constructor of base class will invoke first and then the constructor of derived class and so on. The destructor will invoke in reverse order than constructor. That is destructor of derived class will invoke first and then base class. Example

```
Class a
{
a()
{
cout<<"constructor a";
}
~a()
{
cout<<"destructor a";
}
};
class b : public a
{
b()
{
cout<<"constructor b";
}


~b()
{
cout<<"destructor b";
}
};
void main()
{
b b1;
}
```

output
constructor a
constructor b
destructor b
destructor a

Q.72 Define a class Publication which has a title. Derive two classes from it – a class Book which has an accession number and a class Magazine which has volume number. With these two as bases, derive the class Journal. Define a function print() in each of these classes. Ensure that the derived class function always invokes the base(s) class function. In main() create a Journal called IEEEOOP with an accession number 681.3 and a volume number 1. Invoke the print() function for this object. **(10)**

Ans:

```
#include<iostream.h>
class publication
{
 char title[40];
 float price;
public:
 void getdata()
 {
            cout<<"Enter publication:";
                    cin>>title;
            cout<<"Enter price";
            cin>>price;
 }
 void print()
 {
            cout<<"\tPublication ="<<title<<endl;
            cout<<"\tPrice="<<price<<endl;
 }
};
 class book:public publication
 {
            int accessionno;
 public:
     void getdata()
  {
              publication::getdata();
              cout<<"Enter Accession number";
              cin>>accessionno;
  }
```

```
  void print()
  {
            publication::print();
            cout<<"Accession number is"<<accessionno;
  }
};
class magazine:public publication
{
          int volumeno;
public:
    void getdata()
  {
            publication::getdata();
            cout<<"Enter Volume number of magazine";
            cin>>volumeno;
  }
  void print()
  {
            publication::print();
            cout<<"Volume number is"<<volumeno<<endl;
  }
};
class journal:public book,public magazine
{
          char name[20];
public:
          void getdata()
          {

                    book::getdata();
                    magazine::getdata();
                    cout<<"enter name of journal";
                    cin>>name;
          }
          void print()
          {
                    cout<<"name of journal is"<<name<<endl;
                    book::print();
                    magazine::print();
          }
};
void main()
{
          journal j;
          j.getdata();
          j.print();
}
```

Q.73 With an example highlight the benefit of operator overloading. **(2)**

Ans:
The operator facilitates overloading of the c++ operators. The c++ operator overloaded to operate on member of its class.

Q.74 Explain the difference between operator member functions and operator non member functions. **(2)**

Ans:
The operator overloaded in a class is known as operator member function. Using friend functions in operator overloading then it becomes operator non member function.

Q.75 Define a class Date. Overload the operators += and <<. **(10)**

Ans:
Date class

```
#include<iostream.h>
#include<conio.h>
class time
    {
     private:  int hr,min,sec;

     public:
            void getdata();
 void operator +=(time a1);
 friend ostream & operator << (ostream &out,time &t)
            {
                    out<<t.hr<<":";
                            out<<t.min<<":";
                            out<<t.sec;
                    return out;
            }
   };

 void time::getdata()
    {
 cout<<"ENTER THE HOURS"<<endl;
 cin>>hr;
 cout<<"ENTER THE MINUTES"<<endl;
 cin>>min;
 cout<<"ENTER THE SECONDS"<<endl;
 cin>>sec;
    }
```

```
void time::operator +=(time a1)
    {

 hr=hr+a1.min;
 min=min+a1.min;
 sec=sec+a1.sec;
 if(sec>=60)
 {
          min++;
          sec=sec-60;
 }
 if(min>=60)
 {
          hr++;
          min-=60;
 }

 }


 void main()
   {
    clrscr();
    time t1,t2;
    t1.getdata();
    t2.getdata();
    t1+=t2;
    cout<<"\n RESULT OF ADDITION"<<endl;
    cout<<t1;
    getch();
   }
```

Q.76 What are abstract classes? How do you define one and how is it useful in a typical library? **(4)**

Ans: A class that contains at least one pure virtual function is considered an abstract class. Classes derived from the abstract class must implement the pure virtual function or they, too, are abstract classes. class Account

```
{
public:
            Account( double d );  // Constructor.
   virtual double GetBalance();         // Obtain balance.
   virtual void   PrintBalance() = 0;   // Pure virtual function.
```

```
private:
   double _balance;
};
```

Abstract classes act as expressions of general concepts from which more specific classes can be derived. You cannot create an object of an abstract class type; however, you can use pointers and references to abstract class types.

Q.77 Define a class City which has name and pincode. Define another class Address which has house no and city of type City. Define the constructor, the destructor and a function print() which prints the address. Create an object of type Address in main() and print it. **(10)**

Ans:

```
#include<iostream.h>
class city
{
public:
 char *name;
 int pin;
public:
 city()
 {
 }
 city(char nm[],int pn)
 {
     name=nm;
     pin=pn;
 }
};
class address
{
public:
 city houseno,cty;
 address(char ct[],int hn)
 {
            cty.name=ct;
            houseno.pin=hn;
 }

   void print()
 {
    cout<<"city name is"<<cty.name;
  cout<<"pincode is"<<houseno.pin;
 }
};
```

```
void main()
{
 address ad("delhi",23),ad1("Rohini",85);
ad.print();
ad1.print();
}
```

Q.78 a. Define a class template Queue with put() and get() operations. Using this create a Queue of integers in main() and add two elements to the queue. **(7)**

b. Implement exception handling for handling both when the queue is empty and when the queue is full. **(7)**

Ans:

```
#include<iostream.h>
const int max=3;
template<class type>
class queue
{
 type qu[max];
 int head,tail,count;
 public:
        class full
        {
        };
        class empty
        {
        };
        queue()
        {
                head=-1;
                tail=-1;
                count=0;
        }
        void put(type var)
        {
                if(count>=max)
                        throw full();
                qu[++tail]=var;
                ++count;
                if(tail>=max-1)
                        tail=-1;
        }
```

```
        type get()
        {
                if(count<=0)
                        throw empty();
                type temp=qu[++head];
                --count;
                if(head>=max-1)
                        head=-1;
                return temp;
        }
};
int main()
{
 queue<float> q1;
 float data;
 char choice='p';
 do
 {
        try
        {
                cout<<"\enter 'x' to exit, 'p'for put,'g'for get:";
                cin>>choice;
                if(choice=='p')
                {
                        cout<<"Enter data value:";
                        cin>>data;
                        q1.put(data);
                }
                if(choice=='g')
                        cout<<"Data="<<q1.get()<<endl;
        }
        catch(queue<float>::full)
        {
                cout<<"Error:queue is full."<<endl;
        }
        catch(queue<float>::empty)
        {
                cout<<"Error: queue is empty"<<endl;
        }
 }while(choice!='x');
 return 0;
}
```

Q.79 Describe the different modes in which files can be opened in C++. **(4)**

Ans: Different modes in which files can be opened in C++.

ios::in	Open file for reading
ios::out	Open file for writing
ios::ate	Initial position: end of file
ios::app	Every output is appended at the end of file
ios::trunc	If the file already existed it is erased
ios::binary	Binary mode

Q.80 Define a class Car which has model and cost as data members. Write functions
(i) to read the model and cost of a car from the keyboard and store it a file CARS.
(ii) to read from the file CARS and display it on the screen. **(10)**

Ans:

```
#include<iostream.h>
#include<fstream.h>
class car
{
 char model[10];
 float cost;
public:
 void read()
 {
            cout<<"\n Enter model of car";
            cin>>model;
            cout<<"\n Enter cost of car";
            cin>>cost;
                    ofstream of("car.dat",ios::app);
        of<<model;
             of<<cost;
             of.close();        }
 void read_file()
 {
            ifstream ifs("car.dat");
            while(ifs)
            {                               ifs>>model>>cost;

            }
            ifs.close();
            cout<<"Model"<<model<<endl;
```

```
            cout<<"Cost"<<cost;
 }};
void main()
{           car c;
 c.read();
 c.read_file();}
```

Q.81 What is the output of the following program segment?

```
float pi = 3.14167234; int i=1, j=2;
cout.fill('$'); cout.ios::precision(5); cout.ios::width(10);
cout<<i*j*pi<<'\n';
```

(2)

Ans: The output is $$$$6.2833.

Q.82 Given the following definition

```
class X { public: int a; };
class Y1 : public X { };
class Y2 : protected X { };
class Y3 : private X { };
```

Point out the errors

```
void f(Y1* py1, Y2* py2, Y3* py3)
{
  X* px = py1;
  py1->a = 7;
  px = py2;
  py2->a = 7;
  px = py3;
  py3->a = 7;
}
```

(2)

Ans:

There ia an error class y3 derived privately from X so it cannot access the public or private members;

Q.83 Given

```
class Confusion { };
```

What is the difference between Confusion C; and Confusion C( ); **(2)**

Ans:

The difference between Confusion C; and Confusion C() is that Confusion C() invoke constructor with an argument whereas Confusion C invoke default constructor;

Q.84 What are the shortcomings of procedure oriented approach? **(8)**

Ans: Programming, using languages such as COBOL, FORTRAN and C is known as procedure oriented programming (POP).The problem with POP is that in this approach things are done in a sequential manner. A number of functions are written for this. The drawbacks of POP are as follows:

i. In POP groups of instructions are written which are executed by the compiler in a serial manner.
ii. POP contains many functions, and important data items are made global so that they are easily accessible by all the functions. Global data are more prone to undesirable changes which can be accidently made by a function
iii. Another drawback is that it "does not model real world problems" very well.
iv. In POP, data moves openly around the system from one function to another.

Functions can transform data into different forms.

Q.85 Given

```
const char *s = "This";
char * const ptr = "That";
```

Point out the errors, if any in the following statements

```
*s = 'A';
*ptr = 'A';
ptr++;
s++;
```

(2)

Ans: the error is l-value specifies const object

Q.86 The output of the following piece of code is

```
for (x=6,y=3; x && x!=y; x--)
        y++;
cout<< "x = "<< x << "y = "<< y;
```

(2)

Ans:

The output is z=3,y=3

Q.87 Write expressions to represent the following:

(i) p is a function whose argument is a pointer to an array of characters and which returns a pointer to an integer.

(ii) p is a function whose argument is a pointer to character and which returns a pointer to an array of ten integers. **(2)**

Ans:

i) int *p(char **arr[])

ii) int *p(char *ch)

Q88 What is encapsulation? What are its advantages? How can encapsulation be enforced in C++? **(6)**

Ans: Encapsulation: The wrapping up of data and functions into a single unit is called encapsulation. The data can only be accessed by the function with which they have been tied up and not by the outside world. It creates an interface between the object's data and the program.
A common use of information hiding is to hide the physical storage layout for data so that if it is changed, the change is restricted to a small subset of the total program. For example, if a three-dimensional point (x, y, z) is represented in a program with three floating point scalar variables and later, the representation is changed to a single array variable of size three, a module designed with information hiding in mind would protect the remainder of the program from such a change.
In object-oriented programming, information hiding reduces software development risk by shifting the code's dependency on an uncertain implementation (design decision) onto a well-defined interface. Through encapsulation, we can provide security to our data function.
Encapsulation can be enforced in C++ using the concept of classes which binds data and function together. Private data and member functions will not be visible to outside world. For example, in the following class two members x, y declared as private can be accessed only by the member functions, not even in main( ) we can call these data members.

```
class XY
 {
        int x,y;
   public:
        void getdata();
      void dispdata();
};
 void  XY::getdata()
{   cout<< "Enter two values\n";
    cin>>a>>b;
     }
  void XY::dispdata()
{  cout<<a<<b;
}

   void main()
     {    XY  xy1,xy2;
          xy1.getdata();
          xy2.dispdata();
     }
```

Q.89 Write a C++ program which reads in a line of text word by word in a loop using *cin* statement and outputs on the screen the words in the reverse order using *cout* statement. For example, if the input is 'Jack Prat could eat no fat', then the output is 'fat no eat could prat Jack'. **(8)**

Ans: A C++ program to read a text and output words of the text in reverse order:

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
void main()
{
 char words[25][15],wo[15];
 int len=0,i=0;
 cout<<endl<<"\nEnter words separated by space and type XXX to terminate \n";
 while (1)
 {
            cin>>wo;
            if (strcmp(wo,"XXX")==0)
                        break;
            strcpy(words[i++],wo);
            if (i==24)break;
 }
 cout<<endl<<"Output in Reverse order\n";
 while (1)
 {
            cout<<words[--i]<<' ';
            if (i==0)break;
 }
 getch();
}
```

Q 90 How are member functions different from other global functions? **(2)**

Ans: A member function and a non –member function are entirely different from each other.
A member function has full access to the private data members of a class. A member function can directly access the data members without taking the help of an object. For calling member function in main(), dot operator is required along with an object. A member function cannot be called without an object outside the class.
A non member function cannot access the private members of a class. Non-member function are called in main( ) without using an object of a class.

Q.91 Define a class Word which can hold a word where the length of the word is specified in the constructor. Implement the constructor and the destructor. **(6)**

Ans:

```
#include<iostream.h>
 #include<conio.h>
 #include<stdio.h>
 #include<string.h>
 class Word
```

```
{
  char *word;
  int length;
  public:
            Word()
            {
               length=0;
               word=new char[length+1];
            }
            Word(char *s)
            {
                     length=strlen(s);
                     word=new char[length+1];
                     strcpy(word,s);

            }
            void display()
            { cout<<word<<"\n";
            }
};
void main()
{
  char *w1="Girl";
  Word word1(w1);
  Word word2("Good");
  word1.display();
  word2.display();
}
```

Q.92 Differentiate and give examples to bring out the difference between

(i) function overloading and function overriding
(ii) function template and inline functions
(iii) default constructor and copy constructor
(iv) public and *private* access specifiers **(3.5 x 4 = 14)**

Ans:

(i). **Function overloading and function overriding:**

Function overloading means the use of the same function name to create functions that perform a variety of different tasks. The functions would perform different operations depending on the argument list in the function call. The correct function to be invoked is selected by seeing the number and type of the arguments but not the function type. For example an overloaded function volume() handles different types of data as shown below:

```
int  volume(int s);
double volume(double r, int h);
long volume(long l, long b,int h);
```

Function overridding is used to describe virtual function redefinition by a derived class. This is useful when we want to have multiple derived classes implementing a base class function. We can put the common code to all of the derived classes in the base class function, and then in each of the derived class functions we can add the code specific to each one, and then just invoke the parent method. It differs from function overloading in the sense that all aspects of the parameters should be same when a overridden function is redefined in derived class.

(ii)**Function template and inline functions**

Function templates are those functions which can handle different data types without separate code for each of them. For a similar operation on several kinds of data types, a programmer need not write different versions by overloading a function. It is enough if he writes a C++ template based function. This will take care of all the data types. An example that finds the maximum value of an array using template is given below:

```
#include<iostream.h>
template<class T>
void max(T a[],T &m,int n)
{
 for(int i=0;i<n;i++)
            if(a[i]>m)
                        m=a[i];
}
int main()
{
 int x[5]={10,50,30,40,20};
 int m=x[0];
 max(x,m,5);
 cout<<"\n The maximum value is : "<<m;
 return 0;
}
```

Inline Functions: An inline function is one for which the compiler copies the code from the function definition directly into the code of the calling function rather than creating a separate set of instructions in memory. Instead of transferring control to and from the function code segment, a copy of the function body may be substituted directly for the function call. In this way, the performance overhead of a function call is avoided. A function is declared inline by using the inline function specifier or by defining a member function within a class or structure definition. The inline specifier is only a suggestion to the compiler that an inline expansion can be performed; the compiler is free to ignore the suggestion.

The following code fragment shows an inline function definition.

```
inline int add(int i, int j) { return i + j;}
```

(iii) **Default constructor and copy constructor**

Dafault constructor: A constructor that accepts no argument is called default constructor. This default constructor takes no parameters, or has default values for all parameters. The default constructor for the class A is A::A(). If no such constructor is defined, then the compiler supplies a default constructor. A default parameter is one where the parameter value is supplied in the definition. For example in the class A defined below 5 is the default value parameter.

```
Class A
{    int value;
  Public:
    A(int param=5)
   {
     value = param;
   }
};
```

A copy constructor is a special constructor in the C++ programming language used to create a new object as a copy of an existing object. This constructor takes a single argument: a reference to the object to be copied. Normally the compiler automatically creates a copy constructor for each class (known as an implicit copy constructor) but for special cases the programmer creates the copy constructor, known as an explicit copy constructor. In such cases, the compiler doesn't create one.

For example the following will be valid copy constructors for class A.

```
A(A const&);
A(A&);
A(A const volatile&);
A(A volatile&);
```

(iv)**Public and Private access specifiers**:

The keywords public and private are visibility labels. When the data members of a class are declared as private, then they will be visible only within the class. If we declare the class members as public then it can be accessed from the outside world also. Private specifier is used in data hiding which is the key feature of object oriented programming. The use of the keyword private is optional. When no specifier is used the data member is private by default. If all the data members of a class are declared as private then such a class is completely hidden from the outside world and does not serve any purpose.

```
class class-name
{
   private:
 int x,y;//No entry to private area
```

```
  private:
 int a,b;//entry allowed to public area
}
```

Q.93 The keyword 'virtual' can be used for functions as well as classes in C++. Explain the two different uses. Give an example each. **(6)**

Ans: Virtual function: virtual functions are important because they support polymorphism at run time. A virtual function is a member function that is declared within the base class but redefined inside the derived class. To create a virtual function we precede the function declaration with the keyword virtual. Thus we can assume the model of "one interface multiple method". The virtual function within the base class defines the form of interface to that function. It happens when a virtual function is called through a pointer.The following example illustrates the use of a virtual function:

```
#include<iostream.h>
using namespace std;
class base
{
  public:
  int i;
  base(int x)
  {
            i=x;
  }
  virtual void func()
  {
            cout<<"\n using base version of function";
            cout<<i<<endl;
  }
};
class derived: public base
{
  public:
  derived(int x):base(x){}
  void func()
      {
            cout<<"\n using derived version";
            cout<<i*i<<endl;
      }
};
class derived1: public base
{
  public:
  derived1(int x): base(x){}
  void func()
```

```
  {
        cout<<"\n using derived1 version";
        cout<<i+i<<endl;
  }
};
int main()
{
  base *p;
  base ob(10);
  derived d_ob(10);
  derived1 d_ob1(10);
  p=&ob;
  p->func();
  p=&d_ob;
  p->func();
  p=&d_ob1;
  p->func();
  return 0;
}
```

Virtual class: The concept of virtual class is very important as far as inheritance is concerned. Basically, the need for a making a class virtual arises when all the three basic types of inheritance exist in the same program- Multiple, multilevel and hierarchical inheritance.

For instance suppose we create a class child which inherit the properties of two other classes parent1 and parent2 and these two parent classes are inherited from a grandparent class. The grandfather class is the indirect base class for child class. Now the class child will have properties of the grandfather class but inherited twice, once through both the parents. The child can also inherit the properties straight from its indirect base class grandfather. To avoid the ambiguity caused due to the duplicate sets of inherited data we declare the common base class as virtual. Those classes that directly inherit the virtual class will be declared as follows:

```
class grandparents
{
  ....
  ....
};
class parent1: virtual public grandparents
{
  ....
  ....
};
class parent2: public virtual grandparents
{
  ....
  ....
};
class child: public parent1,public parent2
{
```

```
    ....//only one copy of grandparents
    ....//will be inherited.
};
```

Q.94 Define a class Array which can hold 10 elements of type int. Define a member function
int Get(int index);
which returns the indexth element of the array if index is between 0 to 9 and throws an exception if index is out of bounds. Catch the exception in the main program and print an error. **(8)**

Ans:

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>

class Array
{
 int no[10],i;
 public:
     Array();
     void getdata();
     int get (int index)
          {
               if (index>=0 && index<10)
                  return no[index];
               else
                 throw index;
          }
};
Array::Array()
{
  for (i=0;i<10;i++)
  no[i]=0;

}
void Array::getdata()
{
               cout<<endl<<"Accepting Input from User"<<endl;
               for (i=0;i<10;i++)
               {

                    cout<<"please enter an element:-> ";
                    cin>>no[i];
               }
          }

void main()
```

```
{
 class Array obj;
 obj.getdata();
 //char ch='y';
 try
 {

            cout<<endl<<"Enter the index;-> ";
            int ind,value=0;
            cin>>ind;
            value=obj.get(ind);
            cout<<endl<<"Value at that index = "<<value;
            getch();
  }
  catch(int m)
  {
                        cout<<endl<<"U have opted for wrong index";
                        getch();
  }
}
```

Q 95 Define a class Bag(duplicate values permitted) of 50 integers. The values of the integers is between 1 and 10. Overload the subscript operator[]. **(6)**

Ans:

```
#include <iostream.h>
class Bag
{
  int val[50];
  int size;
   public:
  Bag();
  Bag(int *x);
  int &operator[](int i)
  { return val[i];
  }
};
Bag::Bag()
{    size=50;
  for(int i=0;i<50;i++)
       val[i]=0;
}
Bag::Bag(int *x)
{  int size=sizeof(x);
  for(int i=0;i<size;i++)
       val[i]=x[i];
}
```

```
void main()
{  Bag b1;
   int x[50],n;
   cout<<"How many values u want to put in bag";
   cin>>n;
   cout<<"Enter values between 1 to 10";
   for(int i=0;i<n;i++)
    cin>>x[i];
    b1=x;
   cout<<"values in the bags\n";
   for(i=0;i<n;i++)
   cout<<x[i]<<"\n";
}
```

Q.96 What is the output of the following program segment?

```
void main(){
      char buffer[80]; cout<< "Enter a line with *: "<<endl;
      cin.get(buffer, 8, '*');
      cout << "The buffer contains " << buffer << endl;}
```

Input : Hello World*

Ans: output:- The buffer contains HELLO W

Q.97 What does the poem.txt contain?

```
void main()
{
ofstream outfile("poem.txt");
outfile << "Mary had a lamb";
outfile.seekp(11, ios::beg);
outfile << "little";
                }
```

Ans: poem.txt contain : Mary had a little lamb

Q.98 Given

```
int * entry = new int [10];
```

Write code to fill the array with 10 numbers entered through the keyboard.

Ans:

```
#include<iostream.h>
#include<conio.h>
void main()
{
 clrscr();
 int *entry=new int[10];
 cout<<"\n Enter ten numbers; ";
```

```
for(int i=0;i<10;i++)
        cin>> entry[i];
for(i=0;i<10;i++)
        cout<<entry[i]<<"\n";
getch();
}
```

Q.99 What is the output of the following program segment?

```
void main() {
   int i = 6;
   double d = 3.0;
   char * str = "OOP";
   cout.setf(ios::left);cout.setf(ios::fixed);
   cout << setfill('#') << setw(5) << str;
   cout << setprecision(1) << i + d;
}
```

Ans: output:- OOP##9

Q.100 What is the output of the following program segment?

```
void main ( )
        { int xyz = 333;
         cout << endl <<
         "The output is \n" << endl
          ; cout << "x\
            y\
        z\
        = "
        << xyz;
}
```

Ans: output:- the output is
xyz=333

Q.101 When do you rethrow any (caught) exception? Give the syntax.

Ans: An exception handler can rethrow an exception that it has caught without even processing it. It simply invokes the throw without passing any arguments.

```
throw;
```

This exception is then thrown to the next try/catch and is caught by a catch statement listed after that enclosing try block.

Q.102 Point out the error and correct it

```
class X( static int a; char b; int star() {return a * b } };
```

(2 x 7)

Ans: the error in:

```
class X( static int a; char b;int star{return a*b;}};
```

is that class X should be followed by an opening curly bracket and not a curve bracket.

Q.103 Assume that the cell users are two kinds – those with a post paid option and those with a prepaid option. Post paid gives a fixed free talk time and the rest is computed at the rate of Rs.1.90 per pulse. Prepaid cards have a fixed talk time.

Define a class Cell_user as a base class and derive the hierarchy of classes. Define member functions and override them wherever necessary to

(i) retrieve the talk time left for each user.

(ii) print the bill in a proper format containing all the information for the post paid user. **(14)**

Ans:

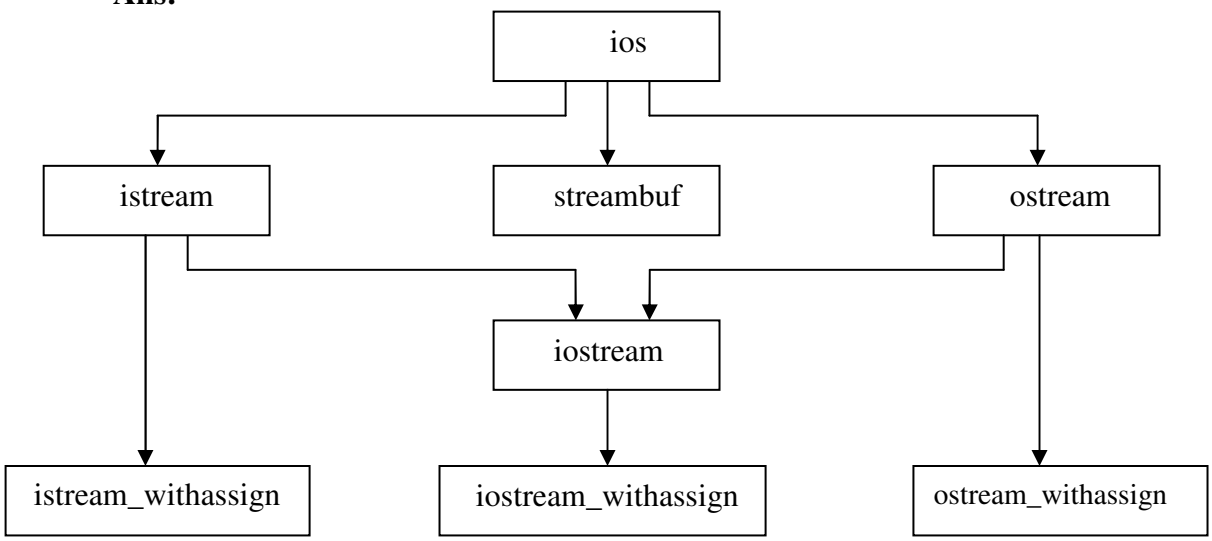
```
class Cell_User
{       protected:
 char cellno[15];
 char name[25], add[50];
 public:
  char plan;
  void getdata();
  void showdata();
};
void Cell_User::getdata()
 {
            cout<<endl<<"Enter your Name:-> ";
            cin.getline(name,25);
            cout<<endl<<"Enter your Address:-> ";
            cin.getline(add,50);
            cout<<endl<<"Enter Cell Number:-> ";
            cin>>cellno;
            cout<<endl<<"Enter Plan ('O' for Post Paid and 'R' for Pre Paid:-> ";
            cin>>plan;
}
void Cell_User::showdata()
{             cout<<"\nName:"<<'\t'<<name;
            cout<<"\nCellno:"<<'\t'<<cellno;
            cout<<"\nAddress:"<<'\t'<<add<<"\n\n";
            cout<<"Plan:"<<'\t'<<plan;
}
class post : public Cell_User
{
 int freepulse, usedpulse,extra;
 public:
  post()
  {
   freepulse=50;
   usedpulse=0;
```

```
  }
  void getdata()
  {cout<<"\nEnter used Time for this user in Pulses:-> ";
   cin>>usedpulse;
  }
  void showdata();
};
void post::showdata()
  {
                cout<<"\nPostpaid User\n";
                cout<<"\nFreepulse="<<freepulse;
                cout<<"\nUsedpulse="<<usedpulse;
                extra=usedpulse-freepulse;
                cout<<"\n Extra pulses:"<<'\t'<<extra;
                if (extra>0)
                {
                     cout<<"\nMonthly Rent="<<600;
                          cout<<"\nCall charges="<<extra*1.90;
                          cout<<"\nTotal="<<600+extra*1.90;
                }
              else
              {  cout<<"\nMonthly Rent="<<600;
                    cout<<"\nNo extra charges\n";
              }
        }
class pre : public Cell_User
{
 int talktime, usedtime,lefttime;

 public:

          pre()
          {
                talktime=50;
                usedtime=0;
          }
          void getdata()
          {
                cout<<"\nEnter used Time for this user in Pulses:-> ";
                cin>>usedtime;
          }
          void showdata()
          {
                cout<<"\nFree talk time="<<talktime;
                cout<<"\nUsed talk time"<<usedtime;
                lefttime=talktime-usedtime;
                if(lefttime>0)
```

```
        cout<<"\n Time left:"<<'\t'<<lefttime;
                        else
                        cout<<"\n Please recharge your coupan\n";
                }
        };
        void main()
        {       clrscr();
                Cell_User c;
         c.getdata();
         if(c.plan=='O')
          {
           post obj1;
           obj1.getdata();
           c.showdata();
                obj1.showdata();
          }
          if(c.plan=='R')
          {       pre obj2;
                obj2.getdata();
                c.showdata();
                obj2.showdata();
          }
          getch();
}
```

Q.104 Explain the I/O stream hierarchy in C++. **(4)**

Ans:



The above hierarchies of classes are used to define various streams which deal with both the console and disc files. These classes are declared in the header file iostream. ios is the base for istream, ostream and iostream base classes. ios is declared as a virtual base class so that only one copy of its members are inherited by the iostream. The three classes istream_withassign,

ostream_withassign and iostream_withassign are used to add assignment operators to these classes.

Q.105 Define a class License that has a driving license number, name and address. Define constructors that take on parameter that is just the number and another where all parameters are present. **(6)**

Ans:

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>
#include<fstream.h>
#include<stdlib.h>
class License
{
 long dlno;
 char name[25],add[50];
 public:
        License()
        {
                dlno=0;
                strcpy(name, " ");
                strcpy(add, " ");

        }
        License(long no)
        {
                dlno=no;
                strcpy(name," ");
                strcpy(add, " ");
        }
        License(long no, char n[25], char ad[50])
        {
                dlno=no;
                strcpy(name,n);
                strcpy(add,ad);

        }

        void getdata()
        {
                cout<<endl<<"Enter your name:-> ";
                cin>>name;
                cin.get();
                cout<<endl<<"Enter your address:-> ";
                cin.getline(add,50);
```

```
                cout<<endl<<"Enter License Number:-> ";
                cin>>dlno;
        }
        void showdata()
        {
                cout<<endl<<dlno<<'\t'<<name<<'\t'<<add;
        }
};
```

Q.106 Create a file and store the License information. **(4)**

Ans:

```
void main()
{
 class License obj1(101),obj2(102,"Ramesh","Laxminagar");
 fstream file;
 file.open("List.TXT",ios::in | ios::out);
 if (file==NULL)
 {
        cout<<endl<<"Memory Allocation Problem.";
        getch();
        exit(1);
 }
 class License obj;
 obj.getdata();
 file.write((char *) & obj, sizeof(obj));
 file.seekg(0);
 cout<<endl<<"Following informations are stored in the file:\n";
 while(1)
 {
        file.read((char *) &obj, sizeof(obj));
        if (file)
                obj.showdata();
        else
                break;
 }
}
```

Q.107 Define a class template Pair which can have a pair of values. The type of the values is the parameter of the template. Define a member function Add() which adds the two values and returns the sum. **(10)**

Ans:

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
template <class T>
```

```
class pair
{
 T val1, val2, val3;
 public:
        void Add()
        {
                val3=val1 + val2;
        }
        void getdata()
        {
                cout<<endl<<"Enter two input:-> ";
                cin>>val1>>val2;
        }
        void showdata()
        {
                cout<<val3;
        }
};

void main()
{
 pair <int> obj1;
 cout<<endl<<"Enter two integer input.\n";
 obj1.getdata();
 obj1.Add();
 obj1.showdata();
 pair <float> obj2;
 cout<<endl<<"Enter two float input.\n";
 obj2.getdata();
 obj2.Add();
 obj2.showdata();
 getch();
}
```

Q.108 Check if the template works for char* where Add() must concatenate the two strings. If not find a solution. **(4)**

Ans:

It will not work for char * where Add() must concatenate the two strings.
This can be done by overloading the function add() as

```
void Add(char *str1, char *str2)
{
   char *str3;
    strcpy(str3,str1);
    strcat(str3,str2);
    cout<<str3;
}
```

Q.109 Write short notes on
(i) Constructor
(ii) Stream
(iii) Access control.
(iv) Friend. **(3.5 x4)**

Ans:

i) **Constructor:**

A constructor is a member function which has the same name as at that of the class. It is used to initialize the data members of the class. A constructor is automatically invoked by the compiler when an object is created. A constructor which does not take any arguments is called default constructor.

The restrictions applied to a constructor are:

- A constructor cannot have any return type as it does not return any value.
- A constructor should be declared in the public part of the class.
- A derived class can call the constructor of the base class but a constructor is never inherited.
- A constructor function cannot be made virtual.
- That object which has a constructor cannot be used as a member of the union.

C++ provides us with very helpful feature i.e., copy constructor. It is the mechanism in which we can declare and initialize an object from another both belonging to the same class. For example, the following statement will create an object obj2 and also initialize it by taking values from obj1.

integer obj2(obj1);

We can also use the following instead the above for the same purpose.

integer obj2=obj1;

This process of initializing objects using copy constructor is known as “copy initialization”.

Another statement obj2=obj1 will not assign the values using the copy constructor. Although the statement is valid this can be done using operator overloading.

A copy constructor accepts a reference to an object of the same class as an argument. We cannot pass an argument by value to a copy constructor.

The following is an example program:

```
#include<iostream.h>
class item
{
                    int code;
  public:
                    item(){}
                    item(int n){code=n;}
```

```
                item(item & a)
                {
                        code=a.code;
                }

                void show(void)
                {cout<<code;}
};
int main()
{
                item obj(12);
                item obj2(obj1);
                item obj3=obj1;
                item obj4;
                obj4=obj1;//copy constructor not called
                cout<<"\n code of obj1 : ";
                obj1.show();
                cout<<"\n code of obj2 : ";
                obj2.show();
                cout<<"\n code of obj3 : ";
                obj3.show();
                cout<<"\n code of obj4 : ";
                obj4.show();
        }
```

ii) Stream: Stream is defined as an interface supplied by the I/O system to the programmer and that which is independent of the actual device being used. Stream is actually a sequence of bytes. These are basically of two types: one is when it is a source through which input can be taken and the other is when it acts as a destination to which output can be sent. The source stream is called the input stream and the destination stream is called the output stream. There are several streams likewise cin is an input stream which extracts input from the keyboard and cout is the output stream which inserts the ouput to the screen. The three streams used are:-

- istream(input stream).this stream inherits the properties of ios and declares input functions such get(), getline() and read().the extraction operator used for this >>
- ostream(output stream). This also inherits properties from ios and declared output functions such as put() and write(). It uses the insertion operator <<.
- Iostream(input/output stream). This inherits properties from ios istream and ostream and through multiple inheritance contains all the input output functions.

iii). **Access control:** This is done using three keywords public, private and protected. These keywords are visibility labels. When the data members of a class are declared as private, then they will be visible only within the class. If we declare the class members as public then it can be accessed from the outside world also. Private specifier is used in data hiding which is the key feature of object oriented programming. The use of the keyword private is optional. When no specifier is used the data member is private by default. If all the data members of a class are declared as private then such a class is completely hidden from the outside world and does not serve any purpose.

```
class class-name
{
   private:
    int x,y;//No entry to private area
   private:
    int a,b;//entry allowed to public area
}
```

These access specifiers determine how elements of the base class are inherited by the derived class. When the access specifier for the inherited base class is public, all public members of the base class become public members of the derived class. If the access specifier is private, all public members of the base class become private members for the derived class. Access specifier is optional. In case of a derived 'class' it is private by default. In case of a derived 'structure' it is public by default.

The protected specifier is equivalent to the private specifier with the exception that protected members of a base class are accessible to members of any class derived from the base. Outside the base or derived class, protected members are not accessible. The protected specifier can appear anywhere in the class.

When a protected member of a base class is inherited as public by the derived class, it becomes protected member of the derived class. If the base is inherited a private, the protected member of the base becomes private member of the derived class.

iv) **Friend:**

Friend is keyword in C++ which is used to declare friend function and friend classes. It can access the private data members of the class it has to take the help of the object of that class. A friend function is not the member of the class. When we create an object memory is not located for the friend function. It is defined like a normal function. While calling a friend function we not use object as friend function is not the part of the object. Thus is called like any normal non member function. It also helps in operator overloading.

The following is an example program:

```
#include<iostream.h>
using namespace std;
class one;

class two
{
    int a;
  public:
    void setvalue(int n){a=n;}
    friend void max(two,one);
};
class one
{
    int b;
  public:
    void setvalue(int n){b=n;}
    friend void max(two,one);
};
void max(two s,one t)
{
    if(s.a>=t.b)
                cout<<s.a;
    else
                cout<<t.b;
}
int main()
{
    one obj1;
    obj1.setvalue(5);
    two obj2;
    obj2.setvalue(10);
    max(obj2,obj1);
    return 0;
}
```

When we declare all the member functions of a class as friend functions of another then such a class is known as a friend class.

```
class Z
{
friend class X;//all member functions af X are friends of Z
}
```

Q.110 What is the difference between passing a parameter by reference and constant reference? **(2)**

Ans: By passing a parameter by reference, the formal argument in the function becomes alias to the actual argument in the calling function. Thus the function actually works on the original data. For example:

```
void swap(int &a, int &b)//a and b are reference variables
{
 int t=a;
 a=b;
 b=t;
}
```

The calling function would be:

```
swap(m,n);
```

the values of m and n integers will be swapped using aliases.

Constant Reference: A function may also return a constant reference. Example:

```
int & max(int &x,int &y)
{
 If((x+y)!=0)
          return x;
 return y;
}
```

The above function shall return a reference to x or y.

Q.111 Define the following terms and give an example to show its realization in C++

(i) Encapsulation (ii) Class variables and class functions
(iii) Repeated inheritance (iv) Overloading **(3.5 x 4)**

Ans:

(i) **Encapsulation:** The term encapsulation is often used interchangeably with information hiding. Not all agree on the distinctions between the two though; one may think of information hiding as being the principle and encapsulation being the technique. A software module hides information by encapsulating the information into a module or other construct which presents an interface.

A common use of information hiding is to hide the physical storage layout for data so that if it is changed, the change is restricted to a small subset of the total program. For example, if a three-dimensional point (x,y,z) is represented in a program with three floating point scalar variables and later, the representation is changed to a single array variable of size three, a module designed with information hiding in mind would protect the remainder of the program from such a change.

(ii) **Class variables and class functions:** Static variables are associated with the class rather than any class objects, so they are known as class variables. The type and scope of each class variable must be defined outside the class definition as they are stored separately rather than as a part of an object.

Static functions can have access to only other static members (functions and variables) declared in the same class. These functions are known as class functions. A class function is called by using the class name instead of its objects like

Class-name:: function-name;

(iii)**Repeated Inheritance**: C++ requires that the programmer state which parent class the feature to use should come from. C++ does not support explicit repeated inheritance since there would be no way to qualify which superclass to use. Repeated inheritance means not being able to explicitly inherit multiple times from a single class. Multiple Inheritance usually corresponds to this-a relationship. It is a common mistake to use multiple inheritance for has-a relationship. For example, a Filter circuit may consist of Resistors, Capacitors and Inductors. It'll be a mistake to declare Filter as a subclass of resistor, capacitor and inductor (multiple inheritance). Instead, it is better to declare component objects of resistors, capacitors, and inductors within the new Filter class, and reference the component methods indirectly through these component objects.

(iv)**Overloading:** The process of making a function or an operator behave in different manners is known as overloading. Function overloading means the use of same function name to perform a variety of different tasks. The correct function to be invoked is selected by seeing the number and type of the arguments but not the function type.

Operator overloading refers to adding a special meaning to an operator. The operator doesn't lose its original meaning but the new meaning is added for that particular class. For example, we can add two strings as:

String1+String2=string3;

"good"+ "girl"= "goodgirl";

An example program of function overloading:

```
//Function volume is overloaded three times.
#include<iostream.h>
using namespace std;

int vol(int);
double vol(double,int);
long vol(int,int,int);
```

```
int main()
{
  cout<<volume(2)<<"\n";
  cout<<volume(2.5,8)<<"\n";
  cout<<volume(2,3,5)<<"\n";
  return 0;
}

int volume(int s)
{
  return(s*s*s);
}

double volume(double r,int h)
{
  return(3.14519*r*r*h);
}

long volume(int l,int b,int h)
{
  return(l*b*h);
}
```

The output would be:

```
8
157.26
30
```

Q.112 Define a class Point which is a three dimensional point having x, y and z coordinates. Define a constructor which assigns 0 if any of the coordinates is not passed. **(6)**

Ans:

```
#include <iostream.h>
 #include<conio.h>
 class Point {
    int X;
    int Y;
    int Z;
 public:
```

```
  Point() {X=Y=Z=0;}
  Point(int x, int y, int z)
  { X=x; Y=y;Z=z;}
  int getX(){ return X;}
  int getY(){return Y;}
  int getZ(){return Z;}
};
```

Q.113 Define a class Sphere which has a radius and a center. Use the class Point defined in (Q 112) to define the center. Define the constructor. Define a member function Volume (which is given as πr^3) which computes the volume. Use the class in main() to create an instance of sphere. Compute the volume and print it.

(10)

Ans:

```
class Sphere
{
  int radius;
  Point center;
  public:
   Sphere(){}
   Sphere(int rad,Point &p)
   {  radius=rad;
     center=p;
   }
   void showdata()
   {  cout<<"Radius="<<radius;

cout<<"\nCenter="<<"("<<center.getX()<<","<<center.getY()<<","<<center.get
Z()<<")";
    }
  float volume()
  {  return (3.14*radius*radius*radius);
  }
};
void main()
{  clrscr();
  Point YourPoint();
  Point p(3,4,5);
  Sphere s1(1,p);
```

```
    s1.showdata();
    float f=s1.volume();
    cout<<"\nVolume="<<f;
    getch();
}
```

Q.114 What is the access specification that should precede a ***friend*** function? **(2)**

Ans: The access specifier that should precede a friend function can be either public or private. The meaning and working of the friend function is not altered by using any of the two access specifiers.

Q.115 Define a class Vector which can hold 20 elements of type *int*. Write a member function read() which reads the values from the input stream and assigns them to the Vector object. **(6)**

Ans:

```
#include<conio.h>
#include<iostream.h>
const size = 20;
class vector
{
 int v[size];
 public:
 vector();
 void read();
 vector(int *x);
 friend vector operator +(vector a, vector b);
 friend istream & operator >> (istream &, vector &);
 friend ostream & operator << (ostream &, vector &);
};
vector :: vector()
{
 for(int i=0; i<size ; i++)
  v[i] = 0;
}
vector :: vector(int *x)
{
 for(int i=0; i<size; i++)
     v[i] = x[i];
}

void vector :: read()
{
 int x[20];
 cout<<"\nEnter 20 values";
```

```
 for(int i=0;i<size;i++)
     cin>>x[i];
 for(i=0; i<size; i++)
     v[i] = x[i];
}
```

Q.116 Overload the operator + for the class Vector defined in Q115 above which adds two vectors. Define the addition operator as a ***friend*** function. The result must be returned as an object of the class Vector.

(8)

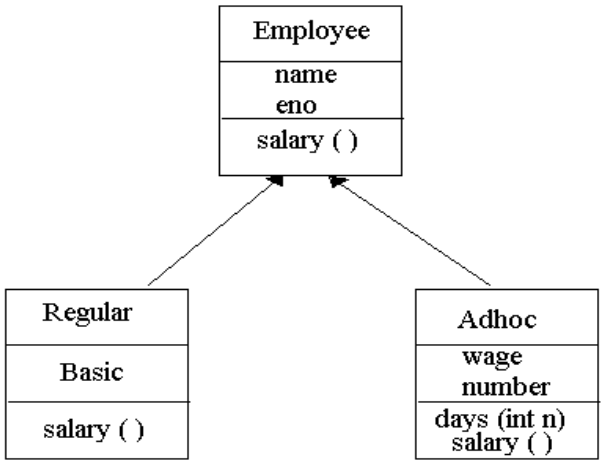
Ans:

```
vector operator +(vector a, vector b)
{
 vector c;
 for(int i=0;i<size;i++)
    c.v[i] = a.v[i] + b.v[i];
 return c;
}
istream & operator >> (istream&x, vector&b)
{
 for(int i=0;i<size;i++)
    x >> b.v[i];
    return(x);
}
ostream & operator << (ostream&dout, vector&b)
{  dout<<"("<<b.v[0];
 for(int i=1;i<size;i++)
    dout <<  " , " << b.v[i];
    dout <<" ) ";
    return(dout);
}

void main()
{
 clrscr();
 vector m;
 m.read();
 vector n;
 cout<< "enter elements of vector n"<<'\n';
 cin>>n;
 cout<<'\n';
 cout<<"m = "<<m<<'\n';
 cout<<"\nn="<<n<<'\n';
```

```
 vector p;
 p = m+n;
cout<<'\n';
 cout<<"p = "<<p<<'\n';
getch();
}
```

Q.117 An organization has two types of employees: regular and adhoc. Regular employees get a salary which is basic + DA + HRA where DA is 10% of basic and HRA is 30% of basic. Adhoc employees are daily wagers who get a salary which is equal to Number * Wage.

(i) Define the classes shown in the following class hierarchy diagram:



(ii) Define the constructors. When a regular employee is created, basic must be a parameter. When adhoc employee is created wage must be a parameter.

(iii) Define the destructors.

(iv) Define the member functions for each class. The member function days ( ) updates number of the Adhoc employee.

(v) Write a test program to test the classes. **(4+4+2+4+2)**

Ans:

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

class Employee
{
 char name[25];
 int eno;
 public:
```

```
 Employee()
 {
        strcpy(name, " ");
        eno=0;
 }
 void getdata()
 {
        cout<<endl<<"Name of Employee:-> ";
        cin>>name;
        cout<<endl<<"Employee Number:-> ";
        cin>>eno;
 }

 void salary()
 {
        cout<<endl<<eno<<'\t'<<name;
 }

 ~Employee()
 {

 }
};
class Regular : public Employee
{
 float Basic;
 public:
        Regular(float bs)
        {
                Basic=bs;
        }

        void getdata()
        {
                Employee::getdata();
        }
        void salary()
        {
                //call Base showdata();
                Employee::salary();
                cout<<endl<<"Basic = "<<Basic;
                cout<<endl<<"DA    = "<<Basic*0.1;
                cout<<endl<<"HRA   = "<<Basic*0.3;
                cout<<endl<<"Total Payble = "<<(Basic + (Basic*0.1) + (Basic*0.3));

        }
```

```
        ~Regular()
        {
        }
};
class Adhoc : public Employee
{
 float wage;
 int number;
 public:
        Adhoc(float wg)
        {
                wage=wg;
                number=0;
        }

        void days(int n)
        {
                number+=n;
        }

        void salary()
        {
                //call Base class Showdata
                Employee::salary();
                cout<<endl<<"Total Payble = "<<wage*number;
        }
        void getdata()
        {
                Employee::getdata();
        }
        ~Adhoc()
        {
        }
};
void main()
{
 class Regular robj(5000);
 class Adhoc aobj(65);
 robj.getdata();
 aobj.getdata();
 aobj.days(25);
 cout<<endl<<"Details of Regular Employee1\n";
 robj.salary();
 cout<<endl<<"Details of Adhoc Employee1\n";
 aobj.salary();
 getch();
}
```

Q.118 Write a function template for the function Power ( ) which has two parameters base and exp and returns $base^{exp}$. The type of base is the parameter to the template and exp is *int*. If exp is negative then it must be converted to its positive equivalent. For example, 2^3 and 2^{-3} must both return 8. **(6)**

Ans:

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<math.h>
template <class T>
class math
{
  int exp;
  T base,s;
  public:
  math(T x, int y)
  {
            exp=abs(y);
            base=x;
            s=1;
  }
  T Power()
  {
     s=pow(base,exp);
     return s;
  }
};
```

Q.119 Instantiate the template function defined above for *int* type. **(2)**

Ans:

```
void main()
{       clrscr();
        math <int>obj1(10,-2);
        cout<<endl<<"Value= "<<obj1.Power();
        getch();
}
```

Q.120 What is exception handling? **(2)**

Ans: Exception Handling:

Using exception handling we can more easily manage and respond to run time errors. It helps us in coping up with the unusual predictable problems that arise while executing a program.

Q.121 What are the keywords on which exception handling is built? Explain each one of them. Give an example. **(6)**

Ans: C++ exception handling is built upon 3 keywords: try, catch and throw.

- The 'try' block contains program statements that we want to monitor for exceptions.
- The 'throw' block throws an exception to the 'catch' block if it occurs within the try block.
- The 'catch' blocks proceeds on the exception thrown by the 'throw' block.

When an exception is thrown, it is caught by its corresponding catch statement which processes the exception. There can be more that one catch statement associated with a try. The catch statement that is used is determined by the type of the exception. An example illustrates the working of the three blocks.

```
#include<iostream.h>
using namespace std;
int main()
{
 cout<<"\n Start "<<endl;
 try
 {
        cout<<"\n Inside try block "<<endl;
        throw 10;
        cout<<"\n this will not execute ";
 }
 catch(int i)
 {
        cout<<"\n catch number "<<endl;
        cout<<i<<endl;
 }
 cout<<"End";
 return 0;
}
```

Q.122 Explain the following:

(i) ios class (ii) setf( ) function
(iii) rdstate() function (iv) writing of objects to files **(8)**

Ans:

i. **ios class:** ios is the base class meant for istream(input sream), ostream(output stream) and iostream(input/output stream).ios is declared as the virtual base class so that only one copy of its members are inherited by the iostream.

ii. **setf() function**: This function is used to specify format flags that can control the form of output display (such as left-justification) and (right justification). The syntax is:

setf(arg1,arg2);

where arg1 is a formatted flag and arg2 is a bit field.

iii. rdstate() function:
ios::rdstate() const;
This function returns the current internal error state flags of the stream. The internal error state flags are automatically set by calls to input/output functions to signal certain types of errors that happened during their execution. It does not contain any parameter.

iv. Writing objects to a file: C++ allows us to read from and write to the disk files objects directly. The binary function write() is used to write object to files. The function write copies a class object from memory byte to byte with no conversion. It writes only the data members and the member functions.

Q.123 Write a program which asks for a file name from the keyboard, opens a file with that name for output, reads a line from the keyboard character by character and writes the line onto the file. **(8)**

Ans:

```
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
#include<fstream.h>
#include<stdlib.h>
void main()
{
 cout<<endl<<"Enter the file name for output:-> ";
 char name[25];
 cin>>name;
 ofstream out(name);

 if (out==NULL)
 {
        cout<<endl<<"\nUnable to open file.";
        getch();
        exit(1);
 }

 char ch;
 cout<<endl<<"Press 'T' to terminate input.\n";

 while(1)
 {
        cin.get(ch);
        if (ch=='T')break;
        out.put(ch);
 }
```

```
out.close();
ifstream in(name);
if (in==NULL)
{
        cout<<endl<<"\nUnexpected Error.";
        getch();
        exit(1);
}
cout<<endl<<"U have entered following\n";
while(in)
{
        in>>ch;
        cout<<ch;
}
in.close();
cout<<endl<<"Thanks";
getch();
}
```

Q.124 Differentiate between

(i) int* q = p; and n = *p;
(ii) Constructor and destructor
(iii) Interface and implementation of a class
(iv) *public* and *private* member of a class
(v) Overloaded pre-increment operator and overloaded post increment operator
(vi) Abstract and concrete class
(vii) *get* ( ) and *getline* ( ) function
(viii) Composition and inheritance **(16)**

Ans:

(i) int* q=p;
the value of p will be assigned to pointer q which will be treated as an address.
int n=*p;
in this an integer variable n is initialized with the value stored at the address to which the pointer p points.

(ii) A constructor is a member function which is used to initialize the data members of a class. It has the same name as that of the class. It does not have any return type and is invoked automatically on the creation of an object.
test() //constructor for the class test
A destructor is a member function that is used to destroy the objects that's have been created by the constructor. It frees the memory space for future use. It also has no return type but its declaration is preceded by a tilde.
~test(); //destructor function

(iii) The keywords public and private are visibility labels. When the data members of a class are declared as private, then they will be visible only within the class. If we declare the class members as public then it can be accessed from the outside world also. Private specifier is used in data hiding which is the key feature of object oriented programming. The use of the keyword private is optional. When no specifier is used the data member is private by default. If all the data members of a class are declared as private then such a class is completely hidden from the outside world and does not serve any purpose.

(iv) In earlier version of C++ when an increment operator was overloaded there was no way of knowing whether the operator precedes or follows its operand. That is assuming that the two statement ob++ and ++ob had identical meanings. However the modern specification for C++ has defined a way through which compiler distinguishes between these operator. For this two versions were used. One defined as

return-type class-name :: operator #( ) and the second is defined as
rerurn-type class-name :: operator #(int notinuse). In (int notinuse) zero value is passed.

(v) In C++ an abstract class is one which defines an interface, but does not necessarily provide implementations for all its member functions. An abstract class is meant to be used as the base class from which other classes are derived. The derived class is expected to provide implementations for the member functions that are not implemented in the base class. A derived class that implements all the missing functionality is called a concrete class.

(vi) get() function extracts a character from the stream and returns its value (casted to an integer).The C++ string class defines the global function getline() to read strings from an I/O stream. The getline() function reads a line from is and stores it into s. If a character delimiter is specified, then getline() will use delimiter to decide when to stop reading data. Example: getline(cin, s);

(vii)The difference between inheritance and composition is the difference between “is-a” and “has-a”. For example, we would not design a “car” object to be inherited from “engine”, “chassis”, “transmission”, and so on (“car is-a engine” makes no sense). We would “compose” the car object from its constituent parts (“car has-a engine” makes a lot of sense). Once we start thinking in terms of “is-a” versus "has-a", it's a lot easier to do proper design.

Q.125 What is the wrong with the following program?

```
#include <iostream.h>
void main ( )
{
   class Test {
```

```
    int x;
  public: int getX ( ) {return x;}
          void setX ( int a) { x = a;}
          void showX{ cout << getX ( ) << endl;}
 };
}
```

Ans:
Parameter names are used with function body.

Q.126 What is the output of the following program, if any?

```
#include <iostream.h>
class A {
   int a;
  public: A( int aa) { a = aa;}
          void print ( ) { cout << " A = " << a; }
};
class B : public A {
     int b;
public: B ( int aa, int bb) : A (aa) { b = bb;}
      void print ( ) { cout << " B = " << b; }
};
void main ( )
{ A objA(10);
  B objB(30, 20);
  objA = obj B;
  objA.print ( );
}
```

Ans: A=30

Q.127 What is wrong with the following code?

```
int *p = new int;
int *p = new int;
cout << "p = "  << p << ", q = " << q << ", p+q = " << p+q << endl;
```

Ans: p and q both the variables are not declared or initialized.

Q.128 How many times is the copy constructor called in the following code?

```
class copy_const{
};
copy_const f(copy_const u)
{ copy_const v(u);
  copy_const w = v;
   return w;
```

```
}
main ( )
{
  copy_const x;
  copy_const y = f (f (x));
                                    }
```

(4 x 4)

Ans: Copy constructor is not called in the given function.

Q.129 Discuss the role of inheritance in object-oriented programming. What is public, private and protected derivation? **(8)**

Ans:

Inheritance: it is the property by which one class can be derived from another class such that it acquires the properties of its base class. Just like a child inherits the properties of its parents in the same way OOP allows us to reuse the members and functions of class to be derived by another class. There are many different kinds of inheritance like multiple inheritance, multilevel inheritance and hierarchical inheritance.

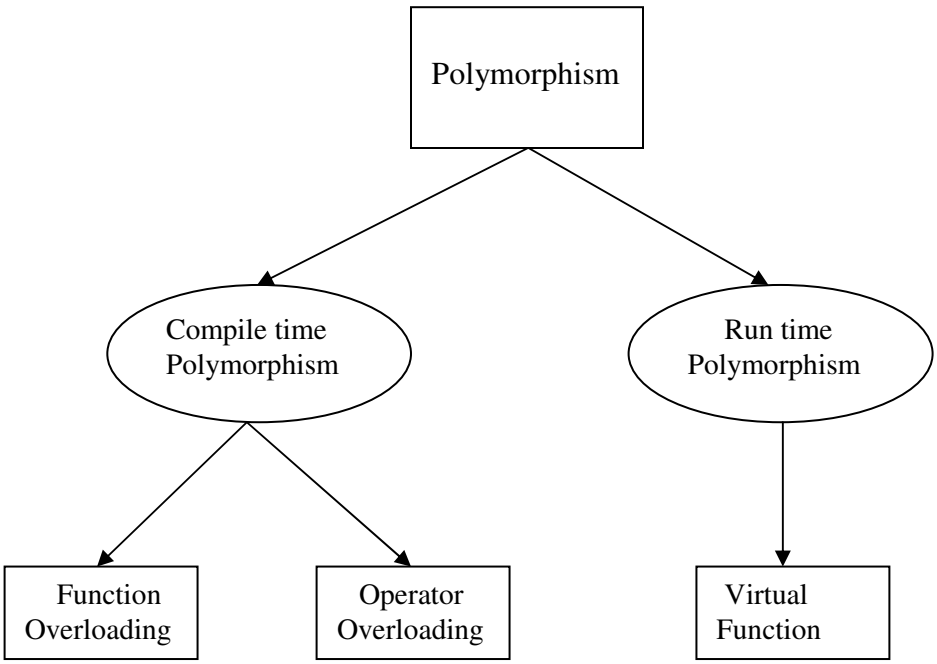
The visibility mode specified while declaring a derived class plays a very important role in the access control of the data members of the class. The visibility mode is one of the three keywords: public, private and protected. These access specifiers determine how elements of the base class are inherited by the derived class. When the access specifier for the inherited base class is public, all public members of the base class become public members of the derived class. If the access specifier is private, all public members of the base class become private members for the derived class. Access specifier is optional. In case of a derived 'class' it is private by default. In case of a derived 'structure' it is public by default.

The protected specifier is equivalent to the private specifier with the exception that protected members of a base class are accessible to members of any class derived from the base. Outside the base or derived class, protected members are not accessible. The protected specifier can appear anywhere in the class.

When a protected member of a base class is inherited as public by the derived class, it becomes protected member of the derived class. If the base is inherited a private, the protected member of the base becomes private member of the derived class.

Q.130 Explain the meaning of polymorphism. Describe how polymorphism is accomplished in C++ taking a suitable example? **(8)**

Ans:



Polymorphism is the property of representing one operation into many different forms. One operation may express different in different situations. The process of making a function or an operator behave in different manners is known as overloading the function or the operator.
Polymorphism is one of the most useful features of object oriented programming. It can be achieved both at run time and at compile time. At compile time polymorphism is achieved using function overloading and operator overloading. At the time of compilation the compiler knows about the exact matching as to which function to call or invoke. This is known as compile time polymorphism or early binding.
Polymorphism can also be achieved at run time. This is done using the concept of virtual functions. Which class function is to be invoked is decided at run time and then the corresponding object of that class is created accordingly. This is also called as late binding.
The following example program explains how polymorphism can be accomplished using function overloading.

```
//Function volume is overloaded three times.
#include<iostream.h>
int volume(int);
double volume(double,int);
long volume(long,int,int);
```

```
int main()
{
   cout<<volume(10)<<"\n";
   cout<<volume(2.5,8)<<"\n";
   cout<<volume(100L,75,15)<<"\n";
   return 0;
}

int volume(int s)
{
   return(s*s*s);
}

double volume(double r,int h)
{
   return(3.14519*r*r*h);
}

long volume(long l,int b,int h)
{
   return(l*b*h);
}
```

The output would be:

```
1000
157.26
112500
```

Q.131 Define a class **Fraction** whose objects represent rational numbers (i.e. fractions). Include integer data members **num** and **den** for storing the numerator and denominator respectively.

a. Write the following functions for fraction:

(i) parameterised constructors with one and two arguments.
(ii) copy constructors.
(iii) a function, eval-fract() for evaluating the value of the rational number.
(iv) a function, invert() for inverting the given rational number.

Ans:

a.

```
#include<conio.h>
#include<iostream.h>

class fract
{
 int num;
```

```
 int den;
 public:
fract(int a);
 fract(int a,int b);
 fract(fract &);
 float eval_fract();
 void invert();
 void display();
};

fract::fract(int a)
{

  num=den=a;
}

fract :: fract(int a,int b)
{

  num=a;
  den=b;
}

fract :: fract(fract &f)
{
  cout<<"copy constructor called\n";
    num=f.num;
  den=f.den;
}

float fract :: eval_fract()
{
  float res;
  res=(float)num/den;
  return res;
}

void fract ::invert()
{
  int t=num;
  num=den;
  den=t;
}
```

```
void fract ::display()
{

 cout<<num<<"/"<<den;
}

void main()
{
 clrscr();
 fract f1(2);
 f1.display();
 cout<<"\n";
 fract f3(2,3);
 fract f2=f3;
 float f=f2.eval_fract();
 cout<<"The value of fraction is:"<<f<<"\n";
 f2.invert();
 f2.display();
 getch();
}
```

b. Overload >> and << operators for reading and printing the fraction object from the keyboard. **(2+8+6)**

Ans: Function declared inside the class fract for overloading << and >> operator:

```
friend istream & operator >> (istream &, fract &);
friend ostream & operator << (ostream &, fract &);
```

Functions defined outside the class:

```
istream & operator >> (istream&x, fract&b)
{
   x >> b.num>>b.den;
   return(x);
}
ostream & operator << (ostream&dout, fract&b)
  {  dout << b.num;
   dout <<" / "<<b.den;
   return(dout);
}
```

Statements in main() to invoke overloading of << and >> are:

```
cout<< "enter num and den of the fraction f"<<'\n';
 cin>>f;
 cout<<'\n';
 cout<<"fraction = "<<f<<'\n';
```

Q.132 What do you mean by static variable and static function? Give an example? **(5)**

Ans: Static class member variables are used commonly by the entire class. It stores values. No different copy of a static variable is made for each object. It is shared by all the objects. It is just like the C static variables.

- It has the following characteristics:
- On the creation of the first object of the class a static variable is always initialized by zero.
- All the objects share the single copy of a static variable.
- The scope is only within the class but its lifetime is through out the program.

A static function is just like a static variable. a static function has the following properties:

- A function can access only the static variable where both belong to the same class.
- A static function is called using the class name directly and not the object like

class-name :: function-name;

The example given below demonstrates the use of static variable 'count' and static function 'display()':

```
#include<iostream.h>
using namespace std;
class demo
{
  int num;
  static int count;
  public:
  void set(void)
  {
        num=++count;
  }
  void show(void)
  {
        cout<<"Number : "<< num<<"\n";
  }
  static void display(void)
  {
        cout<<"count : "<<count<<"\n";
  }
};
int demo::count;
int main()
{
  demo d1,d2;
  d1.set();
  d2.set();
  demo::display();//accessing static function
  demo d3;
  d3.set();
  d1.show();
```

```
  d2.show();
  d3.show();
  return 0;
}
```

Q.133 Write a template function to find the maximum number from a template array of size N. **(8)**

Ans:

```
using namespace std;
template<class T>
void max(T a[],T &m, int n)
{
  for(int i=0;i<n;i++)
        if(a[i]>m)
              m=a[i];
}

int main()
{
  int x[5]={10,50,30,40,20};
  int m=x[0];
  max(x,m,5);
  cout<<"\n The maximum value is : "<<m;
  return 0;
}
```

Q.134 Describe the basic characteristics object-oriented programming. **(3)**

Ans: Object Oriented Programming (OOP) is an approach that provides a way of modularizing programs by creating partitioned memory area for both data and functions that can be used as templates for creating copies of such modules on demand.
Following are the basic characteristics of Object Oriented Programming:

- Encapsulation: The wrapping up of data and functions into a single unit is called encapsulation. The data can only be accessed by the function with which they have been tied up and not by the outside world. It creates an interface between the object's data and the program.
- Inheritance: It is the property by which one class can be derived from another class such that it acquires the properties of its base class. Just like a child inherits the properties of its parents in the same way OOP allows us to reuse the members and functions of class to be derived by another class. There are many different kinds of inheritance like multiple inheritance, multilevel inheritance and hierarchical inheritance.

- Polymorphism: Polymorphism is the property of representing one operation into many different forms. One operation may express different in different situations. The process of making a function or an operator behave in different manners is known as overloading the function or the operator.
- Data Abstraction: Abstraction is the method of representing only the essential features and only which is necessarily required by the user. The important and crucial data is encapsulated.

Q.135 What is meant by exceptions? How an exception is handled in C++? Explain with the help of an example. **(4)**

Ans:

Exception Handling: Using exception handling we can more easily manage and respond to run time errors. C++ exception handling is built upon 3 keywords: try, catch and throw.

- The 'try' block contains program statements that we want to monitor for exceptions.
- The 'throw' block throws an exception to the 'catch' block if it occurs within the try block.
- The 'catch' blocks proceeds on the exception thrown by the 'throw' block.

When an exception is thrown, it is caught by its corresponding catch statement which processes the exception. There can be more that one catch statement associated with a try. The catch statement that is used is determined by the type of the exception. An example below illustrates the working of the three blocks.

```
#include<iostream.h>
using namespace std;
int main()
{
  cout<<"\n Start "<<endl;
  try
  {
        cout<<"\n Inside try block "<<endl;
        throw 10;
        cout<<"\n this will not execute ";
  }
  catch(int i)
  {
        cout<<"\n catch number "<<endl;
        cout<<i<<endl;
  }
  cout<<"End";
  return 0;
}
```

Q.136 What do you mean by operator overloading? How unary and binary operators are implemented using the member and friend functions? **(8)**

Ans: When an operator is overloaded it doesn't lose its original meaning but it gains an additional meaning relative to the class for which it is defined. We can overload unary and binary operators using member and friend functions.
When a member operator function overloads a binary operator the function will have only one parameter. This parameter will receive the object that is on right of the operator. The object on the left side is the object that generates the call to the operator function and is passed implicitly by "this".
Overloading a unary operator is similar to overloading a binary operator except that there is only one operand to deal with. When you overload a unary operator using a member function the function has no parameter. Since there is only one operand, it is this operand that generates the call to the operator function.
Overloading operator using a friend function: as we know that a friend function does not have a 'this' pointer. Thus in case of overloading a binary operator two arguments are passed to a friend operator function. For unary operator, single operand is passed explicitly. We cannot use a friend function to overload an assignment operator(=).

Q.137 Explain the importance of using friend function in operator overloading with the help of an example. **(4)**

Ans: We can overload operators using both member function and friend functions. There are certain cases when we prefer using a friend function more than a member function. For examples suppose we wish to pass two different types of operands to the operator function, one of the operands may be an object and the other may be an integer. In case of overloading a binary operator:
A = ob + 4; or A = ob * 2;
The above statement is valid for both a member function and a friend function. But the opposite is not valid for a member function:
A = 4 + ob; or A = 2 * ob;
This statement is valid only in case of a friend function. This is because the left hand operand should be an object as it is used to invoke a member operator function.
The following program illustrates the use of a friend function:

```
#include<iostream.h>
class coord
{
  int x,y;
  public:
  coord()
  {
            x=0;y=0;
  }
```

```
  coord(int i,int j)
  {
        x=i;y=j;
  }
  void get_xy(int &i,int &j)
  {
        i=x;j=y;
  }
  friend coord operator +(int n,coord ob);
}
coord operator +(int n,coord ob)
{
  coord temp;
  temp.x= n + ob.x;
  temp.y= n + ob.y;
  return temp;
}
int main()
{
  coord ob(5,3),obj;
  int x,y,n;
  obj = n + obj;
  obj.get_xy(x,y);
  cout<<"\n x = "<<x<<"\n y = "<<y;
  return 0;
}
```

Q.138 Define a class **Date** with three variables for **day**, **month** and **year**.

a. Write the default and parameterised constructors,
b. Overload the operators <<,>> to read and print Date object,
c. Overload > to compare two dates.
d. Write the destructor that sets values to zero for all three variables. Define object of the class in main and call the above defined functions. **(2+2+3+3+4+2)**

Ans:

```
#include<iostream.h>
#include<conio.h>
class Date
{  int day,month,year;
  public:
a.    Date()
   { day=month=year=0;
   }
   Date(int d,int m,int y)
   { day=d;
```

```
        month=m;
        year=y;
       }
       Date(Date &d1)
       { day=d1.day;
        month=d1.month;
        year=d1.year;
       }
b.      friend istream & operator >> (istream &, Date &);
     friend ostream & operator << (ostream &, Date &);
c.      friend int operator >(Date &, Date &);
d.      ~Date(){day=month=year=0;}
};

istream & operator >> (istream&x, Date&b)
{
   x >> b.day>>b.month>>b.year;
   return(x);
}
ostream & operator << (ostream&dout, Date&b)
{
dout<<b.day<<"/"<<b.month<<"/"<<b.year;

   return(dout);
}
int operator>(Date &d1, Date&d2)
{  if(d1.year>d2.year)
   return 1;
  else if(d1.year<d2.year)
   return 0;
   else
   {  if(d1.month>d2.month)
     return 1;
     else
   if(d1.month<d2.month)
    return 0;
   else
    {if(d1.day>d2.day)
     return 1;
    else
     return 0;
    }
   }
}
```

```
void main()
        {  clrscr();
           Date d1(3,4,2005);
           cout<<d1;
           Date d2;
           cout<<"\n Enter day, month and year:\n";
           cin>>d2;
           cout<<d2;

           if(d1>d2)
            cout<<"\nBigger date is:"<<d1;
           else
             cout<<"\nBigger date is:"<<d2;
             getch();

        }
```

Q.139 Define the class **Student** which has name (char name[20]) and age(int). Define the default constructor, member functions **get_data()** for taking the name and age of the Student, **print()** for displaying the data of Student. **(5)**

Ans:

```
include<iostream.h>
#include<conio.h>
#include<string.h>
#include<fstream.h>

class Student
{
 char name[20];
 int age;

 public:

 Student()
 {
        strcpy(name,"");
        age=0;
 }
 void getdata()
 {
        cout<<endl<<"Enter Name:-> ";
        cin>>name;
        cout<<endl<<"Enter Age:-> ";
        cin>>age;
 }
```

```
 void print()
 {
          cout<<endl<<name<<'\t'<<age;
 }
 friend void sort(Student *s1,int N);
};
```

Q.140 For the above defined class (of Q139) create an array of students of size N and write the friend function sort(Student arr[N]) which sorts the array of Students according to their age. **(6)**

Ans:
```
void sort(Student *s1, int N)
        {
          Student temp;
          for(int i=0;i<N-1;i++)
           for(int j=N-1;j>i;j--)
           {
              if (s1[j].age < s1[j-1].age)
               {  temp=s1[j];
                     s1[j]=s1[j-1];
                     s1[j-1]=temp;

                    }
           }
            cout<<"\n Sorted Data is:\n";
            for(i=0;i<N;i++)
             s1[i].print();
        }
```

Q.141 Create a file namely "STUDENT.DAT" for storing the above objects in sorted order. **(5)**

Ans:
```
void main()
{         Student obj[100];
 int N,i,j;
 clrscr();
 cout<<endl<<"Enter maximum number of records u want to enter:-> ";
 cin>>N;
 for (i=0;i<N;i++)
          obj[i].getdata();
 cout<<endl<<"U have entered following records:";
 for (i=0;i<N;i++)
          obj[i].print();
 sort(obj,N);
 fstream inoutfile;
 inoutfile.open("STUDENT1.DAT",ios::ate|ios::in|ios::out|ios::binary);
```

```
inoutfile.seekg(0,ios::beg);
for(i=0;i<N;i++)
inoutfile.write((char *) &obj,sizeof obj);
getch();

}
```

Q.142 Consider a publishing company that markets both book and audio cassette version to its works. Create a class **Publication** that stores the title (a string) and price(type float) of a publication. Derive the following two classes from the above Publication class: **Book** which adds a page count (int) and **Tape** which adds a playing time in minutes(float). Each class should have get_data() function to get its data from the user at the keyboard.
Write the main() function to test the Book and Tape classes by creating instances of them asking the user to fill in data with get_data() and then displaying it using put_data().
(16)

Ans:

```
#include<conio.h>
#include<iostream.h>
#include<string.h>

class Publication
{
 protected:
  char title[20];
  float price;
 public:
  virtual void get_data(){}
 virtual void put_data(){}
};

class Book:public Publication
{
  int page_count;
  public:
  void get_data();
  void put_data();
};

class Tape:public Publication
{
  float play_time;
  public:
  void get_data();
  void put_data();
};
```

```
void Book::get_data()
{
  cout<<"\nTitle? ";
  cin.getline(title,20);
  cout<<"\nPrice? ";
  cin>>price;
  cout<<"\nPage_count";
  cin>>page_count;
}
void Book::put_data()
{
  cout<<"\nTitle:  "<<title;
  cout<<"\nPrice:  "<<price;
  cout<<"\nPage_count: "<<page_count;
}

void Tape::get_data()
{  cin.get();
  cout<<"\nTitle? ";
  cin.getline(title,20);
  cout<<"\nPrice? ";
  cin>>price;
  cout<<"\nPlay_time?";
  cin>>play_time;
}
void Tape::put_data()
{
  cout<<"\nTitle:  "<<title;
  cout<<"\nPrice:  "<<price;
  cout<<"\nPlay_time:"<<play_time;
}

void main()
{
 clrscr();
 Book book1;
 cout<<"Please enter book details:\n";
 book1.get_data();
 Tape tape1;
 cout<<"Please enter Tape details:\n";
 tape1.get_data();
 Publication* list[2];
 list[0]=&book1;
 list[1]=&tape1;
 cout<<"\n Book details:\n";
 list[0]->put_data();
```

```
cout<<"\n\n Tape Details:\n";
list[1]->put_data();
getch();
}
```

Q.143 Write short notes on the following:

(i) Nested class.
(ii) Scope resolution operator.
(iii) **this** pointer. **(4*3=12)**

Ans:

(i) Nested class:

Inheritance, one of the major features of OOP, is to derive certain properties of the base class in the derived class. A derived class can make use of all the public data members of its base classes.

When one class is defined inside another class, it is known as containership or nesting. It is also one of the inheriting properties. We can thus imply that one object can be a collection of many different objects. Both the concepts i.e. inheritance and containership have a vast difference. Creating an object that contains many objects is different from creating an independent object. First the member objects are created in the order in which their constructors occur in the body. Then the constructor of the main body is executed. An initialization list of the nested class is provided in the constructor. In the following example the arglist is the list of arguments that is to be supplied when a gamma object is defined. These parameters are used to define the members of gamma. The arglist1 and arglist2 are meant for the constructors a and b respectively.

```
class alpha{....};
class beta{....};

class gamma
{
  alpha a;
  beta b;
  public:
  gamma(arglist): a(arglist1), b(arglist2)
  {
        //constructor body
  }
};
```

(ii). Scope Resolution Operator:

The :: (scope resolution) operator is used to qualify hidden names so that we can use them. We can use the unary scope operator if a namespace scope or global scope name is hidden by an explicit declaration of the same name in a block or class. For example:

```
int count = 0;
```

```
int main(void) {
int count = 0;
::count = 1;  // set global count to 1
count = 2;    // set local count to 2
return 0;
}
```

The declaration of count declared in the main() function hides the integer named count declared in global namespace scope. The statement ::count = 1 accesses the variable named count declared in global namespace scope.

We also use the class scope operator to qualify class names or class member names. If a class member name is hidden, we can use it by qualifying it with its class name and the class scope operator.

In the following example, the declaration of the variable X hides the class type X, but you can still use the static class member count by qualifying it with the class type X and the scope resolution operator.

```
#include <iostream>
using namespace std;
class X
{
public:
static int count;
};
int X::count = 10;              // define static data member

int main ()
{
int X = 0;                  // hides class type X
      cout << X::count << endl;   // use static member of class X
}
```

(iii) this pointer:

C++ contains a special pointer that is called 'this'. 'this' is a pointer that is automatically passed to any member function when it is called and it is a pointer to the object that generates the call.

```
#include<iostream.h>
#include<string.h>
class inventory
{
  char item[20];
  double cost;
  int on_hand;
  public:
  inventory(char *i,double c,int o)
  {
            strcpy(this→item, i);
```

```
            this->cost=c;
            this->on_hand=o;
  }

  void show()
};
void inventory::show()
{
  cout<<this->item;
  cout<<this->cost;
  cout<<this->on_hand;
}

int main()
{
  inventory ob("Wrench",495,4);
  ob.show();
  return 0;
}
```

Q.144 Define the following terms related to Object Oriented Paradigm:
Encapsulation, Inheritance, Polymorphism, Data Abstraction **(12)**

Ans: The following are the features of Object Oriented Programming:

- **Encapsulation:** The wrapping up of data and functions into a single unit is called encapsulation. The data can only be accessed by the function with which they have been tied up and not by the outside world. It creates an interface between the object's data and the program.
- **Inheritance:** It is the property by which one class can be derived from another class such that it acquires the properties of its base class. Just like a child inherits the properties of its parents in the same way OOP allows us to reuse the members and functions of class to be derived by another class. There are many different kinds of inheritance like multiple inheritance, multilevel inheritance and hierarchical inheritance.

- **Polymorphism:** polymorphism is the property of representing one operation into many different forms. One operation may express different in different situations. The process of making a function or an operator behave in different manners is known as overloading the function or the operator.
- **Data Abstraction:** Abstraction is the method of representing only the essential features and only which is necessarily required by the user. The important and crucial data is encapsulated.

Q.145 Identify the errors in the following code fragment **(4)**

```
            ...
char ch; int vowels = 0, others = 0;
cout<<"enter character";
while((ch>='a' &&ch<=z) ||  (ch>=A && ch<=Z))
{
switch(ch)
{
case a:case A:case e:case E:case I:case I:case e:case O:case u:case U:++vowels:
     break;
default:++others;
}}
cout<<"vowels are:"<<vowels;
cout<<"others are:"<<others;
….
```

Ans: In the given code, there is no cin statement to read the character input 'ch' from the user. The char constants should be enclosed in '' (single quotes).

Q.146 Declare a class to represent bank account of customers with the following data members:Name of the depositor, Account number, Type of account(S for saving and C for Current), balance amount. The class also contains member functions to do the following:
(i) To initialise the data member (ii) To deposit money (iii) To withdraw money after checking the balance (minimum balance is Rs.1000) (iv) To display the data members. **(6)**

Ans:

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
class bank
{
 char name[25],type;
 long ac_no;
 float bal,amt;
 public:
 class bank *next;
```

```
bank()
{
        strcpy(name," ");
        type=' ' ;
        ac_no=0;
        bal=0;
        next=NULL;
}
void deposit()
{
        cout<<endl<<"Amount to be deposited:-> ";

        cin>>amt;
        bal+=amt;
        cout<<endl<<"Amount deposited successfully";
}
void withdraw()
{
        cout<<endl<<"How much u want to withdraw:-> ";
        cin>>amt;
        if (bal-amt>=1000)
        {
                bal-=amt;
                cout<<endl<<"Withdraw Done";
        }
        else
        {
                cout<<endl<<"Amount exceeds the Available I";
                getch();
        }
}
void showdata()
{
        cout<<endl<<name<<'\t'<<ac_no<<'\t';;
        if (type=='S')cout<<"Saving";
        else if (type=='C')cout<<"Current";
        cout<<'\t'<<bal;

}
int search(long x)
{
        if (ac_no==x)
                return 1;
        else
                return 0;
}
```

```
        void getdata();

};
 void bank::getdata()
        {
                cout<<endl<<"Enter name of Account Holder:-> ";
                cin>>name;
                cout<<endl<<"Enter Account Number:-> ";
                cin>>ac_no;
                rep:
                cout<<endl<<"Enter type of Account S=Saving, C=Current):-> ";
                cin>>type;
                if (type!='S' && type!='C')
                {
                        cout<<endl<<"Wrong Input";
                        getch();
                        goto rep;
                }
                cout<<endl<<"Enter Amount:-> ";
                cin>>bal;
                next=NULL;
        }


void main()
{
 class bank *bptr=NULL,*tmp=NULL,*counter=NULL;
 int i=0,found=0,ch=0;
 long acno=0;

 while (1)
 {
        clrscr();
        cout<<endl<<"1.Add a new customer";
        cout<<endl<<"2.View Customer Details";
        cout<<endl<<"3.Deposite Amount";
        cout<<endl<<"4.Withdraw Amount";
        cout<<endl<<"5.Exit";
        cout<<endl<<"Enter your choice:-> ";
        cin>>ch;
        switch (ch)
        {
        case 1: if (bptr==NULL)
                        {
                                bptr= new bank;
                                if (bptr==NULL)
```

```
                    {
                        cout<<endl<<"Memory Allocation Problem";
                        getch();
                        exit(1);
                    }
                    bptr->getdata();
                }
                else
                {
                    tmp=new bank;
                    if (tmp==NULL)
                    {
                        cout<<endl<<"Memory Allocation problem";
                        getch();
                        exit(1);
                    }
                    tmp->getdata();
                for (counter=bptr;counter->next!=NULL;counter=counter->next);
                    counter->next=tmp;
                }
                break;
case 2:
                cout<<endl<<"Name\tAc. No.\tType\tBalance\n";
                for (counter=bptr;counter!=NULL;counter=counter->next)
                    counter->showdata();
                getch();
                break;
case 3:         cout<<endl<<"Enter the Account Number to Deposite:-> ";
                long acno;
                cin>>acno;
                for (counter=bptr;counter!=NULL;counter=counter->next)
                {
                    found=counter->search(acno);
                    if (found==1)
                    {
                        counter->deposit();
                        getch();
                        break;
                    }
                }
                if (found==0)
                {
                    cout<<endl<<"Account not exists";
                    getch();
                }
                break;
```

```
        case 4:  cout<<endl<<"Enter the Account Number to Withdraw:-> ";

                        cin>>acno;
                        for (counter=bptr;counter!=NULL;counter=counter->next)
                        {
                                found=counter->search(acno);
                                if (found==1)
                                {
                                        counter->withdraw();
                                        break;
                                }
                        }
                        if (found==0)
                        {
                                cout<<endl<<"Account not exists";
                                getch();
                        }
                        break;
        case 5:  exit(1);
        default: cout<<endl<<"Wrong Choice";
        }
 }

 getch();
}
```

Q.147 How is the working of member function different from friend function and a non member function? **(6)**

Ans: A member function, a non –member function and a friend function, all the three are entirely different from each other.

A member function is that which is declared and defined inside the class. It has full access to the private data members of a class. We can directly access the data members without taking the help of the object. It is not the case with a friend function. Although it can access the private data members of the class it has to take the help of the object of that class. A friend function is not the member of the class. When we create an object of a class, memory is not allocated for the friend function. It is defined like a normal function. While calling a friend function we not use object as friend function is not the part of the object. Thus is called like any normal non member function.

A non member function cannot access the private members of a class. It is not declared inside the class definition.

Q.148 Explain a pure virtual function with an example. **(4)**

Ans: A pure virtual function has no definition related to the base class. Only the function prototype is included. To make a pure virtual function the following syntax is used:
virtual return_type function_name(par_list) = 0;
when a class contains atleast one pure virtual function then it is called an abstract class.
The following is an example program:

```
#include<iostream.h>
using namespace std;
class area
{
   double dim1,dim2;
  public:
   void set_area(int d1,int d2)
   {
            dim1=d1;
            dim2=d2;
   }
   void get_dim(double &d1,double &d2)
   {
            d1=dim1;
            d2=dim2;
   }
   virtual double get_area()=0;
};
class rectangle: public area
{
  public:
   double get_area()
   {
            double d1,d2;
            get_dim(d1,d2);
            return d1*d2;
   }
};
class triangle: public area
{
  public:
   double get_area()
   {
            double d1,d2;
            get_dim(d1,d2);
            return 0.5*d1*d2;
   }
};
```

```
int main()
{
   area *p
   rectangle r;
   triangle t;
   r.set_area(3.3,4.5);
   t.set_area(4.0,5.0);
   p=&r;
   cout<<"\n the area of rectangle is "<<p->get_area()<<endl;
   p=&t;
   cout<<"\n area of triangle is "<<p->get_area()<<endl;
   return 0;
}
```

Q.149 Discuss the various situations when a copy constructor is automatically invoked. **(6)**

Ans:

A copy constructor is invoked in many situations:

- When an object is used to initialize another in a declaration statement.
- When an object is passed as a parameter to a function.
- When a temporary object is created for use as a return value of a function.

A copy constructor only applies to initialization. It does not apply to assignment.

```
Class-name(const class name ob)
{
      //body of copy constructor
}
```

When an object is passed to a function a bitwise copy of that objet is made and given to the function parameter that receives the object. However there are cases in which this identical copy is not desired. For example, if the object contains a pointer to allocated memory. The copy will point to same memory as does the original object. Therefore, if the copy makes a change to the content of this memory it will be changed for the original object too. Also when the function terminates the copy will be destroyed causing its destructor to be called.

Q.150 What are the advantages and disadvantages of inline functions? **(6)**

Ans: Inline functions:

Advantage of an inline function is that they have no overhead associated with function call and return statements. This means that inline function can be executed much faster than the normal function.

The disadvantage of an inline function is that if they are too large and called regularly, program grows larger. For this reason, only short functions are declared as inline.

Q.151 Define Rules for operator Overloading. Write a program to overload the subscript operator '[ ]'. **(8)**

Ans:

The rules for operator overloading are as follows:

- New operators are not created. Only the current operators are overloaded.
- The overloaded operator must have at least one operand of user defined type.
- We cannot alter the basic meaning of the operator.
- When binary operators are overloaded through a member function then the left hand operand is implicitly passed as and thus it must be an object.
- Unary operators when overloaded through member functions will not take any explicit argument.

The subscript operator can be overloaded only by means of a member function:

```
#include<iostream.h>
const int size=5;
class arrtype
{
   int a[size];
  public:
   arrtype()
   {
            int i;
            for(i=0;i<size;i++)
                     a[i]=i;
   }
   int operator [](int i)
   {
            return a[i];
   }
};
int main()
{
   arrtype ob;
   int i;
   for(i=0;i<size;i++)
            cout<<ob[i]<<" ";
   return 0;
}
```

Q.152 How is a class converted into another class? Explain with one example. **(8)**

Ans: C++ allows us to convert one class type to another class type. For example

```
obj1 = obj2;
```

The conversion of one class to another is implemented using a constructor or a conversion function. The compiler treats both of them in the same manner. To decide which form to use depends upon where we want the type conversion function to be located in the source file or in the destination class.

```
#include<iostream.h>
using namespace std;
class invent2;
class invent1;
{
        int code;
        int items;
        float price;
  public:
        invent1(int a,int b,float c)
        {
                code=a;
                items=b;
                price=c;
        }
        void putdata()
        {
                cout<<"Code : "<<code<<"\n";
                cout<<"Items : "<<items<<"\n";
                cout<<"Price : "<<price<<"\n";
        }
        int getcode(){return code;}
        int getitems(){return items;}
        float getprice(){return price;}
        operator float(){return (items*price);}
        /*operator invent2()
        {
                invent2 temp;
                temp.code=code;
                temp.value=price*items;
                return temp;
        }*/
};
class invent2
{
        int code;
        float value;
  public:
        invent2()
        {
                code=0;
                value=0;
```

```
        }
        invent2(int x,float y)
        {
                code=x;
                value=y;
        }
        void putdata()
        {
                cout<<"Code : "<<code<<"\n";
                cout<<"Value : "<<value<<"\n\n";
        }
        invent2(invent1 p)
        {
                code=p.getcode();
                value=p.getitems()*p.getprice();
        }
};
int main()
{
        invent1 s1(115,9,178.6);
        invent2 d1;
        float total_value;
        total_value=s1;
        d1=s1;
        cout<<"\n Product details - invent1 type"<<"\n";
        s1.putdata();
        cout<<"\n Stock Value"<<"\n";
        cout<<"\n Value = "<<total_value<<"\n";
        cout<<"\n Protype details - invent2 type"<<"\n";
        d1.putdata();
        return 0;
}
```

Q.153 How does visibility mode control the access of members in the derived class? Explain with an example. **(6)**

Ans:

The visibility mode specified while declaring a derived class plays a very important role in the access control of the data members of the class. The visibility mode is one of the three keywords: public, private and protected. These access specifiers determine how elements of the base class are inherited by the derived class. When the access specifier for the inherited base class is public, all public members of the base class become public members of the derived class. If the access specifier is private, all public members of the base class become private members for the derived class. Access specifier is optional. In case of a derived 'class' it is private by default. In case of a derived 'structure' it is public by default.

The protected specifier is equivalent to the private specifier with the exception that protected members of a base class are accessible to members of any class derived from the base. Outside the base or derived class, protected members are not accessible. The protected specifier can appear anywhere in the class.

When a protected member of a base class is inherited as public by the derived class, it becomes protected member of the derived class. If the base is inherited a private, the protected member of the base becomes private member of the derived class.

Example:

```
#include<iostream.h>
using namespace std;
class one
{
        int x;
  public:
        int y;
        void get_xy();
        int get_x(void);
        void dis_x(void);
};
class two: public one
{
        int z;
  public:
        void prod(void);
        void display(void);
};
void one:: get_xy(void)
{
        x=7;
        y=9;
}
int one:: get_x()
{
        return x;
}
void one::dis_x()
{
        cout<<"x= "<<x<<"\n";
}
void two::prod()
{
        z=y*get_x;
}
void two::display()
{
        cout<<"x= "<<get_x()<<"\n";
```

```
        cout<<"y= "<<y<<"\n";
        cout<<"z= "<<z<<"\n";
}
int main()
{
        two m;
        m.get_xy();
        m.prod();
        m.dis_x();
        m.display();
        d.prod();
        d.display();
        return 0;
}
```

Q.154 What is run time polymorphism? How it is achieved? Explain with an example? **(4)**

Ans: Polymorphism is one of the most useful features of object oriented programming. Polymorphism means to have many forms of one name. it can be achieved both at run time and at compile time.

Polymorphism can also be achieved at run time. This is done using the concept of virtual functions. Which class' function is to be invoked is decided at run time. And then the corresponding object of that class is created accordingly. This is also called as **late binding**. For this concept we always need to make use of pointers to objects.

The example illustrates the concept of virtual functions:

```
#include<iostream.h>
using namespace std;
class base
{
  public:
            int i;
            base(int x)
            {
              i=x;
            }
            virtual void func()
            {
              cout<<"\n using base version of function";
              cout<<i<<endl;
            }
};
```

```
class derived: public base
{
  public:
            derived(int x):base(x){}
            void func()
            {
              cout<<"\n using derived version";
              cout<<i*i<<endl;
            }
};
class derived1: public base
{
  public:
            derived1(int x): base(x){}
            void func()
            {
              cout<<"\n using derived1 version";
              cout<<i+i<<endl;
            }
};
int main()
{
            base *p;
            base ob(10);
            derived d_ob(10);
            derived1 d_ob1(10);
            p=&ob;
            p->func();
            p=&d_ob;
```

```
        p->func();
        p=&d_ob1;
        p->func();
        return 0;
}
```

Q.155 Define a class loan with Principle, Period and Rate as data members. Incr (float) is a member function, which increases the rate of interest by the parameter value. If rate of interest is same for all loans, use appropriate declarations and implement the class. **(6)**

Ans:

```
#include<iostream.h>
#include<conio.h>

class loan
{
 float principle,rate;
 int period;

 public:
 loan(float f,int p)
 { principle=f;
   rate=0.15;
   period=p;
 }
 void Incr(int N)
 {
     rate=rate+N;
 }

 float calculatevalue(loan &l)
 {
  int year=1;
  float sum=l.principle;
  if(l.principle>5000)
  l.Incr(1);
  while(year<=period)
  {  sum=sum*(1+rate);
     year=year+1;
          }
  return sum;
}
};
```

```
void main()
{ float amount;
  loan l1(6000,5);

  amount=l1.calculatevalue(l1);
  cout<<"\nAmount="<<amount;

  getch();

}
```

Q.156 Write a program having a base class Student with data member rollno and member functions getnum() to input rollno and putnum() to display rollno.
A class Test is derived from class Student with data member marks and member functions getmarks() to input marks and putmarks() to display marks. Class Sports is also derived from class Student with data member score and member functions getscore() to input score and putscore() to display score. The class Result is inherited from two base classes, class Test and class Sports with data member total and a member function display() to display rollno, marks, score and the total(marks + score). **(12)**

Ans:

```
#include<iostream.h>
#include<conio.h>
using namespace std;
class student
{
  protected:
        int roll_num;
  public:
        void get_num(int a)
        {
           roll_num=a;
        }
        void put_num(void)
        {
           cout<<"Roll number: "<<roll_num<<"\n";
        }
};
class test: public student
{
  protected:
        float marks;
  public:
```

```
        void get_marks(float m)
        {
           marks=m;
        }
        void put_marks(void)
        {
           cout<<"Marks obtained: "<<marks<<"\n";
        }
};
class sports
{
  protected:
        float score;
  public:
        void get_score(float s)
        {
           score=s;
        }
        void put_score(void)
        {
           cout<<"Score: "<<score<<"\n";
        }
};
class result: public test,public sports
{
        float total;
  public:
        void display(void)
        {
           total=marks+score;
           put_num();
           put_marks();
           put_score();
           cout<<"Total score: "<<total<<"\n";
        }
};
void main()
{
        result stu;
        stu.get_num(12);
        stu.get_marks(87.6);
        stu.get_score(5.5);
        stu.display();
        getch();
}
```

Q.157 What will be the result of following expressions when they are executed in sequence? **(4)**

```
…
…
int a = 10;
int b = 20;
c = ++a    +   ++a  +  ++a;
b=b++      +   b++;
e=a++      +   --a     + b--;
f=b--  & ++a  +  b++;
cout<<c<<d<<e<<f;
…
…
```

Ans:
The variables c,d,e,f are not declared. If these variables are declared as int then output of statements will be
39 40 46 1

c=++a + ++a + ++a=13+13+13=39
d=b++ +b++=20+20=40
e=a++ + --a + b--=13+12+21=46
f=b-- & ++a + b++ =1

Q.158 What are generic classes? Why are they useful? Explain with an example how these are implemented in c++. **(8)**

Ans:
Generic class:
When we declare a generic class we are able to define all algorithms used by that class but the actual type data being manipulated will be specified as a parameter when object of that class is created. Generic classes are useful when a class contains generalised logic. For example the algorithm that is used to maintain a queue of integer can also be used for a queue of character. The compiler will automatically generate the correct type of object based upon the type you specify when the object is created. General form generic class:

```
template <class type> class class-name
{
 //body;
}
```

Here the type is place holder type-name that will be specified when the object of the class is created. We can define more than one generic data-type by using comma separated list.
To create an object
Class-name <type> ob;
Member functions of a generic class are themselves automatically generic. They need not be explicitly specified by using template.

Q.159 Explain the following functions(with example) for manipulating file pointers: seekg(),seekp(),tellg(),tellp(). **(8)**

Ans: A file pointer is used to navigate through a file. We can control this movement of the file pointer by ourselves. The file stream class supports the following functions to manage such situations.

- **seekg ():**moves get pointer (input) to a specified location.
- **seekp ():**moves put pointer (output) to a specified location.
- **tellg ():**gives the current position of the get pointer.
- **tellp ():** gives the current position of the put pointer.

For example:

infile.seekg (10);will move the file pointer to the byte number 10.

```
ofstream fileout;
fileout.open("morning", ios::app);
int p = fileout.tellp();
```

the above statements will move the output pointer to the end of the file morning and the value of p will represent the number of bytes in the file.

seekg and seekp can also be used with two arguments as follows.

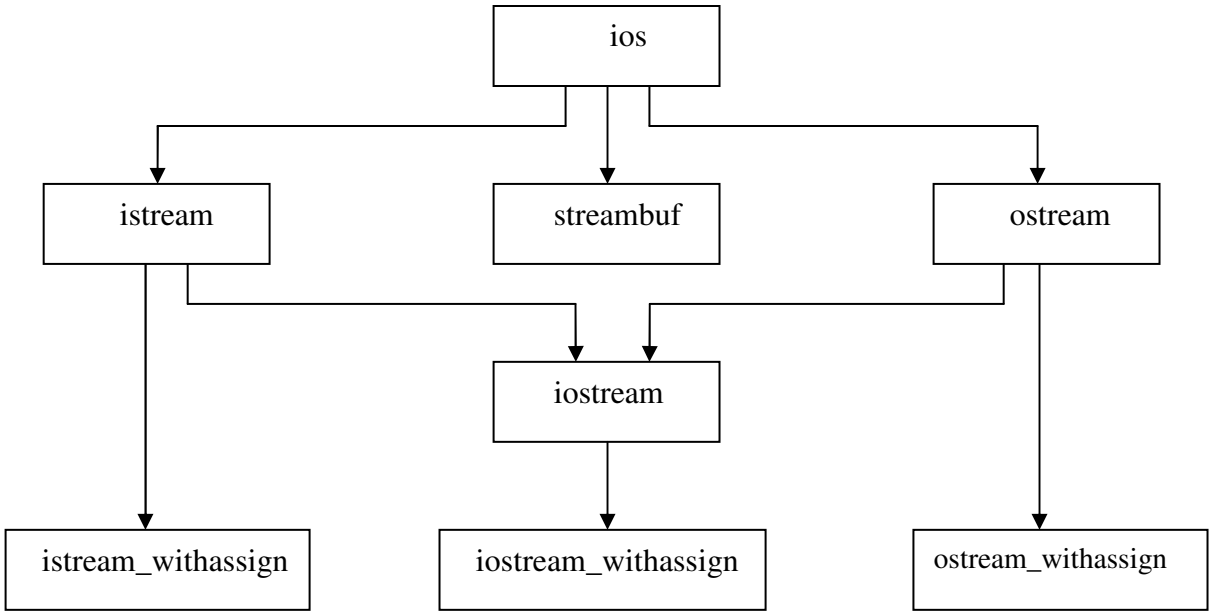
```
seekg (offset, ref position);
seekp (offset, ref position);
```

where offset referes to the number of bytes the file pointer is to be moved from the location specified by the parameter ref position.

Q.160 Explain the following:

i) Stream class hierarchy.
ii) Exception handling.
iii) Abstract class **(4*4=16)**

Ans:
(i)



The above hierarchy of classes are used to define various streams which deals with both the console and disc files. These classes are declared in the header file iostream. Ios is the base for istream, ostream and iostream base classes. Ios is declared as a virtual base class so that only one copy of its members are inherited by the iostream. The three classes istream_withassign, ostream_withassign and iostream_withassign are used to add assignment operators to these classes.

ii) Exception Handling:

using exception handling we can more easily manage and respond to run time errors. C++ exception handling is built upon 3 keywords: try, catch and throw.

- The 'try' block contains program statements that we want to monitor for exceptions.
- The 'throw' block throws an exception to the 'catch' block if it occurs within the try block.
- The 'catch' block proceeds on the exception thrown by the 'throw' block.

When an exception is thrown, it is caught by its corresponding catch statement which processes the exception. there can be more that one catch statement associated with a try. The catch statement that is used is determined by the type of the exception. An example illustrates the working of the three blocks.

```
#include<iostream.h>
using namespace std;
int main()
{
   cout<<"\n Start "<<endl;
   try
   {
            cout<<"\n Inside try block "<<endl;
            throw 10;
            cout<<"\n this will not execute ";
   }
   catch(int i)
   {
            cout<<"\n catch number "<<endl;
            cout<<i<<endl;
   }
   cout<<"End";
   return 0;
}
```

(iii) Abstract class:

When a class contains at least one pure virtual function, it is referred to as an abstract class. An abstract class is something which is to be hidden by people and on which modifications can be done in future. Since an abstract class contains at least one function for which no body exists, technically it can be considered as incomplete and therefore no object is created. Thus we can also define an abstract class as a class for which no object is created. We may conclude that an abstract class exists only to be inherited. We may still create pointer to an abstract class. In the following example, class area is an abstract class because it has a pure virtual function get_area().

```
#include<iostream.h>
using namespace std;
class area
{
   double dim1,dim2;
  public:
   void set_area(int d1,int d2)
   {
            dim1=d1;
            dim2=d2;
   }
   void get_dim(double &d1,double &d2)
   {
            d1=dim1;
            d2=dim2;
   }
   virtual double get_area()=0;
};
```

Q.161 What are the benefits of object oriented programming? Explain. **(8)**

Ans:

Object Oriented Programming (OOP) is an approach that provides a way of modulating programs by creating partitioned memory area for both data and functions that can be used as templates for creating copies of such modules on demand.

The advantages of OOP are as follows:

- Function and data both are tied together in a single unit.
- Data is not accessible by external functions as it is hidden.
- Objects may communicate with each other only through functions.
- It is easy to add new data and functions whenever required.
- It can model real world problems very well.

- Though inheritance we can eliminate redundant code and extend the use of existing classes.
- The principle of data hiding helps the programmer to build secure programs that cannot be invaded by code in other parts of the program.
- It is possible to have multiple instances of an object to co-exist without any interference.
- The data centric design approach enables us to capture more details of a model in implemented form.
- It is possible to map objects in the problem domain to those in the program.
- Message passing techniques for communication between objects makes the interface descriptions with external systems much simpler.
- Software complexity can be managed.

Q.162 Explain Inline functions and the situations where inline expansion may not work and why? **(8)**

Ans: Inline functions: Whenever we write functions, there are certain costs associated with it such as jumping to the function, saving registers, pushing arguments into the stack and returning to the calling function. To remove this overhead we write an inline function. It is a sort of request to the compiler. If we declare and define a function within the class definition then it is automatically inline. We do not need to sue the term inline.

Advantage of inline functions is that they have no overhead associated with function call or return mechanism.

The disadvantage is that if the function made inline is too large and called regularly our program grows larger.

There are certain conditions in which an inline function may not work:

- If a function is returning some value and it contains a loop, a switch or a goto statement.
- If a function is not returning a value and it contains a return statement.
- If a function contains a static variable.
- If the inline function is made recursive.

Q.163 Define a class Coord having two members type int as x and y. Use this class to define another class Rectangle which has two members of type Coord as UpperLeftCoord and BottomRightCoord. Define the constructors and member functions to get the length and breadth of rectangle. Write a global function which creates an instance of the class Rectangle and computes the area using the member functions. **(10)**

Ans:

```
#include <iostream.h>
 #include<conio.h>
#include<math.h>
 class coord {
    int X;
    int Y;
 public:
   coord() {X=Y=0;}
   coord(int x, int y)
```

```
    { X=x; Y=y;}
    int getX(){ return X;}
    int getY(){return Y;}
    //int getZ(){return Z;}
  };

  class rectangle
  {
    coord UpperLeftCoord;
    coord BottomRightCoord;
    public:
     rectangle(){}
     rectangle(coord &p1,coord &p2)
     { UpperLeftCoord=p1;
       BottomRightCoord=p2;
     }
     void showdata()
     {
  cout<<"UpperLeftCoordinate="<<"("<<UpperLeftCoord.getX()<<","<<UpperLeftCoord.getY(
  )<<")";
  cout<<"\nBottomLeftCoordinate="<<"("<<BottomRightCoord.getX()<<","<<BottomRightCoo
  rd.getY()<<")";
     }

  int getLength()
    { return(abs(UpperLeftCoord.getX()- BottomRightCoord.getX()));
    }
    int getBreath()
    { return(abs(UpperLeftCoord.getY()- BottomRightCoord.getY()));
    }
    int Area()
    {  return (getLength()*getBreath());
 }
  };

  void main()
  {  clrscr();
    coord c1(2,4);
    coord c2(6,2);
    rectangle r1(c1,c2);
       r1.showdata();
    int f=r1.Area();
  cout<<"\nArea="<<f;
    getch();
  }
```

Q.164 What is the output of following codes:

(i)
```
void main()
{
int a[] = {10,20,30,40,50};
int i;
for (i=0;i<5;i++)
{
cout<<"\nelement is "<<*(i+a)<<" "<<*(a+i)<<" "<<i[a]<<" "<<a[i];
}
}
```

(ii)
```
void main()
{
char *ptr = "xyzw";
char ch;
ch = ++ *ptr ++;
cout <<ch;
cout << ch++;
}
```
(4+4)

Ans:

(i) output:

element is10 10 10 10
element is20 20 20 20
element is30 30 30 30
element is40 40 40 40
element is50 50 50 50

(ii) output:

yy

Q.165 Write a C++ program using classes and objects to simulate result preparation system for 20 students. **(8)**

The data available for each student includes:
Name (includes first, mid and last name each of size 20 characters),
Rollno,
Marks in 3 subjects.
The percentage marks and grade are to be calculated from the following information

Percentage marks	**grade**
<50	'F'
≥50<60	'D'
≥60<75	'C'
≥75<90	'B'
≥90<100	'A'

Ans:

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
#include<process.h>
class students
{                       struct name
                        {
                                char first[20],mid[20],last[20];
                        };
                        struct name e[20];
                        int rollno[20],marks[20][3],counter;
                        int per[20];
                        char grade[20];
                        public:
                        students();
                        void getdata();
                        void calculate();
                        void showdata();
};
students :: students()
{                       counter=0;
                        for (int i=0;i<20;i++)
                        {                       strcpy(e[i].first,"");
                                strcpy(e[i].mid,"");
                                strcpy(e[i].last,"");
                                rollno[i]=0;
                                for (int j=0;j<3;j++)
                                        marks[i][j]=0;
                                per[i]=0;
                                grade[i]=' ';
                        }}
```

```
void students :: getdata()
{
                    cout<<endl<<"Enter First Name of the Student:-> ";
                    cin>>e[counter].first;
                    cout<<endl<<"Enter Middle Name of the Student:-> ";
                    cin>>e[counter].mid;
                    cout<<endl<<"Enter Last Name of the Student:-> ";
                    cin>>e[counter].last;
                    cout<<endl<<"Enter Roll Number of the student:-> ";
                    cin>>rollno[counter];
                    cout<<endl<<"Enter the Marks in Subject 1;-> ";
                    cin>>marks[counter][0];
                    cout<<endl<<"Enter the Marks in Subject 2;-> ";
                    cin>>marks[counter][1];
                    cout<<endl<<"Enter the Marks in Subject 3;-> ";
                    cin>>marks[counter][2];
                    per[counter]=(marks[counter][0]+marks[counter][1]+marks[counter][2])/3;

                    if (per[counter]>=90 && per[counter]<=100)
                            grade[counter]='A';
                    else if (per[counter]>=75)
                            grade[counter]='B';
                    else if (per[counter]>=60)
                            grade[counter]='C';
                    else if (per[counter]>=50)
                            grade[counter]='D';
                    else
                            grade[counter]='F';
                    counter++;

}

void students :: showdata()
{
                    cout<<endl<<"First\tMiddle\tLast   Marks1   Marks2   Marks3   Percentage
                    Grade\n";
                    for (int i=0;i<counter;i++)
                    {
                            cout<<endl<<e[i].first;
                            cout<<' '<<e[i].mid;
                            cout<<' '<<e[i].last;
                            cout<<' '<<rollno[i];
                            cout<<' '<<marks[i][0];
```

```
                        cout<<' '<<marks[i][1];
                        cout<<' '<<marks[i][2];
                        cout<<' '<<per[i];
                        cout<<' '<<grade[i];
                }
}
void main()
{               students obj1;
                int i,n;
                cout<<endl<<"Enter how many records u want to enter:-> ";
                cin>>n;
                if (!(n<=20))
                {
                        cout<<endl<<"Maximum number of records can be 20\n";
                        getch();
                        exit(1);
                }
                for (i=0;i<n;i++)
                        obj1.getdata();
                cout<<endl<<"Records added successfully.";
                obj1.showdata();
                getch();
}
```

Q.166 Is it possible that a function is friend of two different classes? If yes, then how it is implemented in C++. **(8)**

Ans: Yes, it is possible that a function can be a friend of two classes. Member functions of one class can be friend with another class. This is done by defining the function using the scope resolution operator.

We can also declare all the member functions of one class as the friend functions of another class. In fact we can also declare all the member functions of a class as friend. Such a class is then known as friend class.

The following program shows how a friend function can be a friend of two classes:

```
#include<iostream.h>
using namespace std;
class one;
class two
{
                int a;
  public:
                void setvalue(int n){a=n;}
                friend void max(two,one);
};
```

```
class one
{
                int b;
  public:
                void setvalue(int n){b=n;}
                friend void max(two,one);
};
void max(two s,one t)
{
                if(s.a>=t.b)
                        cout<<s.a;
                else
                        cout<<t.b;
}
int main()
{
                one obj1;
                obj1.setvalue(5);
                two obj2;
                obj2.setvalue(10);
                max(obj2,obj1);
                return 0;
}
```

Q.167 Write a program to overload << and >> operator to I/O the object of a class. **(8)**

Ans:

```
#include<iostream.h>
class complex
{
                double real,imag;
  public:
                complex()
                {
                }
                complex(double r,double i)
                {
                        real=r;
                        imag=i;
                }
                friend ostream& operator <<(ostream& s,complex& c);
                friend istream& operator >>(istream& s,complex& c);
                };
                ostream& operator <<(ostream& s,complex& c)
                {
                        s<<"("<<c.real<<","<<c.imag<<")";
```

```
        return s;
}
istream& operator >>(istream& s,complex& c);
{
        s>>c.real>>c.imag;
        return s;
}
void main()
{
        complex c1(1.5,2.5),c2(3.5,4.5),c3;
        cout<<endl<<"c1= "<<c1<<endl<<"c2= "<<c2;
        cout<<endl<<"Enter a complex number: ";
        cin>>c3;
        cout<<"c3= "<<c3;
}
```

Q.168 What is the difference between a virtual function and a pure virtual function? Give example of each. **(8)**

Ans: Virtual functions are important because they support polymorphism at run time. A virtual function is a member function that is declared within the base class but redefined inside the derived class. To create a virtual function we precede the function declaration with the keyword virtual. Thus we can assume the model of "one interface multiple method". The virtual function within the base class defines the form of interface to that function. It happens when a virtual function is called through a pointer.

The following is an example that illustrates the use of a virtual function:

```
#include<iostream.h>
using namespace std;
class base
{
  public:
  int i;
  base(int x)
  {
        i=x;
  }
  virtual void func()
  {
        cout<<"\n using base version of function";
        cout<<i<<endl;
  }
};
class derived: public base
{
  public:
  derived(int x):base(x){}
```

```
  void func()
  {
        cout<<"\n sing derived version";
        cout<<i*i<<endl;
  }
};
class derived1: public base
{
  public:
  derived1(int x): base(x){}
  void func()
  {
        cout<<"\n using derived1 version";
        cout<<i+i<<endl;
  }
};

int main()
{
  base *p;
  base ob(10);
  derived d_ob(10);
  derived1 d_ob1(10);
  p=&ob;
  p->func();
  p=&d_ob;
  p->func();
  p=&d_ob1;
  p->func();
  return 0;
}
```

A pure virtual function has no definition related to the base class. Only the function prototype is included to make a pure virtual function the following syntax is used:

```
virtual return_type function_name(par_list) = 0;
```

The following is an example program:

```
#include<iostream.h>
using namespace std;
class area
{
  double dim1,dim2;
  public:
  void set_area(int d1,int d2)
  {
        dim1=d1;
        dim2=d2;
  }
```

```
  void get_dim(double &d1,double &d2)
  {
        d1=dim1;
        d2=dim2;
  }
  virtual double get_area()=0;
};
class rectangle: public area
{
  public:
  double get_area()
  {
        double d1,d2;
        get_dim(d1,d2);
        return d1*d2;
  }
};

class triangle: public area
{
  public:
  double get_area()
  {
        double d1,d2;
        get_dim(d1,d2);
        return 0.5*d1*d2;
  }
};
int main()
{
  area *p
  rectangle r;
  triangle t;
  r.set_area(3.3,4.5);
  t.set_area(4.0,5.0);
  p=&r;
  cout<<"\n the area of rectangle is "<<p->get_area()<<endl;
  p=&t;
  cout<<"\n area of triangle is "<<p->get_area()<<endl;
  return 0;
}
```

Q.169 Write a program for Conversion from Basic to Class Type. **(8)**

Ans:

```
#include<iostream.h>
class time
{
  int hours;
  int minutes;
  public:
  time(){}
  time(int t)
  { hours=t/60;
    minutes=t%60;
  }
  void showtime()
  { cout<<hours<<"hrs "<<minutes<<"min";
  }
};
void main()
{
  time t1;
  int duration=90;
  t1=duration;
  t1.showtime();
}
```

Q.170 What is the output of following programs: **(8)**

(i)
```
#include<iostream.h>
int global = 10;
void func(int & x, int y)
{
x = x – y;
y = x * 10;
cout<<x<<“\t”<< y<< “\n”;
}
void main()
{
int global = 7;
func(::global, global);
cout<<global<<“\t”<<::global<<“\n”;
func(global,::global);
cout<<global<<“\t”<<::global<<“\n”;
}
```

(ii)
```
#include<iostream.h>
void main( )
```

```
{
int i, j, m;
int a[5]={8, 10, 1, 14, 16}
i = ++a[2];
m = a[i++];
cout <<i<<m;
}
```

Ans:

i) 3 30
7 3
4 40
4 3
ii) 3 2

Q.171 Write a function template for sorting a list of arrays. **(8)**

Ans:

```
#include<iostream.h>
using namespace std;
template<class T>

void sort(T a[],int n)
{
  for(int i=0;i<n-1;i++)
        for(int j=i+1;j<n;j++)
        {       if(a[i]>a[j])
                        swap(a[i],a[j]);
        }
}
template<class X>
void swap(X &a,X &b)
{
  X temp=a;
  a=b;
  b=temp;
}
int main()
{
  int v[5]={32,78,12,98,56};
  sort(v,5);
  cout<<"\n the list of sorted array is :- ";
  for(int i=0;i<5;i++)
        cout<<"\n"<<v[i];
  return 0;
}
```

Q.172 How does inheritance influence the working of constructor and destructor?
Given the following set of definitions

```
Class x
{
};
class y: public x
{
};
class z: public y
{
};
z obj;
```

What order will the constructor and destructor be invoked? **(8)**

Ans: Inheritance has a great deal of influence over the working of constructors and destructors. In case of inheritance the base class constructor is invoked first and then the derived class constructor. In case of multiple inheritance, the constructors of base classes are executed in the order in which they occur in the declaration in the derived class. Similarly in case of multilevel inheritance, the base class constructors are executed in the order of their inheritance. Destructors in case of inheritance are executed exactly in the opposite order in which constructors are executed. Likewise in case of multilevel inheritance destructors are executed in the reverse of that is, from derived to base.

```
class x
{
};
class y: public x
{
};
class z: public y
{
};
z obj;
```

In the above definitions, the contructor will be fired in the following sequence-
class x → class y → class z
similarly, destructors will be fired exactly in the reverse sequence-
class z → class y → class x

Q.173 What is meant by a constant member function? How is it declared? Give an example. **(8)**

Ans: Constant Member functions : A const member function is that which does alter the value of any data in the program. It is declared as follows:

```
void area(int, int) const;
float dis() const;
```

when a const function tries to alter any data ,the compiler generates an error message.

Q.174 Write a program to create a database of the students information such as name, roll no and the program should have the following facilities. **(10)**

i) Adds a new records to the file.
ii) Modifies the details of an record.

Displays the contents of the file.

Ans:

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>
#include<fstream.h>
#include<stdlib.h>
#include<iomanip.h>

class Students
{
 int rollno;
 char name[25];
 public:
   void getdata()
        {
                cout<<endl<<"Enter Name:-> ";cin>>name;
                cout<<endl<<"Enter Roll Number:-> ";cin>>rollno;
        }
        void showdata()
        {
                cout<<setw(10)<<rollno<<setw(10)<<name<<endl;
        }
};
void main()
{
 clrscr();
 int r;
 fstream file;
 file.open("Student.TXT",ios::ate|ios::in | ios::out|ios::binary);
 file.seekg(0,ios::beg);
 Students obj;
 char ch;
 while (1)
 {
 cout<<endl<<"1.Add\n2.Modify\n3.Display\n4.Exit\nEnter your choice:-> ";
        cin>>ch;
        switch (ch)
        {
        case '1':
           obj.getdata();
```

```
                file.write((char *) & obj, sizeof(obj));
                break;
            case '2':
                int last = file.tellg();
                int n = last/sizeof(obj);
                cout<<"number of objects = "<< n << '\n';
                cout<<"total bytes in the file = "<<last<<'\n';
 //modify the details of an item
                cout<<"enter object no. to b updtd \n";
                int object;
                cin>>object;
                cin.get(ch);
                int location = (object - 1)*sizeof(obj);
                if(file.eof())
                file.clear();
                file.seekp(location);
                cout<<"enter new values of object \n";
                obj.getdata();
                cin.get(ch);
                file.write((char *)&obj,sizeof obj)<<flush;
                file.seekg(0);
                break;
            case '3':
                 file.seekg(0);
                 while(file.read((char *) &obj,sizeof obj))
                        obj.showdata();
                 break;
            case '4': exit(0);
            default:
                                cout<<endl<<"Input out of reange.";
            }
 }
}
```

Q.175 Write a program to read three numbers x, y and z and evaluate R given by

$$R = z/(x - y)$$

Use exception handling to throw an exception in case division by zero is attempted. **(8)**

Ans:

```
#include<iostream.h>
  using namespace std;
  int main()
  {
    int x,y,z,R;
    cout<<"\n Enter three values x,y and z : ";
```

```
        cin >> x >> y >> z;
        int a=(x-y);
        try
        {
                if(a!=0)
                {
                        R=z/a;
                        cout<<"Result(z/(x-y)) = "<< R << "\n";
                }
                else
                        throw(a);
        }
        catch(int i)
        {
                cout<<"\n Exception caught : (x-y)" << i << "\n";

        }
        cout << "END";
}
```

TYPICAL QUESTIONS & ANSWERS

PART - I

OBJECTIVE TYPE QUESTIONS

Each Question carries 2 marks.

Choose correct or the best alternative in the following:

Q.1 Translator for low level programming language were termed as
(A) Assembler **(B)** Compiler
(C) Linker **(D)** Loader

Ans: (A)

Q.2 Analysis which determines the meaning of a statement once its grammatical structure becomes known is termed as
(A) Semantic analysis **(B)** Syntax analysis
(C) Regular analysis **(D)** General analysis

Ans: (A)

Q.3 Load address for the first word of the program is called
(A) Linker address origin **(B)** load address origin
(C) Phase library **(D)** absolute library

Ans: (B)

Q.4 Symbolic names can be associated with
(A) Information **(B)** data or instruction
(C) operand **(D)** mnemonic operation

Ans: (B)

Q.5 The translator which perform macro expansion is called a
(A) Macro processor **(B)** Macro pre-processor
(C) Micro pre-processor **(D)** assembler

Ans: (B)

Q.6 Shell is the exclusive feature of
(A) UNIX **(B)** DOS
(C) System software **(D)** Application software

Ans: (A)

Q.7 A program in execution is called
(A) Process **(B)** Instruction
(C) Procedure **(D)** Function

Ans: (A)

Q.8 Interval between the time of submission and completion of the job is called

(A) Waiting time **(B)** Turnaround time
(C) Throughput **(D)** Response time

Ans: (B)

Q.9 A scheduler which selects processes from secondary storage device is called

(A) Short term scheduler. **(B)** Long term scheduler.
(C) Medium term scheduler**.** **(D)** Process scheduler.

Ans: (C)

Q.10 The scheduling in which CPU is allocated to the process with least CPU-burst time is called

(A) Priority Scheduling **(B)** Shortest job first Scheduling
(C) Round Robin Scheduling **(D)** Multilevel Queue Scheduling

Ans: (B)

Q.11 The term 'page traffic' describes

(A) number of pages in memory at a given instant.
(B) number of papers required to be brought in at a given page request.
(C) the movement of pages in and out of memory**.**
(D) number of pages of executing programs loaded in memory.

Ans: (C)

Q.12 The "turn-around" time of a user job is the

(A) time since its submission to the time its results become available.
(B) time duration for which the CPU is allotted to the job.
(C) total time taken to execute the job**.**
(D) time taken for the job to move from assembly phase to completion phase.

Ans: (C)

Q.13 Which of the following can be used as a criterion for classification of data structures used in language processing.

(A) nature of a data structure **(B)** purpose of a data structure
(C) lifetime of a data structure **(D)** all of the above**.**

Ans: (D)

Q.14 Memory utilization factor shall be computed as follows

(A) memory in use/allocated memory.
(B) memory in use/total memory connected**.**
(C) memory allocated/free existing memory.
(D) memory committed/total memory available.

Ans: (B)

Q.15 Program 'preemption' is
- **(A)** forced de allocation of the CPU from a program which is executing on the CPU.
- **(B)** release of CPU by the program after completing its task.
- **(C)** forced allotment of CPU by a program to itself.
- **(D)** a program terminating itself due to detection of an error.

Ans: (A)

Q.16 An assembler is
- **(A)** programming language dependent.
- **(B)** syntax dependant.
- **(C)** machine dependant.
- **(D)** data dependant.

Ans: (C)

Q.17 Which of the following is not a fundamental process state
- **(A)** ready
- **(B)** terminated
- **(C)** executing
- **(D)** blocked

Ans: (D)

Q.18 'LRU' page replacement policy is
- **(A)** Last Replaced Unit.
- **(B)** Last Restored Unit.
- **(C)** Least Recently Used**.**
- **(D)** Least Required Unit.

Ans: (C)

Q.19 Which of the following is true?
- **(A)** Block cipher technique is an encryption technique.
- **(B)** Steam cipher technique is an encryption technique.
- **(C)** Both (A) and (B).
- **(D)** Neither of (A) and (B).

Ans: (C)

Q.20 Which of the following approaches do not require knowledge of the system state?
- **(A)** deadlock detection.
- **(B)** deadlock prevention.
- **(C)** deadlock avoidance.
- **(D)** none of the above.

Ans: (D)

Q.21 Program generation activity aims at
- **(A)** Automatic generation of program
- **(B)** Organize execution of a program written in PL
- **(C)** Skips generation of program
- **(D)** Speedens generation of program

Ans: (A)

Q.22 Which amongst the following is not an advantage of Distributed systems?

(A) Reliability **(B)** Incremental growth
(C) Resource sharing **(D)** None of the above

Ans: (A)

Q.23 An imperative statement

(A) Reserves areas of memory and associates names with them
(B) Indicates an action to be performed during execution of assembled program
(C) Indicates an action to be performed during optimization
(D) None of the above

Ans: (B)

Q.24 Which of the following loader is executed when a system is first turned on or restarted

(A) Boot loader **(B)** Compile and Go loader
(C) Bootstrap loader **(D)** Relating loader

Ans: (C)

Q.25 Poor response time is usually caused by

(A) Process busy
(B) High I/O rates
(C) High paging rates
(D) Any of the above

Ans: (D)

Q.26 "Throughput" of a system is

(A) Number of programs processed by it per unit time
(B) Number of times the program is invoked by the system
(C) Number of requests made to a program by the system
(D) None of the above

Ans: (A)

Q.27 The "blocking factor" of a file is

(A) The number of blocks accessible to a file
(B) The number of blocks allocated to a file
(C) The number of logical records in one physical record
(D) None of the above

Ans: (C)

Q.28 Which of these is a component of a process precedence sequence?

(A) Process name **(B)** Sequence operator ';'
(C) Concurrency operator ',' **(D)** All of the above

Ans: (D)

Q.29 Which amongst the following is valid syntax of the **Fork** and **Join** Primitive?

(A) Fork <label>
Join <var>

(B) Fork <label>
Join <label>

(C) For <var>
Join <var>

(D) Fork <var>
join <var>

Ans: (A)

Q.30 Nested Macro calls are expanded using the

(A) FIFO rule (First in first out)
(B) LIFO (Last in First out)
(C) FILO rule (First in last out)
(D) None of the above

Ans: (B)

Q.31 A parser which is a variant of top-down parsing without backtracking is

(A) Recursive Descend.
(B) Operator Precedence.
(C) LL(1) parser.
(D) LALR Parser.

Ans: (A)

Q.32 The expansion of nested macro calls follows

(A) FIFO rule.
(B) LIFO rule.
(C) LILO rule.
(D) priority rule.

Ans: (B)

Q.33. In a two-pass assembler, the task of the Pass II is to

(A) separate the symbol, mnemonic opcode and operand fields.
(B) build the symbol table.
(C) construct intermediate code.
(D) synthesize the target program.

Ans: (D)

Q.34 A linker program

(A) places the program in the memory for the purpose of execution.
(B) relocates the program to execute from the specific memory area allocated to it.
(C) links the program with other programs needed for its execution.
(D) interfaces the program with the entities generating its input data.

Ans: (C)

Q.35 Which scheduling policy is most suitable for a time-shared operating system

(A) Shortest-job First.
(B) Elevator.
(C) Round-Robin.
(D) First-Come-First-Serve.

Ans: (C)

Q.36 A critical section is a program segment
- **(A)** which should run in a certain specified amount of time.
- **(B)** which avoids deadlocks.
- **(C)** where shared resources are accessed**.**
- **(D)** which must be enclosed by a pair of semaphore operations, P and V.

Ans: (C)

Q.37 An operating system contains 3 user processes each requiring 2 units of resource R .The minimum number of units of R such that no deadlocks will ever arise is
- **(A)** 4. **(B)** 3.
- **(C)** 5. **(D)** 6.

Ans: (A)

Q.38 Locality of reference implies that the page reference being made by a process
- **(A)** will always be to the page used in the previous page reference.
- **(B)** is likely to be the one of the pages used in the last few page references**.**
- **(C)** will always be to one of the pages existing in memory.
- **(D)**will always lead to a page fault.

Ans: (B)

Q.39 Which of these is not a part of Synthesis phase
- **(A)** Obtain machine code corresponding to the mnemonic from the Mnemonics table
- **(B)** Obtain address of a memory operand from the symbol table
- **(C)** Perform LC processing
- **(D)** Synthesize a machine instruction or the machine form of a constant

Ans: (C)

Q.40 The syntax of the assembler directive EQU is
- **(A)** EQU <address space> **(B)** **<**symbol>EQU<address space**>**
- **(C)** <symbol>EQU **(D)** None of the above

Ans: (B)

Q.41 The following features are needed to implement top down parsing
- **(A)** Source string marker
- **(B)** Prediction making mechanism
- **(C)** Matching and Backtracking mechanism
- **(D)** All of the above

Ans: (D)

Q.42 A macro definition consists of
- **(A)** A macro prototype statement **(B)** One or more model statements
- **(C)** Macro pre-processor statements **(D)** All of the above

Ans: (D)

Q.43 The main reason to encrypt a file is to _____.

(A) Reduce its size **(B) Secure it for transmission**
(C) Prepare it for backup **(D)** Include it in the start-up sequence

Ans: (B)

Q.44 Which of the following is not a key piece of information, stored in single page table entry, assuming pure paging and virtual memory

(A) Frame number
(B) A bit indicating whether the page is in physical memory or on the disk
(C) A reference for the disk block that stores the page
(D) None of the above

Ans: (C)

Q.45 A UNIX device driver is

(A) Structured into two halves called top half and bottom half
(B) Three equal partitions
(C) Unstructured
(D) None of the above

Ans: (A)

Q.46 The following is not a layer of IO management module

(A) PIOCS (Physical Input Output Control System)
(B) LIOCS (Logical Input Output Control System)
(C) FS (File System)
(D) MCS (Management Control System)

Ans: (D)

Q.47 Which amongst the following is not a valid page replacement policy?

(A) LRU policy (Least Recently Used)
(B) FIFO policy (First in first out)
(C) RU policy (Recurrently used)
(D) Optimal page replacement policy

Ans: (C)

Q.48 Consider a program with a linked origin of 5000. Let the memory area allocated to it have the start address of 70000. Which amongst the following will be the value to be loaded in relocation register?

(A) 20000 **(B)** 50000
(C) 70000 **(D)** 90000

Ans: (None of the above choice in correct**.)**

Q.49 An assembly language is a

(A) low level programming language

(B) Middle level programming language
(C) High level programming language
(D) Internet based programming language

Ans: (A)

Q.50 TII stands for
(A) Table of incomplete instructions
(B) table of information instructions
(C) translation of instructions information
(D) translation of information instruction

Ans: (A)

Q.51 An analysis, which determines the syntactic structure of the source statement, is called
(A) Sementic analysis **(B)** process analysis
(C) Syntax analysis **(D)** function analysis

Ans: (C)

Q.52 Action implementing instruction's meaning are a actually carried out by
(A) Instruction fetch
(B) Instruction decode
(C) instruction execution
(D) Instruction program

Ans: (C)

Q.53 The field that contains a segment index or an internal index is called
(A) target datum **(B)** target offset
(C) segment field **(D)** fix dat

Ans: (A)

Q.54 A program in execution is called
(A) process **(B)** function
(C) CPU **(D)** Memory

Ans: (A)

Q.55 Jobs which are admitted to the system for processing is called
(A) long-term scheduling **(B)** short-term scheduling
(C) medium-term scheduling **(D)** queuing

Ans: (A)

Q.56 A set of techniques that allow to execute a program which is not entirely in memory is called
(A) demand paging **(B)** virtual memory
(C) auxiliary memory **(D)** secondary memory

Ans: (B)

Q. 57 SSTF stands for
(A) Shortest-Seek-time-first scheduling **(B)** small – small-time-first
(C) simple-seek-time-first scheduling **(D)** small-simple-time-first

Ans: (A)

Q.58 Before proceeding with its execution, each process must acquire all the resources it needs is called
(A) hold and wait **(B)** No pre-emption
(C) circular wait **(D)** starvation

Ans: (A)

Q.59 Virtual memory is
(A) simple to implement
(B) used in all major commercial operating systems
(C) less efficient in utilization of memory
(D) useful when fast I/O devices are not available

Ans: (B)

Q.60 **R**elocation bits used by relocating loader are specified by
(A) Relocating loader itself **(B)** Assembler or Translator
(C) Macro processor **(D)** Both **(A)** and **(B)**

Ans: (B)

Q.61 Resolution of externally defined symbols is performed by
(A) Linker **(B)** Loader
(C) Compiler **(D)** Editor

Ans: (A)

Q.62 Relocatable programs
(A) cannot be used with fixed partitions
(B) can be loaded almost anywhere in memory
(C) do not need a linker
(D) can be loaded only at one specific location

Ans: (B)

Q.63 Page stealing
(A) is a sign of efficient system
(B) is taking page frames other working sets
(C) should be the tuning goal
(D) is taking larger disk spaces for pages paged out

Ans: (B)

Q.64 The total time to prepare a disk drive mechanism for a block of data to be read from is its

(A) latency
(B) latency plus transmission time
(C) latency plus seek time
(D) latency plus seek time plus transmission time

Ans: (C)

Q.65 To avoid race condition, the maximum number of processes that may be simultaneously inside the critical section is

(A) zero **(B)** one
(C) two **(D)** more than two

Ans: (B)

Q.66 The memory allocation scheme subject to "external" fragmentation is

(A) segmentation **(B)** swapping
(C) pure demand paging **(D)** multiple fixed contiguous partitions

Ans: (A)

Q.67 Page fault frequency in an operating system is reduced when the

(A) processes tend to the I/O-bound
(B) size of pages is reduced
(C) processes tend to be CPU-bound
(D) locality of reference is applicable to the process

Ans: (D)

Q.68 In which of the following page replacement policies Balady's anomaly occurs?

(A) FIFO **(B)** LRU
(C) LFU **(D)** NRU

Ans: (A)

Q.69 Which of the following are language processors?

(A) Assembler **(B)** Compiler
(C) Interpreter **(D)** All of the above

Ans: (D)

Q.70 Virtual memory can be implemented with

(A) Segmentation **(B)** Paging
(C) None **(D)** all of the above

Ans: (D)

Q.71 Recognition of basic syntactic constructs through reductions, this task is performed by

(A) Lexical analysis **(B)** Syntax analysis

(C) Semantic analysis **(D)** Structure analysis

Ans: (B)

Q.72 A grammar for a programming language is a formal description of

(A) Syntax **(B)** Semantics

(C) Structure **(D)** Code

Ans: (C)

Q.73 _____ is a technique of temporarily removing inactive programs from the memory of computer system

(A) Swapping **(B)** Spooling

(C) Semaphore **(D)** Scheduler

Ans: (A)

Q.74 _____ is a technique of improving the priority of process waiting in Queue for CPU allocation

(A) Starvation **(B)** Ageing

(C) Revocation **(D)** Relocation

Ans: (B)

Q.75 _____ is the time required by a sector to reach below read/write head.

(A) Seek Time **(B)** Latency Time

(C) Access time **(D)** None

Ans: (B)

Q.76 Which of the following is most general phase structured grammar?

(A) Context – Sensitive **(B)** Regular

(C) Context – Free **(D)** None of the above

Ans: (A)

Q.77 File record length

(A) Should always be fixed

(B) Should always be variable

(C) Depends upon the size of file

(D) Should be chosen to match the data characteristics.

Ans: (D)

Q.78 A public key encryption system

(A) Allows only the correct receiver to decode the data

(B) Allows only one to decode the transmission.

(C) Allows only the correct sender to decode the data.

(D) Does not encode the data before transmitting it.

Ans: (A)

PART – II

DESCRIPTIVES

Q.1. Discuss in detail Table management Techniques? **(7)**

Ans:
An Assembler uses the following tables:
OPTAB: Operation Code Table Contains mnemonic operation code and its machine language equivalent.
SYMTAB: Symbol Table maintains symbolic label, operand and their corresponding machine.
LITTAB is a table of literals used in the program
For efficiency reasons SYMTAB must remain in main memory throughout passes I and II of the assembler. LITTAB is not accessed as frequently as SYMTAB, however it may be accessed sufficiently frequently to justify its presence in the memory. If memory is at a premium, only a part of LITTAB can be kept in memory. OPTAB should be in memory during pass I

Q.2 Define the following:
(i) Formal language Grammars.
(ii) Terminal symbols.
(iii) Alphabet and String. **(9)**

Ans:
(i) A formal language grammar is a set of formation rules that describe which strings formed from the alphabet of a formal language are syntactically valid, within the language. A grammar only addresses the location and manipulation of the strings of the language. It does not describe anything else about a language, such as its semantics.
As proposed by Noam Chomsky, a grammar *G* consists of the following components:

- A finite set N of *non terminal symbols*.
- A finite set Σ of *terminal symbols* that is disjoint from N.
- A finite set P of *production rules*, each rule of the form

where * is the Kleene star operator and denotes set union. That is, each production rule maps from one string of symbols to another, where the first string contains at least one non terminal symbol.

- A distinguished non terminal symbol from set N that is the *start symbol*.

(ii)*Terminal* symbols are literal strings forming the input of a formal grammar and cannot be broken down into smaller units without losing their literal meaning. In simple words, terminal symbols cannot be changed using the rules of the grammar; that is, they're the end of the line, or terminal. For example, if the grammar rules are that *x* can become *x*a and *x* can become a*x*, then a is a terminal symbol because it cannot become something else. These are the symbols which can appear as it is in the programme.
(iii) A finite set of symbols is called alphabet. An alphabet is often denoted by sigma, yet can be given any name.
B = {0, 1} says B is an alphabet of two symbols, 0 and 1.
C = {a, b, c} says C is an alphabet of three symbols, a, b and c.

Sometimes space and comma are in an alphabet while other times they are meta symbols used for descriptions. A language is defined over an alphabet. For example binary language is defined over alphabet B.
A finite sequence of symbols from an alphabet is called string or word.
01110 and 111 are strings from the alphabet B above.
aaabccc and b are strings from the alphabet C above.
A null string is a string with no symbols, usually denoted by epsilon has zero length.

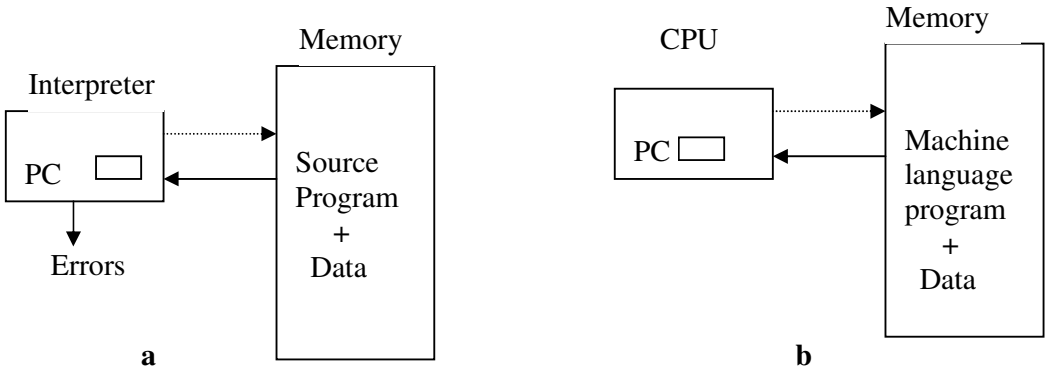
Q.3. What is parsing? Write down the drawback of top down parsing of backtracking. **(7)**

Ans:
Parsing is the process of analyzing a text, made of a sequence of tokens, to determine its grammatical structure with respect to a given formal grammar. Parsing is also known as syntactic analysis and parser is used for analyzing a text. The task of the parser is essentially to determine if and how the input can be derived from the start symbol of the grammar. The input is a valid input with respect to a given formal grammar if it can be derived from the start symbol of the grammar.
Following are drawbacks of top down parsing of backtracking:

(i) Semantic actions cannot be performed while making a prediction. The actions must be delayed until the prediction is known to be a part of a successful parse.
(ii) Precise error reporting is not possible. A mismatch merely triggers backtracking. A source string is known to be erroneous only after all predictions have failed.

Q.4. Give the Schematic of Interpretation of HLL program and execution of a machine language program by the CPU. **(8)**

Ans:



The CPU uses a program counter (PC) to note the address of next instruction to be executed. This instruction is subjected to the instruction execution cycle consisting of the following steps:

1. Fetch the instruction.
2. Decode the instruction to determine the operation to be performed, and also its operands.
3. Execute the instruction.

At the end of the cycle, the instruction address in PC is updated and the cycle is repeated for the next instruction. Program interpretation can proceed in a similar manner. The PC can indicate which statement of the source program is to be

interpreted next. This statement would be subjected to the interpretation cycle, which consists of the following steps:

1. Fetch the statement.
2. Analyse the statement and determine its meaning, viz . the computation to be performed and its operands.
3. Execute the meaning of the statement.

Q.5. Give the difference between multiprogramming and multiprocessing. **(5)**

Ans:

A multiprocessing system is a computer hardware configuration that includes more than one independent processing unit. The term multiprocessing is generally used to refer to large computer hardware complexes found in major scientific or commercial applications. The multiprocessor system is characterized by-increased system throughput and application speedup-parallel processing. The main feature of this architecture is to provide high speed at low cost in comparison to uni- processor.

A multiprogramming operating system is system that allows more than one active user program (or part of user program) to be stored in main memory simultaneously. Multi programmed operating systems are fairly sophisticated. All the jobs that enter the system are kept in the job pool. This pool consists of all processes residing on mass storage awaiting allocation of main memory. If several jobs are ready to be brought into memory, and there is not enough room for all of them, then the system must choose among them. A time-sharing system is a multiprogramming system.

Q.6. Write down different system calls for performing different kinds of tasks. **(4)**

Ans:

A **system call** is a request made by any program to the operating system for performing tasks -- picked from a predefined set -- which the said program does not have required permissions to execute in its own flow of execution. System calls provide the interface between a process and the operating system. Most operations interacting with the system require permissions not available to a user level process, e.g. I/O performed with a device present on the system or any form of communication with other processes requires the use of system calls.

The main types of system calls are as follows:

• **Process Control**: These types of system calls are used to control the processes. Some examples are end, abort, load, execute, create process, terminate process etc.

• **File Management**: These types of system calls are used to manage files. Some examples are Create file, delete file, open, close, read, write etc.

• **Device Management**: These types of system calls are used to manage devices. Some examples are Request device, release device, read, write, get device attributes etc.

Q.7. Differentiate between pre-emptive and non-pre-emptive scheduling. **(4)**

Ans:

In a **pre-emptive scheduling** approach, CPU can be taken away from a process if there is a need while in a non-pre-emptive approach if once a process has been given the CPU, the CPU cannot be taken away from that process, unless the process completes or leaves the CPU for performing an Input Output.

Pre-emptive scheduling is more useful in high priority process which requires immediate response, for example in real time system. While in **nonpreemptive systems,** jobs are made to wait by longer jobs, but treatment of all processes is fairer.

Q.8. CPU burst time indicates the time, the process needs the CPU. The following are the set of processes with their respective CPU burst time (in milliseconds).

Processes	CPU-burst time
P1	10
P2	5
P3	5

Calculate the average waiting time if the process arrived in the following order:

(i) P1, P2 & P3 (ii) P2, P3 & P1 **(6)**

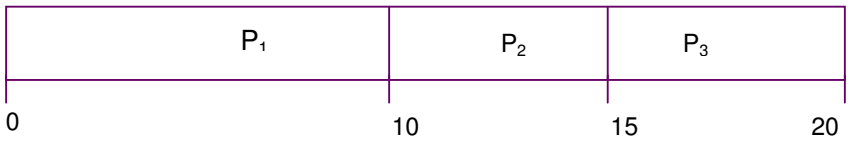
Ans:

Considering FCFS scheduling

Process	Burst Time
P_1	10
P_2	5
P_3	5

(i) Suppose that the processes arrive in the order: $P1$, $P2$, $P3$

The Gantt Chart for the schedule is:

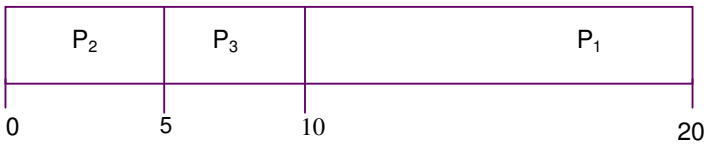


Waiting time for $P1 = 0$; $P2 = 10$; $P3 = 15$

Average waiting time: $(0 + 10 + 15)/3 = 8.33$ unit of time

(ii)Suppose that the processes arrive in the order $P2$, $P3$, $P1$.

The Gantt chart for the schedule is:



Waiting time for $P1 = 10$; $P2 = 0$; $P3 = 5$

Average waiting time: $(10 + 0 + 5)/3 = 5$ unit of time.

Q.9. What is a semaphore? Explain busy waiting semaphores. **(6)**

Ans:

A **semaphore** is a protected variable or abstract data type which constitutes the classic method for restricting access to shared resources such as shared memory in a parallel programming environment.

Weak, Busy-wait Semaphores:

- The simplest way to implement semaphores.
- Useful when critical sections last for a short time, or we have lots of CPUs.
- S initialized to positive value (to allow someone in at the beginning).
- S is an integer variable that, apart from initialization, can only be accessed through 2 *atomic and mutually exclusive* operations:

```
wait(s):
      while (s.value != 0);
      s.value--;
signal(s):
      s.value++;
```

All happens atomically i.e. wrap pre and post protocols.

Q.10. What are the four necessary conditions of deadlock prevention? **(4)**

Ans:

Four necessary conditions for deadlock prevention:

1. Removing the mutual exclusion condition means that no process may have exclusive access to a resource. This proves impossible for resources that cannot be spooled, and even with spooled resources deadlock could still occur. Algorithms that avoid mutual exclusion are called non-blocking synchronization algorithms.

2. The "hold and wait" conditions may be removed by requiring processes to request all the resources they will need before starting up. Another way is to require processes to release all their resources before requesting all the resources they will need.

3. A "no preemption" (lockout) condition may also be difficult or impossible to avoid as a process has to be able to have a resource for a certain amount of time, or the processing outcome may be inconsistent or thrashing may occur. However, inability to enforce preemption may interfere with a priority algorithm. Algorithms that allow preemption include lock-free and wait-free algorithms and optimistic concurrency control.

4. The circular wait condition: Algorithms that avoid circular waits include "disable interrupts during critical sections", and "use a hierarchy to determine a partial ordering of resources" and Dijkstra's solution.

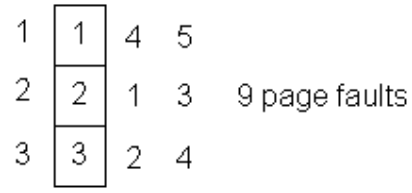
Q.11. Define the following:

(i) FIFO Page replacement algorithm.

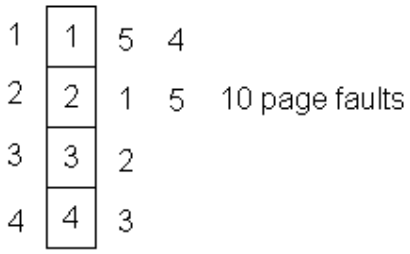
(ii) LRU Page replacement algorithm. **(6)**

Ans:

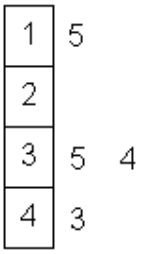
(i) FIFO policy: This policy simply removes pages in the order they arrived in the main memory. Using this policy we simply remove a page based on the time of its arrival in the memory. For example if we have the reference string: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5 and 3 frames (3 pages can be in memory at a time per process) then we have 9 page faults as shown



If frames are increased say to 4, then number of page faults also increases, to 10 in this case.



(ii) LRU policy: LRU expands to least recently used. This policy suggests that we remove a page whose last usage is farthest from current time. For reference string: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5, we have the following page faults



Q.12. List the properties which a hashing function should possess to ensure a good search performance. What approaches are adopted to handle collision? **(8)**

Ans:

A hashing function h should possess the following properties to ensure good search performance:

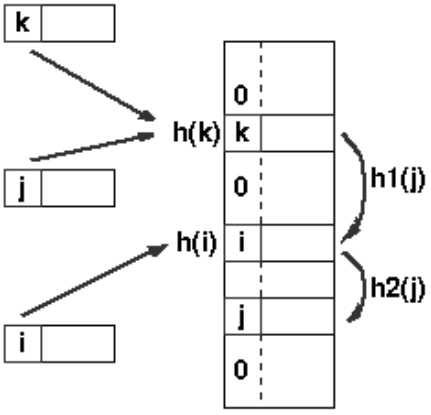
1.The hashing function should not be sensitive to the symbols used in some source program. That is it should perform equally well for different source programs.

2.The hashing function h should execute reasonably fast.

The following approaches are adopted to handle collision are:

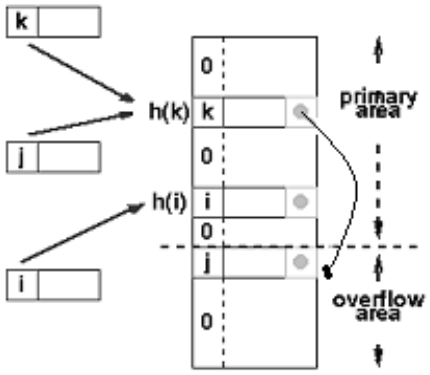
Chaining: One simple scheme is to chain all collisions in lists attached to the appropriate slot. This allows an unlimited number of collisions to be handled and doesn't require *a priori* knowledge of how many elements are contained in the collection. The tradeoff is the same as with linked lists versus array implementations of collections: linked list overhead in space and, to a lesser extent, in time.

Rehashing: Re-hashing schemes use a second hashing operation when there is a collision. If there is a further collision, we *re-hash* until an empty "slot" in the table is found. The re-hashing function can either be a new function or a re-application of the original one. As long as the functions are applied to a key in the same order, then a sought key can always be located.



h(j)=h(k), so the next hash function, **h1** is used. A second collision occurs, so **h2** is used.

Overflow chaining: Another scheme will divide the pre-allocated table into two sections: the *primary area* to which keys are mapped and an area for collisions, normally termed the *overflow area*. When a collision occurs, a slot in the overflow area is used for the new element and a link from the primary slot established as in a chained system. This is essentially the same as chaining, except that the overflow area is pre-allocated and thus possibly faster to access. As with re-hashing, the maximum number of elements must be known in advance, but in this case, two parameters must be estimated: the optimum size of the primary and overflow areas.



Q.13. What is assembly language? What kinds of statements are present in an assembly language program? Discuss. Also highlight the advantages of assembly language. **(8)**

Ans:

Assembly language is a family of low-level language for programming computers, microprocessors, microcontrollers etc. They implement a symbolic representation of the numeric machine codes and other constants needed to program a particular CPU architecture. This representation is usually defined by the hardware manufacturer, and is based on abbreviations (called mnemonic) that help the programmer remember

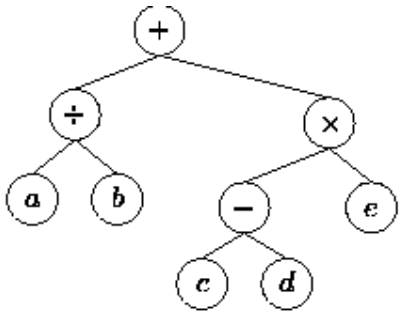
individual instruction, register etc. Assembly language programming is writing machine instructions in mnemonic form, using an assembler to convert these mnemonics into actual processor instructions and associated data.
An assembly program contains following three kinds of statements:
1. **Imperative statements**: These indicate an action to be performed during execution of the assembled program. Each imperative statement typically translates into one machine instruction.
2. **Declaration statements**: The syntax of declaration statements is as follows:
[Label] DS<constant>
[Label] DC '<value>'
The DS statement reserves areas of memory and associates names with them.
The DC statement constructs memory words containing constants.
3. **Assembler directives**: These instruct the assembler to perform certain actions during the assembly of a program. For example
START <constant> directive indicates that the first word of the target program generated by the assembler should be placed in the memory word with address <constant>.
The advantages of assembly language program would be
- reduced errors
- faster translation times
- changes could be made easier and faster

Q.14. What is an expression tree? How an expression is evaluated using an expression tree? Discuss its advantages over the other evaluation techniques. **(8)**

Ans:
Algebraic expressions such as

```
a/b+(c-d)e
```

have an inherent tree-like structure. For example, following figure is a representation of the expression in above equation. This kind of tree is called an *expression tree*.



The terminal nodes (leaves) of an expression tree are the variables or constants in the expression (*a*, *b*, *c*, *d*, and *e*). The non-terminal nodes of an expression tree are the operators (+, -, ×, and ÷)
The expression tree is evaluated using a post-order traversal of the expression tree as follows:
1. If this node has no children, it should return the value of the node
2. Evaluate the left hand child
3. Evaluate the right hand child
4. Then evaluate the operation indicated by the node and return this value

An expression tree is advantageous for:

- Understanding the order of operation. Operations that must be done sooner are further to the right in the tree.
- Counting the number of terms or factors in an expression. Each term or factor is a child node. For example the expression (a+b)/c+2*d contains two terms.

Q.15. Draw an expression tree for the string.

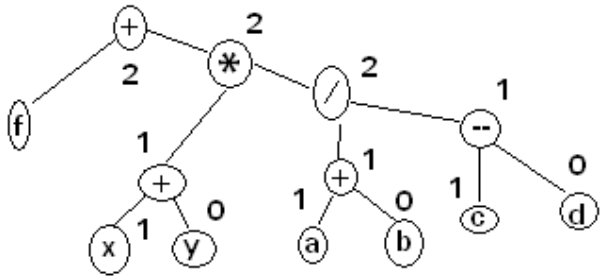
f + (x+y) *((a+b)/(c-d))

Indicate the register requirement for each node and list out the evaluation order for the expression tree. **(8)**

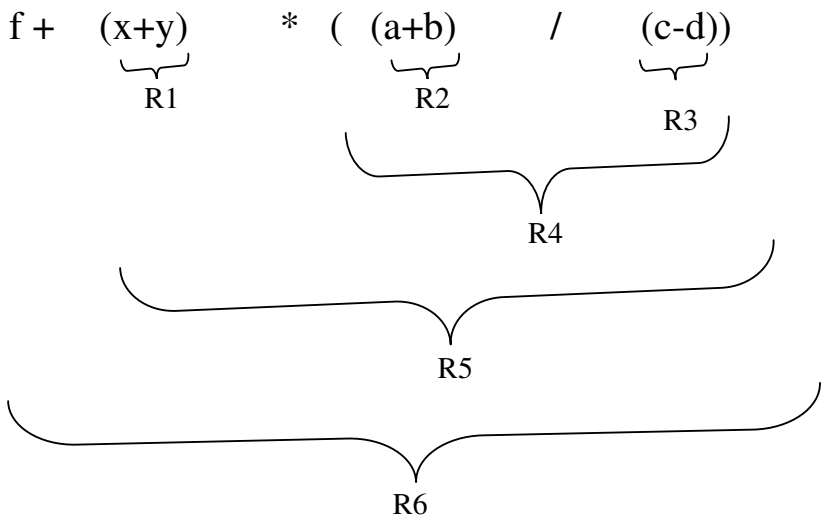
Ans:

An expression tree for the string "f + (x+y) *((a+b)/(c-d))" is given below:

Maximun register requirement is 2.



The expression will be evaluated in the following order: resister R1 first, then register R2, and so on.



Q.16. Explain the following:-

(i) Elimination of common sub expressions during code optimisation.
(ii) Pure and impure interpreters.
(iii) Lexical substitution during macro expansion.
(iv) Overlay structured program.
(v) Facilities of a debug monitor.
(vi) Actions of an interrupt processing routine.
(vii) Real time operating system.
(viii) Fork-join. **(16)**

Ans:

(i) Elimination of common sub expression during code optimization

An optimizing transformation is a rule for rewriting a segment of a program to improve its execution efficiency without affecting its meaning. One of the techniques is "Common sub expression elimination"

In the expression "(a+b)-(a+b)/4", "common subexpression" refers to the duplicated "(a+b)". Compilers implementing this technique realize that "(a+b)" won't change, and as such, only calculate its value once.

(ii) Pure and impure interpreters

In a pure interpreter, the source program is retained in the source form all through its interpretation. This arrangement incurs substantial analysis overheads while interpreting a statement.An impure interpreter performs some preliminary processing of the source program to reduce the analysis overheads during interpretation. The preprocessor converts the program to an intermediate representation (IR), which is used during interpretation. This speeds up interpretation as the code component of the IR i.e the IC, can be analyzed more efficiently than the source form of the program.

(iii)Lexical substitution during macro expansion: Lexical substitution is used to generate an assembly statement from a model statement. A model statement consists of 3 types of strings:

1. An ordinary string, which stands for itself.
2. The name of a formal parameter which is preceded by the character '&'.
3. The name of a preprocessor variable, which is also preceded by the character '&'.

During lexical expansion, strings of type 1 are retained without substitution. Strings of types 2 and 3 are replaced by the 'values' of the formal parameters or preprocessor variables. The value of a formal parameter is the corresponding actual parameter string.

(iv) Overlay structured program: A program containing overlays is referred as overlay structured program where an overlay is a part of program which has the same load origin as some other part(s) of the program. Such a program consists of

1. A permanently resident portion, called the root
2. A set of overlays

The overlay structure of a program is designed by identifying mutually exclusive modules-that is, modules that do not call each other. The basic idea is that such modules do not need to reside simultaneously in memory. Hence they are located in different overlays with the same load origin.

(v) Facilities of a debug monitor are as follows:

a. Setting breakpoints in the program

b. Initiating a debug conversation when control reaches a breakpoint

c. Displaying values of variables

d. Assigning new values to variables

e. Testing user defined assertions and predicates involving program variables.

(vi) Action of an interrupt processing routine are as follows:

1. Save contents of CPU registers. This action is not necessary if the CPU registers are saved by the interrupt action itself.

2. Process the interrupt and take appropriate actions. The interrupt code field of saved PSW information unit corresponding to this interrupt contains useful information for this purpose.

3. Return from interrupt processing.

(vii) Real time operating System

A real-time operating system has well-defined, fixed time constraints. Processing must be done within the defined constraints, or the system will fail. A real time system is considered to function correctly only if it returns the correct result within any time constraints. So the following features are desirable in a real-time operating system:

1. Multi-tasking within an application
2. Ability to define the priorities of tasks
3. Priority driven or deadline oriented scheduling
4. Programmer defined interrupts.

(viii) fork-join are primitives in a higher level programming language for implementing interacting processes. The syntax is as follows:

fork <label>;

join <var>;

where <label> is a label associated with some program statement, and <var> is a variable. A statement fork label1 causes creation of a new process that starts executing at the statement with the label label1. This process is concurrent with the process which executed the statement fork label1.A join statement synchronizes the birth of a process with the termination of one or more processes.

Fork-Join provide a functionally complete facility for control synchronization.

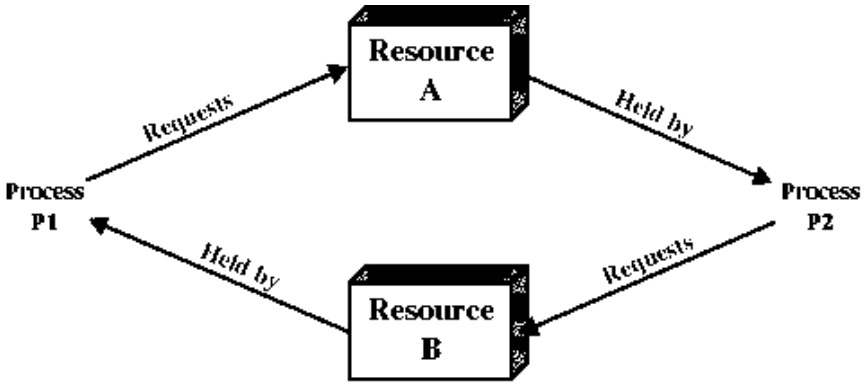
Q.17. List and explain the three events concerning resource allocation. Define the following:

(i) Deadlock.
(ii) Resource request and allocation graph (RRAG)
(iii)Wait for graph (WFG) **(6)**

Ans:

(i) **Deadlock**: Each process in a set of processes is waiting for an event that only a process in the set can cause.

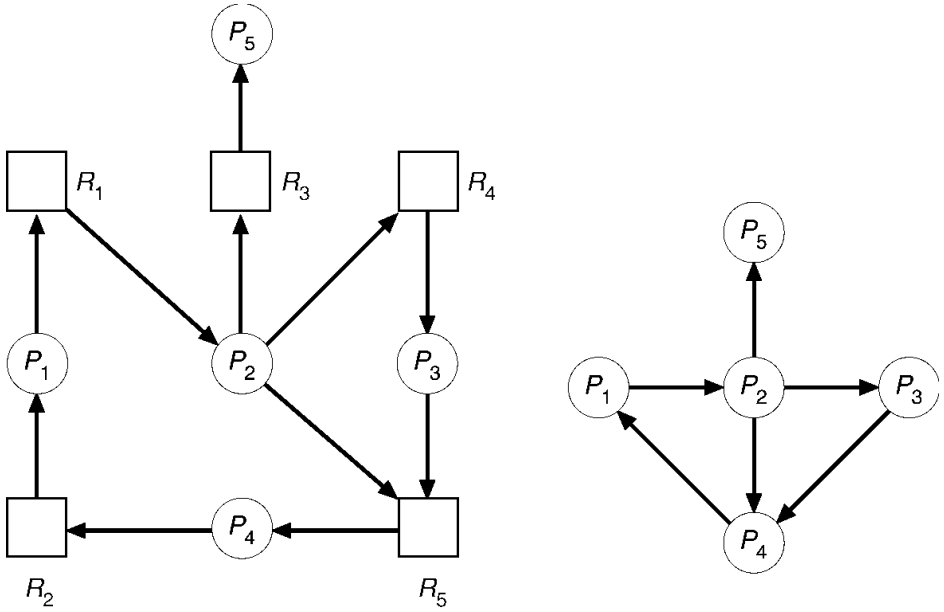
(ii) Deadlocks can be described by a directed bipartite graph called a **Resource-Request-Allocation graph (RRAG)**.A graph G = (V,E) is called bipartite if V can be decomposed into two disjoint sets V_1 and V_2 such that every edge in E joins a vertex in V_1 to a vertex in V_2.Let V_1 be a set of processes and V_2 be a set of resources. Since the graph is directed we will consider:



- an edge (R_j,P_i) (an assignment edge) to mean that resource R_j has been allocated to process P_i
- an edge (P_i,R_j) (called a request edge) to mean that process Pi has requested resource R_j

(iii)

1. Use a resource allocation graph to derive a **wait-for graph**.
2. Wait-for graph obtained by making an edge from p1 to p2 iff p1 is waiting for a resource that is allocated to p2.
3. Deadlock exists iff a cycle exists in resulting wait-for graph.



Q.18. A system contains 10 units of resource class Ru. The resource requirements of three user processes P1, P2 and P3 are as follows

	P1	P2	P3
Maximum requirements	8	7	5
Current allocation	3	1	3
Balance requirements	5	6	2
New request made	1	0	0

Using Banker's algorithm, determine if the projected allocation state is safe and whether the request of P1 will be granted or not. **(6)**

Ans:

From the given data:

Total_alloc=[7]

Total_exist=[10]

The projected allocation state is feasible since the total allocation in it does not exceed the number of resource units of Ru. Since P3 is two units short of its maximum requirements and two unallocated units exits in the system, hence P3 can complete. This will release the resources allocated to it, that is, 5 resources. Now P1 can complete since the number of unallocated units of Ru exceeds the units needed to satisfy its maximum requirement then P2 can be completed. Thus the

processes can finish in the sequence P3, P1, and P2. Hence projected allocation state is safe so algorithm will grant the request made by P1.

Q.19. What is a race condition? Explain how does a critical section avoid this condition. What are the properties which a data item should possess to implement a critical section? **(6)**

Ans:

Race condition: The situation where several processes access – and manipulate shared data concurrently. The final value of the shared data depends upon which process finishes last. To prevent race conditions, concurrent processes must be **synchronized**.

Data consistency requires that only one processes should update the value of a data item at any time. This is ensured through the notion of a critical section. A critical section for data item d is a section of code, which cannot be executed concurrently with itself or with other critical sections for d. Consider a system of n processes (P_0, P_1,…, P_{n-1}), each process has a segment of code called a critical section, in which the proceses may be changing common variables, updating a table, writing a file, and so on. The important feature of the system is that, when one process is executing in its critical section, no other process is to be allowed to execute in its critical section. Thus the execution of critical sections by the processes is mutually exclusive in time.

repeat

Entry section

critical section

Exit section

remainder section

until FALSE

Solution to the Critical Section Problem must satisfy the following three conditions:

1. **Mutual Exclusion**. If process P_i is executing in its critical section, then no other processes can be executing in their critical sections.
2. **Progress**. If no process is executing in its critical section and there exist some processes that wish to enter their critical section, then the selection of the processes that will enter the critical section next cannot be postponed indefinitely.
3. **Bounded Waiting**. A bound must exist on the number of times that other processes are allowed to enter their critical sections after a process has made a request to enter its critical section and before that request is granted.

—Assume that each process executes at a nonzero speed

—No assumption concerning relative speed of the *n* processes.

Q.20. Describe a solution to the Dining philosopher problem so that no races arise. **(4)**

Ans:

A solution to the dining philosopher problem**:**

```
monitor DP
     {
```

```
    enum { THINKING; HUNGRY, EATING) state [5] ;
    condition self [5];
    void pickup (int i) {
           state[i] = HUNGRY;
           test(i);
           if (state[i] != EATING) self [i].wait;
    }
           void putdown (int i) {
           state[i] = THINKING;
                   // test left and right neighbors
            test((i + 4) % 5);
            test((i + 1) % 5);
        }
void test (int i) {
            if ( (state[(i + 4) % 5] != EATING) &&
            (state[i] == HUNGRY) &&
            (state[(i + 1) % 5] != EATING) ) {
                 state[i] = EATING ;
              self[i].signal () ;
             }
     }
        initialization_code() {
           for (int i = 0; i < 5; i++)
           state[i] = THINKING;
    }
}
```

Each philosopher *I* invokes the operations pickup() and putdown() in the following sequence:

```
       dp.pickup (i)
            EAT
        dp.putdown (i)
```

Q.21. Discuss two main approaches to identify and reuse free memory area in a heap. **(6)**

Ans:

Two popular techniques to identify free memory areas as a result of allocation and de-allocations in a heap are:

1. Reference count: the system associates a reference count with each memory area to indicate the number of its active users. This number is incremented when a user accesses that area and decrements when user stops using that. The area is free if the reference counts drops to zero. This scheme is very simple to implement however incurs incremental overheads.

2. Garbage collection: In this technique two passes are made over the memory to identify unused areas. In the first pass it traverses all pointers pointing to allocated areas and marks the memory areas that are in use. The second pass finds all unmarked areas and declares them to be free. The garbage collection overheads are not incremental. They are incurred every time the system runs out of free memory to allocate to fresh requests.

Two main approaches to reuse free memory area in a heap are:

First-fit: Allocate the *first* hole that is big enough. Searching can start either at the beginning of the set of holes or where the previous first-fit search ended. Searching is stopped as soon as a free hole is found that is large enough

Best-fit: Allocate the *smallest* hole that is big enough; Entire list is searched, unless ordered by size. This strategy produces the smallest leftover hole.

Q.22. List the steps needed to perform page replacement. Explain the different page replacement policies. Also list out the main requirements, which should be satisfied by a page replacement policy. **(8)**

Ans:

The steps needed to perform page replacement are:

1.Determine which page is to be removed from the memory.
2.Perform a page-out operation.
3.Perform a page-in operation.

Different page replacement algorithms are briefly described below:

1. First-in, first-out

The first-in, first-out (FIFO) page replacement algorithm is a low-overhead algorithm. Here the operating system keeps track of all the pages in memory in a queue, with the most recent arrival at the back, and the earliest arrival in front. When a page needs to be replaced, the page at the front of the queue (the oldest page) is selected.

Advantage: FIFO is cheap and intuitive.

Disadvantage: 1. Performs poorly in practical application.
2. Suffers from Belady's anomaly.

2. Not recently used

The not recently used (NRU) page replacement algorithm works on the following principle: when a page is referenced, a referenced bit is set for that page, marking it as referenced. Similarly, when a page is modified (written to), a modified bit is set. At a certain fixed time interval, the clock interrupt triggers and clears the referenced bit of all the pages, so only pages referenced within the current clock interval are marked with a referenced bit. When a page needs to be replaced, the operating system divides the pages into four classes:

- Class 0: not referenced, not modified
- Class 1: not referenced, modified
- Class 2: referenced, not modified
- Class 3: referenced, modified.

The NRU algorithm picks a random page from the lowest category for removal.

3. Optimal page replacement algorithm

The optimal page replacement algorithm (also known as OPT)is an algorithm that works as follows: when a page needs to be swapped in, the operating system swaps out the page whose next use will occur farthest in the future. For example, a page that is not going to be used for the next 6 seconds will be swapped out over a page that is going to be used within the next 0.4 seconds.

Disadvantage: This algorithm cannot be implemented in the general purpose operating system because it is impossible to compute reliably how long it will be before a page is going to be used.

The main requirements, which should be satisfied by a page replacement policy, are:

1. Non-interference with the program's locality of reference: The page replacement policy must not remove a page that may be referenced in the immediate future.
2. The page fault rate must not increase with an increase in the memory allocation for a program.

Q.23. What is an I/O buffer? What is the advantage of buffering? Is buffering always effective? Justify your answer with help of an example. **(8)**

Ans:

One kind of I/O requirement arises from devices that have a very high character density such as tapes and disks. With these characteristics, it is not possible to regulate communication with devices on a character-by-character basis. The information transfer, therefore, is regulated in blocks of information. Additionally, sometimes this may require some kind of format control to structure the information to suit the device and/or data characteristics. For instance, a disk drive differs from a line printer or an image scanner. For each of these devices, the format and structure of information is different. It should be observed that the rate at which a device may provide data and the rates at which an end application may consume it might be considerably different. In spite of these differences, the OS should provide uniform and easy to use I/O mechanisms. Usually, this is done by providing a I/O buffer. The OS manages this buffer so as to be able to comply with the requirements of both the producer and consumer of data. Basically, the buffers absorb mismatch in the data transfer rates of processor or memory on one side and device on the other.

Q.24. Discuss the different techniques with which a file can be shared among different users. **(8)**

Ans:

Some popular techniques with which a file can be shared among different users are:

1. Sequential sharing: In this sharing technique, a file can be shared by only one program at a time, that is, file accesses by P1 and P2 are spaced out over time. A lock field can be used to implement this. Setting and resetting of the lock at file open and close ensures that only one program can use the file at any time.

2. Concurrent sharing: Here a number of programs may share a file concurrently. When this is the case, it is essential to avoid mutual interference between them. There are three categories of concurrent sharing:

a. **Immutable files**: If a file is shared in immutable mode, none of the sharing programs can modify it. This mode has the advantage that sharing programs are independent of one another.

b. **Single image immutable files**: Here the changes made by one program are immediately visible to other programs. The Unix file system uses this file-sharing mode.

c. **Multiple image mutable files**: Here many programs can concurrently update the shared file. Each updating program creates a new version of the file, which is different from the version created by concurrent programs. This sharing mode can only be used in applications where concurrent updates and the existence of multiple versions are meaningful.

Q.25. Differentiate between protection and security. Explain the techniques used for protection of user files. **(8)**

Ans:

Operating system consists of a collection of objects, hardware or software. Each object has a unique name and can be accessed through a well-defined set of operations. Protection problem - ensure that each object is accessed correctly and only by those processes that are allowed to do so.

Security must consider external environment of the system, and protect it from:

- Unauthorized access.
- malicious modification or destruction
- Accidental introduction of inconsistency.

It is easier to protect against accidental than malicious misuse.
Protection of user files means that file owner/creator should be able to control: what can be done and by whom. Various categories of access to files are:

- Read
- Write
- Execute
- Append
- Delete
- List

Q.26. What is parsing? Explain any three parsing techniques. **(8)**

Ans

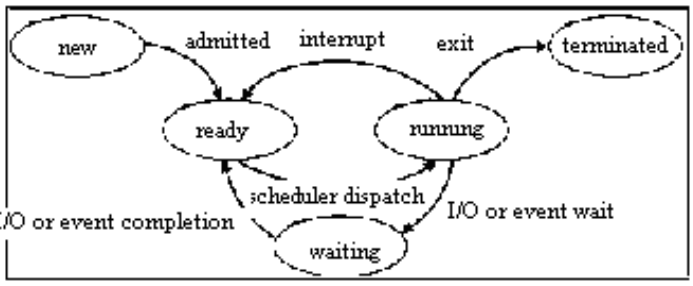
Parsing is the process of analyzing a text, made of a sequence of tokens, to determine its grammatical structure with respect to a given formal grammer. Parsing is also known as syntactic analysis and parser is used for analyzing a text. The task of the parser is essentially to determine if and how the input can be derived from the start symbol of the grammar.
Following are three parsing techniques:
Top-down parsing - Top-down parsing can be viewed as an attempt to find left-most derivations of an input-stream by searching for parse trees using a top-down expansion of the given formal grammar rules. Tokens are consumed from left to right. Inclusive choice is used to accommodate ambiguity by expanding all alternative right-hand-sides of grammar rules.
Bottom-up parsing - A parser can start with the input and attempt to rewrite it to the start symbol. Intuitively, the parser attempts to locate the most basic elements, then the elements containing these, and so on. LR parsers are examples of bottom-up parsers. Another term used for this type of parser is Shift-Reduce parsing.
Recursive descent parsing- It is a top down parsing without backtracking. This parsing technique uses a set of recursive procedures to perform parsing. Salient advantages of recursive descent parsing are its simplicity and generality. It can be implemented in any language supporting recursive procedures.

Q.27. Draw the state diagram of a process from its creation to termination, including all transitions, and briefly elaborate **every state** and **every transition**. **(8)**

Ans:

As a process executes, it changes *state*

- **new**: The process is being created.
- **running**: Instructions are being executed.
- **waiting**: The process is waiting for some event to occur.
- **ready**: The process is waiting to be assigned to a processor.
- **terminated**: The process has finished execution.



Q.28. Consider the following system snapshot using data structures in the Banker's algorithm, with resources A, B, C, and D, and process P0 to P4:

	Max				Allocation				Need				Available			
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
P0	6	0	1	2	4	0	0	1								
P1	1	7	5	0	1	1	0	0								
P2	2	3	5	6	1	2	5	4								
P3	1	6	5	3	0	6	3	3								
P4	1	6	5	6	0	2	1	2								
													3	2	1	1

Using Banker's algorithm, answer the following questions.

(a) How many resources of type A, B, C, and D are there? **(2)**

(b) What are the contents of the Need matrix? **(3)**

(c) Is the system in a safe state? Why **(4)**

(d) If a request from process P4 arrives for additional resources of (1,2,0,0,), can the Banker's algorithm grant the request immediately? Show the new system state and other criteria. **(7)**

Ans:

(a) A-9; B-13;C-10;D-11

(b) Need[i, j]=Max[i,j]-Allocation[i,j] so content of Need matrix is

	A	B	C	D
P0	2	0	1	1
P1	0	6	5	0
P2	1	1	0	2
P3	1	0	2	0
P4	1	4	4	4

(c) The system is in a safe state as the processes can be finished in the sequence P0, P2, P4, P1 and P3.

(d) If a request from process P4 arrives for additional resources of (1,2,0,0,), and if this request is granted, the new system state would be tabulated as follows.

	Max				Allocation				Need				Available			
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
P0	6	0	1	2	4	0	0	1	2	0	1	1				
P1	1	7	5	0	1	1	0	0	0	6	5	0				
P2	2	3	5	6	1	2	5	4	1	1	0	2				
P3	1	6	5	3	0	6	3	3	1	0	2	0				
P4	1	6	5	6	1	4	1	2	0	2	4	4				
													2	0	1	1

After P_O completes P_3 can be allocated. 1020 from released 6012 and available 2011(Total 80 23) and <P_o, P_3, P_4, P_2, P_1> is a safe sequence.

Q.29. Define the following
(i) Process;
(ii) Process Control Block; (PCB)
(iii) Multi programming;
(iv)Time sharing. **(8)**

Ans:
(i) Process: Process is a program in execution; process execution must progress in sequential fashion. A process includes:
- program counter
- stack
- data section

(ii) Process Control Block (PCB): Information associated with each process is stored in Process control Block.
Process state
Program counter
CPU registers
CPU scheduling information
Memory-management information
Accounting information
I/O status information

(iii) Multiprogramming: A multiprogramming operating system is system that allows more than one active user program (or part of user program) to be stored in main memory simultaneously. Multi programmed operating systems are fairly sophisticated. All the jobs that enter the system are kept in the job pool. This pool consists of all processes residing on mass storage awaiting allocation of main memory. If several jobs are ready to be brought into memory, and there is not enough room for all of them, then the system must choose among them. A time-sharing system is a multiprogramming system.

(iv) Time Sharing: Sharing of a computing resource among many users by means of multiprogramming and multi-tasking is known as timesharing. By allowing a large number of users to interact concurrently with a single computer, time-sharing dramatically lowered the cost of providing computing capability, made it possible for individuals and organizations to use a computer without owning one, and promoted the interactive use of computers and the development of new interactive applications.

Q.30. Why are Translation Look-aside Buffers (TLBs) important? In a simple paging system, what information is stored in a typical TLB table entry? **(8)**

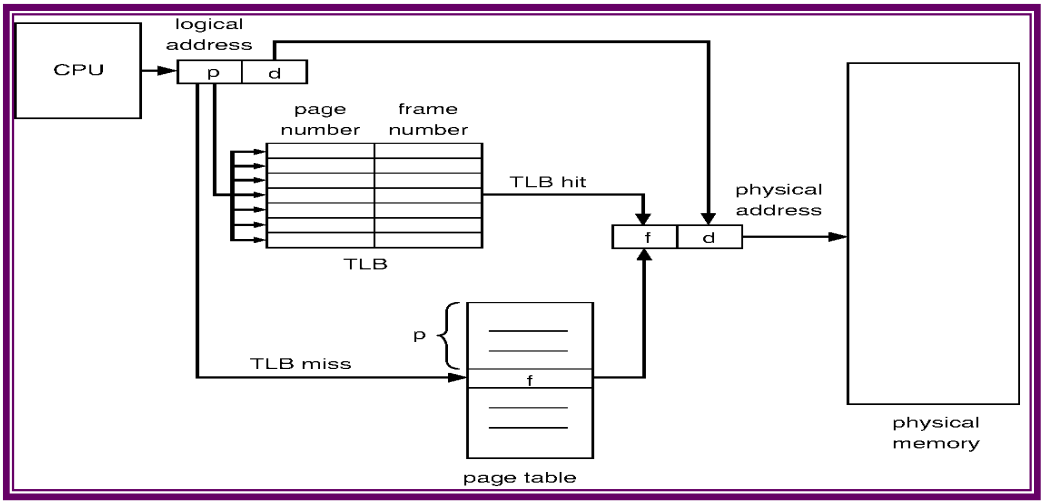
Ans:

The implementation of page-table is done in the following way:

- Page table is kept in main memory.
- *Page-table base register (*PTBR) points to the page table.
- *Page-table length register* (PRLR) indicates size of the page table.
- In this scheme every data/instruction access requires two memory accesses.

One for the page table and one for the data/instruction.

The two-memory access problem can be solved by the use of a special fast-lookup hardware cache called *associative memory* or *translation look-aside buffers (TLBs)*. A set of associative registers is built of high-speed memory where each register consists of two parts: a key and a value. When the associative registers are presented with an item, it is compared with all keys simultaneously. If the item is found, the corresponding value field is the output.

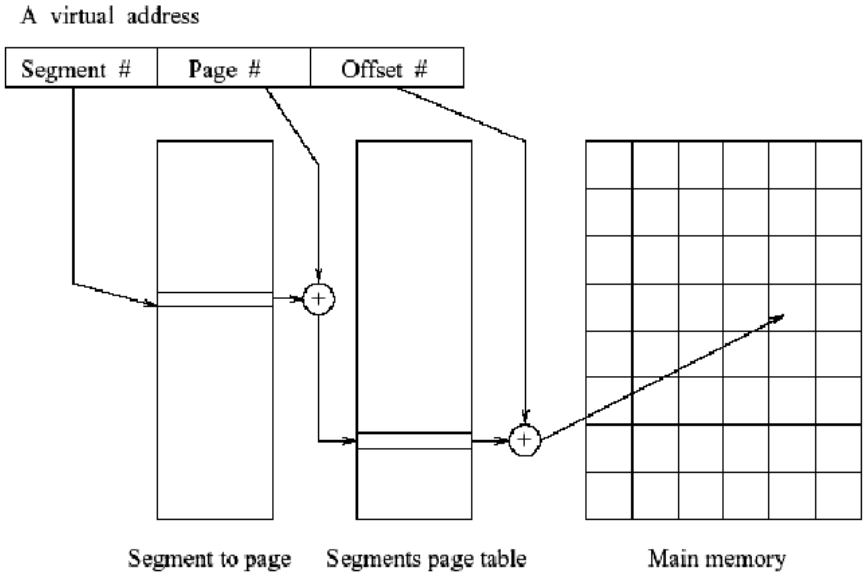


A typical TLB table entry consists of page# and frame#, when a logical address is generated by the CPU, its page number is presented to a set of associative registers that contain page number along with their corresponding frame numbers. If the page number is found in the associative registers, its frame number is available and is used to access memory. If the page number is not in the associated registers, a memory reference to the page table must be made. When the frame number is obtained, it can be used to access memory and the page number along with its frame number is added to the associated registers.

Q.31. Why is segmented paging important (as compared to a paging system)? What are the different pieces of the virtual address in a segmented paging? **(6)**

Ans:

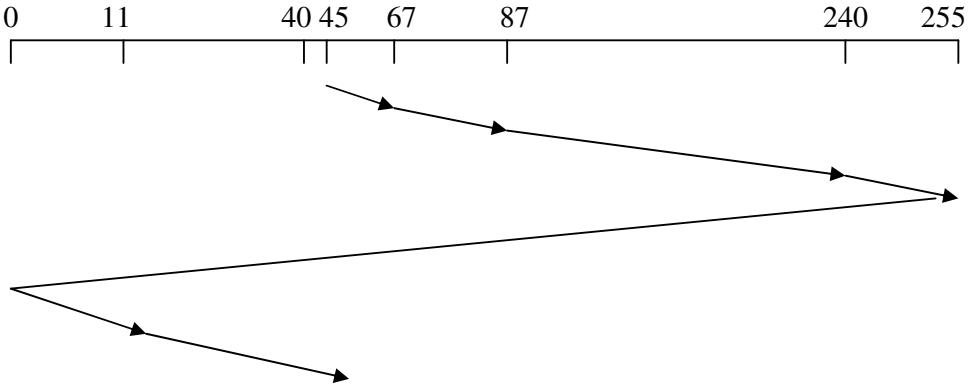
Paging can be superimposed on a segment oriented addressing mechanism to obtain efficient utilization of the memory. This is a clever scheme with advantages of both paging as well as segmentation. In such a scheme each segment would have a descriptor with its pages identified. So we have to now use three sets of offsets. First, a segment offset helps to identify the set of pages. Next, within the corresponding page table (for the segment), we need to identify the exact page table. This is done by using the page table part of the virtual address. Once the exact page has been identified, the offset is used to obtain main memory address reference. The final address resolution is exactly same as in paging. The different pieces of virtual address in a segmented paging is as shown below:



Q.32. Consider the situation in which the disk read/write head is currently located at track 45 (of tracks 0-255) and moving in the positive direction. Assume that the following track requests have been made in this order: 40, 67, 11, 240, 87. What is the order in which optimised C-SCAN would service these requests and what is the total seek distance? **(6)**

Ans:

Disk queue: 40, 67, 11, 240, 87 and disk is currently located at track 45.The order in which optimised C-SCAN would service these requests is shown by the following diagram.



Total seek distance=(67-45)+(87-67)+(240-87)+(255-240)+255+(11-0)+(40-11)
=22+20+153+15+255+11+29
=505

Q.33. Explain any three policies for process scheduling that uses resource consumption information. What is response ratio? **(8)**

Ans:

Three policies for process scheduling are described below in brief:

1. First-come First-served (FCFS) (FIFO)

– Jobs are scheduled in order of arrival
– Non-preemptive

• **Problem:**

– Average waiting time can be large if small jobs wait behind long ones
– May lead to poor overlap of I/O and CPU and convoy effects

2. Shortest Job First (SJF)

– Choose the job with the shortest next CPU burst
– Provably optimal for minimizing average waiting time

• **Problem:**

– Impossible to know the length of the next CPU burst

3. Round Robin(RR)

– Often used for timesharing
– Ready queue is treated as a circular queue (FIFO)
– Each process is given a time slice called a *quantum*
– It is run for the quantum or until it blocks
– RR allocates the CPU uniformly (fairly) across all participants. If average queue length is n, each participant gets 1/n
– As the time quantum grows, RR becomes FCFS
– Smaller quanta are generally desirable, because they improve response time

• **Problem:**

– Context switch overhead of frequent context switch

Highest Response Ratio Next (HRRN) scheduling is a non-preemptive discipline, similar to Shortest Job First (SJF) in which the priority of each job is dependent on its estimated run time, and also the amount of time it has spent waiting. Jobs gain higher priority the longer they wait which prevents indefinite postponement . In fact, the jobs that have spent a long time waiting compete against those which are estimated to have short run times.

$$Priority = \frac{waiting\ time + estimated\ run\ time}{estimated\ run\ time} = 1 + \frac{waiting\ time}{estimated\ run\ time}$$

Q.34. What is a semaphore? Explain a binary semaphore with the help of an example? **(4)**

Ans:

A **semaphore** is a synchronization tool that provides a general-purpose solution to controlling access to critical sections.

A semaphore is an abstract data type (ADT) that defines a nonnegative integer variable which, apart from initialization, is accessed only through two standard operations: wait and signal. The classical definition of wait in pseudo code is

```
wait(S){
      while(S<=0)
            ; // do nothing
      S--;
}
The classical definitions of signal in pseudocode is
signal(S){
      S++;
}
```

A binary semaphore is one that only takes the values 0 and 1. These semaphores are used to implement mutual exclusion.

Q.35. Consider the following page reference and reference time strings for a program:
Page reference string: 5,4,3,2,1,4,3,5,4,3,2,1,5,…..
Show how pages will be allocated using the FIFO page replacement policy. Also calculate the total number of page faults when allocated page blocks are 3 and 4 respectively. **(8)**

Ans:
Page reference string is: 5,4,3,2,1,4,3,5,4,3,2,1,5,…..
For allocated page blocks 3, we have following FIFO allocation. Page reference marked with '+' cause page fault and result in page replacement which is performed by replacing the earliest loaded page existing in memory:

		3	3	3	4	4	4	4	4	2	2	2
	4	4	4	1	1	1	5	5	5	5	5	5
5	5	5	2	2	2	3	3	3	3	3	1	1

5^+	4^+	3^+	2^+	1^+	4^+	3^+	5^+	4	3	2^+	1^+	5

Page Reference

For allocated page blocks 4, we have following FIFO allocation. Page reference marked with '+' cause page fault and result in page replacement.

			2	2	2	2	2	2	3	3	3	3
		3	3	3	3	3	3	4	4	4	4	5
	4	4	4	4	4	4	5	5	5	5	1	1
5	5	5	5	1	1	1	1	1	1	2	2	2

5^+	4^+	3^+	2^+	1^+	4	3	5^+	4^+	3^+	2^+	1^+	5^+

Total number of page faults =10 when allocated page blocks=3
Total number of page faults =11 when allocated page blocks=4

Q.36. What are the different parameter passing mechanisms to a function? Explain with the help of example? **(8)**

Ans:

The various parameter-passing mechanisms are:
1. Call by value
2. Call by value-result
3. Call by reference
4. Call by name

In **call by value** mechanism, the values of actual parameters are passed to the called function. These values are assigned to the corresponding formal parameters. If a function changes the value of a formal parameter, the change is not reflected on the corresponding actual parameter. This is commonly used for built-in functions of the

language. Its main advantage is its simplicity. The compiler can treat formal parameter as a local variable. This simplifies compilation considerably.

Call by value-result: This mechanism extends the capabilities of the call by value mechanism by copying the values of formal parameters back into corresponding ctual parameters at return. This mechanism inherits the simplicity of the call by value mechanism but incurs higher overheads.

Call by reference: Here the address of an actual parameter is passed to the called function. If the parameter is an expression, its value is computed and stored in a temporary location and the address of the temporary location is passed to the called function. If the parameter is an array element, its address is similarly computed at the time of call. This mechanism is very popular because it has 'cleaner' semantics than call by value-result.

Call by name: This parameter transmission mechanism has the same effect as if every occurrence of a formal parameter in the body of the called function is replaced by the name of the corresponding actual parameter. The actual parameter corresponding to a formal parameter can change dynamically during the execution of a function. This makes the call by name mechanism immensely powerful. However the high overheads make it less attractive in practice.

Q.37. What is meant by inter process communication? Explain the two fundamental models of inter process communication. **(8)**

Ans:

Inter process Communication: The OS provides the means for cooperating processes to communicate with each other via an inter process communication (IPC) facility.

IPC provides a mechanism to allow processes to communicate and to synchronize their actions without sharing the same address space. IPC is particularly useful in a distributed environment where the communicating processes may reside on different computers connected with a network.

IPC is best implemented by message passing system where communication among the user processes is accomplished through the passing of messages. An IPC facility provides at least the two operations:

send(message) and receive(message).

Two types of message passing system are as follows:

(a) **Direct Communication:** With direct communication, each process that wants to communicate must explicitly name the recipient or sender of the communication. In this scheme, the send and receive primitives are defined as:

- send(P, message)- Send a message to process P.
- receive(Q, message)- Receive a message from process Q.

A communication link in this scheme has the following properties:

- A link is established automatically between every pair of processes that want to communicate. The processes need to know only each other's identity to communicate.
- A link is associated with exactly two processes.
- Exactly one link exists between each pair of processes.

(b)With indirect communication, the messages are sent to and received from mailboxes, or ports. Each mailbox has a unique identification. In this scheme, a process can communicate with some other process via a number of different

mailboxes. Two processes can communicate only if they share a mailbox. The send and receive primitives are defined as follows:

- send (A, message)- Send a message to mailbox A
- receive (A, message)- Receive a message from mailbox A.

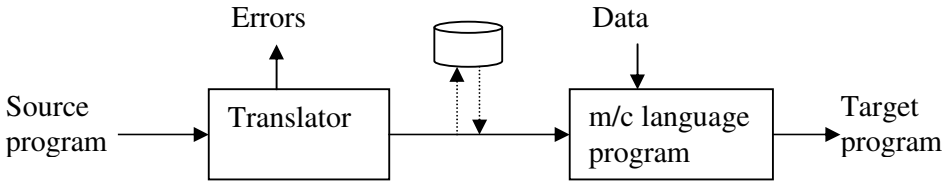
In this scheme, a communication link has the following properties:

- A link is established between a pair of processes only if both members of the pair have a shared mailbox.
- A link may be associated with more than two processes.
- A number of different links may exist between each pair of communicating processes, with each link corresponding to one mailbox.

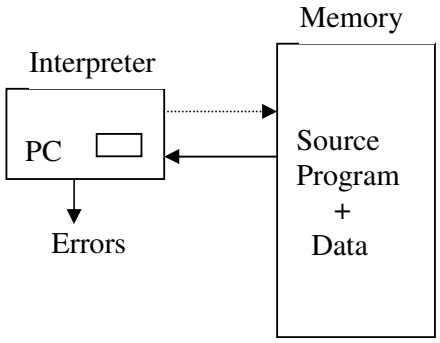
Q.38. Differentiate between program translation and program interpretation. **(6)**

Ans:

The **program translation model** bridges the execution gap by translating a program written in a programming language, called the source program (SP), into an equivalent program in the machine or assembly language of the computer system, called the target program (TP). Following diagram depicts the program translation model.



In a program interpretation process, the interpreter reads the source program and stores it in its memory. It bridges an execution gap without generating a machine language program so we can say that the interpreter is a language translator. However, it takes one statement of higher-level language at a time, translates it into machine language and executes it immediately. Translation and execution are carried out for each statement. The absence of a target program implies the absence of an outer interface of the interpreter. Thus language-processing activity of an interpreter cannot be separated from its program execution activities. Hence we can say that interpreter executes a program written in a programming language. In essence, the execution gap vanishes. Following figure depicts the program interpretation model.



Characteristics of the program translation model are:

- A program must be translated before it can be executed.
- The translated program may be saved in a file. The saved program may be executed repeatedly.
- A program must be retranslated following modifications.

Characteristics of the program interpretation model:

- The source program is retained in the source form itself i.e. no target program form exists.
- A statement is analyzed during its interpretation.

Q.39. Explain the differences between macros and subroutines. **(4)**

Ans:

Macros Vs Subroutines

(i) Macros are pre processor directives i.e. it is processed before the source program is passed to the compiler.

Subroutines are blocks of codes with a specific task, to be performed and are directly passed to the compiler.

(ii) In a macro call the pre processor replaces the macro template with its macro expansion, in a literal way.

As against this, in a function call the control is passed to a function along with certain arguments, some calculations are performed in the function and a useful value is returned back from the function.

(iii) Macro increases the program size. For example, if we use a macro hundred times in a program, the macro expansion goes into our source code at hundred different places. Whereas, functions make the program smaller and compact. For example, if a function is used, the even if it is called from hundred different places in the program, it would take the same amount of space in the program.

(iv) Macros make the program run faster as they have already been expanded and placed in the source code before compilation. Whereas, passing arguments to a function and getting back the returned values does take time and would therefore slow down the program.

(v) Example of macro

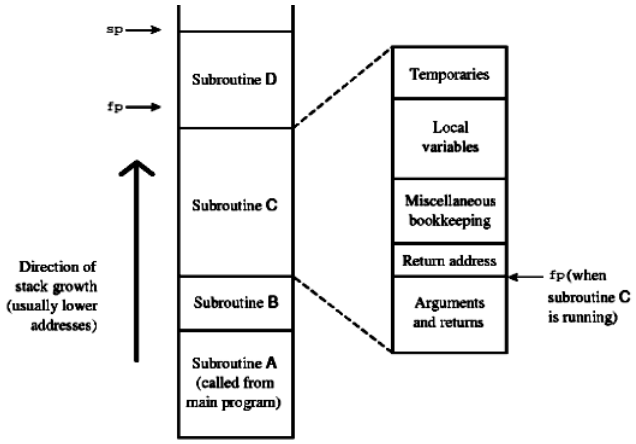
```
#define AREA(x) (3.14*x*x) // macro definition
main(){
  float r1=6.25, r2=2.5, a;
  a=AREA(r1); // expanded to (3.14 * r1 * r1)
  printf("\n Area of circle =%f",a);
  a=AREA(r2); // // expanded to (3.14 * r2 * r2)
 printf("\n Area of circle= %f",a);}
Example of subroutine
main(){
  float r1=6.25, r2=2.5, a;

  a=AREA(r1); // calls AREA()
  printf("\n Area of circle =%f",a);
  a=AREA(r2); // calls AREA()
 printf("\n Area of circle= %f",a);}
float AREA(float r) // subroutine{
  return 3.14*r*r;}
```

Q.40. Explain the stack storage allocation model. **(6)**

Ans:

In **stack-based allocation**, objects are allocated in last-in, first-out data structure, a *stack*.–*E.g.* Recursive subroutine parameters. The stack storage allocation model

- Grow and shrink on procedure calls and returns.
- Register allocation works best for stack-allocated objects.
- Memory allocation and freeing are partially predictable.
- Restricted but simple and efficient.
- Allocation is hierarchical: Memory freed in opposite order of allocation. That is If alloc (A) then alloc (B) then alloc (C), then it must be free(C) then free(B) then free(A).



Q.41. Give an account of the issue pertaining to compilation of if statement in C language (6)

Ans

Control structures like if cause significant gap between the PL domain and the execution domain because the control transfers are implicit rather than explicit. The semantic gap is bridged in two steps as follows:

1. Control structure is mapped into an equivalent program containing explicit goto's. The compiler generates its own labels and put them against the appropriate statements. For example, the equivalent of (a) given below is (b) where int1, int2 are labels generated by compiler for its own purposes.

```
if (e1) then                    if(e1) then goto int1;
    S1;                             S2;
else                                goto int2;
    S2;                         int1:S1;
S3;                             int2:S3;
```

(a) **(b)**

2. These programs are translated into assembly programs.

Q.42. Differentiate between non-relocatable, relocatable and self relocatable programs. (6)

Ans:

A **non- relocatable program** is one that cannot be executed in any memory area other than the area starting on its translated origin. For example a hand coded machine language program.

A relocatable program is one that can be processed to relocate it to a desired area of memory. For example an object module. The difference between a relocatable and a non-relocatable program is the availability of information concerning the address sensitive instructions in it. A self-relocating program is the one that can perform the relocation of its own address sensitive instructions. **A self-relocating program can** execute in any area of memory. This is very important in time-sharing operating system where load address of a program is likely to be different for different executions.

Q.43. Explain briefly any three of the commonly used code optimisation techniques. (6)

Ans:

1.Common sub expression elimination:

In the expression "(a+b)-(a+b)/4", "common sub expression" refers to the duplicated "(a+b)". Compilers implementing this technique realize that "(a+b)" won't change, and as such, only calculate its value once and use the same value next time.

2.Dead code Elimination:

Code that is unreachable or that does not affect the program (e.g. dead stores) can be eliminated. In the example below, the value assigned to i is never used, and the dead store can be eliminated. The first assignment to global is dead, and the third assignment to global is unreachable; both can be eliminated.

```
int global;
void f (){
  int i;
  i = 1;          /* dead store */
  global = 1;     /* dead store */
  global = 2;
  return;
  global = 3;     /* unreachable */}
       Below is the code fragment after dead code
elimination.
int global;
void f (){
  global = 2;
  return;}
```

3. Loop-invariant code motion

If a quantity is computed inside a loop during every iteration, and its value is the same for each iteration, it can vastly improve efficiency to hoist it outside the loop and compute its value just once before the loop begins. This is particularly important with the address-calculation expressions generated by loops over arrays. For correct implementation, this technique must be used with loop inversion, because not all code is safe to be hoisted outside the loop.

for example:

```
for (i=1; i≤10, i++){
x=y*2-(1)
.
.
.
}
```

statement 1 can be hoisted outside the loop assuming value of x and y does not change in the loop.

Q.44. Write short notes on:

i. YACC.

ii. Debug monitors. **(8)**

Ans:

(i)YACC stands for "Yet another Compiler-Compiler" : Computer program input generally has some structure; in fact, every computer program that does input can be thought of as defining an "input language" which it accepts. An input language may be as complex as a programming language, or as simple as a sequence of numbers. Unfortunately, usual input facilities are limited, difficult to use, and often are lax about checking their inputs for validity.

YACC provides a general tool for describing the input to a computer program. The YACC user specifies the structures of his input, together with code to be invoked as each such structure is recognized. YACC turns such a specification into a subroutine that handles the input process; frequently, it is convenient and appropriate to have most of the flow of control in the user's application handled by this subroutine. The output from YACC is LALR parser for the input programming laughing

(ii)Debug monitors provide debugging support for a program. A debug monitor executes the program being debugged under its own control thereby providing execution efficiency during debugging. There are debug monitors that are language independent and can handle programs written in many languages. For example-DEC-10. Debug monitor provide the following facilities for dynamic debugging:

1. Setting breakpoints in the program
2. Initiating a debug conversation when control reaches a breakpoint.
3. Displaying values of variables
4. Assigning new values to variables.
5. Testing user defined assertions and predicates involving program variables.

Q.45. What is an operating system? List the typical functions of operating systems. (4)

Ans:

An **operating system** is system software that provides interface between user and hardware. The operating system provides the means for the proper use of resources (CPU, memory, I/O devices, data and so on) in the operation of the computer system.

An operating system provides an environment within which other programs can do useful work.

Typical functions of operating system are as follows:

(1)Process management: A process is a program in execution. It is the job, which is currently being executed by the processor. During its execution a process would require certain system resources such as processor, time, main memory, files etc. OS supports multiple processes simultaneously. The process management module of

the OS takes care of the creation and termination of the processes, assigning resources to the processes, scheduling processor time to different processes and communication among processes.

(2)Memory management module: It takes care of the allocation and deallocation of the main memory to the various processes. It allocates main and secondary memory to the system/user program and data. To execute a program, its binary image must be loaded into the main memory.

Operating System decides.

(a) Which part of memory are being currently used and by whom.

(b) which process to be allocated memory.

(c) Allocation and de allocation of memory space.

(3)I/O management: This module of the OS co-ordinates and assigns different I/O devices namely terminals, printers, disk drives, tape drives etc. It controls all I/O devices, keeps track of I/O request, issues command to these devices.

I/O subsystem consists of

(i) Memory management component that includes buffering, caching and spooling.

(ii) Device driver interface

(iii) Device drivers specific to hardware devices.

(4)File management: Data is stored in a computer system as files. The file management module of the OS would manage files held on various storage devices and transfer of files from one device to another. This module takes care of creation, organization, storage, naming, sharing, backup and protection of different files.

(5)Scheduling: The OS also establishes and enforces process priority. That is, it determines and maintains the order in which the jobs are to be executed by the computer system. This is so because the most important job must be executed first followed by less important jobs.

(6)Security management: This module of the OS ensures data security and integrity. That is, it protects data and program from destruction and unauthorized access. It keeps different programs and data which are executing concurrently in the memory in such a manner that they do not interfere with each other.

(7)Processor management: OS assigns processor to the different task that must be performed by the computer system. If the computer has more than one processor idle, one of the processes waiting to be executed is assigned to the idle processor.

OS maintains internal time clock and log of system usage for all the users. It also creates error message and their debugging and error detecting codes for correcting programs.

Q.46. What are interrupts? How are they handled by the operating system? (5)

Ans:

Interrupt: An interrupt is a hardware mechanism that enables an external device, typically I/O devices, to send a signal to the CPU. An interrupt signal requests the CPU to interrupt its current activities and attend to the interrupting device's needs. A CPU will check interrupts only after it has completed the processing of one instruction and before it fetches a subsequent one. The basic interrupt mechanism works as follows:

The CPU hardware has wire called the interrupt-request line that the CPU senses after executing instruction. The device controller raises an interrupt by asserting a signal on the interrupt request line. CPU detects that a controller has asserted a signal on the interrupt request line.

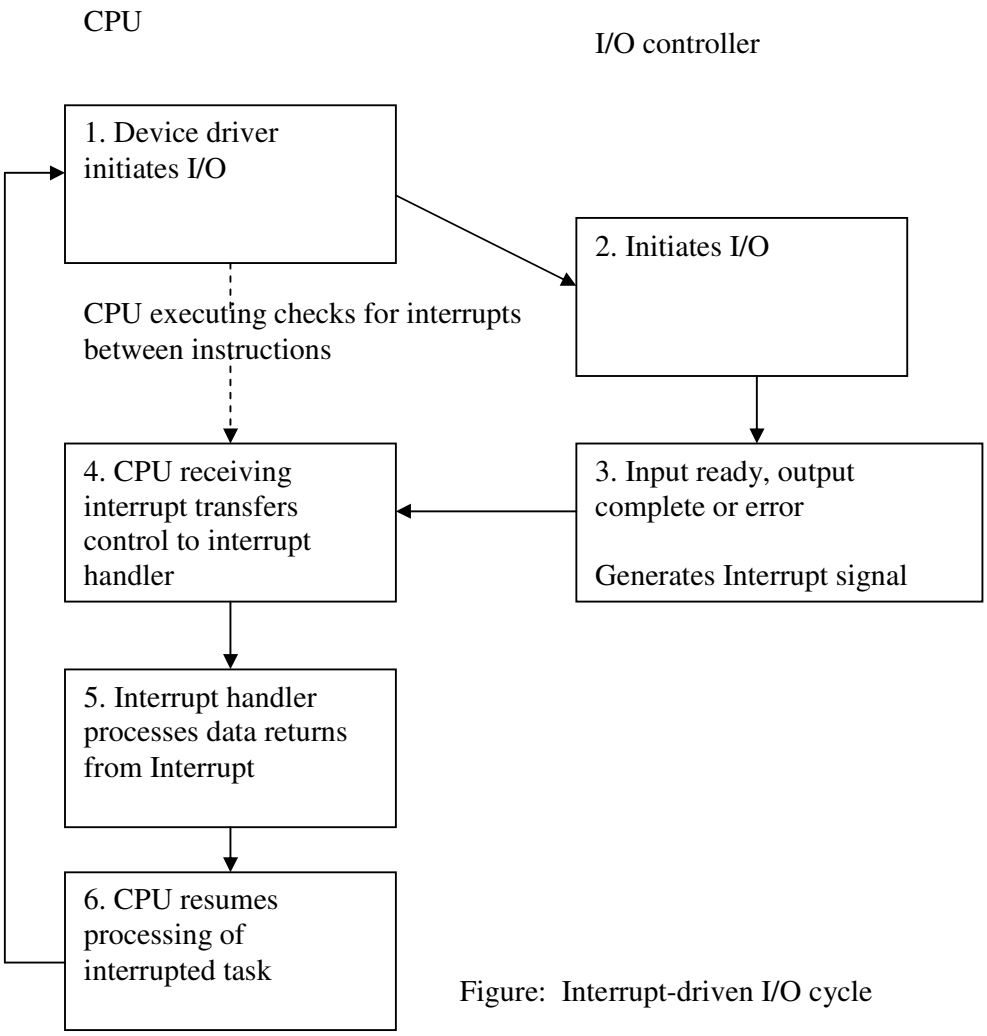


Figure: Interrupt-driven I/O cycle

The CPU saves small amount of state, such as the current value of the instruction pointer i.e. return address, and jumps to the interrupt handler routine at fixed address in memory. The interrupt handler determines the cause of the interrupt, performs the necessary processing, and executes a return from interrupt instruction, thereby clearing the interrupt. The CPU resumes to the execution state prior to the interrupt.

Q.47. Define process. Describe the contents of a Process Control Block (PCB). (5)

Ans:

Process: A process is a program in execution.

A process is an active entity, represented by the value of the program counter and the contents of the processor's registers. A process generally includes the process stack, which contains temporary data (such as method parameters, return addresses, and local variables. Two processes (may or may not be associated with the same program) are two separate execution sequences with its own text and data sections. A process may spawn many processes as it runs. Process Control Block (PCB): Each process is represented in the operating system by a process control block or task control block.

It contains many pieces of information associated with a specific process such as:

(1) Process state: The state may be new, ready, running, waiting, halted and so on.

(2) Program counter: The counter indicates the address of the next instruction to be executed for this process.

(3) CPU registers: The registers vary in number and type, depending on the computer architecture. They include accumulators, index registers, stack pointers and general-purpose registers, plus any condition-code information. Along with the program counter, this state information must be saved when and interrupt occurs, to allow the process to be continued correctly afterward.

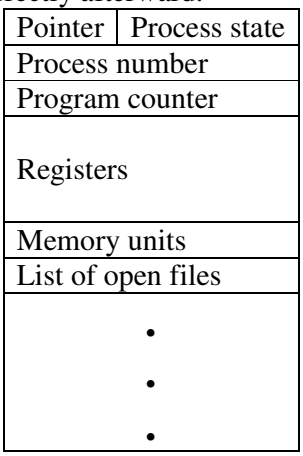


Figure: Process Control Block(PCB)

(4) CPU-scheduling information: This information includes a process priority, pointers to scheduling queues, and any other scheduling parameters.

(5) Memory-management information: This information may include such information as the value of the base and limit registers, the page tables, or the segment tables, depending on the memory system used by the OS.

(6) Accounting information: This information includes the amount of CPU and real time used, time limits, account numbers, job or process numbers and so on.

(7) I/O status information: The information includes the list of I/o devices allocated to this process, a list of open files, and so on.

The PCB simply serves as the repository for any information that may vary from process to process

Q48. What are interacting processes? Explain any two methods of implementing interacting processes. (8)

Ans:

Interacting processes: The concurrent processes executing in the operating system are interacting or cooperating processes if they can be affected by each other.

Any process that shares data with other processes is an interacting process.

Two methods of implementing interacting process are as follows:

(i) **Shared memory solution:** This scheme requires that these processes share a common buffer pool and the code for implementing the buffer be written by the application programmer.

For example, a shared-memory solution can be provided to the bounded-buffer problem. The producer and consumer processes share the following variables:

```
#define BUFFER_SIZE 10
Typedef struct{
   ……. .
```

```
}item;
Item buffer[BUFFER_SIZE];
int in=0;
int out=0;
```

The shared buffer is implemented as a circular array with two logical pointers: in and out. The variable in points to the next free position in the buffer; out points to the first full position in the buffer. The buffer is empty when in==out; the buffer is full when ((in + 1)%BUFFER_SIZE)==out.

The producer process has a local variable nextProduced in which the new item to be produced is stored:

```
while(1){
    /* produce and item in nextProduced */
    While(((in + 1)%BUFFER_SIZE)==out)
     ; // do nothing
     Buffer[in]=nextProduced;
     in =(in+1)% BUFFER_SIZE;}
```

The consumer process has a local variable nextConsumed in which the item to be consumed is stored:

```
while(1){
  while(in==out)
  ; //do nothing
 nextConsumed = buffer[out];
 out=(out +1)% BUFFER_SIZE;
/* consume the item in nextConsumed */}
```

(ii) **Inter process Communication:** The OS provides the means for cooperating processes to communicate with each other via an interprocess communication (IPC) facility. IPC provides a mechanism to allow processes to communicate and to synchronize their actions without sharing the same address space. IPC is particularly useful in a distributed environment where the communicating processes may reside on different computers connected with a network. IPC is best implemented by message passing system where communication among the user processes is accomplished through the passing of messages. An IPC facility provides at least the two operations: send(message) and receive(message).

Some types of message passing system are as follows:

Direct or Indirect Communication: With direct communication, each process that wants to communicate must explicitly name the recipient or sender of the communication. In this scheme, the send and receive primitives are defined as:

- send(P, message)- Send a message to process P.
- receive(Q, message)- Receive a message from process Q.

A communication link in this scheme has the following properties:

- A link is established automatically between every pair of processes that want to communicate. The processes need to know only each other's identity to communicate.
- A link is associated with exactly two processes.
- Exactly one link exists between each pair of processes.

With indirect communication, the messages are sent to and received from mailboxes, or ports. Each mailbox has a unique identification. In this scheme, a process can communicate with some other process via a number of different

mailboxes. Two processes can communicate only if they share a mailbox. The send and receive primitives are defined as follows:

- send (A, message)- Send a message to mailbox A
- receive (A, message)- Receive a message from mailbox A.

In this scheme, a communication link has the following properties:

- A link is established between a pair of processes only if both members of the pair have a shared mailbox.
- A link may be associated with more than two processes.
- A number of different links may exist between each pair of communicating processes, with each link corresponding to one mailbox.

Q.49. Consider the following set of jobs with their arrival times, execution time (in minutes), and deadlines.

Job Ids	Arrival Time	Execution time	Deadline
1	0	5	5
2	1	15	25
3	3	12	10
4	7	25	50
5	10	5	12

Calculate the mean turn-around time, the mean weighted turn-around time and the throughput for FCFS, SJN and deadline scheduling algorithms. (6)

Ans:

Chart for First Come First Served scheduling

1	2	3	4	5
0 5	20	32	57	62

Turnaround time = Terminated time – Arrival time

i.e, T = Tr - Ta

So, turnaround time for various jobs are

For job 1, T1 = 5-0=5 unit time

For job 2, T2 = 20-1=19 unit time

For job 3, T3 = 32-3=29 unit time

For job 4, T4 = 57-7=50 unit time

For job 5, T5 = 62-10=52 unit time

Mean turnaround time, Tm = (T1+T2+T3+T4+T5)/5= 155/5=31 unit time/job

Throughput= no of process completed per unit time= 5/62= 0.081 jobs/unit time

Chart for Shortest Job Next scheduling

1	3	5	2	4
0 5	17	22	37	62

Turnaround time for various jobs are

For job 1, T1 = 5-0=5 unit time

For job 2, T2 = 37-1=36 unit time

For job 3, T3 = 17-3=14 unit time
For job 4, T4 = 62-7=55 unit time
For job 5, T5 = 22-10=12 unit time
Mean turnaround time, Tm = (T1+T2+T3+T4+T5)/5= 122/5= 24.4unit time/job
Throughput= no of process completed per unit time= 5/62= 0.081 jobs/unit time

Q.50. What are the differences between user level threads and kernel supported threads? (4)

Ans:

A **thread,** sometimes called a lightweight process(LWP), is a basic unit of CPU utilization; it comprises a thread ID, a program counter, a register set and a stack.
A thread shares with other threads belonging to the same process its code section, data section and other operating-system resources, such as open files and signals.
If the process has multiple threads of control, it can do more than one task at a time.
User Level Threads Vs Kernel Supported Threads

i. User threads are supported above the kernel and are implemented by a thread library at the user level.
Whereas, kernel threads are supported directly by the operating system.

ii. For user threads, the thread library provides support for thread creation, scheduling and management in user space with no support from the kernel as the kernel is unaware of user-level threads. In case of kernel threads, the kernel performs thread creation, scheduling and management in kernel space.

iii. As there is no need of kernel intervention, user-level threads are generally fast to create and manage. As thread management is done by the operating system, kernel threads are generally slower to create and manage that are user threads.

iv. If the kernel is single-threaded, then any user-level thread performing blocking system call, will cause the entire process to block, even if other threads are available to run within the application.
However, since the kernel is managing the kernel threads, if a thread performs a blocking system call, the kernel can schedule another thread in the application for execution.

v. User-thread libraries include POSIX P threads, Mach C-threads and Solaris 2 UI-threads.

Some of the cotemporary operating systems that support kernel threads are Windows NT, Windows 2000, Solaris 2, BeOS and Tru64 UNIX(formerly Digital UNIX).

Q.51. Define deadlock? Explain the necessary conditions for deadlock to occur. (5)

Ans:

Deadlock is a situation, in which processes never finish executing and system resources are tied up, preventing other jobs from starting. A process requests resources; if the resources are not available at that time, the process enters a wait state. Waiting processes may never again change state, because the resources they have requested are held by other waiting processes, thereby causing deadlock.Necessary conditions for deadlock to occur are:

i. **Mutual exclusion:** At least one resource must be held in a nonsharable mode; that is, only one process at a time can use the resource. If another process requests

that resource, the requesting process must be delayed until the resource has been released.

ii. **Hold and wait:** A process must be holding at least one resource and waiting to acquire additional resources that are currently being held by other processes.

iii. **No pre-emption:** Resources cannot be pre-empted; that is, a resource can be released only voluntarily by the process holding it, after the process holding it has completed its task.

iv. **Circular wait:** A set{P0, P1,……, Pn) of waiting processes must exist such that P0 is waiting for a resource that is held by P1, P1 is waiting for a resource that is held by P2, ……., Pn-1 is waiting for a resource that is held by Pn and Pn is waiting for a resource that is held by P0.

All four conditions must hold simultaneously for a deadlock to occur and conditions are not completely independent. For example, the circular-wait implies the hold-and-wait condition.

Q52. An operating system contains 3 resource classes. The number of resource units in these classes is 7, 7 and 10. The current resource allocation state is shown below:

Processes	Allocated resources			Maximum requirements		
	R1	R2	R3	R1	R2	R3
P1	2	2	3	3	6	8
P2	2	0	3	4	3	3
P3	1	2	4	3	4	4

(i) Is the current allocation state safe?

(ii) Can the request made by process P1 (1, 1, 0) be granted? (5)

Ans:

(i) In the given question,

Available matrix for resources [R1 R2 R3] = No of resource unit -Total Allocation = [7 7 10]-[5 4 10]= [2 3 0]

Need matrix is defined as (Max – Allocation),

Processes	Need of resources		
	R1	R2	R3
P1	1	4	5
P2	2	3	0
P3	2	2	0

Using Safety Algorithm, we get sequence:

Processes	Available resources after satisfying need		
	R1	R2	R3
	2	3	0
P2	4	3	3
P3	5	5	7
P1	7	7	10

The sequence <P2, P3, P1> satisfies the safety criteria. So current allocation state is safe.

(ii) Request made by process P1, Request(P1)= [1 1 0]

Here, Request(P1)< Need(P1) < Available

i.e. [1 1 0]< [1 4 5] < [2 3 0]

Pretending that request can be fulfilled, we get new state:

Processes	Allocation			Need			Available		
	R1	R2	R3	R1	R2	R3	R1	R2	R3
P1	3	3	3	0	3	5	1	2	0
P2	2	0	3	2	3	0			
P3	1	2	4	2	2	0			

As Need > Available for all process, no need can be fulfilled

So allocation is not thread safe i.e. request made by Process P1 can't be granted.

Q.53. What are semaphores? How do they implement mutual exclusion? (6)

Ans:

Semaphore: A semaphore is a synchronization tool that provides a general-purpose solution to controlling access to critical sections. A semaphore is an abstract data type (ADT) that defines a nonnegative integer variable which, apart from initialization, is accessed only through two standard operations: wait and signal. The classical definition of wait in pseudo code is

```
wait(S){
     while(S<=0)
          ; // do nothing
     S--; }
```

The classical definitions of signal in pseudo code is

```
signal(S){
     S++; }
```

When one process modifies the semaphore value, no other process can simultaneously modify that same semaphore value. In addition, in the case of the wait(S), the testing of the integer value of S(S<=0), and its possible modification(S--), must also be executed without interruption.

Mutual-exclusion implementation with semaphores:

Let there are n-processes and they share a semaphore, mutex (standing for mutual exclusion), initialized to 1.Each process Pi is organized as shown below:

```
do{
     wait(mutex);
     critical section
     signal(mutex);
          remainder section   }while(1);
```

Disadvantage: Mutual-exclusion solutions given by semaphores require busy waiting. That is, while a process is in its critical section, any other process that tries to enter its critical section must loop continuously in the entry code. Hence, busy waiting wastes CPU cycles that some other process might able to use productively.

Advantage: This type of semaphore is also called spinlock because the process "spins" while waiting for the lock. Spinlocks are useful in multiprocessor systems as no context switch is required when a process must wait on a lock. Thus, when locks are expected to be held for short times, spinlocks are useful.

Q.54. Give a solution for readers-writers problem using conditional critical regions. (8)

Ans:

Readers-writers problem: Let a data object(such as a file or record) is to be shared among several concurrent processes. Readers are the processes that are interested in only reading the content of shared data object. Writers are the processes that may want

to update (that is, to read and write) the shared data object. If two readers access the shared data object simultaneously, no adverse effects will result. However if a writer and some other process (either a reader or writer) access the shared object simultaneously, anomaly may arise. To ensure that these difficulties do not arise, writers are required to have exclusive access to the shared object. This synchronization problem is referred to as the readers-writers problem.

Solution for readers-writers problem using conditional critical regions. Conditional critical region is a high level synchronization construct. We assume that a process consists of some local data, and a sequential program that can operate on the data. The local data can be accessed by only the sequential program that is encapsulated within same process. One process cannot directly access the local data of another process. Processes can, however, share global data.

Conditional critical region synchronization construct requires that a variable v of type T, which is to be shared among many processes, be declared as

v: shared T;

The variable v can be accessed only inside a region statement of the following form:

region v when B do S;

This construct means that, while statement S is being executed, no other process can access the variable v. When a process tries to enter the critical-section region, the Boolean expression B is evaluated. If the expression is true, statement S is executed. If it is false, the process releases the mutual exclusion and is delayed until B becomes true and no other process is in the region associated with v.

Now, let A is the shared data object.

Let readcount is the variable that keeps track of how many processes are currently reading the object A.

Let writecount is the variable that keeps track of how many processes are currently writing the object A. Only one writer can update object A, at a given time.

Variables readcount and writecount are initialized to 0.

A writer can update the shared object A when no reader is reading the object A.

```
region A when( readcount = = 0 AND writecount = = 0){
                  ……
        writing is performed
           ……    }
```

A reader can read the shared object A unless a writer has obtained permission to update the object A.

```
region A when(readcount >=0 AND writecount = = 0){
              ……
      reading is performed                    ……    }
```

Q.55. Given memory partitions of 100k, 500k, 200k, 300k, and 600k (in order), apply first fit and best fit algorithms to place processes with the space requirement of 212k, 417k, 112k and 426k (in order)? Which algorithm makes the most effective use of memory? (3)

Ans:

Given memory partitions of 100k, 500k, 200k, 300k, and 600k (in order), applying first fit algorithms to place processes with the space requirement of 212k, 417k, 112k and 426k (in order), we have the following status:

Memory → / Request ↓	100K	500K	200K	300K	600K
212K	100K	288K	200K	300K	600K
417K	100K	288K	200K	300K	183K
112K	100K	176K	200K	300K	183K
426K	Can't fulfil request of 426K,so memory status will remain same				

And applying best fit algorithm the status is as follows:

Memory → / Request ↓	100K	500K	200K	300K	600K
212K	100K	500K	200K	88K	600K
417K	100K	83K	200K	88K	600K
112K	100K	83K	88K	88K	600K
426K	100K	83K	88K	88K	174K

Best fit makes the most efficient use of memory.

Q.56. Differentiate between
(i) Problem-oriented and procedure-oriented language
(ii) Dynamic and static binding
(iii) Scanning and parsing (9)

Ans:
(i) **Problem-oriented and procedure-oriented language:** The programming languages that can be used for specific applications are called problem oriented

languages. Such languages have large execution gaps and this gap is bridged by the translator or interpreter and does not concern the software designer.
A procedure-oriented language provides general purpose facilities required in most application domains. Such a language is independent is independent of specific application domains and results in a large specification gap which has to be bridged by an application designer.
(ii) Dynamic and static binding: A dynamic binding is a binding performed after the execution of a program has just begun while static binding is a binding performed before the execution of a program begins.
Static bindings lead to more efficient execution of a program than dynamic bindings.
(iii) Scanning and parsing: Scanning is the process of recognizing the lexical components in a source string while parsing is the process of checking the validity of a source string, and to determine its syntactic structure. The reason for separating scanning from parsing is that the lexical features of a language can be specified using Type-3 grammars. Each Type-3 production specifying lexical components is also a Type-2 production. However, a recognizer for Type-3 productions is simple, easier to build and more efficient during execution than a recognizer for Type-2 productions. Hence it is better to handle the lexical and syntactic components of a source language separately.

Q.57. Define Grammar of a language. Identify the different classes of grammar. Explain their characteristics and limitations. (10)

Ans:A **formal language grammar** is a set of formation rules that describe which strings formed from the alphabet of a formal language are syntactically valid, within the language. A grammar only addresses the location and manipulation of the strings of the language. It does not describe anything else about a language, such as its semantics.

As proposed by Noam Chomsky, a grammar *G* consists of the following components:

- A finite set N of *non terminal symbols*.
- A finite set Σ of *terminal symbols* that is disjoint from N.
- A finite set P of *production rules*, each rule of the form

where * is the Kleene star operator and denotes set union. That is, each production rule maps from one string of symbols to another, where the first string contains at least one non terminal symbol.
A distinguished symbol that is the *start symbol*. The Chomsky hierarchy consists of the following levels:

- **Type-0 grammars** (unrestricted grammars) include all formal grammars. They generate exactly all languages that can be recognized by a Turing machine. The language that is recognized by a Turing machine is defined as all the strings on which it halts. These languages are also known as the recursively enumerable languages.
- **Type-1 grammars** (context-sensitive grammars) generate the context-sensitive languages. These grammars have rules of the form $\alpha A\beta \rightarrow \alpha\gamma\beta$ with A a non terminal and α, β and γ strings of terminals and non terminals. The strings α and β may be empty, but γ must be nonempty. The rule $S \rightarrow \varepsilon$ is allowed if S does not appear on the right side of any rule. The languages described by these grammars are exactly all languages that can be recognized by a non-deterministic

- **Type-2 grammars** (context-free grammars) generate the context-free languages. These are defined by rules of the form $A \rightarrow \gamma$ with A a non terminal and γ a string of terminals and non terminals. These languages are exactly all languages that can be recognized by a non-deterministic pushdown automaton. Context free languages are the theoretical basis for the syntax of most programming languages.
- **Type-3 grammars** (regular grammars) generate the regular languages. Such a grammar restricts its rules to a single non terminal on the left-hand side and a right-hand side consisting of a single terminal, possibly followed by a single non terminal. The rule $S \rightarrow \varepsilon$ is also here allowed if S does not appear on the right side of any rule. These languages are exactly all languages that can be decided by a finite state automaton. Additionally, this family of formal languages can be obtained by regular expressions. Regular languages are commonly used to define search patterns and the lexical structure of programming languages.

Q.58. Enumerate the data structures used during the first pass of the assembler. Indicate the fields of these data structures and their purpose/usage. (8)

Ans:

Three major data structures used during the first pass of the assembler are:

- LOCCTR (Location counter)
- OPTAB (operation code table)
- SYMTAB (Symbol table)

LOCCTR: Location Counter keeps track machine addresses of symbolic tables.

- Initialized by START.
- Increased for each instruction
  - Pseudo Instruction: BYTE, WORD,RESB, RESW.
  - Machine Instruction: Fixed length (3 bytes) for SIC, variable length for SIC/XE (looking up OPTAB).
  - Assign the value (address) of LOCCTR to corresponding symbol table.

OPTAB: operation table contains mnemonic operation code and its machine language equivalent.

ADD	18
LDS	6C
. . . . .	. . . . .

OPTAB can be implemented using hashing function for fast access.

SYMTAB: Symbol table maintain symbolic label, operand and their corresponding machine.

Addresses

First	1000
CLOOP	1003
RDREC	2039
⋮	⋮

•SYMTAB is dynamic, constructed during Pass 1:
- for symbolic label, fill in symbol and address (according to current LOCCTR).
- for symbolic operand, fill in symbol and mark it as undefined. Address field will be defied later.

•SYMTAB can be implemented using hashing function for fast access.

Q.59. What is macro-expansion? List the key notions concerning macro expansion. Write an algorithm to outline the macro-expansion using macro-expansion counter. (8)

Ans:

macro call leads to macro expansion. During macro expansion, the macro call statement is replaced by a sequence of assembly statements. Two key notions concerning macro expansion are:

1.**Expansion time control flow-** this determines the order in which model statements are visited during macro expansion.

2.**Lexical substitution:** Lexical substitution is used to generate an assembly statement from a modal statement.

The flow of control during macro expansion can be implemented using a macro-expansion counter (MEC). The outline of algorithm is as follows:

1. MEC:=statement number of first statement following the prototype statement;
2. While statement pointed by MEC is not a MEND statement
   (a) If a model statement then
      (i) expand the statement.
      (ii) MEC:=MEC+1;
   (b) Else (i.e. a pre processor statement)
      (i) MEC:=new value specified in the statement;
3. Exit from macro expansion.

Q.60 What is a heap? Name and explain the popular techniques to identify free memory areas as a result of allocation and de-allocations in a heap. **(8)**

Ans:

The **heap** is an area of memory, which is dynamically allocated. Like a stack, it may grow and shrink during runtime. Unlike a stack, a heap is not LIFO implies more complicated to manage. Two popular techniques to identify free memory areas as a result of allocation and de-allocations in a heap are:

1. Reference count: the system associates a reference count with each memory area to indicate the number of its active users. This number is incremented when a

user accesses that area and decrements when user stops using that. The area is free if the reference counts drops to zero. This scheme is very simple to implement however incurs incremental overheads.

2. Garbage collection: In this technique two passes are made over the memory to identify unused areas. In the first pass it traverses all pointers pointing to allocated areas and marks the memory areas that are in use. The second pass finds all unmarked areas and declares them to be free. The garbage collection overheads are not incremental. They are incurred every time the system runs out of free memory to allocate to fresh requests.

Q.61. What are threads? Why are they required? Discuss the differentiate between Kernel level and user level threads? (8)

Ans:

A thread, sometimes called a lightweight process(LWP), is a basic unit of CPU utilization; it comprises a thread ID, a program counter, a register set and a stack.

A thread shares with other threads belonging to the same process its code section, data section and other operating-system resources, such as open files and signals.

If the process has multiple threads of control, it can do more than one task at a time.

User Level Threads Vs Kernel Supported Threads

i. User threads are supported above the kernel and are implemented by a thread library at the user level.
   Whereas, kernel threads are supported directly by the operating system.

ii. For user threads, the thread library provides support for thread creation, scheduling and management in user space with no support from the kernel as the kernel is unaware of user-level threads.
   In case of kernel threads, the kernel performs thread creation, scheduling and management in kernel space.

iii. As there is no need of kernel intervention, user-level threads are generally fast to create and manage.
   As thread management is done by the operating system, kernel threads are generally slower to create and manage that are user threads.

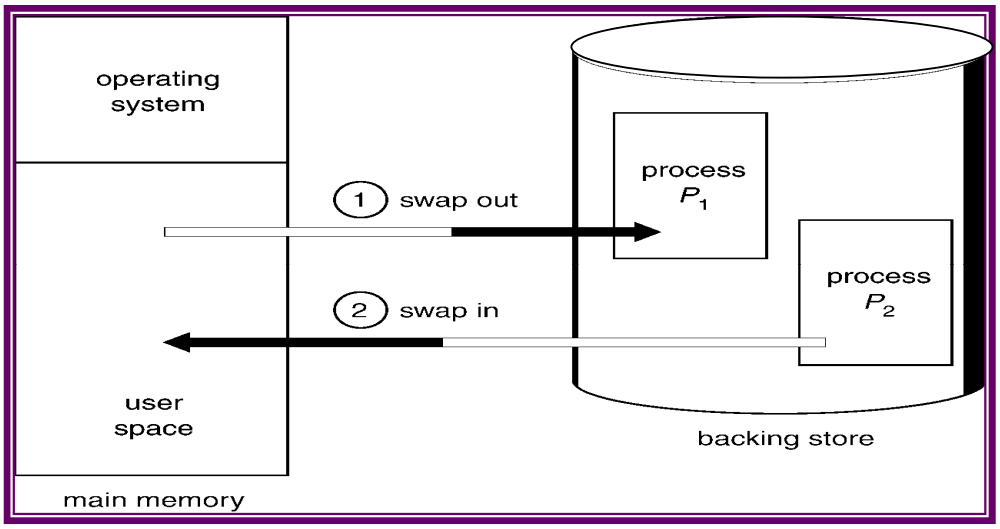
iv. If the kernel is single-threaded, then any user-level thread performing blocking system call, will cause the entire process to block, even if other threads are available to run within the application.
   However, since the kernel is managing the kernel threads, if a thread performs a blocking system call, the kernel can schedule another thread in the application for execution.

v. User-thread libraries include POSIX P threads, Mach C-threads and Solaris 2 UI-threads.
   Some of the cotemporary operating systems that support kernel threads are Windows NT, Windows 2000, Solaris 2, BeOS and Tru64 UNIX(formerly Digital UNIX).

Q.62. What is swapping? Does swapping increase the Operating Systems' overheads? Justify your answer. (8)

Ans:

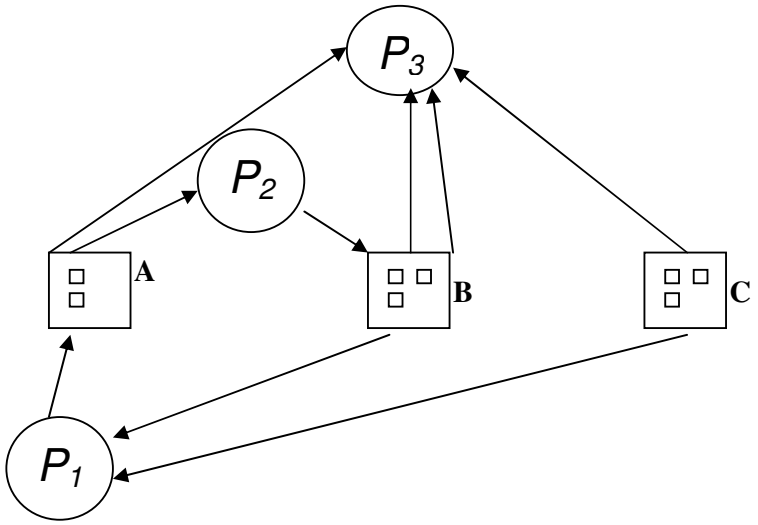
A process can be *swapped* temporarily out of memory to a *backing store*, and then brought back into memory for continued execution.



Major part of swap time is transfer time; total transfer time is directly proportional to the *amount* of memory swapped. Modified versions of swapping are found on many systems, i.e., UNIX, Linux, and Windows.

Q.63. Suppose there are 2 copies of resource A, 3 copies of resource B, and 3 copies of resource C. Suppose further that process 1 holds one unit of resources B and C and is waiting for a unit of A; that process 2 is holding a unit of A and waiting on a unit of B; and that process 3 is holding one unit of A, two units of B, and one unit of C. Draw the resource allocation graph. Is the system in a deadlocked state? Why or why not? (8)

Ans:
Resource allocation graph is given below:



There is a cycle in the resource allocation graph implies system is not in a safe state. System is in deadlock as required resources are 1 unit of resource A and 1 unit of resource B while available resource is 1 unit of resource C. None of the process can be completed.

Q.64 what is system programming? Explain the evolution of system software. (6)

Ans:

System software is collection of system programs that perform a variety of functions, viz file editing, recourse accounting, IO management, storage management etc.

System programming is the activity of designing and implementing SPs.

System programs which are the standard component of the s/w of most computer systems; The two fold motivation mentioned above arises out of single primary goal viz of making the entire program execution process more effective.

Q.65 Give difference between assembler, compiler and interpreter. (6)

Ans:

An **assembler** is the translator for an assembly language of a computer. An assembly language is a low-level programming language which is peculiar to a certain computer.

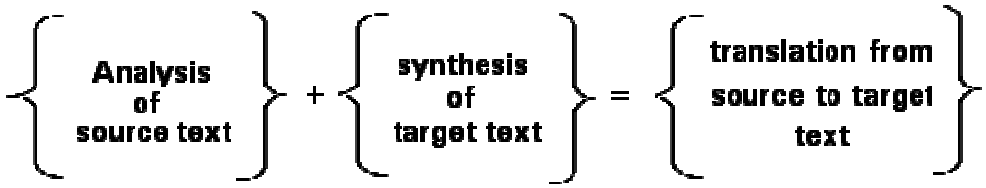
A **compiler** is a translator for machine independent HLL like say FORTRAN, COBOL etc.

An **interpreter** analysis the source program statement by statement and it self carries out the actions implied by each statement.

Q.66 Write down the general model for the translation process. (4)

Ans:

General model for the translation process can be represented as follows:



Q.67 Pass I of the assembler must also generate the intermediate code for the processed statements. Justify your answer. (8)

Ans: Criteria for selection of an appropriate intermediate code form are;

(i) Ease of use: It should be easy to construct the intermediate code form and also easy to analyze and interpret it during pass II, i.e. the amount of processing required to be done during its construction and analysis should be minimal.

(ii) Economy of storage: It should be compact as the target code itself .This will reduce the overall storage requirements of assembler.

Q.68 what are the advantages and disadvantages of macro pre-processor? (8)

Ans: The **advantage** of macro pre-processor is that any existing conventional assembler can be enhanced in this manner to incorporate macro processing. It would reduce the programming cost involved in making a macro facility available.

The **disadvantage** is that this scheme is probably not very efficient because of the time spent in generating assembly language statement and processing them again for the purpose of translation to the target language.

Q.69. What is parsing? Give difference between top down parsing and bottom up parsing. (6)

Ans:

The goal of **parsing** is to determine the syntactic validity of a source string. If the string is valid, a tree is built for use by subsequent phase of compiler.

Top down parsing: Given an input string, top down parsing attempts to derive a string identical to it by successive application of grammar rules to the grammar's distinguished symbol. When such a string is obtained , a tree representing its derivation would be the syntax tree for an input string. Thus if ά is input-string, a top down parse determines a derivation sequence.

Bottom up parsing: A bottom up parse attempts to develop syntax tree for an input string through a sequence of reduction. If the input string can be reduced to the distinguished symbol, the string is valid. If not, error would be detected and indicated during the process of reduction itself.

Q.70 How non-relocatable programs are different from relocatable programs? (4)

Ans: A **non relocatable** program is one which cannot be made to execute in any area of storage other than the one designated for it at the time of its coding or translation. A **relocatable** program form is one which consists of a program and relevant information for its relocation. Using this information it is possible to relocate the program to execute from a storage area then the one designated for it at the time of its coding or translation.

Q.71 what are the fundamental steps in program development? Discuss program testing and debugging in detail. (6)

Ans:

The fundamental steps in program development are:

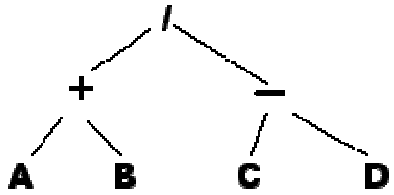
(i) Program design, coding and documentation.
(ii) Preparation of the program in machine readable form, and initial editing to adopt two required formats.
(iii) Program translation and linking/ loading
(iv) Program testing and debugging.
(v) Program modification for performance enhancement.
(vi) Reformatting programs data and/or results to suite other programs which process them.

In program testing and debugging important steps are as follows:

(i) Construction of test data for the program
(ii) Analysis of test results to detect program errors.
(iii) Localization of errors and modification of the program to eliminate them, i.e. debugging.

Q.72 Give LOAD-STORE optimization based on expression trees for the expression (A+B)/(C-D) (8)

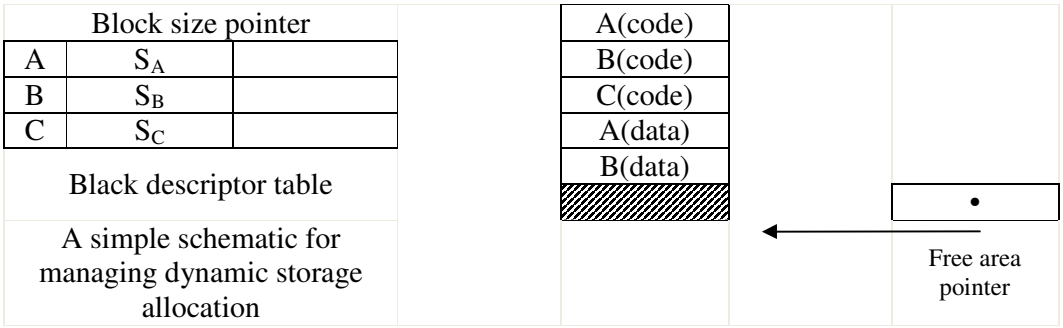
Ans:



Load	A		LOAD	C
ADD	B		SUB	D
STORE	TEMP		STORE	TEMP1
LOAD	C		LOAD	A
SUB	D		ADD	B
STORE	TEMP2		DIU	TEMP1
LOAD	TEMP1			
DIU	TEMP2			

Q.73 Draw a simple schematic for managing dynamic storage allocation. (8)

Ans:



A simple schematic for managing dynamic storage allocation

Q.74. Differentiate between synchronous and asynchronous input / output with the help of an example. (8)

Ans: The I/O operation is asynchronous input output operation because after the start of input/output, control is returned to the user program without waiting for the input/output to complete. The input/output continues and on its completion, an interrupt is generated by the controller to attain CPU's attention.
CPU execution waits, while I/O proceeds, in which case, there is a possibility that at most one request I/O request is outstanding at a time. This is known as synchronous I/O.

Q.75. List the major activities of an operating system with respect to memory management, **secondary storage management and process management.** (8)

Ans:
Operating system is responsible for following activities in connection with management of memory;
(i) Allocation and de allocation of memory as and when needed

(ii) Keeping track of used and unused memory space.
(iii) Deciding what process to be loaded into memory in case space becomes available.

For secondary space management:
(i) Swap space and free space management
(ii) Disk scheduling
(iii) Allocating space to the data and programs onto the secondary storage device.

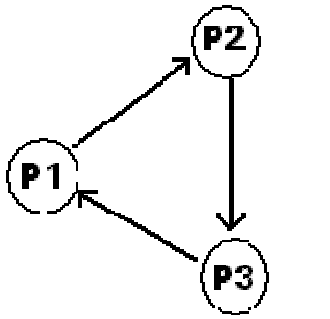
For process management:
(i) Creation, deletion of both user and system process.
(ii) Handling process synchronization.
(iii) Deadlock handling.

Q.76. What are the disadvantages of FCFS scheduling algorithm as compared to shortest job first (SJF) scheduling? (8)

Ans:
Disadvantages:
(i) Waiting time can be large if short requests wait behind the long ones.
(ii) It is not suitable for time sharing systems where it is important that each user should get the CPU for an equal amount of time interval.
(iii) A proper mix of jobs is needed to achieve good results from FCFS scheduling.

Q.77 Explain deadlock detection algorithm for single instance of each resource type. (8)

Ans:
(i) Maintain a wait: nodes in a graph represent process. If process i is waiting for resource hold by process j. Then there is an edge from I to j.



WAIT- FOR- GRAPHS

(ii) Periodically invokes an algorithm that searches for cycles in the graph. If there is a cycle in the wait-for-graph a dead lock is said to be exist in the system.

Q.78 Discuss the concept of segmentation? What is the main problem with segmentation? (8)

Ans:
Segmentation is techniques for the non contiguous storage allocation. It is different from paging as it supports user's view of his program.
Problem with segmentation
(i) Is with paging, this mapping requires two memory references per logical address, which slows down the computer system by a factor of two. Caching is the method used to solve this problem.

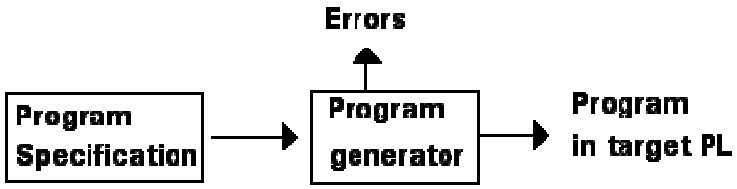
(ii) Problem of external fragmentation.

Q79. What is the difference between absolute and relative path name of a file? (8)

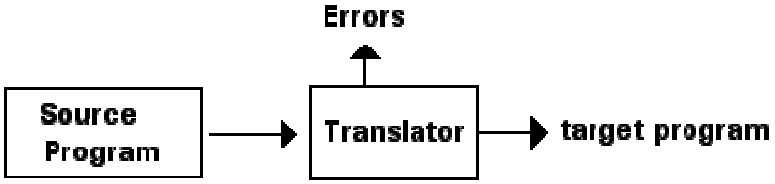
Ans:
Absolute path name:
It is listing of the directories and files from the root directory to the intended file.
Relative path name :
A user can specify a path particular directory as his current working directory and all the path names instead of being specified from the root directory are specified relative to the working directory.

Q.80. Describe language processing activities? **(8)**

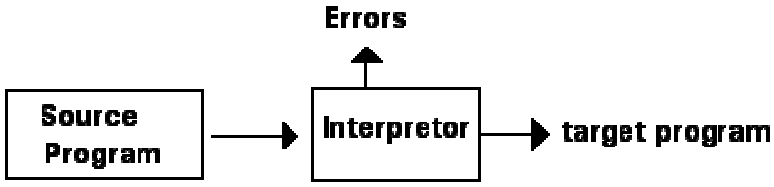
Ans:
There are two different types of language processing activities:
1. Program generation activities
2. Program execution activities
Program generation activities: A program generation activity aims at automatic generation of a program. The source language is a specification language of an application domain and the target language is typically a procedure oriented programming language. the following figure shows program generation activity



The program generator is a software system which accepts the specification of a program to be generated and generates a program in the target. PL. The program generator introduces a new domain between the application and PL domains. We call this the program generator domain. The specification gab is now gab between the application domain and the program generator domain. This gab is smaller than the gab between the application domain and PL domain.
Program execution activities: A program execution activity organizes system. Two model program executions are:
1. Translation 2. Interpretation
Translation: The program translation models bridges execution gap by translating a program written in a PL, called the source program into an equivalent program in the machine or assembly language of the computer system.



Interpretation: The interpreter reads the source program and stores it in its memory. During interpretation it takes a statement, determines its meaning and performs actions which implement it.



Q.81 Explain the criteria to classify data structures used for language processors? **(8)**

Ans:
The data structures used in language processing can be classified on the basis of the following criteria:
1. Nature of data structure (whether a linear or non-linear data structure)
2. Purpose of a data structure (whether a search data structure or an allocation data structure)
3. Life time of a data structure (whether used during language processing or during target program execution)
A linear data structure consists of a linear arrangement of elements in the memory. A linear data structure requires a contiguous area of memory for its elements. This poses a problem in situations where the size of a data structure is difficult to predict. the elements of non linear data structures are accessed using pointers. Hence the elements need not occupy contiguous area of memory.
Search Data structures are used during language processing to maintain attribute information concerning different entities in the source program. In this the entry for an entity is created only once, but may be searched for large number of times. Allocation data structures are characterized by the fact that the address of memory area allocated to an entity is known to the users. So no search operations are conducted.

Q.82 Explain macro definition, macro call and macro expansion? **(6)**

Ans :
A unit of specification for a program generation is called a macro. It consists of name, set of formal parameters and body of code. When a macro name is used with a set of actual parameters it is replaced by a code generated from its body. This code is called macro expansion. There are two types of expansions:
1. lexical expansion
2. Semantic expansion
Lexical expansion: It means a replacement of character string by another string during program generation. It is generally used to replace occurrences of formal parameters by corresponding actual ones.
Semantic Expansion: It implies generation of instructions build to the requirements of specific usage. It is characterized by the fact that different uses of a macro can lead to codes which differ in the number, sequence and opcodes of instructions.
The macro definition is located at the beginning of the program is enclosed between a macro header and macro end statement.
A macro statement contains macro name and parameters

< macro name > { < parameters>}

A macro call: A macro is called by writing the macro name in the mnemonic field of an assembly statement.

Q.83. What are the advantages of code optimization? Explain optimizing transformations? **(10)**

Ans:

Code optimization aims at improving the execution efficiency of a program. This is achieved in two ways. Redundancies in a program are eliminated and computations in a program are rearranged to make it execute efficiently. The optimized program occupies 25 percent less storage and execute three times as fast as the unoptimized program.

Optimizing transformations:

An optimizing transformation is a rule for rewriting a segment of a program to improve its execution efficiency without affecting its meaning.

Commonly used optimizing transformations are

(i) Compile time evaluation: Execution efficiency can be improved by performing certain action specified in a program during compilation itself. Constant folding is the main optimization of this kind. When all operands in an operation are constants, the operation can be performed at compilation time.

For Example: a: =3.141557/2 can be replaced by a:=1.570785 eliminating a division operation.

(ii) Elimination of common subexpressions:

Common subexpressions are occurrences of expressions yielding the same value.

For Example: a:=b*c t: = b*c

λ: = b*c + 5.2 ⟹ a: = t

λ: = t + 5.2

(iii) **Dead code elimination:** Code which can be omitted from a program without affecting its results is called dead code. For example An assignment statement x: = < exp > constitutes dead code if the value assigned to x is not used in the program.

(iv)Frequency and strength reduction:

Execution time of a program can be reduced by moving code from a part of a programs which is executed very frequently to another part of the program which is executed fewer times.

The strength reductions replaces the occurrance of a time consuming operation by an occurrance of a faster operation.

Q.84 Explain the following terms

(i) Translated address
(ii) Linked address
(iii) Load address

Explain the relationship amongst these. **(6)**

Ans:

(i)Translated address: Address assigned by the translator

(ii)Linked address: Address assigned by the linker

(iii)Load time address: Address assigned by the loader.

While compiling a program P, a translator is given an original specification for P. This is called the translated origin of P. The translator uses the value of the translated origin to perform memory allocation for the symbols declared in P. This results in the assignment of a translated time address to each symbol in the program. The origin of a program may have to be changed by the linker or loader for one of the following reasons:

1. Object modules of library routines often have the same translated origin so memory allocation to such programs would conflict unless their origins are changed.

2. Operating system requires a program to be executed from a specific location so this may require a change in its origin.

Q.85 What are the functions of passes used in two-pass assembler? Explain pass-1 algorithm? **(10)**

Ans :
Two pass translation of an assembly language program can handle forward references early.
The following tasks are performed by the passes of a two pass assembler are as follows:
Pass I: (i) Separate the symbol, mnemonic opcode and operand fields
(ii) Build the symbol table
(iii) Perform LC processing
(iv) Construct intermediate representation.
Pass II: Synthesize the target program
Pass I uses the following data structures:
OPTAB : A table of mnemonic opcodes and related information
SYMTAB: symbol table
LITTAB: A table literally used in the program
OPTAB contains the fields mnemonic opcode, class and mnemonic information. The class field indicated whether the opcode corresponds to an imperative statement (IS), a declaration statement (DL) or an assembler directive (AD).
(SYMTAB entry contains the fields address and length. A LITTAB entry contains literals and address.)

Q.86 What data structure is used by an operating system to keep track of process information? Explain **(4)**

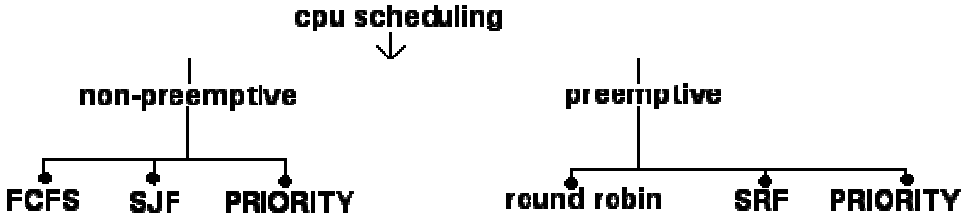
Ans:
A process is a program in execution. An operating system considers a process to be the fundamental unit for resource allocation. Following resources could be allocated to a process
(i) Memory (ii) Secondary memory (iii) I/O Devices (iv) files opened by the process (v) CPU time consumed by process
A data structure called process control block (PCB) is used by an OS to keep track of all information concerning a process. The PCB of a process contains the following information.

Process ID
Priority
Process state
PSR
Registers
Event information
Memory allocation
Resources held
PCB pointer

(i) Process scheduling information: This information consists of three fields process ID, priority, process state.
(ii) PSR and machine registers: These fields hold contents of the processor states register (PSR) and the machine registers when the execution of the process was last suspended.
(iii) Event information: when a process is in blocked state, this field contains information concerning the event for which the process is waiting.
(iv) Memory and resource information: This information is useful for the deallocating memory and resources when the process terminates.
(v) PCB pointer: It is a pointer to the next PCB in the process scheduling list.

Q.87. Categorize the CPU scheduling algorithms? Explain non-pre-emptive algorithms? **(4)**

Ans
The various **CPU scheduling algorithms** are classified as follows:



Non preemptive algorithms: In this method a job is given to CPU for execution as long as the job is non completed the CPU cannot be given to other processes.
There are three types of non preemptive algorithms.
(i) First-come-first-serve (FCFS):
This is simplest CPU scheduling algorithm . With this scheme, the process that requests the CPU at first is given to the CPU at first. The implementation of FCFS is easily managed by with a FIFO queue.
(ii) Shortest-job-first (SJF): This is also called SPN (shortest process next). In this the burst times of all the jobs which are waiting in the queue are compared. The job which is having the least CPU execution time will be given to the processor at first. In this turnaround time and waiting times are least. This also suffers with starvation. Indefinite waiting time is called as starvation. It is complex than FCFS.
(iii) Priority: In this algorithm every job is associated with CPU execution time, arrival time and the priority. Here the job which is having the higher priority will be given to the execution at first. This also suffers with starvation. And by using aging technique starvation effect may be reduced.

Q.88. Differentiate between Batch Operating System and Time Sharing Operating System? **(6)**

Ans
Batch operating systems: A batch is a sequence of jobs. This batch is submitted to batch processing operating systems, and output would appear some later time in the form of a program or as program error. To speed up processing similar jobs are batched together. The major task of batch operating systems is to transfer control automatically from one job to next. Here the operating is always in the memory.
(i) It is lack of interaction between user and job while executing
(ii) Turnaround time is more.
(iii) CPU is often idle, because of 1/0 devices are very slow.

Time sharing: Time sharing or multi tasking is a logical execution of multiprogramming. Multiple jobs are executed by the CPU switching between them. Here the computer system provides on line communication between the user and the system.

Here the CPU is never idle. Time shared operating system allows many users to share the computer simultaneously.

Time sharing systems requires some sort of memory management and protection.

Q.89. What is a Deadlock? Write an algorithm for deadlock detection. **(8)**

Ans.

Deadlock is a situation, in which processes never finish executing and system resources are tied up, preventing other jobs form starting.

A process requests resources; if the resources are not available at that time, the process enters a wait state. Waiting processes may never again change state, because the resources they have requested are held by other waiting processes, thereby causing deadlock.

An algorithm for deadlock detection:

1. Let Work and Finish be vectors of length m and n, respectively. Initialize:

(a) Work = Available

(b) For i = 1,2, …, n, if Allocation $_i$ $^1\neq$ 0, then
Finish[i] = false; otherwise, Finish[i] = true.

2. Find an index i such that both:

(a) Finish[i] == false

i. (b) Request$_i$ ≤ Work

If no such i exists, go to step 4.

3. Work = Work + Allocation$_i$
Finish[i] = true
go to step 2.

4. If Finish[i] == false, for some i, 1 ≤ i ≤ n, then the system is in deadlock state. Moreover, if Finish[i] == false, then P$_i$ is deadlocked.

Q 90 Develop a regular expression for

(i) Integer
(ii) Real number
(iii) Real number with optional fraction
(iv) Identifier **(8)**

Ans :

A regular expression for

(i) integer is $[+ \mid -] (d)^+$
(ii) real number is $[+ \mid -] (d)^+ . (d)^+$
(iii) real number with optional fraction is $[+ \mid -] (d)^+ . (d)^*$
(iv) identifier is $l(l \mid d)^*$

Q.91. What is critical section problem? Give two solutions for critical section problem? **(8)**

Ans :

A race condition on a data item arises when many processes concurrently update its value data consistency, requires that only one process should update the value of a data item at any time. This ensured through the notion of a critical section.

A critical section for a dataitem d is a section of code, which cannot be executed concurrently with itself or with other critical section(s) for d.
Consider a system of n processes (P_0, P, .P_{n-1}).
Each process has a segment of code called a critical section, in which the process may be changing common variables, updating a table, waiting a file and so on. Important feature of the system is, when one process is executing in its critical section, no other process is to be allowed to execute its critical section. Thus the execution of critical sections by the processes is mutually exclusive in time.
A solution to the critical section problem must specify the following requirements.
(i) Mutual exclusion (ii) Progress (iii) Bounded waiting
One solution for critical section problem is provided by semaphores.
Another solution for critical section problems is by Moniters

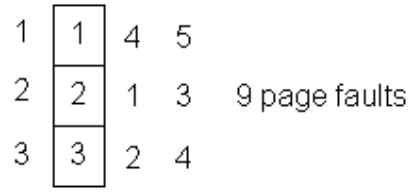
Q.92. Explain difference between Security and Protection? Describe the scheme of capability lists to implement protection? **(8)**

Ans :
Protection mechanism: The following mechanisms are commonly used for protecting files containing programs and data.
(i) Access controls lists (ACL's)
(ii) Capability lists (C- lists)
These lists are used to ensure that uses only access files which are explicitly authorized access. These files include
(i) files created by a user himself/herself
(ii) files owned by others, for which a user process explicit access privileged granted by other owners.
Security mechanisms: Authentication is the primary security mechanism. Authentication is the act of verifying the identity of a user. Authentication is typically performed through passwords at login time. The system stores the password information in a system as set as pair of the form (user id, password info)
The password information is protected by encryption
C-list: A capability is a file access privilege concerning capabilities possessed by a user is stored in a capability list. A C-list is a set of pairs{ (file.id, access privileges),…..}
C-lists are usually small in size. This limits the space and time over heads in using them to control file accesses. A c-list is a token representing certain access privileges for an object. An object is any hardware or software entity in the system. A capability possessed by a process. A process possessing a capability for an object can access the object in a manner consistent with the access privileges described in the capability. Thus maintaining C-lists provide:
1. A uniform addressing mechanism for long and short life objects
2. It does not explicitly associate memory with processes. It associates C-lists with processes.
3. A process may access objects existing anywhere in the system.

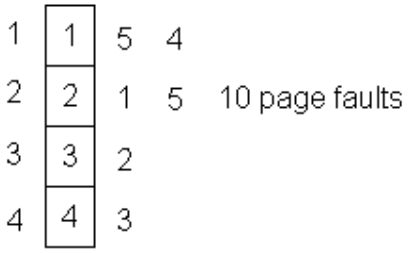
Q.93. Explain with the help of examples FIFO and LRU page replacement algorithms? **(10)**

Ans :
FIFO policy: This policy simply removes pages in the order they arrived in the main memory. Using this policy we simply remove a page based on the time of its arrival in the memory.

For example if we have the reference string: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5 and 3 frames (3 pages can be in memory at a time per process) then we have 9 page faults as shown



If frames are increased say to 4, then number of page faults also increases, to 10 in this case.



LRU policy: LRU expands to least recently use. This policy suggests that we remove a page whose last usage is farthest from current time.

Q.94. Describe the necessary conditions for Deadlock. **(8)**

Ans:

Necessary conditions for deadlock
1. Mutual exclusion
2. Hold and wait
3. No preemption
4. Circular wait

Mutual exclusion: The mutual exclusion condition must hold for non sharable resources. For example, a printer cannot be simultaneously shared by several processes. Shared resources on the other hand, do not require mutually exclusive access, and thus cannot be involved in the deadlock. In general, however it is not possible to prevent deadlocks by denying the mutual exclusion condition.

Hold and wait: To ensure that the hold and wait condition never occurs in the system, we must guarantee that, whenever a process requests a resource, it does not hold any other resources. A process may request some resources and use them before it can request any other resources, however it must release all the resources that it is currently allocated.

No preemption: The third necessary condition is that there be no preemption of resources that have already been allocated. If a process that is holding some resources requests another resource that it cannot be immediately allocated to it. Then all resources currently being held are preempted.

Circular Wait: One way to ensure that the circular wait condition never holds is to impose a total ordering of all resources types, and to ensure that each process requests resources in an increasing order of enumeration.

Q.95 What is a Process Scheduling? Explain the different sub-functions of Process Scheduling. **(8)**

Ans:

Scheduling is a key part of the workload management software which usually perform some or all of:

- Queuing
- Scheduling
- Monitoring
- Resource management
- Accounting

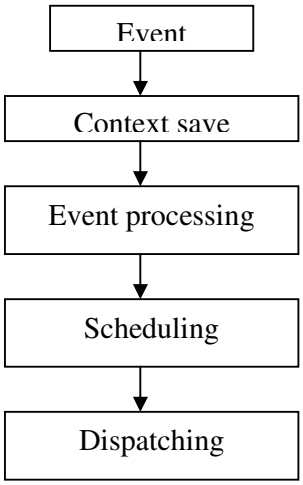
The difficult part of scheduling is to balance policy enforcement with resource optimization in order to pick the *best* job to run. Essentially one can think of the scheduler performing the following loop:

- Select the *best* job to run, according to policy and available resources.
- Start the job.
- Stop the job and/or clean up after completion.
- repeat.

Process scheduling consists of the following sub-functions:

1. Scheduling: Selects the process to be executed next on the CPU. The scheduling function uses information from the PCB's and selects a process based on the scheduling policy in force.
2. Dispatching: Sets up execution of the selected process on the CPU. This function involves setting up the execution environment of the selected process, and loading information from the PSR and registers fields of the PCB into the CPU.
3. Context save: Saves the status of a running process when its execution is to be suspended. This function performs housekeeping whenever a process releases the CPU or is pre-empted.

The following diagram illustrates the use of scheduling sub functions: Occurrence of an event invokes the context save function. The kernel now processes the event that has occurred. The scheduling function is now invoked to select a process for execution on the CPU. The dispatching function arranges execution of the selected function on the CPU.



Q.96. Describe the essential properties of the following operating systems

Real Time and Distributed Operating System **(8)**

Ans:

Real time operating system:

1. Time constraint result
2. Priority driven or deadline oriented scheduling
3. Programmer defined interrupts
4. Hard Real time and soft real time system
5. Multitasking
6. Event driven
7. Embedded systems
8. Robotics

Distributed operating systems:

1. Resource sharing
2. Computation speed up
3. Reliability
4. Communication

Q.97. Describe Data structures used during passes of assembler and their use. **(10)**

Ans:

Data structure during passes of assembler and their use.

Pass 1 data base

1. Input source program
2. A location counter (LC)
3. A table, the machine-operation table (MOT), that indicates the symbolic mnemonic for each instruction and its length.
4. Pseudo- operation table
5. Symbol table
6. Literal table
7. Copy of the input to be used later by pass 2

Pass 2

1. Copy of source program input to pass 1
2. Location counter (LC)
3. MOT
4. POT
5. ST
6. Base table that indicates which registers are currently specified as base register.
7. A work space, INST, that's used to hold instruction as its various parts are being assembled together
8. Punch line, used to produce a printed listing.
9. Punch card for converting assembled instructions into the format needed by the loader.

Q.98. what is parsing and specify the goals of parsing **(6)**

Ans:

Source programmed statements are regarded as tokens, building block of language the task of scanning the source statement, recognizing and classifying the various tokens is known as lexical analysis. The part of the compiler that performs this task is commonly called a scanner.

After the token scan, each statement in the program must be recognized as some language constructs, such as declaration or an assignment statement described by the grammar.

This process is called Syntactic analysis or parsing is performed by the part of compiler called parser.

Goals:

1. to check the validity of source string
2. to determine the syntactic structure of a source string.

For invalid string it reports error, for a valid string it builds a parse tree to reflect the sequences of derivations or reductions performed during parsing.

Q.99. Write short note on code optimization **(8)**

Ans:

Code optimization is the optional phase designed to improve the intermediate code so that the

Ultimate object program runs faster or takes less space. Code optimization in compilers aims at improving the execution efficiency of a program by eliminating redundancies and by rearranging the computations in the program without affecting the real meaning of the program.

Scope – First optimization seeks to improve a program rather than the algorithm used in the program. Thus replacement of algorithm by a more efficient algorithm is beyond the scope of optimization. Also efficient code generation for a specific target machine also lies outside its scope.

The structure of program and the manner in which data is defined and used in it provide vital clues for optimization.

Optimization transformations are classified into local and global transformations.

Q.100. Explain the Features of Major scheduling algorithms **(8)**

Ans:

1. FCFS – First come first served scheduling
2. Shortest job – First scheduling
3. Priority scheduling
4. Round robin scheduling

FCFS-

1. Process that request the CPU first is allocated CPU first
2. Managed by FIFO queue.
3. Average waiting time is generally long
4. Non preemptive
5. Troublesome for time sharing systems

Shortest job first

1. The process which has the smallest CPU burst time gets first
2. It increases the waiting time of long processes.
3. Problem – difficult to predict the length of next CPU request
4. May be preemptive or non preemptive

Priority

1. Highest priority process gets CPU allocation first
2. The larger the CPU burst, lower the priority
3. Priority can be defined internally or externally
4. Can be preemptive or non-preemptive
5. Problem-starvation, blocking of process
6. Solution – aging – increases the priority

Round robin

1. Time quantum from 10 100 millisecond is defined
2. Processes are considered in circular queue
3. CPU allocation is divided among processes accordingly
4. Performance depends heavily on time quantum
5. Preemptive

Q.101. List the criteria on the basis of which data structures used in language processing can be classified. **(3)**

Ans:
The data structures used in language processing can be classified on the following criterion:

1. **Nature of a data structure-**whether a linear or nonlinear data structure. A linear data structure consists of a linear arrangement of elements in memory and elements require a contiguous area of memory. The elements of a non-linear data structure are accessed using pointers and hence the elements need not occupy contiguous areas of memory.
2. **Purpose of a data structure-**whether a search data structure or an allocation data structure. Search data structures are used during language processing to maintain attribute information concerning different entities in the source program. Allocation data structures are characterized by the fact that the user of that entity knows the address of the memory area allocated to an entity thus no search operations are conducted on them.
3. **Lifetime of a data structure-**whether used during language processing or during target program execution.

Q.102. Differentiate between logical address and physical address. **(4)**

Ans:
A logical address is the address of the instruction or data word as used by a program (this includes the use of index, base or segment register).
A physical address is the effective memory address of an instruction or data word. The set of physical addresses generated during operation of system constitute the **physical address** space of the system. The compile time and the load time address binding schemes result in environment where the logical and physical address are same, whereas during execution time addresses differ.

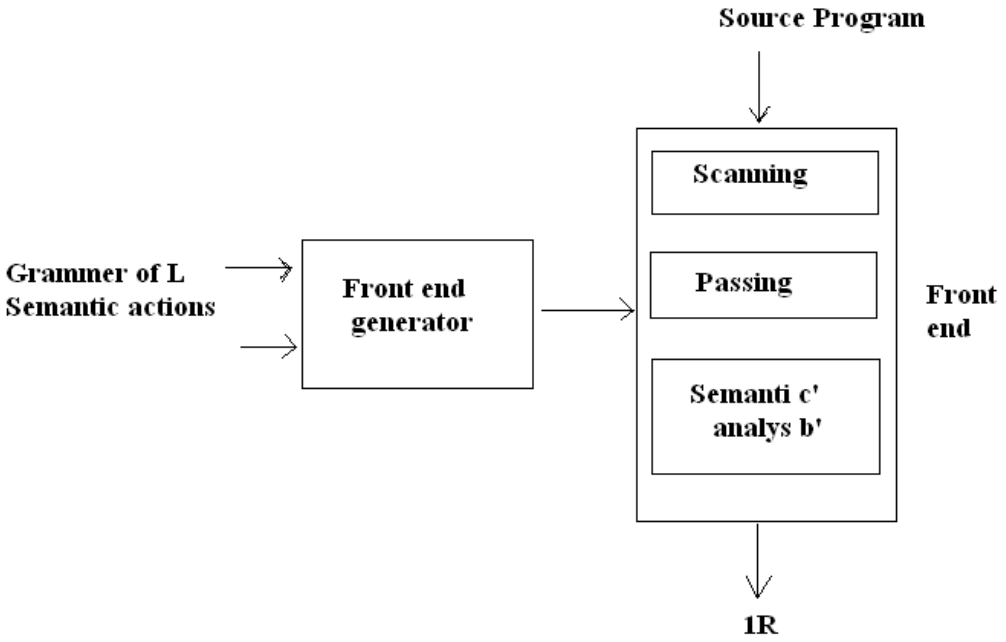
Q103. What are Language Processor Development Tools (LPDTs)? Explain through schematic diagram. Name the widely used LPDTs. **(4)**

Ans:
Language processor development tools (LPDTs) focuses on generation of the analysis phase of language processors. The LPDT requires the following two inputs:

1. Specification of a grammar of language L
2. Specification of semantic actions to be performed in the analysis phase.

It generates programs that perform lexical, syntax and semantic analysis of the source program and construct the IR. These programs collectively form the analysis phase of the language

processor.



Widely used LPDT's are:

Lex - A Lexical Analyzer Generator

Lex helps write programs whose control flow is directed by instances of regular expressions in the input stream. It is well suited for editor-script type transformations and for segmenting input in preparation for a parsing routine.

Lex source is a table of regular expressions and corresponding program fragments. The table is translated to a program, which reads an input stream, copying it to an output stream and partitioning the input into strings which match the given expressions. As each such string is recognized the corresponding program fragment is executed. The recognition of the expressions is performed by a deterministic finite automaton generated by Lex. The program fragments written by the user are executed in the order in which the corresponding regular expressions occur in the input stream.

YACC: Yet another Compiler-Compiler

Computer program input generally has some structure; in fact, every computer program that does input can be thought of as defining an ``input language" which it accepts. An input language may be as complex as a programming language, or as simple as a sequence of numbers. Unfortunately, usual input facilities are limited, difficult to use, and often are lax about checking their inputs for validity.

YACC provides a general tool for describing the input to a computer program. The YACC user specifies the structures of his input, together with code to be invoked as each such structure is recognized. YACC turns such a specification into a subroutine that han- dles the input process; frequently, it is convenient and appropriate to have most of the flow of control in the user's application handled by this subroutine.

Q.104. Explain the following term

(i) Overlays **(4)**

(ii) Macro definition and call **(4)**

Ans:

(i) An overlay is a part of program which has the same load origin as some other part of the program. Overlays are used to reduce the main memory requirement of a program. When the problem arises where the size of program exceeds the memory size, then overlays are used. Structure of such programs consists of a **permanently resident portion, called the root.**

A set of overlays

To start with, the root is loaded in memory and given control for the purpose of execution. Other overlays are overwrites a previously loaded overlay with same load origin, this reduces memory requirement of a program. It is also possible to execute programs whose size exceeds the memory size.

(ii) Macro: The assembly language programming often finds it necessary to repeat certain piece of code many times during the course of program. In such situations we find macro facility useful. Whole process consists of three steps:

1. Macro definition instruction
2. macro call
3. macro expansion

A macro definition is enclosed between a macro header statement and a macro end statement. Macro definitions are typically located at the start of a program. It consists of

1. A macro prototype statement
2. One or more model statements
3. Macro preprocessor

A macro call is called by writing the macro name in the mnemonics field of an assembly statement. Syntax for the same is

<macro name>[<actual parameter spec>[,..]]

A macro call leads to macro expansion, during macro call expansion the macro call statement is replaced by the sequence of assembly statements.

Q.105. Can the operand expression in an ORG statement contain forward references? If so, outline how the statement can be processed in a two-pass assembly scheme. (9)

Ans: (ORG (origin) is an assembler directive that

- Indirectly assign values to symbols
- Reset the location counter to the specified value-ORG value
- Value can be: constant, other symbol, expression
- No forward reference

Assemblers scan the source program, generating machine instructions. Sometimes, the assembler reaches a reference to a variable, which has not yet been defined. This is referred to as a **forward reference** problem. It is resolved in a **two-pass assembler** as follows:

On the first pass, the assembler simply reads the source file, counting up the number of locations that each instruction will take, and builds a **symbol table** in memory that lists all the defined variables cross-referenced to their associated memory address. On the second pass, the assembler substitutes opcodes for the mnemonics,

and variable names are replaced by the memory locations obtained from the symbol table.

Consider the following program

```
          ORG   $800
N:        DS    1
M:        DS    1
COLUMN:   DS    50
ODD:      DS    50
*
          CLR   M              ; initialize M
          LDAB  N              ; Put N into B
          LDX   #COLUMN        ; Point X to COLUMN
          LDY   #ODD           ; Point Y to ODD
LOOP:     LDAA  1,X+           ; Next number of COLUMN into A
          BPL   JUMP           ; Go to next number if positive
          BITA  #1             ; Z = 1 if, and only if, A is even
          BEQ   JUMP           ; Go to next number if even
          STAA  1,Y+           ; Store odd, negative number
          INC   M              ; Increment length of ODD
JUMP:     DBNE  B,LOOP         ; Decrement counter; loop if not done
          BGND                 ; Halt
```

On the first pass, the ORG statement sets the location counter to $800. Thus the label N has the value $800, the label M has the value $801, the label COLUMN has the value $802, and the label ODD has the value $834. The instruction CLR M will take three bytes , the instruction LDAB N will take three bytes, and so forth. Similarly, we see that the first byte of instruction we see that the first byte of instruction

```
LOOP:   LDAA  1,X+
```

will be at location $872. Thus the symbolic address LOOP has the value $872 and so on. Continuing in this way, we come to

```
BPL   JUMP
```

This program searches the array COLUMN looking for odd, negative, one-byte numbers, which then are stored in array ODD. The length of COLUMN is N and the length of ODD is M, which the program calculates. We do not know the second byte of this instruction because we do not know the value of the address JUMP yet. (This is called a forward reference, using a label whose value is not yet known.) However, we can leave this second byte undetermined and proceed until we see that the machine code for DBNE is put into location $87f, thus giving JUMP the value $87f. As we continue our first pass downward, we allocate three bytes for DBNE B,LOQP.

We do not find this instruction's offset yet, even though we already know the value of LOOP.

Scanning through the program again, which is the second pass, we can fill in all the bytes, including those not determined the first time through, for the instructions BPL JUMP, BEQ JUMP, and DBNE B,LOOP. At this time, all object code can be generated.

Q.106. What criteria should be adopted for choosing type of file organization **(8)**

Ans:
Choosing a file organization is a design decision, hence it must be done having in mind the achievement of good performance with respect to the most likely usage of the file. The criteria usually considered important are:
1. fast access to single record or collection of related records
2. Easy record adding/update/removal, without disrupting (1)
3. Storage efficiency
4. Redundancy as a warranty against data corruption.

Needless to say, these requirements are in contrast with each other for all but the most trivial situations, and it's the designer job to find a good compromise among them, yielding an adequate solution to the problem at hand. For example, easiness of adding/ etc. is not an issue when defining the data organization of CD ROM product, whereas fast access is given the huge amount of data that this media can store. However as it will become apparent shortly, fast access techniques are based on the use additional information about the records, which in turn competes with the high volumes of data to be stored.
Logical data Organization is indeed the subject of whole shelves of books in the "Database" section of your library. Here we'll briefly address some of the simpler used techniques, mainly because of their relevance to data management from the lower-level (with respect to a database's) point of view of an OS. Five organization models will be considered:

(i) Pile.
(ii) Sequential.
(iii) indexed-sequential.
(iv)Indexed
(v) Hashed.

Q.107 Compare pre-emptive and non-preemptive scheduling policies. (4)

Ans:
In preemptive scheduling we preempt the currently executing process. In non-preemptive we allow the current process to finish its CPU burst time.

Q.108. Explain the differences between:

(i) Logical and physical address space.
(ii) Internal and external fragmentation.
(iii)Paging and segmentation. (6)

Ans:

Logical Vs physical address space

(1) An address generated by the CPU is commonly referred to as a logical a logical address. The set of all logical addresses generated by a program is known as logical address space. Whereas, an address seen by the memory unit- that is, the one loaded into the memory-address register of the memory- is commonly referred to as physical address. The set of all physical addresses corresponding to the logical addresses is known as physical address space.

(2) The compile-time and load-time address-binding methods generate identical logical and physical addresses. However, in the execution-time address-binding scheme, the logical and physical-address spaces differ.

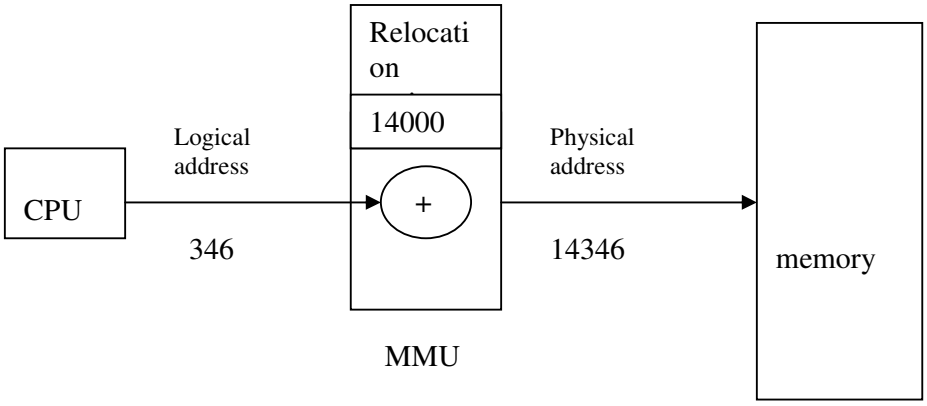
(3) The user program never sees the physical addresses. The program creates a pointer to a logical address, say 346, stores it in memory, manipulate it, compares it to other logical addresses- all as the number 346.

Only when a logical address is used as memory address, it is relocated relative to the base/relocation register. The memory-mapping hardware device called the memory-management unit(MMU) converts logical addresses into physical addresses.

(4) Logical addresses range from 0 to max. User program that generates logical address thinks that the process runs in locations 0 to max.

Logical addresses must be mapped to physical addresses before they are used. Physical addresses range from (R+0) to (R + max) for a base/relocation register value R.

(5) Example:



Mapping from logical to physical addresses using memory management unit (MMU) and relocation/base register

The value in relocation/base register is added to every logical address generated by a user process, at the time it is sent to memory, to generate corresponding physical address.

In the above figure, base/ relocation value is 14000, then an attempt by the user to access the location 346 is mapped to 14346.

(ii) Internal and external fragmentation

(1) When memory allocated to a process is slightly larger than the requested memory, space at the end of a partition is unused and wasted. This wasted space within a partition is called as internal fragmentation. When enough total memory space exists to satisfy a request, but it is not contiguous; storage is fragmented into a large number of small holes. This wasted space not allocated to any partition is called external fragmentation.

(2) Internal fragmentation is found in multiple fixed partition schemes where all the partitions are of the same size. That is, physical memory is broken into fixed-sized blocks.
External fragmentation is found in multiple variable partition schemes. Instead of dividing memory into a fixed set of partitions, an operating system can choose to allocate to a process the exact amount of unused memory space it requires.
(3) In multiple fixed partition scheme, the partition table needs to store either the starting address for each process or the number of the partition allocated to each process.
In multiple variable partition scheme, the overhead of managing more data increases. The partition table must store exact starting and ending location of each process and data about which memory locations are free must be maintained.
(4) In multiple fixed partition schemes, size/limit register is set at boot time and contains the partition size. Each time a process is allocated control of CPU, the operating system only needs to reset the relocation register. In multiple variable partition schemes, each time a different process is given control of the CPU, the operating system must reset the size/limit register in addition to the relocation register. The operating system must also make decisions on which partition it should allocate to a process.
(5) Internal fragmentation can be reduced using multiple variable partition method. However, this solution suffers from external fragmentation. External fragmentation can be solved using compaction where the goal is to shuffle the memory contents to place all free memory together in one large block. Another possible solution to the external fragmentation problem is to permit the logical address space of a process to be non contiguous. This solution is achieved by paging and segmentation.

(iii) Paging and segmentation

Paging	Segmentation
Computer memory is divided into small partitions that are all the same size and referred to as, page frames. Then when a process is loaded it gets divided into pages which are the same size as those previous frames. The process pages are then loaded into the frames.	Memory-management scheme that supports user view of memory. Computer memory is allocated in various sizes (segments) depending on the need for address space by the process.
Address generated by CPU is divided into: *Page number (p)* – used as an index into a *page table* which contains base address of each page in physical memory. *Page offset (d)* – combined with base address to define the physical memory address that is sent to the memory unit.	Logical address consists of a two tuple: <segment-number, offset>
Transparent to programmer (system allocates memory)	Involves programmer (allocates memory to specific function inside code)
No separate protection	Separate protection
No separate compiling	Separate compiling
No shared code	Share code

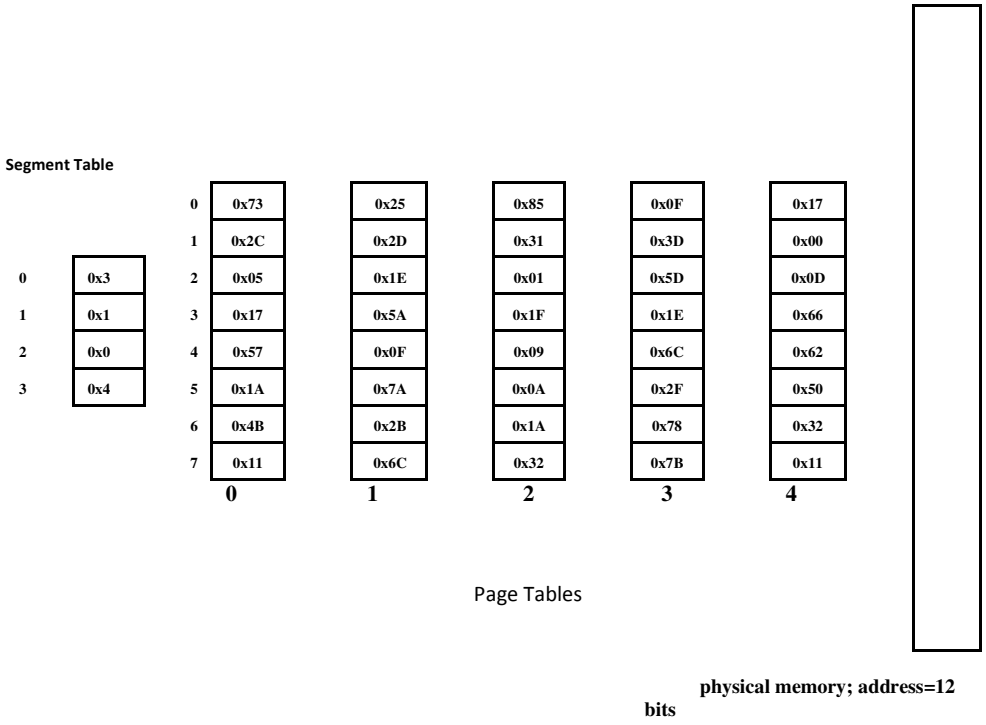
Q.109. Suppose that a process scheduling algorithm favors those processes that have used the least processor time in the recent past. Why will this algorithm favour /IO-bound processes, but not starve CPU-bound processes? (6)

Ans: It will favor the I/O-bound programs because of the relatively short CPU burst request by them; however, the CPU-bound programs will not starve because the I/O-bound programs will relinquish the CPU relatively often to do their I/O.

Q.110 List one advantage and one disadvantage of having large block size. (4)

Ans: The advantage of using a large block of memory is accommodation of maximum processes resulting is less number of page faults.
The disadvantage of using a large block of memory is occurrence of internal fragmentation i.e allocated memory may be slightly larger than requested memory.
e.g., suppose memory is allocated in blocks of 4K, a 1K process will waste 3K of space in its partition, as will a 5K process.

Q.111. Consider the following segmented paging memory system. There are 4 segments for the given process and a total of 5 page tables in the entire system. Each page table has a total of 8 entries. The physical memory requires 12 bits to address it; there are a total of 128 frames.



(i) How many bytes are contained within the physical memory?
(ii) How large is the virtual address?
(iii) What is the physical address that corresponds to virtual address 0x312?
(iv) What is the physical address that corresponds to virtual address 0x1E9? (8)

Ans:

(i) Number of bytes in physical memory is equal to 2 ^(7 + 7) = 16K bytes. This is because 12 bits are required to address physical memory location out of which 3 bits are to refer frame no. within page table + 2 bits to locate page from segment and remaining 7 bits for offset with frame.
(ii) The size of virtual memory is 2 ^ 20 (20 = 2 for segment index, + 3 for page table index + 3 for frame index in page table + 7 for frame number and 5 for offset within frame).
(iii) 312 (Hex) = 001100010010 = 00(segment) in table number 3 (refer to the data in question at entry 0 in segment table then find page 3, in page 3 find 110 (6th entry) which is 78; that is 120 th frame then the offset with the frame is given by the last 7 bits 0010010.
(iv) 312 (Hex) = 001100010010 = 00(segment) in table number 3 (refer to the data in question at entry 0 in segment table then find page 3, in page 3 find 110 (6th entry) which is 78; that is 120 th frame then the offset with the frame is given by the last 7 bits 0010010.

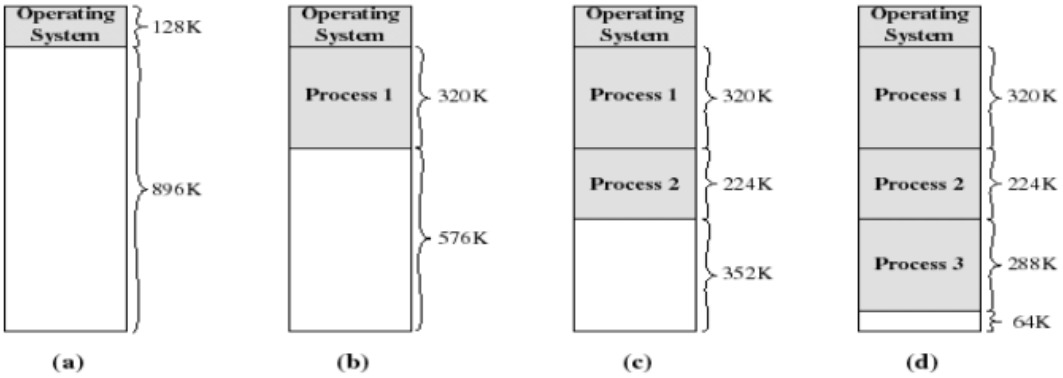
Q 112. Explain analysis and synthesis phase of a compiler. (7)

Ans:
The analysis and synthesis phases of a compiler are:
Analysis Phase: Breaks the source program into constituent pieces and creates intermediate representation. The analysis part can be divided along the following phases:
1. **Lexical Analysis**- The program is considered as a unique sequence of characters. The Lexical Analyzer reads the program from left-to-right and sequence of characters is grouped into **tokens**–lexical units with a collective meaning.
2. **Syntax Analysis**- The **Syntactic Analysis** is also called **Parsing**. Tokens are grouped into grammatical phrases represented by a **Parse Tree, which** gives a hierarchical structure to the source program.
3. **Semantic Analysis**- The **Semantic Analysis** phase checks the program for semantic errors (**Type Checking**) and gathers type information for the successive phases. **Type Checking** check types of operands; No real number as index for array; etc.
Synthesis Phase: Generates the target program from the intermediate representation. The synthesis part can be divided along the following phases:
1. **Intermediate Code Generator**- An intermediate code is generated as a program for an abstract machine. The intermediate code should be easy to translate into the target program.
2. **Code Optimizer**- This phase attempts to improve the intermediate code so that faster-running machine code can be obtained. Different compilers adopt different optimization techniques.
3. **Code Generator**- This phase generates the target code consisting of assembly code. Here
1. Memory locations are selected for each variable;
2. Instructions are translated into a sequence of assembly instructions;
3. Variables and intermediate results are assigned to memory registers.

Q.113. Explain the concept of variable-partition contiguous storage allocation. (6)

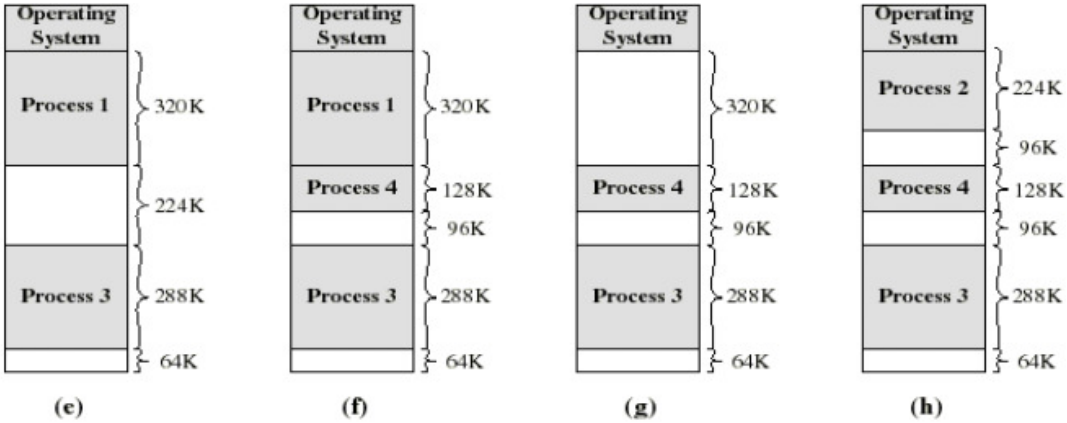
Ans:

Assume that we have 1024K main memory available in which 128K is occupied by operating system program. There are 4 jobs waiting for memory allocation in a job queue Applying FCFS scheduling policy, Process 1, Process 2 and Process 3 can be immediately allocated in memory. Process 4 cannot be accommodated because there is there is not enough space.



•A hole of 64K is left after loading 3 processes: not enough room for another process.

•Eventually each process is blocked. The OS swaps out process 2 to bring in process 4.



•Another hole of 96K is created.

•Eventually each process is blocked. The OS swaps out process 1 to bring in again process 2 and another hole of 96K is created.

Q.114. Suppose two processes enter the ready queue with the following properties:
Process 1 has a total of 8 units of work to perform, but after every 2 units of work, it must perform 1 unit of I/O (so the minimum completion time of this process is 12 units). Assume that there is no work to be done following the last I/O operation.
Process 2 has a total of 20 units of work to perform. This process arrives just behind P1. Show the resulting schedule for the shortest-job-first (preemptive) and the round-robin algorithms. Assume a time slice of 4 units of RR. What is the completion time of each process under each algorithm? (8)

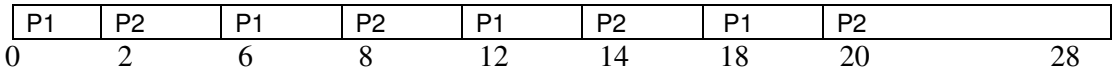
Ans:
S-J-F(pre emtive):

P1	P2	P1	P2	P1	P2	P1	P2	P2

0 2 3 5 6 8 9 11 12 28

Between 2-3unit of time, p2 is in ready queue and P1 has gone for I /O. So P2 should be executed as it is pre-emptive SJF algorithm.
Completion time of process p1 = 12 unit
Completion time of process p2 = 28 unit

Round- robin (R R) algorithm



Completion time of process p1 = 20 unit
Completion time of process p2 = 28 unit

Q.115. What are the relocation requirements in segmented addressing? (8)

Ans:
The relocation requirements of a program are influenced by the addressing structure of the computer system on which it is to execute. Use of the segmented addressing structure reduces the relocation requirements of a program
Consider the following assembly program of 8088. The ASSUME statement declares the Segment registers CS and DS to be available for memory addressing.

S.No.		Statement		offset
0001	DATA-HERE	SEGMENT		
0002	ABC	DW	25 0000	
0003	B	DW?	0002	
.				
.				
.				
0012	SAMPLE	SEGMENT		
0013		ASSUME	CS: SAMPLE, DS: DATA-HERE	
0014		MOV	AX, DATA-HERE	0000
0015		MOV	DS, AX	0003
0016		JMP	A	0005
0017		MOV	AL, B	0008
.		.		
.		.		
.		.		
0027	A	MOV	AX, BX	0196
.				
.				
.				
0043	SAMPLE	ENDS		
0044		END		

Translation time address of A is 0196. At S.NO 16, a reference to A is assembled as a displacement of 196 from the contents of the CS register. This avoids the use of an absolute address. Now no relocation is needed if segment SAMPLE is to be loaded in the memory starting at the address 2000 because CS register would be loaded with the address by a calling program (or by the OS). The effective operand address would be calculated as <CS> + 0196 = 2196. A similar situation for B in S.NO. 17. The reference to B is assembled as a displacement of 0002 from the contents of the DS register. Since the DS register would be loaded with the execution time address of DATA –HERE, the reference to B would be automatically relocated to the correct address.

TYPICAL QUESTIONS & ANSWERS

PART - I,

OBJECTIVE TYPE QUESTIONS

Each Question carries 2 marks.
Choose correct or the best alternative in the following:

Q.1 Literal means

(A) a string. **(B)** a string constant.
(C) a character. **(D)** an alphabet.

Ans:B

Q.2 Choose the correct answer

(A) Casting refers to implicit type conversion.
(B) Coercion refers to implicit type conversion.
(C) Casting means coercion.
(D) Coercion refers to explicit type conversion.

Ans:B

Q.3
```
printf ("%d", printf ("tim"));
```

(A) results in a syntax error **(B)** outputs tim3
(C) outputs garbage **(D)** outputs tim and terminates abruptly

Ans:B
printf statement will print tim3, "tim" due to inner printf statement and 3 as length due to outer printf statement.

Q.4 Output of the following program fragment is

```
x = 5;
y = x++;
printf("%d%d", x, y);
```

(A) 5, 6 **(B)** 5, 5
(C) 6, 5 **(D)** 6, 6

Ans:C
x is incremented by 1 and before that increment the value is assigned to y so value of x=6 and y=5.

Q.5 The value of an automatic variable that is declared but not initialised will be

(A) 0 **(B)** -1
(C) unpredictable **(D)** none of these

Ans:C

Q.6 Consider the following program

```
main ( )
{        float a = 0.5, b = 0.7;
                  if (b < 0.8)
             if (a < 0.5) printf ("ABCD");
                           else printf ("PQR");
                  else printf ("JKLF);
    }
```

The output is

(A) ABCD **(B)** PQR
(C) JKLF **(D)** None of these

Ans:B

Since b=0.7<0.8, the control goes to second "if" statement where (a<0.5) is false to printf statement in else part executed printing "PQR"

Q.7 The following program fragment

```
int *a;
*a = 7;
```

(A) assigns 7 to a **(B)** results in compilation error
(C) assigns address of a as 7 **(D)** segmentation fault

Ans:D

Q.8 A pointer variable can be

(A) passed to a function as argument.
(B) changed within function.
(C) returned by a function.
(D) assigned an integer value.

Ans:C

Q.9 'C' is often called a

(A) Object oriented language **(B)** High level language
(C) Assembly language **(D)** Machine level language

Ans:B

Q.10 The loop in which the statements within the loop are executed at least once is called

(A) do-while **(B)** while
(C) for **(D)** goto

Ans:A

Q.11 The control automatically passes to the first statement after the loop in

(A) continue statement **(B)** break statement
(C) switch statement **(D)** if statement

Ans:B

Q.12 A self contained block of statements that perform a coherent task of some kind is called a
(A) Monitor **(B)** Function
(C) Program **(D)** Structure

Ans:B

Q.13 Recursion is sometimes called
(A) Circular definition **(B)** Complex definition
(C) Procedure **(D)** Union

Ans:A

Q.14 Unsigned integer occupies
(A) Two bytes **(B)** Four bytes
(C) One byte **(D)** Eight bytes

Ans:B

Q.15 Each C preprocessor directive begins with
(A) # **(B)** include
(C) main() **(D)** {

Ans:A

Q.16
```
main() {
   long i = 30000;
   printf("%d", i); }
```
the output is
(A) 3000 **(B)** 30000
(C) 0 **(D)** -1

Ans:B

Q.17 The directive that can be used to test whether an expression evaluates to a nonzero value or not is
(A) #if **(B)** #elif
(C) #endif **(D)** #exit

Ans:A

Q.18
```
main() {
   printf("%p\n", main());
   }
```
(A) Prints the address of main function.

(B) Prints 0.
(C) Is an error.
(D) Is an infinite loop.

Ans:A

Q.19 The << operator is used for
(A) Right shifting **(B)** Left shifting
(C) Bitwise shifting **(D)** Bitwise complement

Ans:B

Q.20 The C language includes the header file standard input & output in
(A) stdlib.h library **(B)** stdio.h library
(C) conio.h library **(D)** #include library

Ans:B

Q.21 The value that follows the keyword CASE may only be
(A) constants **(B)** variable
(C) number **(D)** semicolon

Ans:A

Q.22 The statement which is used to terminate the control from the loop is
(A) break **(B)** continue
(C) goto **(D)** exit

Ans:A

Q.23 The machine registers are sometimes called
(A) local variables **(B)** global variables
(C) accumulators **(D)** static variables

Ans:A

Q.24 Set of values of the same type, which have a single name followed by an index is called
(A) function **(B)** structure
(C) array **(D)** union

Ans:C

Q.25 An array of pointers is same as
(A) pointer to array **(B)** pointers to pointers
(C) pointer to function **(D)** pointer to structure

Ans:B

Q.26 What is the output of the following program segment?

```
main()
{
  long i = 65536;
  printf("%d\n", i);
}
```

(A) 0 **(B)** 65536
(C) -1 **(D)** 65

Ans:A

Q.27 What is the output of the following program segment?

```
main()
{
  int i = 1;
do
  { printf("%d..", i);
  } while(i--);
}
```

(A) 0..1.. **(B)** 1..0..
(C) 0 **(D)** -1

Ans:B

Q.28 What is the output of the following program segment?

```
main()
{
  int i = ++2;
  printf("%d\n", i);
}
```

(A) 3 **(B)** 2
(C) 0 **(D)** -1

Ans:A
It is a compilation error. However if we write "i=2; ++i;" then value of i is printed as 3.

Q.29 The name of all functions end with a
(A) pair of parenthesis **(B)** semicolon
(C) braces **(D)** colon

Ans:A

Q.30 A float variable can store any variable within the range of
(A) -1.7×10^{38} to 1.7×10^{38} **(B)** -3.4×10^{38} to 3.4×10^{38}

(C) -7.2×10^{38} to 7.2×10^{38} **(D)** -1.2×10^{38} to 1.2×10^{38}

Ans:B

Q.31 scanf() can be used for reading
(A) double character **(B)** single character
(C) multiple characters **(D)** no character

Ans:C

Q.32 'C' allows a three-way transfer of control with the help of
(A) unary operator **(B)** relational operator
(C) ternary operator **(D)** comparison operator

Ans:C

Q.33 The statement that transfers control to the beginning of the loop is called
(A) break statement **(B)** exit statement
(C) continue statement **(D)** goto statement

Ans:C

Q.34 A variable which is visible only in the function in which it is defined, is called
(A) static variable **(B)** auto variable
(C) external variable **(D)** local variable

Ans:D

Q.35 The number of arguments supplied from the command line, by convention, is known as
(A) arg c **(B)** arg v
(C) #define **(D)** #include

Ans:A

Q.36 Output of the program given below is

```
int i;
main()
{
  printf("%d", i);
}
```

(A) 1 **(B)** 0
(C) -1 **(D)** Null

Ans:B

Q.37 What will be the output of the following program?

```
main()
{
  char *p = "ayqm";
  printf ("%c", ++*(p++));}
```

(A) b **(B)** z
(C) q **(D)** n

Ans:A

Q.38 What will be the output of the following program?

```
main()
{
  int i = 5;
  printf("%d", i=++i==6);
}
```

(A) 0 **(B)** 1
(C) 7 **(D)** 6

Ans:B

Q.39 Determine which of the following is a valid character constant

(A) '\\' **(B)** '\0'
(C) 'xyz' **(D)** '\052'

Ans:A

Q.40 The maximum value that an integer constant can have is

(A) .32767 **(B)** 32767
(C) 1.7014e+38 **(D)** –1.7014e+38

Ans:B

Q.41 The expression X=4+2%-8 evaluates

(A) –6 **(B)** 6
(C) 4 **(D)** None

Ans:B

Q.42 What will be the output of following program?

```
main()
{
   int x=15;
   printf("\n%d%d%d", x!=15, x=20, x<30);
}
```

(A) 0, 20, 1 **(B)** 15, 20, 30
(C) 0, 0, 0 **(D)** Error

Ans:A

Q.43 How many times the following program would print ("abc")?

```
main()
{
   printf("\nabc");
   main();
}
```

(A) Infinite number of times **(B)** 32767 times
(C) 65535 times **(D)** Till the stack does not overflow

Ans:A

Q.44 What would be output of the following program?

```
# define SQR(X) (X*X)
main()
{
   int a, b=3;
   a = SQR(b+2);
   printf("\n%d", a);
}
```

(A) 25 **(B)** 11
(C) Error **(D)** Garbage value

Ans:B

Q.45 What would be output of the following program?

```
#include "stdio.h"
main()
{
   printf("%d%d", size of (NULL!), size of (" "));
}
```

(A) 2 1 **(B)** 1 2
(C) 2 2 **(D)** 1 1

Ans:C

Q.46 What would be output of the following program, if the array begins at 65486?

```
main()
{
   int arr[ ] = {12, 14, 15, 23, 45};
   printf("%u%u", arr+1, &arr+1);
}
```

(A) 65486, 65486 **(B)** 65488, 65488
(C) 65488, 65496 **(D)** None of the above

Ans:C

Array begins at address 65486 so arr+1=65488 as size of int is 2 bytes and &arr+1=65486+10=65496 as there are 5 elements in array.

Q.47 Given the statement, maruti.engine.bolts=25;
which of the following is true?
(A) Structure bolts is nested within structure engine
(B) Structure engine is nested within structure maruti
(C) Structure maruti is nested within structure engine
(D) Structure maruti nested within structure bolts

Ans:B

Q.48 Which amongst the following is not a keyword?
(A) external **(B)** int
(C) float **(D)** double

Ans:A

Q.49 If a = 5 and b = 7 then the statement p = (a > b) : a ? b
(A) assigns a value 5 to p **(B)** assigns a value 7 to p
(C) assigns a value 8 to p **(D)** gives an error message

Ans:D

Q.50 The expression P >> 6 shifts all bits of P six places to right. What is the value of P >> 6 if P = 0×6db7 ?
(A) 0×1234 **(B)** 0×0001
(C) 0×0000 **(D)** 0×1B6

Ans:D

Q.51 If an integer occupies 4 bytes and a character occupies 1 bytes of memory, each element of the following structure would occupy how many bytes?

```
struct name
       {
       int age;
                                          char name [20];
       }
```

(A) 5 **(B)** 24
(C) 21 **(D)** 22

Ans:B

Q.52 If an array is used as function argument, the array is passed
(A) by value. **(B)** by reference

(C) by name. **(D)** the array cannot be used as a function argument.

Ans:B

Q.53 To access a structure element using a pointer, _____ operator is used

(A) dot (.) **(B)** pointer (&)
(C) pointer (*) **(D)** arrow ($\rightarrow$)

Ans:D

Q.54 The library function sqrt( ) operates on a double precision argument. If, i is an integer variable, which one of the following calls would correctly compute sqrt(i)?

(A) sqrt((double)i) **(B)** (double) sqrt(i)
(C) (double) (sqrt(i)) **(D)** sqrt(i)

Ans:A

Q.55 What will happen if the following loop is executed?

```
int num = 0;
do
{
   --num;
      printf("%d", num);
   num++;
   }while (num >= 0);
}
```

(A) The loop will run infinite number of times.
(B) The program will not enter the loop.
(C) There will be a compilation error.
(D) There will be runtime error.

Ans:C

Q.56 The break statement causes an exit

(A) Only from the innermost loop.
(B) Only from the innermost switch.
(C) From the innermost loop or switch.
(D) From the program.

Ans:C

Q.57 It is necessary to declare the type of function in the calling program if

(A) Function returns an integer.
(B) Function returns a non-integer value.
(C) Function is not defined in the same file.
(D) Function is called number of times.

Ans:B

Q.58 The function fprintf is used in a program

(A) When too many printf calls have been already used in the program.
(B) In place of printf, since printf uses more memory.
(C) When the output is to be printed on to a file.
(D) When the type of variables to be printed are not known before.

Ans:C

Q.59 The following statement displays

```
float x = 2000.53;
printf ("%e", x);
```

(A) 2.00053e+04 **(B)** 2.00053e+03
(C) 2.00053+e04 **(D)** 2.0005e+03

Ans:B

Q.60 The output of the following is

```
int a = 75;
printf ("%d%%", a);
```

(A) 75 **(B)** 75%%
(C) 75% **(D)** None of the above

Ans:D

Q.61 C language was invented by

(A) Abacus **(B)** Charles babage
(C) Thomson **(D)** Dennis Ritchie

Ans:D

Q.62 The given FOR loop is

```
for ( ; ; )
{
    printf(" ");
}
```

(A) valid **(B)** indefinite
(C) invalid **(D)** displays runtime errors

Ans:D
The given for loop displays runtime errors because no test condition is given. Test condition is must inside for loop.

Q.63 The following code displays

```
main( )
```

```
{
   int *p;
   p = (int*) malloc(sizeof(int));
   *p = 10;
   printf("p = %d\n", *p);
}
```

(A) 10 **(B)** 1542 (address of p)
(C) 20 **(D)** None of the above

Ans:A

Q.64 The _____ operator is a technique to forcefully convert one data type to the others

(A) Cast **(B)** Conversion
(C) Type **(D)** Uniary

Ans:A

Q.65 The output of the following will be

```
for (x=1, y=5; x+y<=10; x++)
{
   printf("%d%d", x,y);
   y++;
}
```

(A)
1 5
2 6
3 7

(B)
1 5
2 6
3 7
4 8

(C)
1 5
1 6
1 7
1 8
1 9

(D)
1 5
2 5
3 5
4 5
5 5

Ans:A

Q.66 The _____ statement causes immediate exit from the loop overriding the condition test

(A) Exit **(B)** Break
(C) Goto **(D)** None of the above

Ans:B

Q.67 The output of the following code is

```
a = 5;
a << 1;
printf("%d", a);
```

(A) 5 **(B)** 6
(C) 2 **(D)** 3

Ans:A

Q.68 The purpose for mode "w+b" in file operation is
(A) create a binary file for write
(B) create a binary file for read/write
(C) open a binary file for writing
(D) open a binary file for reading/writing

Ans:B

Q.69 The operators << and >> are
(A) assignment operator **(B)** relational operator
(C) logical operator **(D)** bitwise shift operator

Ans D

Q.70 Which of the following numerical value is invalid constant
(A) .75 **(B)** 9.3e2
(C) 27,512 **(D)** 123456

Ans C

Q.71 A C program contains the following declaration int i=8, j=5 what would be the value of following expression?
abs(i-2*j)
(A) 2 **(B)** 4
(C) 6 **(D)** 8

Ans A

Q.72 What will be the output of the following program

```
main( )
{
        int k,  num=30;
        k=(num>5 ? (num<= 10 ? 100:200):500);
        printf("\n%d", num);
}
```

(A) 100 **(B)** 5
(C) 30 **(D)** 500
Ans C

Q.73 What is the output of the following code

```
int n = 0, m=1;
```

```
do
{
printf("%d", m);
m++;
}
while(m<=n);
```

(A) 0 **(B)** 2
(C) 1 **(D)** 4

Ans C

Q.74 If a=8 and b=15 then the statement

```
x= (a>b) ? a:b;
```

(A) assigns a value 8 to x **(B)** gives an error message
(C) assigns a value 15 to x **(D)** assigns a value 7 to x

Ans C

Q.75 What is the output of the following code

```
int n=0, m;
for (m=1;  m<=n+1;  m++)
printf("%d", m);
```

(A) 2 **(B)** 1
(C) 0 **(D)** 6

Ans B

Q.76 How many times the following loop be executed

```
{
…
ch = 'b';
while(ch >= 'a' && ch <= 'z')
ch++;    }
```

(A) 0 **(B)** 25
(C) 26 **(D)** 1

Ans B

Q.77 Which of the following is FALSE in C

(A) Keywords can be used as variable names
(B) Variable names can contain a digit
(C) Variable names do not contain a blank space
(D) Capital letters can be used in variable names

Ans A

Q.78 int **ptr; is
(A) Invalid declaration **(B)** Pointer to pointer
(C) Pointer to integer **(D)** none of the above

Ans B

Q.79 A Pixel is _____.
(A) a computer program that draws picture
(B) a picture stored in secondary memory
(C) the smallest resolvable part of a picture
(D) None of these

Ans C

Q.80 Which number system is usually followed in a typical 32-bit computer?
(A) 2 **(B)** 10
(C) 16 **(D)** 32

Ans A

Q.81 Which technology is used in optical disks?
(A) Mechanical **(B)** Electrical
(C) Electro Magnetic **(D)** Laser

Ans D

Q.82 Which of the following storage devices can store maximum amount of data?
(A) Floppy Disk **(B)** Hard Disk
(C) Compact Disk **(D)** Magneto Optic Disk

Ans B

Q.83 EPROM can be used for
(A) Erasing the contents of ROM
(B) Reconstructing the contents of ROM
(C) Erasing and reconstructing the contents of ROM
(D) Duplicating ROM

Ans C

Q.84 Memory unit is one part of
(A) Input device **(B)** Control unit
(C) Output device **(D)** Central Processing Unit

Ans D

Q.85 MS-WORD is a
(A) system software **(B)** high level language
(C) spread sheet application **(D)** word processing package

Ans D

Q.86 The grammar dialog box can be involved by choosing grammar from _____ menu.
(A) insert **(B)** file
(C) tools **(D)** view

Ans C

Q.87 A template stores
(A) styles, macros
(B) Auto Text entires, Customized word command Settings
(C) graphics, text
(D) All of the above

Ans A

Q.88 To return the remainder after a number is divided by a divisor in EXCEL we use the function
(A) ROUND( ) **(B)** FACT( )
(C) MOD( ) **(D)** DIV( )

Ans C

Q.89 _____ unit controls the flow and manipulation of data and information.
(A) Arithmetic logic **(B)** Central
(C) Middle **(D)** Control

Ans D

Q.90 Usually, an algorithm will contain a number of procedural steps which are dependent on results of previous steps and is called _____.
(A) Flowchart **(B)** Chart
(C) Drawing Chart **(D)** Food Chart

Ans A

Q.91 The performance of cache memory is frequently measured in terms of a quantity called:
(A) bit ratio **(B)** nor ratio
(C) no ratio **(D)** hit ratio

Ans D

Q.92 In addition to communicating with I/O, the processor must communicate with the _____ unit.
(A) control **(B)** memory
(C) Arithmetic **(D)** process

Ans B

Q.93 Which of the following was not associated with second generation computers?
(A) high level procedural language **(B)** operating system
(C) magnetic core and transistor **(D)** All of the above were associated

Ans D

Q.94 Which of the following number system/code uses only 0s and 1s
(A) decimal **(B)** octal
(C) hexadecimal **(D)** none of the above

Ans D

Q.95 A group of characters that has some meaning as a unit is known as a
(A) field **(B)** file
(C) record **(D)** word

Ans A

Q.96 The process of production of customer list in alphabetical order falls under the category of
(A) editing **(B)** sorting
(C) updating **(D)** calculating

Ans B

Q.97 State in case of following statement whether it is true or false?
Magnetic drums and disks are quiet similar in operation.
(A) True **(B)** False

Ans A

Q.98 State in case of following statement whether it is true or false?
Ink-jet printers are classified as impact printers.
(A) True **(B)** False

Ans B

Q.100 A Compiler is _____.
(A) a combination of computer hardware
(B) a program which translates from one high-level language to another

(C) a program which translates from one high-level to a machine level
(D) None of these

Ans C

Q.101 When a key is pressed on the keyboard, which standard is used for converting the keystroke into the corresponding bits
(A) ANSI **(B)** ASCII
(C) EBCDIC **(D)** ISO

Ans B

Q.102 Which of the following is not an output device?
(A) Scanner **(B)** Printer
(C) Flat Screen **(D)** Touch Screen

Ans A

Q.103 The memory location address are limited to
(A) 00000 to 9ffff(16) **(B)** 00001 to 9ffff(16)
(C) 00010 to 9ffff(16) **(D)** 10000 to 9ffff(16)

Ans A

Q.104 The programs which are as permanent as hardware and stored in ROM is known as
(A) Hardware **(B)** Software
(C) Firmware **(D)** ROMware

Ans C

Q.105 Memory is made up of
(A) Set of wires **(B)** Set of circuits
(C) Large number of cells **(D)** All of these

Ans C

Q.106 Using Find command in Word, we can search
(A) characters **(B)** formats
(C) symbols **(D)** All of the above
Ans D

Q.107 MS-Word automatically moves the text to the next line when it reaches the right edge of the screen and is called
(A) Carriage Return **(B)** Enter
(C) Word wrap **(D)** None of the above

Ans C

Q.108 MS-EXCEL can be used to automate
(A) Financial statements, Business forecasting
(B) Transaction registers, inventory control
(C) Accounts receivable, accounts payable
(D) Any of the above

Ans D

Q.109 Command to delete all files and folders is _____.
(A) Deltree **(B)** Del
(C) Remove **(D)** CD

Ans A

Q.110 Machine language is a language
(A) Directly understood by a computer.
(B) Which needs to be translated.
(C) Which uses mnemonics.
(D) In which programs are written first.

Ans A

Q.111 Which of the following device have a limitation that one can only add information to it but cannot erase or modify.
(A) Floppy Disk **(B)** Hard Disk
(C) Tape Drive **(D)** CDROM

Ans D

Q.112 Which amongst the following devices can store the maximum amount of data
(A) Floppy Disk **(B)** Hard Disk
(C) Compact Disk **(D)** Magnetic Optic Disk

Ans B

Q.113 EPROM can be used for
(A) Erasing the contents of ROM.
(B) Reconstructing the contents of ROM.
(C) Erasing and reconstructing the contents of ROM.
(D) Duplicating ROM.

Ans A

Q.114 The earliest calculating devices are

(A) Abacus. **(B)** Clock.
(C) Difference . **(D)** None of these.

Ans A

Q.115 The language that the computer can understand and execute is called
(A) Machine Language. **(B)** Application Software.
(C) System Program. **(D)** Assembly language.

Ans A

Q.116 The three sequential function of CPU operation are
(A) Decode, fetch, execute. **(B)** Execute, decode, fetch.
(C) Fetch, execute, decode. **(D)** Fetch, decode, execute.

Ans D

Q.117 Moving process from a main memory to a disk is called
(A) Scheduling **(B)** Caching
(C) Swapping **(D)** Spooling

Ans C

Q.118 While running DOS on a PC, the command that is used to duplicate the entire diskette?

(A) COPY **(B)** DISK COPY
(C) CHK DISK **(D)** TYPE

Ans B

Q.119 If data is processed as it arrives, this type of data processing is called
(A) Real time processing. **(B)** Batch processing.
(C) Off line processing. **(D)** Distributed processing.

Ans A

Q.120 Client/Server architecture can be used in
(A) LAN **(B)** WAN
(C) MAN **(D)** All of these.

Ans D

Q.121 Which of the following is not a programming language?
(A) UNIX **(B)** LISP
(C) BASIC **(D)** ADA

Ans A

Q.122 The process of retaining data for future use is called
(A) Reading **(B)** Writing
(C) Storing **(D)** Coding

Ans C

Q.123 The process of manipulating the data to achieve some meaningful results is called
(A) Information handling. (B)Data sharing.
(C) Data distribution. (D)Data processing.

Ans D

Q.124 Windows is a(n)
(A) Operating system. **(B)** User interface
(C) Operating environment. **(D)** Programming platform.

Ans C

Q.125 The register that keeps track of the program during execution
(A) Address register **(B)** Program counter.
(C) Data register **(D)** Accumulator

Ans B

Q.126 Typical processing speed of Super Computer is of the order of
(A) 100 MIPS **(B)** 200 MIPS
(C) 300 MIPS **(D)** 400 MIPS and above

Ans D

Q.127 While using DIR command,_____switch can not be used with it
(A) /P. **(B)** /W.
(C) /S. **(D)** /T.

Ans D

Q.128 _____ is not a utility program
(A) Debugger **(B)** Editor
(C) Spooler **(D)** Defragmenter

Ans C

Q.129 The access method used for magnetic tape is
(A) Direct **(B)** Random

(C) Sequential **(D)** Indexed

Ans C

Q.130 _____ unit coordinates the sequencing of events within the central processor of a computer.
(A) Logic unit. **(B)** Arithmetic unit.
(C) Storage unit. **(D)** Control unit.

Ans D

Q.131 Which of the following is not a direct entry input device?
(A) Optical scanner. **(B)** Digitizer.
(C) Keyboard. **(D)** Light pen.

Ans C

Q.132 A network, which is used for sharing data, software and hardware among several users of microcomputers, is called
(A) Wide Area Network. **(B)** Metropolitan Area Network.
(C) Local Area Network. **(D)** Value Added Network.

Ans C

Q.133 The instructions / programs that are loaded into main memory when a computer is booted are
(A) Internal commands. **(B)** External commands.
(C) Utility Programs. **(D)** Loader.

Ans A

Q.134 Which of the following types of memory loses data when power is switched off?
(A) Magnetic tape **(B)** Static Random Access Memory
(C) Magnetic disk **(D)** CD-ROM

Ans B

Q.135 Which is the latest write-once optical storage media
(A) Digital Page. **(B)** CD - ROM disk.
(C) WORM disk. **(D)** Magneto- optical disk.

Ans B

Q.136 The factor that does not affect the storage capacity of a hard disk
(A) Track density. **(B)** Height of the hard disk drive.
(C) Recording density. **(D)** Number of plates.

Ans B

Q.137 The operating system that reads and reacts in terms of actual time is
(A) Batch system. **(B)** Quick response system.
(C) Real time system. **(D)** Time sharing system.

Ans C

Q.138 Which of the following is a spreadsheet package?
(A) Corel Draw. **(B)** Wordstar
(C) EXCEL. **(D)** MS-WORD.

Ans C

Q.139 Find and Replace option is placed under the_____ menu.
(A) Edit. **(B)** Insert.
(C) View. **(D)** File.

Ans A

Q.140 Ink-jet printers are
(A) Impact printers **(B)** Laser printers
(C) Non-impact printers **(D)** Optical printers

Ans C

Q.141 A refreshing circuit is required in
(A) SRAM **(B)** EPROM
(C) DRAM **(D)** EEPROM

Ans C

Q.142 The first component of the machine language statement is called
(A) Lines of code. **(B)** Op-code.
(C) Pseudocode. **(D)** Operators and operands.

Ans B

Q.143 10001 x 101 =
(A) 101101. **(B)** 1010101.
(C) 100101. **(D)** 101010.

Ans A

Q.144 In heap files, records are placed in
(A) Random order. **(B)** Sequential order.
(C) Indexed order **(D)** Printer-referenced order.

Ans A

Q.145 In immediate addressing mode the immediate data is

(A) Constant data. **(B)** Variable data
(C) Symbolic data. **(D)** **(A)** & **(C)** both

Ans A

Q.146 Conditional jump instructions in 8086 are always
(A) Short-jumps ranging from –128 bytes to +127 bytes.
(B) Jumps ranging from –256 bytes to +255 bytes.
(C) Short-jumps ranging from –64 bytes to +63 bytes.
(D) Jumps ranging from –256 to +256 bytes.

Ans A

Q.147 Lotus notes is a
(A) Spreadsheet package **(B)** Programming tool.
(C) Notepad **(D)** Groupware.

Ans D

PART – II

DESCRIPTIVES

Q.1 Write a C program that reads a fixed point real number in a character array and then displays rightmost digit of the integral part of number. **(6)**

Ans: A C program to display rightmost digit of the integral part of a read real number:

```
#include<stdio.h>
#include<conio.h>
void main()
{
   char a[10];
   int i;
   clrscr();
   printf("enter a fixed point real no. : ");
   scanf("%s",a);
   i=0;
   while(a[i]!='.')
   i++;
   printf("%c",a[--i]);
   getch();
}
```

Q.2 Output the number x = 56.1624 under the following format specification

(i) printf ("%7.2f", x)
(ii) printf (%f", x)
(iii) printf("8.2e", x)
(iv) printf("%e", x) **(4)**

Ans: The output of number 56.1624 under various format specification is:

(i) 56.16
(ii) 56.162399
(iii) 5.62e+01
(iv) 5.616240e+01

Q.3 Write C statements to read the values of three variables of the type int, float and string and print them if correct data is entered otherwise print "error in input". **(4)**

Ans: A C program that read three values and print these if data input is correct otherwise print "error in input":

```
#include<stdio.h>
#include<conio.h>
void main()
{
```

```
   int n;
   float m;
   char p;
   printf("enter the values of n,m and p");
   ((scanf("%d %f %c",&n,&m,&p)==3)?printf("correct
   input"):printf("error in input"));
   getch();
}
```

Q.4 Write a C program that uses 'for' construct to find the sum of the following harmonic series for a given value of n and display the sum.

1 + 1/2 +1/3 +… + 1/n **(6)**

Ans: A C program to find and display sum of given harmonic series is listed below:

```
 #include<stdio.h>
#include<conio.h>
void main()
{
   int i,n;
   float sum=0.0;
   clrscr();
   printf("enter the value of n: ");
   scanf("%d",&n);
   for(i=1;i<=n;i++)
   sum=sum+(float)1/i;
   printf("sum of the series upto %d = %f",n,sum);
   getch();
}
```

Q.5 Write conditional operators to evaluate the following function

y = 2.4x + 3, for x <= 2
y = 3x –5, for x > 2 **(4)**

Ans: The conditional operator for given problem can be written as y = ((x < = 2)? (2.4 * x + 3): (3 * x + 5)). A program to evaluate this conditional operator is given below:

```
#include<stdio.h>
#include<conio.h>
void main()
{
   int x;
   float y;
   clrscr();
   printf("enter the values of x\n");
   scanf("%d",&x);
   y=((x<=2)?(2.4*x+3):(y=3*x+5));
   printf("y=%f",y);
   getch();}
```

Q.6 Write a C program fragment using "do…while" construct to print out even numbers between 10 to 100 making sure that two numbers are written per line. **(4)**

Ans: A C program to print even numbers between 10 to 100 using "do-while" construct:

```
#include<stdio.h>
#include<conio.h>
void main()
{
   int i,k;
   clrscr();
   k=10;
   i=0;
   do
   {
      if(k>10)
      {
         printf("%d",k);
         printf(" ");
         if(i%2==0)
         printf("\n");
      }
      i++;
      k=k+2;
   }while(k<100);
   getch();
}
```

Q.7 Given an integer, write a C program that displays the number as follows:

first line : All digits of integer
second line : all except first rightmost digit
third line : all except two rightmost digits .
. .
last line : leftmost digit **(8)**

Ans: A C program to display a number as follows:

```
#include<stdio.h>
#include<conio.h>
void main()
{
   int n,k,l,i,r,a[100];
   clrscr();
   printf("enter any integer value");
   scanf("%d",&n);
   while(n>0)
   {
      printf("%d",n);
      printf("\n");
```

```
      n=n/10;
    }
    //to have space between the digits of the integer
    k=n;
    while(k>0)
    {
      l=0;
      n=k;
      while(n>0)
      {
        r=n%10;
        a[l]=r;
        l++;
        n=n/10;
      }
      l--;
      for(i=l;i>=0;i--)
      {
        printf("%d",a[i]);
        printf(" ");
      }
      printf("\n");
      k=k/10;
    }
    getch();
  }
```

Q.8 Describe the output of the following C program fragment

```
void main ()
{  int k = 0, x = 0;
  while (k < 25)
  {          if (k % 5 == 0)
              {    x += k;
                                printf("%d  ", x);
               }
  ++k;
  }
  printf("\nx=%d ", x +k);
  }
```

(6)

Ans: Output of the given program segment is:
05153050
x=75
k=0; test condition of while loop is true; If condition false till k=5
At k=5, (k%5==0) is true so x=x+k so printf prints x=0+5=5;
At k=10, (k%5==0) is true so x=x+k so printf prints x=5+10=15;
At k=15, (k%5==0) is true so x=x+k so printf prints x=15+15=30;

At k=20, (k%5==0) is true so x=x+k so printf prints x=30+20=50; k is incremented till 25 where we are out of while loop.
Outer printf statement prints x+k=50+25=75.

Q.9 Shown below is a Floyd's triangle. Write a C program to print this triangle **(8)**

```
1
2     3
4     5     6
7     8     9     10
11    …                 15

79    …                       91
```

Ans: A C program to print Floyd's triangle upto 91 is given below:

```
#include<conio.h>
void main()
{
        int num=1,k=1,j;
        clrscr();
        while(num<=91)
        {
              for(j=1;j<=k;j++)
                    printf(" %d",num++);
              printf("\n\n");
              k++;
        }
        getch();
}
```

Q.10 Given the string "DATA PROCESSING", write a C program to read the string from terminal and display the same in the following formats.
(i) DATA PROCESSING
(ii) DATA
PROCESSING **(6)**

Ans: A C program that read string "DATA PROCESSING" from the terminal and display that in two formats:
(i)DATA PROCESSING
(ii)DATA

PROCESSING

```
 #include<stdio.h>
#include<conio.h>
void main()
{
```

```
    int i;
    char a[50];
    clrscr();
    printf("enter the string");
    gets(a);
    puts(a);
    i=0;
    while(a[i]!=' ')
    {
       printf("%c",a[i]);
       i++;
    }
    i++;
    printf("\n");
    while(a[i]!='\0')
    {
       printf("%c",a[i]);
       i++;
    }
    getch();
}
```

Q.11 Write a C program to compute the value of sin function. The loop must terminate when the absolute value of the term is less than 0.00001.

$$\sin x = x\ x - x^3/3! + x^5/5! - x^7/7! + \ldots$$ **(8)**

Ans: A C program to compute the sum of sin function is listed below:

```
#include<conio.h>
#include<math.h>
void main()
{
 float x,dig,sum;
 int i,n;
 clrscr();
 printf("\n Enter the value of x : ");
 scanf("%f",&x);
 printf("\n Enter the value of n : ");
 scanf("%d",&n);
 /*convert x into radians*/
 x=x*3.1412/180;
 sum=x;
 dig=x;
 for(i=1;i<=n;i++)
 {
     dig=(dig*pow((double)(-1),(double)(2*i-
1))*x*x)/(2*i*(2*i+1));
```

```
        sum+=dig;
    }
    printf("\n The sum is : %6.2f",sum);
    getch();
}
```

Q.12 Write a switch statement that will examine the value of an integer variable flag and print the following messages: **(6)**

It is hot weather; if flag has value 1
It is a stormy weather; if flag has value 2
It is sticky weather; if flag has value 3
It is a pleasant weather; otherwise

Ans: A Program to demonstrate switch statement:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int flag;
    printf( "Enter any value\n" );
    scanf( "%d", &flag );
    switch ( flag ) {
        case 1:
            printf( "It is hot weather!\n" );
            break;
        case 2:
            printf( "It is a stormy weather!\n" );
            break;
        case 3:
            printf( "It is a sticky weather!\n" );
            break;
        default:
            printf( "It is a pleasant weather!\n" );
            break;
    }
    getch();
}
```

Q.13 Define a structure named 'student' containing two fields 'name' and 'marks'.

Ans: A structure student:

```
struct student
{
   char name[10];
   int marks[6];
   int total;
};
```

Q.14 Declare an array of structure having 50 elements of student type. **(2)**

Ans: An array of structure:
struct student stu[50];

Q.15 Write an input statement for inputting the marks and the names of 50 students defined as above. **(3)**

Ans: Statements for inputting marks of 50 students:

```
for(i=0;i<50;i++)
{
    printf("%d. name : ",i+1);
    scanf("%s",&stu[i].name);
    printf("\nenter marks :");
    scanf("%d",&stu[i].marks);
    printf("\n");
}
```

Q.16 Write a complete C program to compute and print the names of those students who have got more than 80 marks. Also print their marks along with their names. **(7)**

Ans: A C program to compute and print names of students who scored more than 80 marks

```
#include<stdio.h>
#include<conio.h>
struct student
{
   char name[10];
   int marks;
};
void main()
{
   int i,n;
   struct student stu[50];
   clrscr();
   printf("enter no. of students");
   scanf("%d",&n);
   printf("ENTER NAME AND MARKS\n");
   for(i=0;i<n;i++)
   {
      printf("%d. name : ",i+1);
      scanf("%s",&stu[i].name);
      printf("\nenter marks :");
      scanf("%d",&stu[i].marks);
      printf("\n");
   }
   for(i=0;i<n;i++)
   {
```

```
        if(stu[i].marks>80)
        {
           printf("%s",stu[i].name);
           printf("\t");
           printf("%d",stu[i].marks);
           printf("\n");
        }
     }
     getch();
}
```

Q.17 Write a recursive function in C to compute the value of x^n where n is a positive integer and x has a real value. **(7)**

Ans: A recursive function to compute the value of x^n, power(x,n) is given below:

```
#include<stdio.h>
#include<conio.h>
void main()
{
   float x,y;
   int n;
   clrscr();
   printf("enter the value of x : ");
   scanf("%f",&a);
   printf("enter the value of n : ");
   scanf("%d",&n);
   y=power(x,n);
   printf("%f raise to power %d = %f",x,n,y);
   getch();
}
float power(float a, int b)
{
   float  k;
   if(b==1)
   return(a);
   else
   k=a*power(a,b-1);
   return(k);}
```

Q.18 Write a function 'exchange' to interchange the values of two variables say x and y. Illustrate the use of this function in calling program. Assume that x and y are defined as global variables. **(7)**

Ans: A C function exchange to interchange the values of x and y:

```
#include<stdio.h>
#include<conio.h>
```

```
#include<math.h>
int i,j;
int x,y;
void main()
{
    i=10,j=20;
   clrscr();
   printf("The values before exchange is i: %d,
   j:%d\n",i,j);
   exchange();
   printf("The values after exchange is i: %d,
   j:%d\n",i,j);
   printf("\n");
   getch();
}
void exchange()
{ int temp;
   temp = x;
   x = y;
   y = temp;
 }
```

Q.19 Write a C function that returns 1 if the argument is a prime number and returns 0 otherwise. **(7)**

Ans: A C function prime(n) described below returns 1 if the argument is a prime number otherwise 0:

```
#include<stdio.h>
#include<conio.h>
void main()
{
   int n,flag;
   clrscr();
   printf("enter any number");
   scanf("%d",&n);
   flag=prime(n);
   if(flag==1)
   {
      printf("no. is prime");
      else
      printf("%d",flag);
      getch();
   }
   int prime(n)
   { int i;
      for(i=2;i<n;i++)
      {
         if(n%i==0)
```

```
            return 0;
        }
    }
     return 1;
}
```

Q.20 Write a C function for searching an element in an array of size N. Assume that elements in array are stored in ascending order. **(7)**

Ans: A C function to search an element in an array of size n:

```
rec(int a[],int low,int high,int m)
{
   int mid;
   mid=(low+high)/2;
   if(low<=high)
   {
      if(a[mid]==m)
         return 1;
      if(a[mid]>m)
         rec(a,low,(mid-1),m);
      if(a[mid]<m)
         rec(a,(mid+1),high,m);
   else
      printf("element is not present");
     }
}
```

Q.21 Explain the declaration int (*p (char *a))[10]; **(3)**

Ans: *int (*p (char *a))[10];*
interpretation of the above statement is:
*int (*p (char *a))[10];*
char *a :- pointer to character (Character Array)
(*p (Character Array))
int (*p (Character Array))
Integer pointer to character array
*int (*p (Character Array))[10]*
Array of integer pointer to character array
The given statement represent array of int pointer to char array.

Q.22 Suppose a member of a structure is a pointer variable. How can the object of the pointer be accessed in terms of structure variable name and the member name? **(3)**

Ans:

```
struct rec
{
```

```
        int *a;}obj;
```

to use pointer member with structure variable following operator is used.
obj->a;
that is ordinary variables are used with dot (.) operator and pointer variables are used with arrow (->) operator.

Q.23 What happens when a pointer to structure is incremented? What danger is associated with this type of operation? **(2)**

Ans: When pointer to structure is incremented then the pointer points to the next block of memory. As for example

```
struct rec
{
        int a;
        char b[10], *c;
};
main()
{
        struct rec *obj;
        obj=(struct rec *)malloc(sizeof(struct rec));
        obj->a=10;
        strcpy(obj->b, "XYZ");
        strcpy(obj->c, "ABC");
        printf ("\n%d\t%s\t%s",obj->a, obj->b, obj->c);
        //Incrementing pointer to structure
           obj++;
        obj->a=15;
        strcpy(obj->b, "PQR");
        strcpy(obj->c, "MNO");
        printf ("\n%d\t%s\t%s",obj->a, obj->b, obj->c);
}
```

In the above program, dynamic memory allocation is used for assigning one block of structure in obj. Later the pointer to structure (obj) is incremented, so that pointer points to the next block of memory.
Danger associated with increment of pointer to structure:
As in the above program structure member is a char pointer. When pointer to structure is incremented then it may results in memory overwrite, because the size for the pointer member is not defined.

Q.24 Distinguish between the following:

(i) Syntactic error and semantic error
(ii) Run time error and logical error
(iii) Compiler and Interpreter **(6)**

Ans:(i)Syntactic error and semantic error

Syntactic errors also known as compilation errors are caused by violation of the grammar rules of the language. The compiler detects, isolate these errors and give terminate the source program after listing the errors. Some of the common syntactic errors are:

- missing or misplaced ; or }
- missing return type for a procedure
- missing or duplicate variable declaration

Semantic errors are logical errors. If there is a semantic error in a program, it will run successfully, in the sense that the computer will not generate any error messages, but it will not do the right thing. The problem is that the meaning of the program (its semantics) is wrong. Identifying semantic errors can be tricky because it requires working backward by looking at the output of the program and trying to figure out what it is doing.

(ii) Run time error and logical error

Run-time errors: Errors such as mismatch of data types or array out of bound error are known as runtime errors. These errors are generally go undetected by the compiler so programs with run-time error will run but produce erroneous results.

Logical errors: These are the errors related with the logic of the program execution. These errors are not detected by the compiler and are primarily due to a poor understanding of the problem or a lack of clarity of hierarchy of operators. Such errors cause incorrect result.

(iii) Compiler and Interpreter

These are two types of language translators.

A **compiler** converts the source program (user-written program) into an object code (machine language by checking the entire program before execution. If the program is error free, object program is created and loaded into memory for execution. A compiler produces an error list of the program in one go and all have to be taken care even before the execution of first statement begin. It takes less time for execution.

An **interpreter** is also a language translator that translates and executes statements in the program one by one. It work on one statement at a time and if error free, executes the instruction before going to second instruction. Debugging is simpler in interpreter as it is done in stages. An interpreter takes more time for execution of a program as compared to a compiler.

Q.25 The skeletal outline of a C program is shown below:

```
    main()
{ FILE *p;
   int a;
   float b;
   char c;
   p = fopen("sample.dat", "r");
   …
   fclose(p);
}
```

Read the values of a, b and c from the data file and display them on the screen

(3)

Ans:
```
#include<stdio.h>
#include<conio.h>
void main()
{
     FILE *p;
     int a;
     float b;
     char c;
     p=fopen("sample.dat","r");
      fscanf(p,"%d %f %c",&a,&b,&c);
      printf("a=%d b=%8.2f    c=%c", a,b,c);
      fclose(p);
       getch();
   }
```

Q.26 How the program design and program efficiency related to each other. **(2)**

Ans: Design is an important phase in software engineering. Two critical resources used in development process are execution time and memory. The efficiency of a program is measured in terms of these two resources. Inefficiencies in software design can result in complexity that is costly in software maintenance. Program efficiency can be improved with good program design and error free coding.

Q.27 Given the following program

```
main( )
{           static int a[5] = {10,20,30,40,50};
  void find (int *p);
  ….
  find (a);
  ….
}
void find (int *p)
{int j, sum = 0;
  for (j=3;j<5;++j) sum +=*(p+j);
  printf("sum = %d", sum);
  return;
}
```

(i) What kind of argument is passed to find?
(ii) What kind of value is returned?
(iii) What value is displayed by print statement within find? **(9)**

Ans :
```
#include<conio.h>
main()
{       static int a[5]={10,20,30,40,50};
       void find(int *p);
       clrscr();
```

```
        find(a);
        getch();
}
void find(int *p)
{
      int j, sum=0;
      for(j=3;j<5;++j)
      sum+=*(p+j);
      printf("sum=%d",sum);
}
```

(i) a pointer type variable is passed to function find .
(ii) The value returned is void.
(iii)The "printf" statement within find will display 90, which is sum of subscript 3 and 4, that is value 40 and 50 in the array a.

Q.28 What is top down design? Write down the steps to breakdown a problem into sub problems? **(8)**

Ans: A **top-down** design is essentially breaking down a system to gain insight into its compositional sub-systems. In a top-down approach an overview of the system is first formulated, specifying but not detailing any first-level subsystems. Each subsystem is then refined in greater detail, sometimes in many additional subsystem levels, until the entire specification is reduced to base elements. In short the top down approach means decomposing of the solution procedure into subtasks. This approach produces a readable and modular code that can be easily understood and maintained.

Steps for breaking a problem into sub-problems involve the following steps:
1. Start with a simple and short statement of the problem. This is top level in the design.
2. In the next level, describe the program as sequence, selection or repetition of main tasks. These tasks are called modules and these are complete in themselves. However these are described in greater details in next level. A module should have just one entry point and exit point.
3. This process is repeated step by step till all modules are refined. This step-wise refinement stops when there are sufficient details to convert that procedure into program.
4. Pseudo code or flowcharts are used to represent the procedure at intermediate steps.
5. Individual modules are tested at each level to ensure that they are working as per requirement.

Q.29 What are the qualities and capabilities of good algorithms? **(8)**

Ans: Every algorithm should have the following five capabilities and qualities:

1.Input: The algorithm should take zero or more input.
2. Output: The algorithm should produce one or more outputs.

3. Definiteness: Each and every step of algorithm should be defined unambiguously.
4. Effectiveness: A human should be able to calculate the values involved in the procedure of the algorithm using paper and pencil.
5. Termination: An algorithm must terminate after a finite number of steps.

Q.30 Design an algorithm that accepts a positive integer and reverses the order of its digits. **(8)**

Ans: An algorithm to find reverse of an integer is:

```
void reversein()
{
   int n,r;
   printf("enter an integer");
   scanf("%d",&n);
   printf("\nreverse of %d : ",n);
   while(n>0)
   {
      r=n%10;
      printf("%d",r);
      n=n/10;
   }
}
```

Q.31 Given a set of n numbers design an algorithm that adds these numbers and returns the resultant sum. **(8)**

Ans: An algorithm to add n numbers:

```
void main()
{
   int a[10],n,i,sum=0;
   printf("how many nos. u want to enter");
   scanf("%d",&n);
   printf("enter the nos.");
   for(i=0;i<n;i++)
   {
        scanf("%d",&a[i]);
        sum=sum+a[i];
   }
   printf("sum of these numbers = %d",sum);
}
```

Q.32 Determine the hierarchy of operations and evaluate the following expression:
kk = 3 / 2 * 4 + 3 / 8 + 3 **(4)**

Ans: kk=7 because 3/2*4+3/8+3 =1*4+0+3=7; 3/2 evaluates to 1, int/int=int and similarly 3/8 evaluates to 0.

Q.33 While purchasing certain items, a discount of 10% is offered if the quantity purchased is more than 1000. If the quantity and price per item are input through keyboard, write a program to calculate the total expenses. **(6)**

Ans:A C program to calculate the total expenses if the quantity and price per item are input through keyboard is:

```
#include<stdio.h>
#include<conio.h>
void main()
{
   int n,i,price,dis,p,q,sum=0;
   clrscr();
   printf("how many items u purchased");
   scanf("%d",&n);
   i=1;
   while(i!=n+1)
   {
      printf("enter the quantity of %d item",i);
      scanf("%d",&q);
      printf("enter price per item");
      scanf("%d",&p);
      if(q>1000)
      {
         dis=.1*p*q;
         price=(p*q)-dis;
      }
      else
      price=p*q;
      sum=sum+price;
      i++;
   }
   printf("total bill = %d",sum);
   getch();
}
```

Q.34 Write a program to copy input to output, replacing each string of one or more blanks by a single blank. **(6)**

Ans: A C program to copy input to output replacing each string of one or more blanks by a single blank is:

```
#include<stdio.h>
#include<conio.h>
void main()
{
 int i,j;
 char a[50],b[50];
 printf("enter the string");
```

```
    gets(a);
    i=0;
    j=0;
    while(a[i]!=' ')
    {
        b[j]=a[i];
        j++;i++;
    }
    b[j]=' ';
    j++;
    i++;
    while(a[i]!='\0')
    {
     while(a[i]!=' '&&a[i]!='\0')
     {
      b[j]=a[i];
      j++;
      i++;
     }
     while(a[i]==' ')
       i++;
    }
    b[j]='\0';
    printf("%s",b);
    getch();
}
```

Q.35 How many types of logical operators are there in 'C' programming language. **(4)**

Ans: C allows usage of three logical operators:

Operator	Symbol	Example	It evaluates to
AND	&&	exp1 && exp2	True (1) only if both exp1 and exp2 are true; false (0) otherwise
OR	||	exp1 || exp2	True (1) if either exp1 or exp2 is true; false (0) only if both are false
NOT	!	!exp1	False (0) if exp1 is true; true (1) if exp1 is false

For example:
(4 == 4) && (5!= 1) evaluates to True (1), because both operands are true.
(4 > 1) || (9 < 1) evaluates to True (1), because one operand is true (4>1).
!(5 == 4) evaluates to True (1), because the operand is false.

Q.36 Why do we use functions? Give the advance features of functions. **(4)**

Ans: We use functions due to following reasons:

1. A programmer may have a block of code that he has repeated forty times throughout the program. A function to execute that code would save a great deal of space, and it would also make the program more readable.
2. It is easy to locate and isolate a faulty function. Having only one copy of the code makes it easier to make changes.
3. Another reason for functions is to break down a complex program into logical parts. For example, take a menu program that runs complex code when a menu choice is selected. The program would probably best be served by making functions for each of the actual menu choices, and then breaking down the complex tasks into smaller, more manageable tasks, which could be in their own functions. In this way, a program can be designed that makes sense when read. And has a structure that is easier to understand quickly. The worst programs usually only have the required function, main, and fill it with pages of jumbled code.
4. A function may be used by many other programs. A programmer can use already compiled function instead of starting over from scratch.

Q.37 Write a program to test if a character from the keyboard is a lower case letter. **(8)**

Ans: A C program to test if a character from the keyboard is a lower case letter:

```
#include<stdio.h>
#include<conio.h>
#include<process.h>
void main()
{
   char a;
   clrscr();
   printf("\nenter any character :\n");
   a=getchar();
   if(islower(a)>0)
      printf("\nit is lower case letter");
   else
      printf("\nit is not a lower case letter");
   getch();
}
```

Q.38 What are the features of C preprocessor? Give the differences between macros and functions? **(6)**

Ans: A **pre-processor** is a program that processes the source code before it passes through the compiler. It operates under the control of preprocessor directive. These are placed in the source program before the main.

To define a macro, # define statement is used. This statement, also known as macro definition takes the following general form:

#define identifier string

The pre-processor replaces every occurrence of the identifier in the source code by the string. The preprocessor directive definition is not terminated by a semicolon. For example

#define COUNT 100 will replace all occurrences of COUNT with 100 in the whole program before compilation. Similarly we can define small functions with the help of macros. For example, a macro defined as

#define SQUARE(x) (x*x) will calculate square of argument when it is called. This is called macro definition.

Macros Vs Functions:

A macro's definition is expanded into the code each time the macro is encountered in the source code. If your program invokes a macro 100 times, 100 copies of the expanded macro code are in the final program. In contrast, a function's code exists only as a single copy. Therefore, in terms of program size, the better choice is a true function.

When a program calls a function, a certain amount of processing overhead is required in order to pass execution to the function code and then return execution to the calling program. There is no processing overhead in "calling" a macro. In terms of speed, a macro has the advantage.

Q.39 Write a function, which takes an array of real numbers and its size as arguments and returns the maximum. Using the above function write a program to read a set of real numbers from the keyboard and find the maximum number in the array. **(10)**

Ans: A C program to read a set of real numbers from the keyboard and find maximum among them using a max function:

```
#include<stdio.h>
#include<conio.h>
void main()
{
   int i,n;
   float a[100];
   clrscr();
   printf("\nhow many elements u want to enter :\n");
   scanf("%d",&n);
   printf("\nEnter the elements:");
   for(i=0;i<n;i++)
   scanf("%f",&a[i]);
   max(a,n);
   getch();
}

max(float a[],int n)
{
   int i;
   float k,large;
   large=a[0];
   for(i=1;i<n;i++)
   {
      if(a[i]>large)
```

```
            {
               k=a[i];
               a[i]=large;
               large=k;
            }
          }
          printf("largests element is   :   %f",large);
        }
```

Q.40 Define a pointer? Write a program to assign any number at random to an integer variable k and display the same through pointer. **(8)**

Ans: A pointer is a variable which contains the address in memory of another variable. We can have a pointer to any variable type. The unary or monadic operator **&** gives the "address of a variable". The indirection or dereference operator ***** gives the "contents of an object pointed to by a pointer".

A pointer is declared as follows:

```
int *ptr;
```

where *ptr is a pointer of int type.

A Program to display the value of an int variable through pointer is listed below"

```
#include<conio.h>
void main()
{
      int k;
      int *ptr;
      clrscr();
      k=10;
      ptr=&k;
      printf("\n Value of k is %d\n\n",k);
      printf("%d is stored at addr %u\n",k,&k);
      printf("%d is stored at addr %u\n",*ptr,ptr);
            *ptr=25;
      printf("\n Now k = %d\n",k);
      getch();
}
```

Output of the program is:

Value of k is 10

10 is stored at addr 65524

10 is stored at addr 65524

Now k=25.

Q.41 Write a program to read the coordinates of the end points of a line and to find its length. Use a structure variable named 'line' to store the relevant information about its end points. **(8)**

Ans: A Program to read end points of a line and finding its length is:

```
#include<stdio.h>
```

```
#include<conio.h>
#include<math.h>
struct lyn
{
   int a1;
   int b1;
   int a2;
   int b2;
};
void main()
{
   int a,b;
   float length;
   struct lyn line;
   clrscr();
   printf("enter the coordinates of end point A");
   scanf("%d%d",&line.a1,&line.b1);
   printf("\nenter the coordinates of end point B");
   scanf("%d%d",&line.a2,&line.b2);
   a=line.a1-line.a2;
   b=line.b1-line.b2;
   length=sqrt(pow(a,2)+pow(b,2));
   printf("length of the line is %f",length);
   getch();
}
```

Q.42 Explain library string functions. Write a program to copy the contents of the string "HELLO" to another string. **(8)**

Ans: Using C standard library functions, we can copy, concatenate, compare, and search strings. Some of the important Library string functions are described below:

(i) strcat() Function concatenates two strings together and has the following form:

strcat(string1,string2);

When this function is executed, string2 is appended to string1 by removing the null character at the end of string1.

C permits nesting of strcat functions as strcat(strcat(string1,string2),string3);

(ii) strcmp() is the string comparison function defined in string.h header file. It has the following form:.

int strcmp (const char *s1, const char *s2);

strcmp will accept two strings. It will return an integer. This integer will either be:

Negative if s1 is less than s2.

Zero if s1 and s2 are equal.

Positive if s1 is greater than s2.

Strcmp performs a case sensitive comparison; if the strings are the same except for a difference in case, then they're countered as being different. Strcmp also passes the address of the character array to the function to allow it to be accessed.

(iii) strcpy() function is just like a string-assignment operator which take the following form:
char *strcpy (char *dest, const char *src);
strcpy is short for string copy, which means it copies the entire contents of src into dest. The contents of dest after strcpy will be exactly the same as src such that strcmp (dest, src) will return 0.src may be a character array variable or a string constant.

A C program to copy the contents of string "HELLO" to another string:

```
#include<stdio.h>
#include<conio.h>
void main()
{
   char str1[20],*str2;
   int m,i,flag=0,j;
   clrscr();
   printf("enter the 1st string");
   gets(str1);
   str2="Hello";
   printf("enter the index after which u want to insert Hello in
   1st : ");
   scanf("%d",&m);
   i=0;
   while(i<=m)
   {
      i++;
   }
   j=0;
   while(str2[j]!='\0')
   {
      str1[i]=str2[j];
      i++;
      j++;
      if(str1[i]=='\0')
         flag=1;
   }
   if(flag==1)
      str1[i]='\0';
      printf("%s",str1);
   getch();
}
```

Q.43 Explain the following types of errors, which are considered while testing programs.

1. Syntax errors.
2. Run-time errors.

3. Logical errors.
4. Latent errors. **(8)**

Ans:

(i) Syntax errors: Syntax errors also known as compilation errors are caused by violation of the grammar rules of the language. The compiler detects, isolate these errors and terminate the source program after listing the errors.

(ii) Run-time errors: Errors such as mismatch of data types or array out of bound error are known as runtime errors. These errors are generally go undetected by the compiler so programs with run-time error will run but produce erroneous results.

(iii) Logical errors: These are the errors related with the logic of the program execution. These errors are not detected by the compiler and are primarily due to a poor understanding of the problem or a lack of clarity of hierarchy of operators. Such errors cause incorrect result.

(iv)Latent errors: These are hidden errors which come into the picture only when a particular data set is used. For example, consider the following statement:

```
int z=x/x-y;
```

An error will occur when x and y are equal.

Q.44 Define the functions:
(i) fwrite() (ii) fread() **(6)**

Ans:

For binary File I/O we use fwrite and fread .

(i) fwrite( ): it write from memory into a file.The declaration is given below:

```
size_t fwrite(const void *ptr, size_t size_of_elements, size_t number_of_elements, FILE
*a_file);
```

(ii)fread( ): It is used for reading into the memory. The declaration is given below:

```
size_t fread(void *ptr, size_t size_of_elements, size_t number_of_elements, FILE *a_file);
```

Both take four arguments. The first argument is the name of the array or the address of the structure you want to write to the file. The second argument is the size of each element of the array; it is in bytes. For example, if we have an array of characters, the size_of_elements is one. sizeof operator can be used get the size of the various datatypes;
The third argument tells how many elements we want to read or write; for example, if it is a 100 element array, we will pass 100.The final argument is simply the file pointer we've been using. When fread is used, after being passed an array, fread will read from the file until it has filled the array, and it will return the number of elements actually read.

For example,

```
FILE *fp;
fp=fopen("c:\\test.bin", "wb");
char x[10]="ABCDEFGHIJ";
fwrite(x, sizeof(x[0]), sizeof(x)/sizeof(x[0]), fp);
```

Q.45 Give the difference between sprintf() and sscanf() functions. **(6)**

Ans:

sprintf: One can use this function as an alternative to "itoa". "sprintf" takes three arguments- char *variable to hold the converted number, a string containing a format specifier and the number to be converted into a string. "sprinf" returns the number of characters in the string (not included the null character). The following example converts a few numbers into string format, and prints out the result:

```
#include <stdio.h>
void main()
 {
     char str[10];
     int i;
     i = sprintf(str, "%o", 15);
     printf("15 in octal is %s\n",   str);
     printf("sprintf returns: %d\n\n", i);
     i = sprintf(str, "%d", 15);
     printf("15 in decimal is %s\n", str);
     printf("sprintf returns: %d\n\n", i);
}
Output:
15 in octal is 17
sprintf returns: 2
15 in decimal is 15
sprintf returns: 2
```

sscanf converts a string into various formats. It also takes three arguments- a char * variable that contains data to be converted, a string containing a format specifier that determines how the string is converted and a memory location to place the result of the conversion. "sscanf" returns the number of items converted. The following example performs conversions in various formats:

```
#include <stdio.h>
void main()
 {
     char* ints = "20, 40, 60";
     char* floats = "10.4, 24.66";
     int i;
     int n;
     float f;
     i = sscanf(ints, "%d", &n);
     printf("n: %d\n", n);
     printf("sscanff returns: %d\n\n", i);
     i = sscanf(floats, "%f", &f);
     printf("f: %f\n", f);
     printf("sscanff returns: %d\n\n", i);}
```

Output:
n: 20
sscanff returns: 1
f: 10.400000
sscanff returns: 1

Q.46 What are static and dynamic tools? **(4)**

Ans: Static and Dynamic tools:
There are two basic types of verification tools:

- Static verification tools examine the driver code without running the driver. Because these tools do not rely on tests that exercise the code, they can be extremely thorough. Theoretically, static verification tools can examine all of the driver code, including code paths that are rarely executed in practice. However, because the driver is not actually running, they can generate false-positive and false-negative results, that is, they can misinterpret the code, report an error in a code path that cannot occur in practice, or miss an error that can occur.
- Dynamic verification tools examine the driver code while the driver is running, typically by intercepting calls to commonly used driver support routine and substituting calls to their own error-checking versions of the same routines. Because the driver is actually running while the dynamic tools are doing the verification, false-positive results are rare. However, because the dynamic tools detect only the actions that occur while they are monitoring the driver, the tools can miss certain driver defects if the driver test coverage is not adequate.

The best practice is to use a combination of static and dynamic verification tools. Static tools allow you to check code paths that are difficult to exercise in practice, while the dynamic tools find serious errors that are occurring in the driver.

Q.47 Explain divide & conquer strategy to solve any problem. **(4)**

Ans: Divide and Conquer strategy: The original problem is solved by repeatedly solving a divided sub-problem that is of smaller size and can be solved more efficiently. The problem is split into smaller sub-problems recursively until we eventually reach a stage where sub-problem is small enough to break further and can be solved easily. This strategy is called Divide and Conquer strategy. Suppose a problem P is associated with a set S and an algorithm partitions S into smaller sets such that solution of the problem P is reduced to solution of its smaller sub-sets. Then this algorithm is called divide and conquer algorithm. For example, in searching for a key in an ordered list, the size of the list is halved after every search.

Q.48 Explain the common programming errors. **(6)**

Ans: Common programming errors:

- Missing semicolon: Every C statement must end with a semicolon. A missing semicolon is confusion to the compiler and may result in misleading error messages.
- Missing braces: Very common error as it is common to forget a closing brace. Number of opening braces should match number of closing braces.
- Undeclared variables: C requires declaration of variables before their use.

- Forgetting the precedence of operators: Expression are evaluated according to precedence of operators. It is very common for beginners to forget this.
- Mismatch of parameters in function calls: There may be mismatch in actual and formal parameters in function calls.
- Missing '&' operator in scanf call.
- Crossing the bounds of an array.
- Unending and sometimes wrong loops.
- Using uninitialized pointer that points to garbage.
- Improper comment characters.

Q.49 Write down steps, which are included in implementation of algorithms. **(6)**

Ans: The steps included in the implementation of algorithms are:
- Clear understanding of the problem and algorithm designed to solve that problem.
- Decide a set of input and derive the corresponding output.
- Choose an appropriate programming language to convert algorithm to program.
- Use appropriate data structures and identify various procedures.
- Make distinct and separate modules of the whole problem
- Using the chosen language, write clean source code and well-written comments.
- Debug separate modules and check their validity in terms if already designed input set and corresponding output.
- Integrate separate modules into one unit and test it as a whole.

Q.50 Write a C program to find the sum of the following series using a function declaration
sum = x – (x^3)/3! + (x^5)/5! – ... (x^n)/n! **(8)**

Ans: A C program to find the sum of Sine series is listed below:

```
#include<conio.h>
#include<math.h>
void main(){
 float x,dig,sum; int i,n; clrscr();
 printf("\n Enter the value of x : ");
 scanf("%f",&x);
 printf("\n Enter the value of n : ");
 scanf("%d",&n);       /*convert x into radians*/
 x=x*3.1412/180;
 sum=x;
 dig=x;
 for(i=1;i<=n;i++){
     dig=(dig*pow((double)(-1),(double)(2*i-
1))*x*x)/(2*i*(2*i+1));
       sum+=dig;}
 printf("\n The sum is : %6.2f",sum);getch();}
```

Q.51 Write an algorithm to generate Fibonacci series. **(10)**

Ans: An algorithm to generate nth member of Fibonacci sequence is:

1. Start
2. Scan the number 'n' upto which the series to be generated.
3. Initialize Sum=0, x=0, y=1 and i=3.
4. Print x and y as the part of series.
5. Repeat a to e until i<=n
   a. Calculate Sum=x+y
   b. Print Sum as part of series.
   c. Assign y to x
   d. Assign sum to y
   e. i=i+1
6. Stop

Q.52 What is meant by identifiers? How do identifiers differ from keywords? **(6)**

Ans: Each C word can be classified as either a keyword or an identifier. Identifiers refer to the names of variables, functions and arrays. These are user-defined names and consist of a sequence of letters and digits, with a letter as a first character. Both uppercase and lowercase can be used to form identifiers. Maximum length of an identifier is 8 characters. Some compiler allows length upto 40 characters. Comma and blanks are not allowed. No special symbol except underscore is allowed in identifiers name.
All keywords have fixed meanings and these cannot be changed. Keywords are the basic building blocks for program statement. The keywords also known as reserved words cannot be used as variable names. There are 32 keywords available in C.

Q.53 What are the bitwise logical operators? **(4)**

Ans: Operators that are used for manipulation of data at bit level are known as bitwise operator. Bitwise logical operator are binary operator and require two integer type operand. These work on their operand bit by bit starting from the least significant bit. There are following three logical bitwise operators:

- Bitwise AND(&)
- Bitwise OR(|)
- Bitwise exclusive OR(^)

Results of all three logical bitwise operations are tabulated below:

Op1	Op2	Op1&Op2	Op1|Op2	Op1^Op2
1	1	1	1	0
1	0	0	1	1
0	1	0	1	1
0	0	0	0	0

Q.54 Explain the following statements:
(i) getchar() (ii) putchar() **(6)**

Ans:

(i) getchar( ):

The function getchar() obtains the next character from the stream stdin. It provides buffered character input with echo, and its prototype is int getchar(void);

(ii) putchar( ):

The putchar() Function located in stdio.h, is as follows:
int putchar(int c);
This function writes the character stored in c to stdout. Although the prototype specifies a type int argument, we pass putchar() a type char. The function returns the character that was just written, or EOF if an error has occurred.
A program to demonstrate getchar() and putchar() is given below along with output and explanation:

```
#include <stdio.h>
main(){
   int ch;
   while ((ch = getchar()) != `\n')
        putchar(ch);}
```

Today is good weather
Today is good weather
When getchar() function is called, it waits to receive a character from stdin. Because getchar() is a buffered input function, no characters are received until Enter is pressed. However, each pressed key is echoed immediately on the screen. When we press Enter, all the entered characters , including the newline, are sent to stdin by the operating system. The getchar() function returns the characters one at a time, assigning each in turn to ch. Each character is compared to the newline character '\n' and, if not equal, displayed on-screen with putchar().When a newline is returned by getchar(), the while loop terminates.

Q.55 What is the difference between a structure declaration and a structure initialization? **(8)**

Ans:

A structure is defined as follows:

```
     struct coord {
       int x;
       int y;
     };
```

The struct keyword, which identifies the beginning of a structure definition, must be followed immediately by the structure name, or tag (which follows the same rules as other C variable names).Within the braces following the structure name is a list of the structure's member variables. The preceding statements define a structure type named coord that contains two integer variables, x and y. They do not, however, actually create any instances of the structure coord. In other words, they don't set aside any storage for the structures. There are two ways to declare structures. One is to follow
the structure definition with a list of one or more variable names, as is done here:

```
     struct coord {
        int x;
```

```
    int y;
} first, second;
```

These statements define the structure type coord and declare two structures, first and second, of type coord. first and second are each instances of type coord; first contains two integer members named x and y, and so does second. This method of declaring structures combines the declaration with the definition. The second method is to declare structure variables at a different location in your source code from the definition. The following statements also declare two instances of type coord:

```
struct coord {
    int x;
    int y;
};
  struct coord first, second;
```

Structure members can be initialized using a dot operator as follows:

first.x=10;
first.y=20;

Q.56 Write a C program to declare a self referential structure using dynamic allocation and to display the content of the structure. **(8)**

Ans: A linked list is a self referential structure which contain a member field that point to the same structure type. In simple term, a linked list is collections of nodes that consist of two fields, one containing the information about that node, item and second contain the address of next node. Such a structure is represented as follows:

```
struct node
{      int item;
        struct node *next;  };
```

A program to demonstrate a self referential structure using dynamic allocation and display the content of the structure is:

```
//demonstrate a simple linear linked list
#include<stdio.h>
#include<stdlib.h>
#define NULL 0
struct linked_list
{
 int number;
 struct linked_list *next;
};
typedef struct linked_list node;
void main()
{
 node *head;
 void create(node *p);
 void print(node *p);
 clrscr();
 head=(node *)malloc(sizeof(node));
```

```
        create(head);
        printf("\n");
        print(head);
        getch();
      }
      void create(node *list)
      {
        printf("Input a number\n");
        printf("(type -999 to end) : ");
        scanf("%d",&list->number);
        if(list->number == -999)
              list->next=NULL;
        else{
              list->next=(node *)malloc(sizeof(node));
              create(list->next);}
        return;
      }
      void print(node *list)
      {
        if(list->next!=NULL){
              printf("%d - ->",list->number);
        if(list->next->next == NULL)
              printf("%d",list->next->number);
        print(list->next);     }
        return;
}
```

Q.57 Write a program to find the sum & average of the given numbers using the do----while loop. **(6)**

Ans: A C program to find the sum and average of given numbers using do-while loop is:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
   int i=0,n;
   float sum,avg,num;
   clrscr();
   sum=0;
   printf("How many numbers you want to find sum\n");
   scanf("%d",&n);
   printf("Enter the numbers\n");
   do{
       scanf("%f",&num);
        sum=sum+num;
        i++;}while(i<n);
   avg=sum/n;
```

```
printf("Sum=%f\n",sum);
printf("Average=%f\n",avg);
getch();}
```

Q.58 What is a function? List out the advantages & disadvantages of using functions in C? **(4)**

Ans: A **function** is a named, independent section of C code that performs a specific task and optionally returns a value to the calling program. A function is named, each have a unique name. By using that name in another part of the program, one can execute the statements contained in the function.

Advantages of using functions in C code:

1. A programmer may have a block of code that he has repeated forty times throughout the program. A function to execute that code would save a great deal of space, and it would also make the program more readable.
2. It is easy to locate and isolate a faulty function. Having only one copy of the code makes it easier to make changes.
3. Another reason for functions is to break down a complex program into logical parts. For example, take a menu program that runs complex code when a menu choice is selected. The program would probably best be served by making functions for each of the actual menu choices, and then breaking down the complex tasks into smaller, more manageable tasks, which could be in their own functions. In this way, a program can be designed that makes sense when read. And has a structure that is easier to understand quickly. The worst programs usually only have the required function, main, and fill it with pages of jumbled code.
4. A function may be used by many other programs. A programmer can use already compiled function instead of starting over from scratch.

Disadvantages of using functions in C code:

When a program calls a function, a certain amount of processing overhead is required in order to pass execution to the function code and then return execution to the calling program. If we are calling a small function again and again, this processing overhead results in inefficiency.

Q.59 When passing parameters to functions, explain the difference between pass-by-value and pass-by-reference. **(4)**

Ans: *The two ways of parameter passing mechanism are:*

(i) Pass by value or sending the values of the arguments- The value of each of the actual arguments in the calling function is copied into corresponding formal arguments of the called function. The changes made to the formal arguments have no effect on the values of actual arguments in the calling function. This technique of passing arguments is called pass by value illustrated by the following example.

(ii) Pass by reference or sending the addresses of the arguments- the addresses of actual arguments in the calling function are copied into formal arguments of the called function. Using these addresses we are actually working on actual argument so changes will be reflected in the calling function. This technique of passing arguments is called pass by reference, illustrated by following example.

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
   int i=10,j=20;
   clrscr();
   printf("The values before swap is i: %d, j:%d\n",i,j);
   swapv(i,j);
   printf("The values after swap is i: %d, j:%d\n",i,j);
   printf("\n");
   swapr(&i,&j);
   printf("The values after swap is i: %d, j:%d\n",i,j);
   printf("\n");
   getch();
}
swapv(int x,int y)
{ int temp;
   temp=x;
   x=y;
   y=temp;
  }
swapr(int *x,int *y)
{
  int temp;
   temp=*x;
   *x=*y;
   *y=temp;
  }
```

The value of i and j is 10 and 20 only after calling function swapv, that is call by value. However the result of calling swapr(), call by reference is i=20 and j=10.

Q.60 Write a program to find the sum of given non-negative integers using a function declaration. Sum = 1+2+3+4+-------n **(8)**

Ans: A Program to find the sum of given non-negative integers using a function declaration:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
              int Sum( int [],int );
void main()
{
 int i,sum,n;
 int a[100];clrscr();
 printf("How many numbers you want to enter\n");
 scanf("%d",&n);
 printf("Enter Nos.\n");
```

```
 for(i=0;i<n;i++)
    scanf("%d",&a[i]);
 sum=Sum(a,n);
 printf("\n Sum of the %d number is %d\n",n,sum);
 getch();
}
          int Sum(int [],int n)
{
   int i,sum=0;
   for(i=0;i<n;i++)
      sum=sum+a[i];
 return(sum);
}
```

Q.61 What is an array & how is an array variable different from an ordinary variable? **(4)**

Ans:An array is a collection of data storage locations, each having the same data type and the same name. Each storage location in an array is called an array element. A particular value is indicated by writing a number called index or subscript after array name. For example a[5] stands for 6^{th} element in the array a. The complete set of values is called array while the individual values are called elements. Arrays can be of any variable type.

Q.62 Define an *union* which contains two members-**colour** which is an array of 12 characters and **size** which is int. Define two union variables and assign values to them. **(4)**

Ans: Union definition

```
union  item
{
              char colour[12];
         int size;
};
```

Union enables us to treat the same space in memory as a number of different variables. We can define a union member as
union item item1;
We cannot assign different values to the different union elements at the same time. We can do Item1.size=5; or item1.colour[0]=50;

Q.63 Write a program to read a set of lines from the keyboard & remove the white spaces such as tab, space, carriage return, new line, line feed & vertical tab from the text & display onto the screen. **(8)**

Ans: A program to read a set of lines from keyboard and display onto the screen after deleting white spaces is as follows:

```
#include<stdio.h>
#include<conio.h>
void main()
```

```
{
 char a[100],b[100];
 int i=0,j=0;
 clrscr();
 printf("enter the set of lines");
 gets(a);
 while(a[i]!='\0')
 {
  while(a[i]!=' '&&a[i]!='\t'&&a[i]!='\0')
  {
     b[j]=a[i];
     j++;
     i++;
   }
   while(a[i]==' '||a[i]=='\t')
     i++;
 }
 b[j]='\0';
 printf("%s",b);
 getch();
}
```

Q.64 Distinguish between the following functions when operating on files:

- **i.** rewind and ftell
- **ii.** printf and fprintf
- **iii.** feof and ferror
- **iv.** getc and putc **(8)**

Ans:

(i) rewind and ftell

rewind( ) sets the position to the beginning of the file. It also takes a file pointer and reset the position to the start of the file. For example:

rewind(fp);

n=ftell(fp); would assign 0 to n because file pointer has been set to the start of the file by rewind.

ftell( ): gives the current position in the file from the start. ftell takes a file pointer and returns a number of type long, that corresponds to the current position. For example:

n=ftell(p); would give the relative offset n in bytes of the current position. This means that n bytes have already been read (or written).

(ii) printf anf fprintf

The printf() function is the most versatile function in C standard library that is used by a program to display data on-screen. For example, to print a text message on-screen, call the printf() function, passing the desired message enclosed in double quotation marks as follows:

printf("Nice weather!");

fprintf is just like printf with the only difference that this work only on file. The general syntax for fprintf is:

fprintf(fp,"control string",list);

For example:

fprintf(fp,"%s %d",name,age);

where fp is the file pointer of file opened in write mode. Control string contains output specifications for the items in the list. list is the names of the variables. In the example, name and age are the variables. %d and %s are the control strings.

(iii)feof and ferror

feof function is used to test for an end of file condition. It takes a file pointer as an argument and returns a non zero if all of the data from the specified file has been read and a zero otherwise. For example:

if**(feof(fp))**
printf("End of data.");

ferror function reports the status of the file indicated. It also performs in the similar manner. It takes a file pointer as an argument and returns a non zero if and error has been indicated upon that point. It returns a zero otherwise. For example:

if**(ferror(fp)!=0)**
printf("An error has occurred.");

(iv)getc and putc

getc and putc funtions are used to handle one character at a time. getc reads a character from the file that is opened in read mode. For example:

c=getc(fp2);

this statement will read a character from the file whose file pointer is fp2.

The putc function is used to write a character to the file. For example:

putc(c,fp2);

this statement will write the character stored in the character variable c to the file whose file pointer is fp2.

Q.65 Write a program to copy the contents of one string to another string using a pointer method. **(8)**

Ans: A C program to copy the contents of one string to another using a pointer method is:

```
#include<stdio.h>
#include<conio.h>
#include<malloc.h>
#define length 50
void main()
{
 char *s1,*s2,c;
 int i=0;
 clrscr();
 s1=(char*)malloc(length*sizeof(char));
 s2=(char*)malloc(length*sizeof(char));
```

```
 printf("enter string\n");
 gets(s1);
 while((c=*(s1+i))!='\0')
 {
  s2[i]=c;
  i++;
 }
 s2[i]='\0';
 printf("Copied string is\n");
 printf("%s",s2);
 getch();
}
```

Q.66 What is meant by conditional compilation? **(4)**

Ans: C pre-processor offers a feature called conditional compilation that is used to switch on or off a particular line or group of lines in a program. Compiler skips over part of a source code by inserting the pre-processing commands #ifdef and #endif, which have the general form as:

```
#ifdef macro
        Stmt1;
        Stmt2;
#endif
```

If macro has been #defined, the block of code will be processed as usual; otherwise not.

Q.67 What are the different functions involved in ctype.h? **(6)**

Ans:ctype.h: The different functions involved in this header are character testing functions and conversion functions. For example

- isalpha(c) returns an int type data and determine if the argument c is alphabetic. It returns nonzero value if true; 0 otherwise.
- toascii(c): is a conversion function defined in ctype.h that converts value of argument to ascii.

Other character testing functions in ctype.h are:
Isalnum(c), isascii(c),isdigit(c),islower(c),isspace(c),isupper(c),isxdigit(c),ispunct(c) etc.
Other conversion functions in ctype.h are tolower(c),toupper(c).

Q.68 Write a program to read a set of values from the keyboard using a pointer structure operator & to display the contents of the structure onto the screen. **(6)**

Ans: A program to read a set of values using a pointer structure operator and to display the contents of the structure onto the screen:

```
#include<stdio.h>
#include<conio.h>
struct student
{
```

```
  char name[20];
  int roll_no;
};
void main()
{
  struct student stu[3],*ptr;
  clrscr();
  printf("\n Enter data\n");
  for(ptr=stu;ptr<stu+3;ptr++)
  {printf("Name");
    scanf("%s",ptr->name);
    printf("roll_no");
    scanf("%d",&ptr->roll_no);
  }
  printf("\nStudent Data\n\n");
  ptr=stu;
  while(ptr<stu+3)
  {
     printf("%s   %5d\n",ptr->name,ptr->roll_no);
     ptr++;
    }
getch();
}
```

Q.69 How problem definition phase plays an important role in terms of the problem-solving aspect? Justify your answer. **(8)**

Ans: Problem definition plays an important role in terms of the problem-solving aspect. Before start attacking any problem, the problem should be completely understood and that can be done by defining the problem precisely. One should be sure to specify all necessary user interface, data structures, input and corresponding output. In problem definition phase, working should be done with major details, and main tasks can be divided into subtasks. In this way one can go to the minute details slowly. Algorithm is designed and it can be verified in view of predefined inputs and corresponding output. If problem is defined precisely, then one can evaluate the relationships between the data elements, find out the operations that must be performed on the logically related data elements and also one can identify the easy and efficient operation. Program verification can also be done efficiently only if the problem is defined properly.

Q.70 What is an algorithm? Write an algorithm to compute factorial of a number n where $n \geq 0$. **(8)**

Ans: Algorithm is the most fundamental concept in computer science. An algorithm is an ordered sequence of well-defined effective operations which will produce an output given an input and terminate in a finite amount of time. Algorithms can be expressed in many kinds of notation, including natural languages, pseudocode, flowcharts, and programming languages. Natural language expressions of algorithms is verbose and are rarely used for complex or

technical algorithms; while pseudocode and flowcharts are structured ways to express algorithms and also independent of a particular implementation language. Most algorithms are intended to be implemented as programs.

An algorithm for computing factorial of a number is as follows:

1) Start
2) Scan the number 'n' for which factorial is to be calculated.
3) Initialize i=1.
4) Initialize fact=1..
5) Repeat a to b until i<=n
   a. Calculate fact=fact*i.
   b. i=i+1
6) Print the value of fact.
7) Stop.

Q.71 Write a program that reads a character from the keyboard and then prints it in the reverse case i.e. if the input is in upper case, the output will be in lower case and vice versa. **(8)**

Ans:
```
#include<conio.h>
void main()
{
 char ch;
 clrscr();
 printf("\n Enter a character: ");
 scanf("%c",&ch);
 if(ch>=65 && ch<=90)
       ch+=32;
 else if(ch>=97 && ch<=122)
       ch-=32;
 else
       printf("\n The entered character is not an
alphabet!!");
 printf("\n The entered character in reverse case is :
%c",ch);
 getch();
}
```

Q.72 Design an algorithm to evaluate the function sin(x) as defined by the infinite series expansion

$$\sin(x) = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots.$$ **(8)**

Ans:
```
#include<conio.h>
#include<math.h>
void main()
{
 float x,dig,sum;
```

```
 int i,n;
 clrscr();
 printf("\n Enter the value of x : ");
 scanf("%f",&x);
 printf("\n Enter the value of n : ");
 scanf("%d",&n);
 /*convert x into radians*/
 x=x*3.1412/180;
 sum=x;
 dig=x;
 for(i=1;i<=n;i++)
 {
     dig=(dig*pow((double)(-1),(double)(2*i-
1))*x*x)/(2*i*(2*i+1));

       sum+=dig;
 }
 printf("\n The sum is : %6.2f",sum);
 getch();
}
```

Q.73 Write short notes on following:

i. Compilation errors.

ii. Linker errors. **(6)**

Ans:

(i) Compilation errors: Compilation errors are caused by violation of the grammar rules of the language. The compiler detects, isolate these errors and terminate the source program after listing the errors. These are of two types:

a. Syntax errors -- Common syntax errors include
   missing or misplaced ; or },
   missing return type for a procedure,
   missing or duplicate variable declaration.

b. Type errors -- These include
   type mismatch on assignment,
   type mismatch between actual and formal parameters.

(ii) Linker errors: Errors such as mismatch of data types or array out of bound error are known as Linker errors or run. These errors are generally go undetected by the compiler so programs with run-time error will run but produce erroneous results. Following are the types:

a. Output errors -- the program runs but produces an incorrect result. This indicates an error in the meaning of the program (logic error).

b. Exceptions -- the program terminates abnormally. Examples include
   division by zero,
   null pointer,

out of memory.

Q.74 Write a program to read an integer number from the keyboard, add 1 to it if the number read is even & again add 1 to it if the number is less than 20. Otherwise, keep the number unchanged. **(6)**

Ans:
```
#include<stdio.h>
#include<conio.h>
void main()
{
   int n;
   clrscr();
   printf("enter a number");
   scanf("%d",&n);
   if(n%2==0)
   { n=n+1;
    if(n<20)
      n=n+1;
   }
   printf("The Number is:%d",n);
   getch();
 }
```

Q.75 Give the difference between #include <stdio.h> and #include "stdio.h". **(4)**

Ans:#include "filename": The search for the file is made first in the current directory and then in the standard directories as mentioned in the include search path.
#include <filename>: This command would look for the file in the standard list of directories.
Both of these directives cause the entire contents of filename to be inserted into the source code at that point in the program.

Q.76 Explain the use of default in switch block with the help of an example? **(5)**

Ans: The general form of switch statement is as follows:
```
switch (expression)
{
            case value-1:
            block-1
            break;
    case value-2:
            block-2
            break;
      …
      …
    default:
            default block;
```

```
            break;

 }
```

When switch is executed, the value of the expression is successively compared against the value value-1, value-2,…. If a case is found whose value matches with the value of the expression, then the block of statements that follow the case are executed. The default is an optional case. When present, it will be executed if the value of the expression does not match with any of the case values. If it is not present, no action will take place if all matches fail and control will come out of the switch block.

Q.77 Write a program to read a set of real numbers & find the maximum & the minimum number entered with the help of do.. while loop. **(6)**

Ans:

```
#include<conio.h>
     float largest(float*,int);
     float smallest(float*,int);
     void main()
     {
      float a[20],k,l;
      int n,i,m;
      clrscr();
      do
      {
       printf("how many numbers u want to enter");
       scanf("%d",&n);
       printf("enter numbers");
       for(i=0;i<n;i++)
       scanf("%f",&a[i]);
       k=largest(a,n);
       l=smallest(a,n);
       printf("largest element  =  %f",k);
       printf("\nsmallest element =  %f",l);
       printf("\npres 1 to perform again  :");
       scanf("%d",&m);
      }while(m==1);
      getch();
     }
     float largest(float a[], int n)
     {
      float large,t;
      int i=1;
      large=a[0];
      do
      {
       if(a[i]>large)
       {
        t=a[i];
```

```
    a[i]= large;
    large=t;
   }
   i++;
  }while(i<n);
  return (large);
}
float smallest(float a[], int n)
{
  float small,t;
  int i=1;
  small=a[0];
  do
  {
   if(a[i]<small)
   {
    t=a[i];
    a[i]=small;
    small=t;
   }
   i++;
  }while(i<n);
  return (small);
}
```

Q.78 Give the difference between break & continue statement. **(5)**

Ans:

The **break** command will exit the most immediately surrounding loop regardless of what the conditions of the loop are. Break is useful if we want to exit a loop under special circumstances.

```
#include <stdio.h>
void  main()
{
   int a;
    printf("Pick a number from 1 to 4:\n");
   scanf("%d", &a);
    switch (a)
   {
     case 1:
       printf("You chose number 1\n");
       break;
     case 2:
       printf("You chose number 2\n");
       break;
     case 3:
       printf("You chose number 3\n");
```

```
            break;
          case 4:
            printf("You chose number 4\n");
            break;
          default:
            printf("That's not 1,2,3 or 4!\n");
        }
        getch();
      }
```

Continue is another keyword that controls the flow of loops. If we are executing a loop and hit a continue statement, the loop will stop its current iteration, update itself (in the case of for loops) and begin to execute again from the top. Essentially, the continue statement is saying "this iteration of the loop is done; let's continue with the loop without executing whatever code comes after me."

The syntax of continue statement is simple

continue;

Continue cannot be used with switch, like break.

Q.79 Write a program to generate 100 random numbers lying in the range 0.0000 to 1.0000. Evaluate their mean with the help of a function. **(8)**

Ans:

```
#include<stdlib.h>
main()
{
 float a=0,mean=0;
 int i=0;
 for (i=0;i<100;)
 {
       a=rand();
       if (a>0 && a<9999)
       {   a=a*0.0001;
            printf ("%.4f ",a);
            mean+=a;
            i++;
       }
 }
 mean/=100;
 printf ("\nMean of above mentioned numbers=%f",mean);
 getch();
}
```

Q.80 Define a function & explain why function prototype is essential. **(8)**

Ans:A **function** is a named, independent section of C code that performs a specific task and optionally returns a value to the calling program. A function is named, each have a unique

name. By using that name in another part of the program, one can execute the statements contained in the function.

Advantages of using functions in C code:

1. A programmer may have a block of code that he has repeated forty times throughout the program. A function to execute that code would save a great deal of space, and it would also make the program more readable.
2. It is easy to locate and isolate a faulty function. Having only one copy of the code makes it easier to make changes.
3. Another reason for functions is to break down a complex program into logical parts. For example, take a menu program that runs complex code when a menu choice is selected. The program would probably best be served by making functions for each of the actual menu choices, and then breaking down the complex tasks into smaller, more manageable tasks, which could be in their own functions. In this way, a program can be designed that makes sense when read. And has a structure that is easier to understand quickly. The worst programs usually only have the required function, main, and fill it with pages of jumbled code.
4. A function may be used by many other programs. A programmer can use already compiled function instead of starting over from scratch.

The prototype for a function is identical to the function header, with a semicolon added at the end. Like the function header, the function prototype includes information about the function's return type, name, and parameters. The prototype's job is to tell the compiler about the function's return type, name, and parameters. With this information, the compiler can check every time when the function is called to verify that programmer is passing the correct number and type of arguments to the function and using the return value correctly. If there's a mismatch, the compiler generates an error message.

Q.81 Write a program to sort an array of real numbers in ascending order. **(8)**

Ans:
```
#include<stdio.h>
#include<conio.h>
void main()
{
 float a[20];
 int i,j,n,c,flag;
 clrscr();
 printf("how many numbers u want to enter :\n");
 scanf("%d",&n);
 printf("\nenter the numbers :\n");
 for(i=0;i<n;i++)
 scanf("%f",&a[i]);
 for(i=0;i<n-1;i++)
 {
  for(j=0;j<n-1-i;j++)
  {
   if(a[j]>a[j+1])
```

```
      {
       c=a[j];
       a[j]=a[j+1];
       a[j+1]=c;
       flag=0;
      }
     }
     if(flag)
      break;
     else
      flag=1;
   }
   printf("sorted elements :\n");
   for(i=0;i<n;i++)
   printf("%f\t",a[i]);
   printf("\n");
   getch();
  }
```

Q.82 Define the following:

(i) Auto variables. (ii) Static variables.
(iii) External variables. (iv) Register variables. **(8)**

Ans:(i)Auto variables: The features are as follows

Declaration place:-declared inside a function in which they are to be utilized, that's why referred as local or internal variables.

Declaration syntax:- A variable declared inside a function without storage class specification by default is an automatic variable. However, we may use the keyword auto to declare it explicitly.

```
main()
{
     auto int age;
}
```

Default initial value:-Garbage value

Scope:-created when the function is called and destroyed on exit from the function.

Life:- till the control remains within the block in which defined.

(ii) Static variables: The features are as follows

Declaration place:-may be declared internally or externally.

Declaration syntax:-we use the keyword static to declare a static variable.

Static int age;

Default initial value:- Zero

Scope:-in case of internal static variable, the scope is local to the function in which defined while scope of external static variable is to all the functions defined in the program.

Life:- value of variable persists between different function calls.

(iii) External variables: The features of variables are as follows

Storage: Memory
Default initial value: Zero
Scope: global
Life: As long as program execution does not come to an end

(iv)Register variables: The features of variables are as follows
Storage: CPU registers
Default initial value: Garbage value
Scope: Local to the block in which defined
Life: till the control remains within the block in which defined

Q.83 What is a macro & how is it different from a pre processor? **(6)**

Ans:A **macro** is a preprocessor directive. A pre-processor is a program that processes the source code before it passes through the compiler. It operates under the control of preprocessor directive. These are placed in the source program before the main. The preprocessor directive can be divided into three categories:
Macro substitution directives,
File inclusion directives.
Compiler control directives.

Q.84 Differentiate between #include Directive & # define Directive. **(4)**

Ans: To define a macro, # define statement is used. This statement, also known as macro definition takes the following general form:
#define identifier string
The pre-processor replaces every occurrence of the identifier in the source code by the string. The preprocessor directive definition is not terminated by a semicolon. For example
#define COUNT 100 will replace all occurrences of COUNT with 100 in the whole program before compilation.
An external file containing functions or macro definitions can be included as part of the program by using preprocessor directive
#include "filename" where filename is the name of the file containing the required definitions or functions. By using "#include" , the preprocessor inserts the entire contents of the filename into the source code of the program.

Q.85 Write a program to read any number x & to evaluate $x^2 - 2$. Use a function to evaluate $x^2 - 2$. Call this function through its pointer. **(6)**

Ans:
```
#include<conio.h>
#include<math.h>
void main()
{
 int x,val;
 clrscr();
```

```
printf("\n Enter the value of x : ");
scanf("%d",&x);
val=(pow(x,2)-2);
printf("The value of (x^2 - 2) is : %d",val);
getch();
}
```

Using a function:

```
#include<conio.h>
void main()
{
 int x,c;
 clrscr();
 printf("enter the value of x : ");
 scanf("%d",&x);
 c=(power(x,2)-2);
 printf("x^2-2 = %d",c);
 getch();
}
power(int a,int b)
{
 int k;
 if(b==1)
 return(a);
 else
 k=a*power(a,b-1);
 return(k);
}
```

Q.86 Explain in detail most commonly used dynamic memory allocation functions. **(8)**

Ans: Most commonly used dynamic memory allocation functions:

malloc( ): It is a memory allocation function that allocates requested size of bytes and returns a pointer to the first byte of the allocated space. The malloc function returns a pointer of type void so we can assign it to any type of pointer. It takes the the following form:

ptr= (cast type *) malloc(byte-size);

where ptr is a pointer of type cast-type. For example, the statement

x=(int *) malloc(10 *sizeof(int)) means that a memory space equivalent to 10 times the size of an int byte is reserved and the address of the first byte of memory allocated is assigned to the pointer x of int type.

The malloc function can also allocate space for complex data types such as structures. For example:

ptr= (struct student*) malloc(sizeof (struct student)); where ptr is a pointer of type struct student.

calloc( ): It is another memory allocation function that allocates space for an array of elements, initializes them to zero and then returns a pointer to the memory. This function is normally used for requesting memory space at run time. It takes the following form:

ptr= (cast type *) calloc(n, element-size);

This statement allocates contiguous space for n blocks, each of size element-size bytes.

realloc( ): realloc is a memory allocation function that modifies the size of previously allocated space. Sometime it may happen that the allocated memory space is larger than what is required or it is less than what is required. In both cases, we can change the memory size already allocated with the help of the realloc function known as reallocation of memory. For example, if the original allocation is done by statement

ptr= malloc(size);

then reallocation is done by the statement

ptr=realloc(ptr,newsize); which will allocate a new memory space of size newsize to the pointer variable ptr and returns a pointer to the first byte of the new memory block.

Q.87 Write a program to create a link list of a set of integer numbers entered in the ascending order and add a new number to the linked list, such that the numbers in the new list also remain in the ascending order. **(8)**

Ans:
```
#include <stdio.h>
#include <conio.h>
#include <alloc.h>
/* structure containing a data part and link part */
struct node
{
   int data ;
   struct node *link ;
};
void add ( struct node **, int ) ;
void display ( struct node * ) ;
int count ( struct node * ) ;
void delete ( struct node **, int ) ;
void main( )
{
   struct node *p ;
   p = NULL ;  /* empty linked list */
   add ( &p, 15 ) ;
   add ( &p, 1 1) ;
   add ( &p, 16 ) ;
   add ( &p, 14 ) ;
   add ( &p, 17 ) ;
   clrscr( ) ;
   display ( p ) ;
   printf ( "\nNo. of elements in Linked List = %d", count ( p )
  ) ;
}
/* adds node to an ascending order linked list */
void add ( struct node **q, int num )
```

```
{
   struct node *r, *temp = *q ;
   r = malloc ( sizeof ( struct node ) ) ;
   r -> data = num ;
 /* if list is empty or if new node is to be inserted before the
first node */
 if ( *q == NULL || ( *q ) -> data > num )
 {
       *q = r ;
       ( *q ) -> link = temp ;
 }
 else
 {
       /* traverse the entire linked list to search the position
to insert the
            new node */
     while ( temp != NULL )
     {
               if ( temp -> data <= num && ( temp -> link ->
data > num || temp -> link == NULL ))
         {
         r -> link = temp -> link ;
         temp -> link = r ;
         return ;
          }
   temp = temp -> link ;  /* go to the next node */     } }
}
```

Q.88 Which programming design approach is followed by 'C' language (TOP-DOWN/BOTTOM-UP). Justify. **(3)**

Ans: C language follows Top down programming design approach. In C, the user thinks of a problem in terms of function modules or blocks. C is a structured language lends itself to top down approach. The essence of top down approach is to divide the whole problem into number of independent tasks called modules which are further divided into sub modules an so on. These modules form the basis of functions in the program.

Q.89 Evaluate the following expression. Show the hierarchy displaying all steps during evaluation

(i) int i = 2*3/4+4/4+8–2+5/8
(ii) int k = 3/2*4+3/8+3
(iii) float s = q*a/4–6/2+2/3*6/g (q = 4, a=2, g=2); **(6)**

Ans:

(i) output is 8 as
i=2*3/4+4/4+8-2+5/8
=1+1+6+0

=8

(ii) output is 7 as

k=3/2*4+3/8+3

=1*4+0+3

=4+0+3

=7

(iii) output is -1.000000

s=q*a/4-6/2+2/3*6/g

=4*2/4-6/2+2/3*6/2

=8/4-3+0* 3

=2-3

=-1

Since k is in float so output is -1.000000

Q.90 What is the difference between Testing & Debugging? Explain different Debugging Techniques. **(7)**

Ans:Program testing is the process of checking program, to verify that it satisfies its requirements and to detect errors. These errors can be of any type-Syntax errors, Run-time errors, Logical errors and Latent errors. Testing include necessary steps to detect all possible errors in the program. This can be done either at a module level known as unit testing or at program level known as integration testing.

Debugging is a methodical process of finding and reducing the number of bugs in a computer program making it behave as expected. One simple way to find the location of the error is to use print statement to display the values of the variables. Once the location of the error is found, the error is corrected and debugging statement may be removed.

Different debugging techniques are:

- To place print statements throughout the program to display the values of variables.
- Conditional compilation can be used to switch on or off debugging statements.
- Elimination and refinement: location of error is arrived by listing the possible causes of the error.
- Backtrack: The incorrect result is backtracked through the program logic until error is located.

Q.91 Enumerate the steps that need to be taken to design efficient algorithms. **(6)**

Ans: Steps that need to be taken to design efficient algorithms:

1. For any algorithm, the first step should be to prove that it always returns the desired output for all legal instances of the problem.

2. Second step to analyze an algorithm is to determine the amount of resources such as time and storage necessary to execute it. Usually the efficiency or complexity of an algorithm is stated as a function relating the input length to the number of steps (time complexity) or storage locations (space complexity). In theoretical analysis of algorithms it is common to estimate their complexity in asymptotic sense, i.e., to estimate the complexity function for reasonably

large length of input. Big O notation, omega notation and theta notation are used for this purpose.
There are many techniques for solving a particular problem. We must analyze these algorithms and select the one which is simple to follow, takes less execution time and produces required results.

Q.92 Replace the if-else statements by conditional operators

```
main( )
{
     int code;
     scanf("%d", &code);
     if(code>1)
        printf("\nJerusalem");
     else
        if(code<1)
           printf("\nEddie");
        else
           printf("\nC Brain");
}
```

Ans:
Replacement of if-else statements by conditional operator:
code>1?printf("\nJerusalem"):code<1?printf("\nEddie"):printf("\nBrain");

Q.93 Write a program to determine whether a given number is an Armstrong number or not. (Hint: $1^3+5^3+3^3 = 153$) **(4)**

Ans: A program to determine if a given number is an Armstrong number or not:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
 int n,r,k,sum=0;
 clrscr();
 printf("enter any number");
 scanf("%d",&n);
 k=n;
 while(n>0)
 {
  r=n%10;
  sum=sum+pow(r,3);
  n=n/10;
 }
 if(sum==k)
 printf("%d is an armstrong number",k);
```

```
  else
  printf("%d is not an armstrong number",k);
  getch();
 }
```

Q.94 (i) Write a C program to read two matrices and display their sum. **(8)**
(ii) What is meant by the scope of variables and summarise the variables types of storage class in C? **(8)**

Ans: A C program to read two matrices and display their sum is given below:

```
#include<stdio.h>
#include<conio.h>
void main()
{
 int m,n,a[10][10],b[10][10],c[10][10],i,j;
 clrscr();
 printf("enter the value for no.of rows");
 scanf("%d",&m);
 printf("enter the value for no. of columns");
 scanf("%d",&n);
 printf("enter the elements for matrix A :\n");
 for(i=0;i<m;i++)
 {
     for(j=0;j<n;j++)
     scanf("%d",&a[i][j]);
 }
 printf("enter the elements for matrix B :\n");
 for(i=0;i<m;i++)
 {
     for(j=0;j<n;j++)
     scanf("%d",&b[i][j]);
 }
 printf("MATRIX A :\n");
 for(i=0;i<m;i++)
 {
  for(j=0;j<n;j++){
      printf("%d",a[i][j]);
      printf("\t");}
  printf("\n");
 }
 printf("MATRIX B :\n");
 for(i=0;i<m;i++){
  for(j=0;j<n;j++){
      printf("%d",b[i][j]);
      printf("\t");}
  printf("\n");
 }
```

```
 printf("sum of two matrix :\n");
 for(i=0;i<m;i++)
 {
  for(j=0;j<n;j++)
  {
      c[i][j]=a[i][j]+b[i][j];
      printf("%d",c[i][j]);
      printf("\t");
  }
  printf("\n");
 }
 getch();
}
```

(ii): The scope of a variable determines the region of the program in which it is known. An identifier's "visibility" determines the portions of the program in which it can be referenced—its "scope." An identifier is visible only in portions of a program encompassed by its "scope," which may be limited to the file, function or block in which it appears.

File scope: The variables and functions with file scope appear outside any block or list of parameters and is accessible from any place in the translation unit after its declaration. Identifier names with file scope are often called "global" or "external." The scope of a global identifier begins at the point of its definition or declaration and terminates at the end of the translation unit. A function has file scope.

Function scope: A label is the only kind of identifier that has function scope. A label is declared implicitly by its use in a statement. Label names must be unique within a function however a label having the same name in two different functions is allowed.

Block scope: The variables with block scope appear inside a block or within the list of formal parameter declarations in a function definition. It is visible only from the point of its declaration or definition to the end of the block containing its declaration or definition. Its scope is limited to that block and to any blocks nested in that block and ends at the curly brace that closes the associated block. Such identifiers are sometimes called "local variables."

There are four storage classes in C:

a. Automatic storage class: The features of variables are as follows
   - -Storage: Memory
   - -Default initial value: Garbage value
   - -Scope: Local to the block in which defined
   - -Life: till the control remains within the block in which defined.

b. Register storage class: The features of variables are as follows
   - -Storage: CPU registers
   - -Default initial value: Garbage value
   - -Scope: Local to the block in which defined
   - -Life: till the control remains within the block in which defined

c. Static storage class: The features of variables are as follows
   - -Storage: Memory
   - -Default initial value: Zero
   - -Scope: Local to the block in which defined

-Life: value of variable persists between different function calls.

d. External storage class: The features of variables are as follows

-Storage: Memory
-Default initial value: Zero
-Scope: global
-Life: As long as program execution does not come to an end

Q.95 Write 'switch' statement that will examine the value of an integer variable flag & print one of the following messages:

(i) HOT, if flag=1
(ii) LUKE WARM, if flag=2
(iii) COLD, if flag=3
(iv) OUT OF RANGE, if any other value **(5)**

Ans: A program to demonstrate switch statement to display given messages:

```
void main(){
    int flag;
    printf( "Enter any value\n" );
    scanf( "%d", &flag );
    switch ( flag ) {
        case 1:            /* Note the colon, not a
semicolon */
     printf( "HOT\n" );
     break;
 case 2:
     printf( "LUKE WARM\n" );
            break;
        case 3:
     printf( "COLD\n" );
            break;
        default:
     printf( "OUT OF RANGE\n" );
     break;    }
    getchar();}
```

Q.96 What does 'return' statement do in a function? Can a function have more than one return statement? Explain. Write the user-defined code for finding factorial( ) of a given number using Recursion. **(6)**

Ans: 'return' statement in a function is used to return a value to the calling program. The return statement takes one of the following forms:

return; It does not return any value and act as a closing brace of the function.

or

return (expression); It returns the value of the expression to the called function.

A function can have more than one return statement. This happens when the value returned is based on certain conditions. For example:

```
if(x<=y)
  return x;
else
  return y;
```

All functions by default return int type data.

A program for finding factorial of a given number using recursion is as follows:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n,c;
clrscr();
printf("enter the number   :");
scanf("%d",&n);
c=fact(n);
printf("factorial of %d = %d",n,c);
getch();
}
fact(int n)
{
int factorial;
if(n==1||n==0)
return(1);
else
factorial=n*fact(n-1);
return (factorial);
}
```

Q.97 Define an array. Write a program for 2-D Matrix Multiplication using arrays. **(8)**

Ans: An **array** is a collection of data storage locations, each having the same data type and the same name. Each storage location in an array is called an array element. A particular value is indicated by writing a number called index or subscript after array name. For example a[5] stands for 6^{th} element in the array a. The complete set of values is called array while the individual values are called elements. Arrays can be of any variable type.

A C program for 2-D matrix multiplication:

```
#include<stdio.h>
#include<conio.h>
void main()
{
 int a[10][10],b[10][10],c[10][10],i,j,k,m,n,p,q;
 clrscr();
 printf("enter no. of rows and column for 1st matrix\n");
 printf("no.of rows\n");
 scanf("%d",&m);
 printf("no. of columns");
 scanf("%d",&n);
```

```
printf("\nenter no. of rowsand columns for 2nd matrix");
printf("\nno. of rows");
scanf("%d",&p);
printf("\nno. of columns");
scanf("%d",&q);
if(n==p)
{
 printf("enter elements for matrix A");
 for(i=0;i<m;i++)
 {
  for(j=0;j<n;j++)
  scanf("%d",&a[i][j]);
 }
 printf("enter elements for matrix B");
 for(i=0;i<p;i++)
 {
  for(j=0;j<q;j++)
  scanf("%d",&b[i][j]);
 }
 printf("\nMATRIX A:\n");
 for(i=0;i<m;i++)
 {
  for(j=0;j<n;j++)
  {
      printf("%d",a[i][j]);
      printf("\t");
  }
  printf("\n");
 }
 printf("\nMATRIX B:\n");
 for(i=0;i<p;i++)
 {
  for(j=0;j<q;j++)
  {
   printf("%d",b[i][j]);
   printf("\t");
  }
  printf("\n");
 }
 printf("\n");
 printf("MULTIPLICATION :\n");
 for(i=0;i<m;i++)
 {
  for(j=0;j<q;j++)
  {
   for(k=0;k<p;k++)
   {
```

```
            c[i][j]=c[i][j]+(a[i][k]*b[k][j]);
        }
        printf("%d",c[i][j]);
        c[i][j]=0;
        printf("\t");
         }
     printf("\n");
     }
   }
   else
   printf("multiplication is not possible");
   getch();
 }
```

Q.98 Write an algorithm to find greatest common divisor of two positive non-zero integers. **(5)**

Ans: An algorithm to find greatest common divisor of two positive integers is given below:

```
 void gcd()
 {
  int a,b,r,h,k,c;
  clrscr();
  printf("enter two numbers");
  scanf("%d%d",&a,&b);
  h=a;
  k=b;
  while(r!=0)
  {
   r=a%b;
   a=b;
   b=r;
  }
  printf("G.C.D. of %d and %d = %d",h,k,a);}
```

Q.99 Point out and rectify the error(s), if any in the following code:

```
main( )
{
    int code, flag;
    if(code = = 1 & flag = = 0)
       printf(“/n The Eagle has landed);
}
```

(3)

Ans: In the problem bitwise AND (&) operator is used. In fact it should be logical AND '&&'.

Q.100 What would be the output of following code? **(4)**

```
main( )
{
    float a=13.5;
```

```
        float *b, *c;
        b =&a; // suppose the address of 'a' is 1006.
        c = b;
        printf("\n%u%u%u", &a, b, c");
        printf("\n%f %f%f%f%f", a, *(&a), *&a, *b, *c);
    }
```

Ans: The output of the given code assuming address of a is 1006 is:

1006 1006 1006

13.500000 13.500000 13.500000 13.500000 13.500000

The first printf statement giving the address of a, that is assigned to b and c also.
The second printf statement is giving the value at this address that is 13.5 printed as 13.500000 (taking 6 places after decimal)

Q.101 Write at least any 2 differences between malloc( ) & calloc( ) function. **(4)**

Ans: malloc( )and calloc( ):

malloc( ): It is a memory allocation function that allocates requested size of bytes and returns a pointer to the first byte of the allocated space. The malloc function returns a pointer of type void so we can assign it to any type of pointer. It takes the the following form:

ptr= (cast type *) malloc(byte-size);

where ptr is a pointer of type cast-type. For example, the statement

x=(int *) malloc(10 *sizeof(int)) means that a memory space equivalent to 10 times the size of an int byte is reserved and the address of the first byte of memory allocated is assigned to the pointer x of int type.

The malloc function can also allocate space for complex data types such as structures. For example:

ptr= (struct student*) malloc(sizeof (struct student)); where ptr is a pointer of type struct student.

calloc( ): It is another memory allocation function that allocates space for an array of elements, initializes them to zero and then returns a pointer to the memory. This function is normally used for requesting memory space at run time. It takes the following form:

ptr= (cast type *) calloc(n, element-size);

This statement allocates contiguous space for n blocks, each of size element-size bytes.

Q.102 What are the advantages of using pointers. **(4)**

Ans: The **advantages** of using a pointer are as follows:

- Through pointers we can access a variable that is declared outside a function.
- Data tables can be handled in an efficient manner by pointers.
- The length and complexity of a program is reduced.
- The execution speed in reduced.
- By using pointer to an array of character strings helps in saving of data storage space in memory.

Q.103 Differentiate between call by reference and call by value. Use suitable examples to explain. **(4)**

Ans: Call by value and Call by reference

Call by value means sending the values of the arguments- The value of each of the actual arguments in the calling function is copied into corresponding formal arguments of the called function. The changes made to the formal arguments have no effect on the values of actual arguments in the calling function. This technique of passing arguments is called call by value illustrated by swapv(int x, int y) function in the following example.

Call by reference means sending the addresses of the arguments- the addresses of actual arguments in the calling function are copied into formal arguments of the called function. Using these addresses we are actually working on actual argument so changes will be reflected in the calling function. This technique of passing arguments is called call by reference, illustrated by swapr(int *x,int *y) in following example.

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
 int i=10,j=20;
 clrscr();
 printf("The values before swap is i: %d, j:%d\n",i,j);
 swapv(i,j);
 printf("The values after swap is i: %d, j:%d\n",i,j);
 printf("\n");
 swapr(&i,&j);
 printf("The values after swap is i: %d, j:%d\n",i,j);
 printf("\n");
 getch();
}
swapv(int x,int y)
{ int temp;
   temp=x;
   x=y;
   y=temp;
  }
swapr(int *x,int *y)
{
  int temp;
   temp=*x;
   *x=*y;
   *y=temp;
  }
```

The value of i and j is 10 and 20 only after calling function swapv, that is call by value. However the result of calling swapr(), call by reference is i=20 and j=10

Q.104 Write macro definition for the following:

(i) Minimum of 2 values; MIN (a, b)

(ii) To check whether entered character is a digit (or) not; ISDIGIT(Y) **(6)**

Ans:

(i) A macro for finding minimum of two numbers:

```
#include<stdio.h>
#include<conio.h>
#define min(a,b) ((a>b)?b:a)
void main()
{
  .
  .
  .

  z=min(x,y);          //Statement in program can be used as
  ..}
```

(ii) A macro to check whether entered character is a digit or not:

```
#include<stdio.h>
#include<conio.h>
#define isdigit(a) ((48<=a&&a<=57)?1:0)
void main()
{
  .
  .
  .
  y=isdigit(x);          //Statement in program can be used
  as

  .
  .
  .
  .
}
```

Q.105 Write different built-in (library) functions provided by 'C' language for handling I/O operations on files. **(10)**

Ans: There are several functions used for I/O operations in files:

- **getc and putc functions** :

getc and putc funtions are used to handle one character at a time. getc reads a character from the file that is opened in read mode. For example:c=getc(fp2);

this statement will read a character from the file whose file pointer is fp2.

The putc function is used to write a character to the file. For example:putc(c,fp2);

this statement will write the character stored in the character variable c to the file whose file pointer is fp2.

- **getw and putw** :

These are integer oriented functions. Both work just like getc and putc with the difference it is used with integers only.

The general syntax is:

```
getw(fp2);
putw(integer,fp2);
```

In the above staments getw will read an integer from file whose file pointer is fp2. similarly putw will write the integer in file having file pointer fp2.

- **fprintf and fscanf** :

These two functions are just like printf and scanf with the only difference that these work only on files. The general syntax and examples are are the same for both fprintf and fscanf .

syntax:fprintf(fp,"control string",list);

example:fprintf(fp,"%s %d",name,age);

In the above syntax, fp is the file pointer of file opened in write mode. Control string contains output specifications for the items in the list. list is the names of the variables. In the above example, name and age are the variables. %d and %s are the control strings.

Q.106 What would be output of following code?

Justify your answer

(i)
```
f1( ){
    Static int count= 5;
    printf("\n count=%d", count- -);
    if(count != 0)
    f1( );}
```

(ii)
```
int i=0;
main( )
{
    printf("\main's i=%d", i);
    i++;
    val( );
}
val( ){
    i=100;
    printf("\n val's i=%d", i);
    i++;}
```

Ans:

(i) The output is:

count=5
count=4
count=3
count=2
count=1

Value of count is initially 5, it is printed and then count is decremented by 1.

Since count!=0 so function f1 is called again, 4 is printed and same process is repeated till count =0 so the above output.

(ii) The output is:

main's i=0
val's i=100

main function is calling global variable i with value=0 so first printf statement is printing 0 while in the function val ( ), the local variable takes precedence over global and so value 100 is printed.

Q.107 Write a C program to find if a number is present in a list of N numbers or not. **(10)**

Ans: A C program to find if a number is present in a list of N numbers or not:

```
#include<stdio.h>
#include<conio.h>
void main(){
        int i,n,m,flag=0; int a[10];        clrscr();
        printf("how many elements u want to enter");
        scanf("%d",&n);
        printf("enter element in the array");
        for(i=0;i<n;i++)
        scanf("%d",&a[i]);
        printf("enter the element u want to search");
        scanf("%d",&m);
       for(i=0;i<n;i++)   {
       if(a[i]==m)     {
          flag=1;
         break;  }}
     if(flag==0)
           printf("not present");
    else
           printf("present");
  getch();     }
```

Q.108 Write a C function using pointers to exchange the values stored in two memory locations in the memory. **(6)**

Ans: A C program to exchange the values stored in two memory locations:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
int i,j;
void main(){
  i=10,j=20;clrscr();
 printf("The values before exchange is i: %d, j:%d\n",i,j);
 exchange(&i,&j);
 printf("The values after exchange is i: %d, j:%d\n",i,j);
 printf("\n");
 getch();}
exchange(int *x,int *y){ int temp;
   temp=*x;
```

```
*x=*y;
*y=temp;  }
```

Q.109 What are the different types of errors that can occur during I/O operations on a file? **(6)**

Ans: Typical error situations that may occur during I/O operations on a file:

- Trying to read beyond the end–of-file mark.
- Device overflow.
- Trying to use file that has not been opened.
- Opening a file with an invalid filename.
- Attempting to write to a write-protected file.
- Trying to perform an operation on a file when the file is opened for another type of operation.

Q.110 Write user-defined function for copying a string to another **(6)**

Ans: A program to copy a string to another:

```
#include<conio.h>
void main(){
 char str1[20],str2[20];
 int m,i,flag=0,j;
 clrscr();
 printf("enter the 1st string");
 gets(str1);
 printf("enter the 2nd string");
 gets(str2);
 printf("enter the index after which u want to insert 2nd
 string in 1st : ");
 scanf("%d",&m);
 i=0;
 while(i<=m){
     i++;}
 j=0;
 while(str2[j]!='\0'){
  str1[i]=str2[j];
  i++;
  j++;
  if(str1[i]=='\0')
  flag=1;
 }
 if(flag==1)
     str1[i]='\0';
     printf("%s",str1); getch();}
```

Q.111 Explain the salient features of typedef. **(6)**

Ans: Typedef statement allows user to define an identifier that would represent an exiting data type. The user-defined data type identifier can be used further to declare variables. It has the following syntax typedef datatype identifier; where datatype refers to exiting data type and identifier is the new name given to this datatype. For example typedef int nos; nos here symbolizes int type and now it can be used later to declare variables like nos num1,num2,num3;

Q.112 Write a program in C to find the sum and average of the given numbers stored in an array of n values. **(8)**

Ans: A C program to find the sum and average of the given numbers stored in an array of n values:

```
       #include<stdio.h>
#include<conio.h>
         void main()     {
     int i,a[100],n,sum; float average; clrscr();
     printf("enter the value of n less than 100");
     scanf("%d",&n);
     printf("enter  values: \n");
     for(i=0;i<n;i++)
        scanf("%d",&a[i]);
     sum=0;
     for(i=0;i<n;i++)   {
             sum=sum+a[i];     }
     printf("\n%d",sum);
     average=(float)sum/n;
     printf("\n%f",average);
     getch();     }
```

Q.113 Explain goto statement

Ans: C support **"goto"** statement to branch unconditionally from one point to another in the program.The goto keyword is followed by a label, which is basically some identifier placed elsewhere in the program where the control is to be transferred.During running of a program, the statement like "goto label1;" cause the flow of control to the statement immediately following the label "label1". We can have a forward jump or a backward jump.

```
     #include <stdio.h>
     void main() {
      int attempt, number = 46;
      looping: /* a label */
       printf("Guess a number from 0-100\n");
       scanf("%d", &attempt);
       if(number==attempt) {
        printf("You guessed correctly!\n\n");  }
       else {
        printf("Let me ask again...\n\n");
```

```
goto looping; /* Jump to the label*/  }}
```

Q.114 Name and explain the primitive or fundamental data types that Compilers support. **(4)**

Ans: Fundamental data types: Typical primitive types may include-

- Character (`char`)-usually stored in 8 bits(one byte) of internal storage; unsigned char have values between 0 to 255 while signed chars range from -128 to 127.
- Integer (`int`, `short`, `long`, `byte`) with a variety of precisions; C has three classes of internal storage, namely short int, int, and long int, in both signed and unsigned forms.
- Floating-point number (`float`, `double`, `real`, `double precision`); These are stored in 32 bits, with 6 digits of precision.
- Boolean having the values true and false.
- Reference (also called a pointer or handle), a small value referring to another object's address in memory.

Q.115 Write an algorithm to find the sum of the squares of the first n positive integers.

$1^2 + 2^2 + \ldots\ldots\ldots\ldots\ldots\ldots + n^2$ **(8)**

Ans: An algorithm to find sum of the squares of the first n positive integers is given below:

```
void SumSq()
{
 int i,n,sum=0;
 printf("enter the length of the series");
 scanf("%d",&n);
 for(i=1;i<=n;i++)
     sum=sum+i*i;
 printf("sum of squares upto %d = %d",n,sum);
}
```

Q.116 Write an algorithm to find the value of x^n where n is a positive integer greater than 1. **(8)**

Ans: An algorithm power( ) given below finds the value of x^n

```
void power()
{
 int a,b,i,result=1;
 printf("enter value of x :");
 scanf("%d",&x);
 printf("enter value of n :");
 scanf("%d",&n);
 printf("x raise to power n = ");
 for(i=1;i<=n;i++)
 result=result*x;
 printf("%d",result);
 }
```

Q.117 Write an algorithm to find the highest common factor of two positive numbers. **(8)**

Ans: An algorithm to find highest common factor of two positive numbers:

```
void main(){
    int a,b,r,h,k,c;
    printf("enter two numbers");
    scanf("%d%d",&a,&b);
       h=a;
       k=b;
     while(r!=0)    {
        r=a%b;
        a=b;
        b=r;     }
       printf("H.C.F. of %d and %d = %d",h,k,a);
}
```

Q.118 Name two types of program testing and explain them in detail. **(8)**

Ans: Program testing:**White box testing** strategy deals with the internal logic and structure of the code. White box testing also known as glass, structural, open box or clear box testing, tests code written, branches, paths, statements and internal logic of the code etc.
In order to implement white box testing, the tester has to deal with the code and hence is needed to possess knowledge of coding and logic i.e. internal working of the code. White box test also needs the tester to look into the code and find out which unit/statement/chunk of the code is malfunctioning. White box testing is applicable at unit, integration and system level however it is mainly done at unit level.
Black box testing Black-box test design treats the system as a "black-box", so it doesn't explicitly use knowledge of the internal structure. It takes an external perspective of the test object to derive test cases. These tests can be functional or non-functional, though usually functional. The test designer selects valid and invalid input and determines the correct output. There is no knowledge of the test object's internal structure.
This method of test design is applicable to all levels of software testing: unit, integration, functional testing, system and acceptance.

Q.119 Explain the following operators with the help of examples
(i) Conditional operator. (ii) sizeof operator.
(iii) Increment operator. (iv) Cast operator. **(4 × 3)**

Ans:
(i) Conditional operator: It is a ternary operator of the form
exp1? exp2: exp3 where exp1,exp2 and exp3 are expressions. It means that (condition)?(evaluate if condition was true):(evaluate if condition was false). For example:
my_variable = (x > 10)?"a":"b";
To accomplish the same using a standard if/else statement, would take more than one line of code as given below:
if (x > 10) {

```
 my_variable = 'a';
}
else {
my_variable = 'b';}
```

(ii)sizeof operator: It is a compile time operator and when used with an operand, it returns the number of bytes the operand occupies.

```
int m=sizeof(operand1);
```

so integer m will have number of bytes operand1 occupies.

(iii)Increment Operator: This is a unary operator '++' adds 1 to the operand with which it is used. That is ++num is same as num=num+1.

Increment operator is of two types-postfix and prefix operator which mean the same thing when form independent statements. However when used in expression on the right-hand side of an assignment, these behave differently. For example:

```
(i)   int i,j;
        i=2;
        j=++i;
(ii)  int i,j;
        i=2;
        j=i++;
```

In case (i), the value of j and i would be 3 because prefix operator first add 1 to the operand then assign it to the variable on left. While in case (ii), the value of j is 2 and i would be 3 because postfix operator first assign the value to the variable on the left and then increment the value.

(iv)Cast operator: The general form of a cast operator in C is:

(type name) expression, used for casting an expression to type "type name". Type name is one of the standard C data types and expression may be a constant, variable or an expression. For example, the statement

X=(int) 5.6 will convert 5.6 to an integer by truncation.

Y=(int)(a+b), the result of a+b is converted to integer.

Q.120 Write a program to convert the given temperature in Fahrenheit to temperature in Celsius where cel = ((fah – 32) * 5/9); and Fahrenheit is denoted by fah. **(4)**

Ans: A program to convert the given temperature in Fahrenheit to Celcius:

```
#include<stdio.h>
#include<conio.h>
void main()
{
 float fah,cel;
 clrscr();
 printf("enter temp in fahrenheit");
 scanf("%f",&fah);
 cel=((fah-32)*5/9);
```

```
 printf("\ntemp in celsius = %f",cel);
 getch();
}
```

Q.121 Write a program to display the real, imaginary and equal roots of a quadratic equation $ax^2 + bx + c = 0$. **(8)**

Ans: A program to display real, imaginary and equal roots of a quadratic equation:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
 int a,b,c,disc;
 float root1,root2,x1,y1;
 clrscr();
 printf("enter the value of a,b,c");
 printf("\nA  :\t");
 scanf("%d",&a);
 printf("\nB  :\t");
 scanf("%d",&b);
 printf("\nC  :\t");
 scanf("%d",&c);
 disc=((b*b)-(4*a*c));
 if(disc==0)
 {
  printf("/nroots are  real and equal");
  root1=(-b)/(2*a);
  printf("\nroots of the equation are :");
  printf("\nROOT1 : %f",root1);
  printf("\nROOT2 : %f",root1);
 }
 if(disc>0)
 {
  printf("roots are real and distinct");
  root1=((-b)+sqrt(disc))/(2*a);
  root2=((-b)-sqrt(disc))/(2*a);
  printf("\n roots of the equation are :");
  printf("\nROOT1 : %f",root1);
  printf("\nROOT2 : %f",root2);
   }
if(disc<0)
{
 disc=disc*(-1);
 printf("roots are imaginary");
 x1=(-b)/(2*a);
```

```
        y1=sqrt(disc)/(2*a);
        printf("ROOT1 : ");
        printf("%f",x1);
        printf("+i");
        printf("%f",y1);
        printf("\nROOT2 : ");
        printf("%f",x1);
        printf("-i");
        printf("%f",y1);
      }
      getch();
      }
```

Q.122 What do you mean by a loop? Explain the difference between the do loop, while loop, and for loop with the help of an example. **(8)**

Ans: Loop is a control structure used to perform repetitive operation. Some programs involve repeating a set of instruction either a specified number of times or until a particular condition is met. This is done using a loop control structure. A program loop consists of two parts: Body of the loop and control statement. The control statement tests certain conditions and then decides repeated execution or termination of statements. Most real programs contain some construct that loops within the program, performing repetitive actions on a stream of data or a region of memory. There are several ways to loop in C described below:

1.While statement: The basic format of while statement is

```
    while (conditional expression)
    {
 ...block of statements to execute...
    }
```

The `while` loop continues to loop until the conditional `expression` becomes false. Once this expression become false, the control is transferred out of the loop. On exit, the program continues with the statement immediately after the body of the loop. For example:

```
i=0;
        while(i<10)
         {
               printf("Hello world\n");
      i++;  }
```

This statement will print "Hello world" 10 times in a new line and come out of the loop when 'i' become 10.

2. Do statement: This loop construct is of the form:

```
      do
        {    ...block of statements to execute...

      }while(conditional expression);
```

While construct checks the conditional expression before the loop is executed. Sometimes it is necessary to execute the body of the loop before the conditional expression is evaluated. Such

situations are handled by do-while loop construct. On reaching the do statement, the body of the loop is evaluated and at the end of the loop, the conditional expression is checked for true or false. If true,it continues to evaluate the body again and when condition become false, the control is transferred to the statement immediately after the while statement.For example:

```
   do
     {
         printf( "Input a character\n");
ch=getch( );
  }while(ch='n');
```

This segment of program reads a character from the keyboard until 'n' is keyed in.

3. **For statement:** This is another entry-controlled loop having a general form:

```
    for (expression_1; expression_2; expression_3)
    {
  ...block of statements to execute...
    }
```

The expression_1 is for initialization of the control variable. The condition is tested upon using the expression_2. If the condition is true, the body of the loop is executed. Then control is transferred back to the for loop expression_3 where control variable is incremented using an assignment statement and new value of variable is checked through expression_2 and process is repeated till the expression_2 is evaluated to be false. On the termination of loop, the execution continues with the statement immediately following the loop.

The `for` loop is a special case, and is equivalent to the following while loop:

```
expression_1;
   while (expression_2)
     {
  ...block of statements...
  expression_3;
     }
```

For instance, the statement in while lop takes the following form in the 'for' loop:

```
for (i = 1;  i <10;  i = i+1) {
  printf("Hello World\n");
}
```

Q.123 How are values initialized in one – dimensional array? Should the entire array be initialised in the definition? **(4)**

Ans: The values are initialized in one-dimensional array just like an ordinary variable. The general form of initialization of arrays is:

type array-name[size]={list of values};

The values in the list are separated by commas. For example:

int marks[6]={60,70,80,40,79,89};

It is not necessary to initialize all the values in the array. However If they are not given any values, they would be garbage values in it. For example:

int marks[6]={30, 70 ,80,40}; here marks is array of 6 integer where only 4 values are provided; for marks in rest two subject, it will take the garbage value.
If the array is initialized where it is declared, mentioning the dimension of the array is optional. For example:

int marks[]={56,89,67,34,89,90};

Q.124 Write a program to find the average of 10 real numbers in an array. **(4)**

Ans: A program to find average of 10 real numbers:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
 int i=0;
 float sum,avg,num;
 clrscr();
 sum=0;
 while(i<10)
 {
   scanf("%f",&num);
   sum=sum+num;
   i++;
 }
 avg=sum/10;
 printf("Sum=%f\n",sum);
 printf("Average=%f\n",avg);
 getch();
}
```

Q.125 Define a structure for a student having name, roll number and marks obtained in six subjects. Write a program to input the details for 20 students and print the details of the students who have scored more than 70% marks overall. **(12)**

Ans: A C program to input details for 20 students and print the details of students who have scored more than 70% marks overall is given below:

```
#include<stdio.h>
#include<conio.h>
struct student
{
 char name[10];
 int roll_no;
 int marks[6];
 int total;
 int per;
};
void main()
```

```
{
 struct student stu[20];
 int i,j,req;
 clrscr();
 for(i=0;i<20;i++)
 {
 stu[i].total=0;
 printf("enter data for %d students :",i+1);
 printf("\nenter name");
 scanf("%s",stu[i].name);
 printf("\nenter roll no   ");
 scanf("%d",&stu[i].roll_no);
 printf("\nenter marks in subjects\t");
  for(j=0;j<6;j++)
  {
  printf("\nenter marks in %d subject\t",j+1);
  scanf("%d",&stu[i].marks[j]);
  stu[i].total=stu[i].total+stu[i].marks[j];
  }
  stu[i].per=stu[i].total/6;
  printf("\n");
 }
 for(i=0;i<20;i++)
 {
  if(stu[i].per>70)
  {
   printf("\nSTUDENT    %d",i+1);
   printf("\nname  :");
   printf("%s",stu[i].name);
   printf("\nroll no");
   printf("%d",stu[i].roll_no);
   for(j=0;j<6;j++)
   {
    printf("\nmarks in %d subject\t",j+1);
    printf("%d",stu[i].marks[j]);
   }
   printf("\nTOTAL    :%d",stu[i].total);
  }
 }
 getch();
}
```

Q.126 Print the following sequence of integers 1, 3, 9, 27, 81, 243 using (i) For statement (ii) While statement **(8)**

Ans: A C Program to print the sequence using (i) for statement (ii) while statement

(i)
```
#include<stdio.h>
```

```
#include<conio.h>
#include<math.h>
void main(){
     int i,n,l,k;
     clrscr();
     for(i=0;i<=5;i++){
         k=pow(3,i);
         printf("%d\t",k);}
     getch();}
(ii)#include<stdio.h>
#include<conio.h>
#include<math.h>
void main(){
     int i,n,l,k;
     clrscr();
     i=0;
     while(i<=5){
        k=pow(3,i);
        printf("%d\t",k);
        i++;}
getch();}
```

Q.127 If p is a pointer to an address and j is a numeric value, than what does the expression like p + j mean. **(2)**

Ans: If p is a pointer to an address and j is a numerical value, then p+j points to an address j locations after the current location.

Q.128 List the four stages involved in program design. **(2)**

Ans: Four stages involved in program design are:

- Understanding the problem and algorithm designed for the problem.
- Decide what is to be given as input and corresponding output.
- Choose an appropriate programming language and data structures.
- Derive test cases and test exhaustively each and every module of the program.

Q.129 How are the data elements initialized in a multidimensional array? What is the scope of rules for the multidimensional array. **(8)**

Ans: Multidimensional array: Multidimensional arrays can be described as "arrays of arrays". For example, a bidimensional array can be imagined as a bidimensional table made of elements, all of them of a same uniform data type.

int arr[3][5]; represents a bidimensional array of 3 per 5 elements of type `int`.

Similarly a three dimensional array like

int arr[3][4][2]; represent an outer array of three elements , each of which is a two dimensional array of four rows, each of which is a one dimensional array of five elements.

Multidimensional arrays are not limited to two ot three indices (i.e., two dimensional or three dimensional). They can contain as many indices as needed. However the amount of memory needed for an array rapidly increases with each dimension. For example:
char arr [100][365][24][60][60];declaration would consume more than 3 gigabytes of memory.
Memory does not contain rows and columns, so whether it is a one dimensional array or two dimensional arrays, the array elements are stored linearly in one continuous chain. For example, the multidimensional array

```
int arr[3][4][2]= {
                    {  {1,2},{3,4},{5,6},{7,8} },
                  {  {9,1},{1,2},{3,7},{4,7} },
                   {  {6,1},{18,19},{20,21},{22,23} },
                   };  is stored in memory just like an one-dimensional array shown below:
```

1	2	3	4	5	6	7	8	9	1	1	2	3	7	4	7	6	1	18	19	20	21	22	23
101							**117**								**133**								

Multidimensional arrays are just an abstraction for programmers, since we can obtain the same results with a simple array just by putting a factor between its indices.

Q.130 List any four different bitwise operators available in 'C' language. **(6)**

Ans: Bitwise operator: These permit the programmer to access and manipulate individual bits within a piece of data. These operators can operate upon ints and chars but not on floats and doubles. Four different bitwise operators available in C language are as follows:

(i) **One's complement:** denoted by symbol ~ operates on a single variable. On applying this operator on a number,all 1's present in the binary equivalent of that number is changed to 0 and all 0's are changed to 1's.For example:

x=1001 0110 1100 1011
~x=0110 1001 0011 0100

(ii) **Right Shift Operator:** represented by symbol >>. It shifts each bits in the operand to the right. The number of places the bits are shifted to right depends on the number following the operand. For example x>>3 would shift all bits three places to the right. For example suppose x=0100 1001 1100 1011

x>>3=0000 1001 0011 1001

(iii) **Left Shift Operator:** represented by symbol <<. It shifts each bits in the operand to the left. The number of places the bits are shifted to left depends on the number following the operand. For example x<<3 would shift all bits three places to the left.

x<<3 for above x would be 0100 1110 0101 1000

(iv) **Exclusive OR:** It is a binary operator represented by ^. The result of exclusive OR is 1 if only one of the bits is 1; otherwise it is 0. For example:

X=0000 0000 0000 1101
Y=0000 0000 0001 1001

X^Y=0000 0000 0001 0100

Q.131 Explain the method of program verification briefly. **(8)**

Ans: Method of program verification

- is used for answering the question :Have we built the software right? That is does it match the specification?
- Verification is the job of the developer.
- They use testing techniques for verification. In verification various reviews, inspections, and walkthroughs are used.
- The developer does Verification that the module or the project is doing the same as for the intended.

Reviews are made to check whether program goal has been achieved or not. One should distinguish between the reviews made about program resources use, and checking whether a system model or technical proposal is correct. These are two different review processes; one is to achieve correctness and other to ensure effective utilization of resources.

Inspections are made to evaluate program's qualitative features. Roles are allocated to people involved in inspection-Programmer whose product is under review, an inspector who evaluate and a moderator who control the review process.

Walkthrough are usually made to detect errors in the system. It is a procedure that is commonly used to check the correctness of the program produced by structured system analysis.

Q.132 Design an algorithm to compute summation of a set of numbers. **(8)**

Ans: A C algorithm to compute the summation of a set of numbers:

```
void main()
{
int i,sum,n;
int a[100];
clrscr();
printf("How many numbers you want to enter\n");
scanf("%d",&n);
printf("Enter Nos.\n");
for(i=0;i<n;i++)
   scanf("%d",&a[i]);
sum=Sum(a,n);
printf("\n Sum of the %d number is %d\n",n,sum);
getch();
}
int Sum(int a[],int n){
   int i,sum=0;
   for(i=0;i<n;i++)
      sum=sum+a[i];
 return(sum);
}
```

Q.133 Explain the top-down technique for algorithm design. **(8)**

Ans: Top-down design is a strategy that can be applied to take the solution of the computer problem from a vague outline to a precisely defined algorithm and program implementation. The design suggests that the general statements about the solution can be taken one at a time, and can be broken down into a set of more precisely defined subtasks. The subtasks can be implemented as program statements.
To make computer solutions to problems appropriate data structure should be selected. The data structure is selected by considering the points like easy searching, updating, excessive use of storage, recovering of earlier state in the computation etc.
For implementation of subtasks the iterative constructs, or loops and structures are used. To construct any loop, three things must be taken into account, the initial conditions, the invariant relation, termination.

Q. 134 Write an algorithm to reverse the digits of an integer. **(8)**

Ans:
l. Establish n, the positive integer to be reversed.
2. Set the initial condition *for* the reversed integer dreverse.
3. While the integer being reversed is greater than zero do
   (a) use the remainder function to extract the rightmost digit of the number being reversed;
   (b) increase the previous reversed integer representation dreverse by a factor of 10 and add to it the most recently extracted digit to give the current dreverse value;
   (c) use integer division by 10 to remove the rightmost digit from the number being reversed.

Q.135 Explain the following statements
(i) for loop
(ii) while loop
(iii) do-while loop
(8)

Ans:
(i) The for loop is entry-controlled loop that provides a more concise loop control structure. The general form of the for loop is
for(initialization; test-condition; increment) {
body of the loop}

1. initialization -The initialization of the control variable is done first, using assignment statements such as i = 1 and count = ( ).
2. test-condition - The value of the control variable is tested using the test-condition. The test-condition is a relational expression, such as i<10 that determines when the loop will exit. If the condition is true, the body of the loop is executed; otherwise the loop is terminated and the execution continues with the statement that immediately follows the loop.
3. increment - the control variable is incremented using an assignment statement.

(ii) While loop
while (test condition){

body of the loop}

The test condition is evaluated and if the condition is true, then the body of the loop is executed. If the condition is false, the control is transferred out of the loop.

(iii) Do while do

```
{
        body of the loop
}
  while( test-condition);
```

The program evaluate the body of the loop first. At the end of the loop, the test-condition in the while statement is evaluated. If the condition is true, the program continues to evaluate the body of the loop once again. The process continues as long as the condition is true. If the condition becomes false, the loop will be terminated.

Q.136 Write an algorithm to generate prime numbers in the first n positive integers.**(8)**

Ans:

1. Initialize and write out the first 3 primes. Also initialize the square of the 3rd prime.
2. Initialize x to 5.
3. While x less than n do
   (a) get next x value excluding multiples of2 and 3;
   (b) if not past end of multiples list then
      (b.1) if x square of largest prime then
      (1.a) include next prime multiple as its square,
      (1.b) update square by squaring next prime > x
   (c) while have not established x is non-prime with valid prime multiples do
      (c.1) wile current prime multiple is less than x, increment by current prime value doubled,
      (c.2) do prime test by comparing x with current multiple;
   (d) if current x prime then
      (d.1) write out x and if it is less than n store it.

Q.137 Write an algorithm to find the square root of a number. **(8)**

Ans: An algorithm for computing square root of a number is as follows:

1. Start
2. Initialise tol=0.000005
3. Scan the number 'val' for which square root is to be calculated.
4. Initialize oldval=val
   Initialize newval = (oldval + val/oldval)/2
5. while (fabs((newval-oldval)/newval) > tol)
   oldval = newval;
   newval = (oldval + value/oldval)/2;
6. Print "Square root of given number is:" newval
7. Stop

Q.138 Write a program to find the sum of odd and even numbers between 1 and 100.**(8)**

Ans:
```
#include<stdio.h>
main( )
{
int oddsum=( ), evensum=( ), I;
for (i=l; i <= 100; i+=2)
{
  oddsum += i;
  evensum += i+ 1;
}
printf("\n\n sum of odds is %d and evens=%d\n",
oddsum,evensum);
}
```

Q.139 Define pointers and discuss the advantages & disadvantages of pointers. **(6)**

Ans: Advantages of pointers are:

1. Function cannot return more than one value. But when the same function can modify many pointer variables and function as if it is returning more than one variable.
2. Flexible memory management: Allocating and deallocating memory as needed during run time allows you to create large objects, such as arrays, quickly and immediately free the memory when it is no longer required. In the case of arrays, we can decide the size of the array at runtime by allocating the necessary space.
3. Pointers provide a performance advantage by allowing you to access computer memory directly.
4. Pointers are not just for objects in memory; they can also be used for functions, thus allowing a function to be passed as a parameter to another function.

Disadvantages of pointers:

1. If sufficient memory is not available during runtime for the storage of pointers, the program may crash.
2. If the programmer is not careful and consistent with the use of pointers, the program may crash.
3. Direct access to memory means you can do things that perhaps you should not. Sometimes unintentionally (or intentionally) access to memory that is not yours, you could overwrite critical memory, modify the code of a running application, or cause your application or another application to behave or exit
unexpectedly.

Q.140 Write a program to swap two elements using pointer concept. **(6)**

Ans:
```
#include<stdio.h>
#include<conio.h>
#include<math.h>
int i,j;
void main( )
{
i=10, j=20;
```

```
clrscr0;
printf("The values before swap is i: %d, j:%d\n",i,j);
swap(&i,&j);
printf("The values after swap is i: %d,j:%d\n",i,j);
printf("\n");
getch( );
}
swap(int *x,int *y)
{ int temp;
   temp=*x;
   *x=*y;
   *y=temp;
}
```

Q.141 Classify the different types of programming errors. **(4)**

Ans: Programming errors can be classified broadly into following categories:
(i) **Compilation errors**: Compilation errors are caused by violation of the grammar rules of the language. The compiler detects, isolate these errors and terminate the source program after listing the errors.
These are of two types:
a. Syntax errors -- Common syntax errors include
missing or misplaced; or },
missing return type for a procedure,
missing or duplicate variable declaration.
b. Type errors -- These include
type mismatch on assignment,
type mismatch between actual and formal parameters.

(ii) **Linker errors**: Errors such as mismatch of data types or array out of bound error are known as Linker errors or run. These errors are generally going undetected by the compiler so programs with run-time error will run but produce erroneous results. Following are the types:
a. Output errors -- the program runs but produces an incorrect result. This indicates an error in the meaning of the program (logic error).
b. Exceptions -- the program terminates abnormally. Examples include
division by zero,
null pointer,
out of memory.
Some common programming errors are listed below:

- Missing semicolon: Every C statement must end with a semicolon. A missing semicolon is confusion to the compiler and may result in misleading error messages.
- Missing braces: Very common error as it is common to forget a Closing brace. Number of opening braces should match number of closing braces. .
- Undeclared variables: C requires declaration of variables before their use. .
- Forgetting the precedence of operators: Expression are evaluated according to precedence of operators. It is very common for beginners to forget this.

- Mismatch of parameters in function calls: There may be mismatch in actual and formal parameters in function calls.
- Missing '&' operator in scanf call.
- Crossing the bounds of an array.
- Unending and sometimes wrong loops.
- Using uninitialized pointer that points to garbage.
- Improper comment characters.

Q.142 What are the input and output functions used with the files. **(4)**

Ans: Input functions used in files are getc( ) and fscanf( )

getc( ) It is used to read a character from a file that has been opened in read mode.

fscanf( ) It works in the same way as scanf( ). It has one more parameter placed as first, which is the data file pointer.
fscanf(filej)ointer, "control string", list);

put( ) It has two parameters, a character expression and the file pointer declared as FILE. This prints one character by character into the file referenced by the file pointer.
putc(c, fileyointer);

fprintf( ) It works in the same way as printf( ). It has one more parameter placed at first, which is the data file pointer. This file pointer can be user-defined or the standard output device file stdout.
fprintf(file yointer, "control string", list);

Q.143 Write a menu driven program to find perimeter of rectangle, circle, square, triangle. **(8)**

Ans:

```
/* PROGRAM TO FIND PERIMETER OF RECTANGLE, CIRCLE,
 SQUARE, TRIANGLE */
#include<stdio.h>
# include<math.h>
#define pi 3.14
main( );
{
float 1, b, r, side;
int a, c;
printf(" 1 for rectangle");
printf(" 2 for circle");
printf(" 3 for square");
printf(" 4 for triangle");
scanf("%f  %f  %f  %f ", &1, &b, &r, &side);
switch( c)
{
case 1: printf("%f ", 1 * b);
     break;
case 2: printf("%f ", pi * r * r);
     break;
case 3: printf("%f ", 4 * side);
```

```
     break;
case 4: printf("%f ", (a + b + c)/2);
     break;
}
}
```

Q.144 Differentiate between Structure and Array. **(4)**

Ans:
(i) An array is a collection of related data elements of same type. Structure an have elements of different types.
(ii) An array is derived data type whereas a structure is a programmer-defined one.
(iii) Any array behaves like a built-in data type. Declare an array variable field use it. In the case of structure, first, data structure is to be designed and declared before the variables of that type are declared and used.

Q.145 Define a function? Explain Call by value and Call by reference. **(8)**

Ans:A function is a self-contained block of statements.
Function include the following elements function name, type, list of parameters, local variable declarations, function statements and a return statement. function_type function_name(parameter list)
{
return statement;
}
The technique used to pass data from one function to another is known as parameter passing. Parameter passing can be done in 2 ways.
1. Call by value (pass by value)
2. Call by reference (pass by pointers)
In call by value, values of actual parameters are copied to the variables in the parameter are copied to the variables in the parameter list of the called function. The called function works on the copy and not on the original values of the actual parameters.
In call by reference, the memory addresses of the variables rather than the copies of values are sent to the called function. The called function directly works on the data in the calling function and the changed values are available in the calling function for its use.

```
main( )                    main( )
{                          {
   . . . . .                  . . . . .
function1(x,y,z)           function1(x, y, &s,&d);
   . . . . .                  . . . . .
}                          }
functionl(int a,           function1 (int a, int b, int
int b, int c)              *sum, int *dift)
{                          {
   . . . . .                  . . . . .
   . . . . .                  . . . . .
}                          }
```

Q.146 Write a program to accept the elements of the structure
(i) emp_no (ii) basic_pay
and display the same structure along with the DA, CCA and gross salary. DA and CCA are calculated as follows
DA = 51% of basic_pay
CCA = Rs.100/- consolidated **(8)**

Ans:
```
main( )
{
int i, n, cca;
float da, gross;
struct
{
int emp_no, basic_pay;
}emp[10];
printf("Enter the number of employees");
scanf("%d", &n);
for(i=0; i<n; i++)
{
scanf("%d %d", &emp[i].emp _no,  &emp[i].basic_pay);
}
printf(Empno      Basic pay                          DA   CCA
Gross\n");
for (i=0; i<n; i++)
{
da=emp[i].basic-pay * .51;
cca= 100;
gross=emp[i]. basic_pay+da+cca;
printf("\n%4d %5d %7.2f %3d %8.2f ", emp[i].emp_no,
emp[i].basic_pay, da, cca, gross);
}}
```

Q.147 Explain the following operators
(i) Arithmetic Operator (ii) Relational Operator
(iii) Logical Operator (iv) Increment & Decrement Operator
(v) Conditional Operator

(10)

Ans:
(i) Arithmetic operator – C provides all the basic arithmetic operators.
+ addition
– subtraction
* multiplication
/ division
% modulo division
ex. a+b, a*b, a/b ~

(ii) Relational operator – These operators are used to compare two variables or expressions. C provide the following relational operators:

<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal
=	to equal to
!=	not equal to

(iii) Logical operator

&&	AND
||	OR
!	NOT

The logical operator && and || are used to test more than one condition.
Ex. a>b && x = = 10

(iv) Increment and Decrement operator
+ + increment
– – decrement
The operator + + adds 1 to the operand, – – subtracts 1
They both are unary operators.

Ex.	m = 10	m = 10
	X = ++m	x = m++
	Then x = 10, m = 10	then x = 10, m = 11

(v) Conditional Operator - A ternary operator pair "?" is available in C to construct conditional expressions of the form
exp 1 ? exp2 : exp3
where exp1, exp2, exp3 are expressions.
expl is evaluated first. If it is nonzero(true), then the expression exp2 is evaluated and becomes the value of the expression. If exp 1 is false, exp3 is evaluated, and its value becomes the value of the expression.
Ex. a=l0, b=15 x=(a>b)? a:b;

Q.148 Write a program to subtract two matrices. **(6)**

Ans:/* PROGRAM TO FIND THE SUBTRACTION TWO MATRICES */

```
#include<stdio.h>
#include<conio.h>
void main( )
{
int a[3][3], b[3][3], i, j, c[3 ][3 ];
clrscr( );
for(i = ( ); i < 3; i++)
{
for(j = ( );j < 3;j++)
{
printf("enter a[%d][%d]", i, j);
scanf("%d", &a[i][j] );
printf("enter b[%d][%d]", i, j);
scanf("%d ", &b[i]fj]);
```

```
}
}
for(i = 0; i < 3; i++)
{
for(j = 0; j < 3; j++)
     {
     c[i][j] = a[i][j] - b[i][j];
     printf("%d", c[i][j]);
     getch( );
     }
}
}
```

Q.149 Write a C function to find the maximum number in a set of n numbers. **(6)**

Ans:
1. Establish an array a[1..n] of n elements where n>1.
2.Set temporary maximum max to first array element.
3. While less than n array elements have been considered do
(a) ifnext element greater than current maximum max then assign it to max.
4. Return maximum max for the array of n elements.

Q.150 Write a C program to find the sum of digits of a number using recursion. **(8)**

Ans:
```
#include<stdio.h>
main( )
{
int n,sum,add(int);
printf("Enter an integer");
scanf("%d", &n);
sum=add(n);
printf("\nn The sum of the digits of %d is %d",sum);
}
int add(int a)
{
int t;
if(a = = 0)
return(0);
else
t=a%10+add(a/10);
return(t);
}
```

Q.151 Determine the value of the following expression
(i) x=a-b/3+c*2-1 a=15, b=12, c=3
(ii) z=2*x/(3*y) x=8.8, y=3.5 **(2)**

Ans:
(i) x = 15 – 12 / 3 + 3 * 2 – 1 = 15 – 4 + 3 * 2 – 1 = 15 – 4 + 6 – 1

= 11 + 6 – 1
=17 – 1
=16

(ii) z =2*8.8/(3*3.5)
= 2*8.8/(10.5)
= 17.6/10.5
= 1.67619

Q.152 Explain briefly the salient features of the various generation of computers. **(8)**

Ans: Salient Features of various generations of computer are:
i) First generation computers used vacuum tubes or valves, which worked on the principle of thermionic emission. The vacuum tubes contained filaments which, on heating, emitted electrons. Disadvantages of the first generation computers are:

- Extremely slow compared to the speed of the present day computers.
- Huge in size and not portable.
- Needed heavy air conditioning.
- Costly
- The system of processing was batch processing.

ii) Second generation computers replaced vacuum tubes with transistors. The transistors led to the development of new electronic hardware, including the first minicomputers. Advantages of second generation computers over the first generation ones are:

- Small size
- Operation speed relatively high.
- Portable and robust.
- Less cost.

iii) Third generation computers used integrated circuits (ICs). The integrated circuits are very small in size, required nominal power. The advantages of this generation of computers was that the size decreased considerably and speed was very high. They were portable and handy and the cost was also very low.

iv) Fourth generation computers used more powerful leis. like Medium, large and very large scale ICs. The most important electronic device developed during this generation was the microprocessor, which led to the development of micro computers. This led to computers which are :

- Very powerful in mathematical calculations and data processing abilities.
- Very fast, operating in the range of nanoseconds
- Capable of Multitasking
- Cheapest among all generation of computers.

v) Fifth generation computers are thought of to be intelligent ones (using Artificial Intelligence) which is lacking in today's computers. They would probably:

- work in parallel and not serially.

- Do a greater number of tasks at a time.
- Be faster and more powerful than the fourth generation computers and have large memories.
- Will do knowledge-processing in addition to data-processing jobs.

Q.153 Distinguish between minicomputer, microcomputer and mainframe computer. **(8)**

Ans:

Mini Computer	Micro computer	Mainframe Computer
Smaller than mainframe	Smallest in size	Large computers.
Not portable.	Portable	Not portable
It integrates commercial and technical operations better than the more powerful computers.		Great power and storage capacity.

Q.154 Distinguish between the following:
(i) Cache and virtual memory
(ii) Static and dynamic RAM **(8)**

Ans:

(i) Cache and virtual memory.

Cache memory:- A CPU cache is a cache used by the central processing unit of a computer to reduce the average time to access memory. The cache is a smaller, faster memory which stores copies of the data from the most frequently used main memory locations. As long as most memory accesses are to cached memory locations, the average latency of memory accesses will be closer to the cache latency than to the latency of main memory.

Virtual memory is an addressing scheme implemented in hardware and software that allows non-contiguous memory to be addressed as if it were contiguous. All current implementations of virtual memory support that Programs can address data that does not currently reside in main memory. When this occurs, the hardware and operating system automatically load the data at the requested address from auxiliary storage into main memory. This occurs transparently to the user program. As a result, programs can reference a larger amount of (RAM) memory than actually exists in the computer.

(ii) Static and dynamic RAM.

In static type of RAM i.e. SRAM, once the data is stored, it will be retained as it is in the memory as the static charge. It is relatively costlier than Dynamic RAM.
In case of Dynamic RAM i.e. DRAM, data once stored will not be retained forever unless we provide an electrical pulse to 'make it remember' repeatedly after some interval of time and this type of remembering pulse is known as the refresh pulse.

Q.155 What is DVD-ROM? What are the differences between CD-ROM and DVD-ROM? **(8)**

Ans:
a) DVD (also known as "Digital Versatile Disc" or "Digital Video Disc") is a popular optical disc storage media format used for data storage. Primarily uses are for movies, software, and data backup purposes, DVDs are of the same form factor as compact discs (CDs), but allow for 8 times the data storage capacity (single-layer, single-sided).

i) Storage capacity: Going by the storage capacity, the DVD specifications have four disk configurations, ranging from 4.7 GB to 17 GB. It ranges from single-sided, single-layer disks that are much like traditional CD ROM disks to double-sided double-layered DVDs. Normally a DVD ROM with a minimum storage capacity can store up to the data of 7 CD ROMs. Normally the capacity of a CD ROM is only 650 MB. The capacity of DVD ROM starts from 4.4 GB, which is 7 times more than a normal CD ROM. However, this storage capacity of DVD ROM relates to single-layer storage. But with double-layer storage, the storage capacity of DVD ROM can go up to 8 GB, which is 12.5 times higher than a normal CD ROM. Furthermore, double-sided, DVD ROM drives can store 8.8 GB, 14 times greater than a CD ROM's capacity. Double-sided, dual-layer DVD-ROM drives store 15.9 GB, 25 times greater than a CD ROMs capacity. To put it differently a CD ROM can hold the entire encyclopedia, but a DVD ROM can hold the encyclopedia, the dictionary, the thesaurus, and the phone number/address of everyone in whole India with still a lot of space left! With the storage of a DVD ROM, you can see interactive multimedia programs that use hours of full-screen video to help you see movies, play games or even learn your courses.

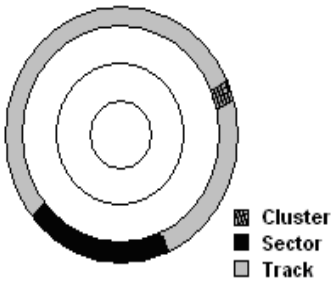
ii) Speed of data transfer: DVD ROM scores over the CD ROM. Going by figures the minimum transfer rate of CD ROM drives is 150 KB per second. However, the minimum transfer rate of DVD ROM drives is 11.08 MB per second. The data transfer rate of DVD ROM is approximately equivalent to a 92X CD ROM.

iii) Reliability: Individually though CD ROMs are very reliable and have a long shelf life yet if compared to the DVD, the DVD scores over the CDs in terms of reliability. Since DVD ROM discs are made of plastics bonded together, the discs are more rigid than CD ROM discs. Though the DVDs can also catch scratch just like CDs yet the scratches in the DVD normally do not matter as they are usually out of focus of the laser and therefore it does not affect the readability of the DVDs. In case of CDs, scratches could affect the data and could also make it useless if the scratches destroy critical tracks. Besides scratches, other factors like error correction also makes DVD much better and reliable in comparison to the CDs. Normally, in case of the DVD, the error correction is 10 times more effective than the CD error correction. Moreover the CDs normally follow RS-CIRC error correction mode, whereas DVD have RS-PC error correction. Likewise the error correction overhead in case of CD ROM is very high at 34 per cent, whereas it is only 13 per cent in case of DVD drive.

Q.156 What is a sector and track in hard disk? **(4)**

Ans:A Sector in the context of computing refers to a small area of a storage device, for example a hard disk drive.
For instance, a typical hard disk drive when low-level formatted is split into tracks, sectors and clusters:

- Tracks are concentric circles around the disk.
- Sectors are segments of a track.
- Clusters are a set of sectors.



Q.157 Briefly describe the function of serial access secondary storage device. **(4)**

Ans: Serial access storage device: These devices are used where data is to be accessed serially. Normally these devices are used for backup of large amount of data as it is cheap. Magnetic tape is example of this device.

Q.158 What are the different types of key switches used in keyboards? Explain their working with the help of neat diagram. **(8)**

Ans:Membrane Contact Keyswitches: The *membrane* contact keyswitch is a variant of the rubber dome design and works in a rather similar way. The basic mechanism is the same: contact pairs on a circuit board, and rubberized boots or "dimples" with a carbon button underneath. Press down the key and the rubber deforms, the carbon touches the contacts and a keystroke is sensed. The big difference here is that individual keycaps and plastic plungers are replaced with a thin membrane that fits over the rubber domes. There may not even be separate rubber domes, just molded "dimples" for each key, the carbon buttons in each dimple. The user presses directly on the membrane to collapse the domes and create contact with the printed circuit board. Travel is very small with this design, since there is no keycap and no plunger.

Capacitive Keyswitches: All of the other keyswitch technologies described in this section are *contact* designs. They all work using different ways of causing physical contact to establish a circuit and register a keypress. There is one keyboard technology, however, that detects keystrokes without using any form of contact at all. The *capacitive* keyswitch design makes use of a design characteristic of capacitors to determine when a key has been pressed. A capacitor is an electronic component that is comprised (at least conceptually) of a pair of parallel metal plates. When an electric field is applied to the plates, a charge is stored there.

Foam and Foil Contact Kevswitches: Like the mechanical contact keyswitch, the *foam and foil contact* keyswitch design also uses contact to complete a circuit and indicate when a keypress is made. However, it takes a very different approach to creating the contact. Each keyswitch is constructed of a (usually plastic) plunger on the top, connected to a foam pad. The foam pad is coated with foil on the bottom. A spring wraps around the plunger at the top, suspending the key in its normal position. Below all the keyswitches is a circuit board, printed with many pairs of copper contacts; one pair is oriented under the foam pad of each keyswitch. When they key is pressed, the foam pad moves down and touches the pair of copper contacts, completing the circuit and telling the keyboard that a key was pressed.

When the key is released, the spring pulls the plunger back up, breaking the contact.
Mechanical Contact Kevswitches: The simplest keyswitch technology in conceptual terms, *mechanical contact keyswitches* are "classical" switches, working in much the same way that many other types of switches do in the world around us. They work simply on the basis of two contacts mechanically touching each other to complete a circuit, not unlike a doorbell in some ways.

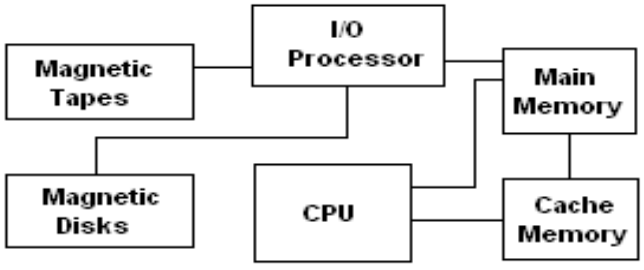
Q.159 Explain the term Hardware, Software and Firmware, specifying the relationship between them? Name at least two items of all categories. **(8)**

Ans:

Hardware	Software	Firmware
The physical components of a computer system, such as the computer itself, its modules and peripherals (the input/output devices and auxiliary storage units) are called hardware.	Software is basically the set of instructions grouped into programs that make the electronic devices in the computer to function in the desired ways. The different categories of software are system software (E.g are Operating system like DOS, UNIX, XENIX etc) and application software (E.g. are Word processors, electronic spreadsheets etc.). Software drives the hardware.	

Q.160 Briefly describe memory hierarchy and explain logic for making memory address map. **(8)**

Ans: Memory hierarchy:- The memory hierarchy system consists of all storage devices employed in a computer system from the slow but high-capacity auxiliary memory to a relatively faster main memory, and to an even smaller and faster cache memory accessible to the high-speed processing logic.
Example diagram of memory hierarchy in a computer system:



Memory address map:- The designer of a computer system must calculate the amount of memory required for the particular application and assign it to either RAM or ROM. The interconnection between memory and processor is then established from knowledge of size of memory required, and the type of RAM and ROM chips available.
The addressing of memory can be established by means of a table that specifies the memory address assigned to each chip. This table is called Memory Address Map, which is a pictorial representation of assigned address space for each chip in the system.

Q.161 What is a cache? How it is mapped with main memory? **(8)**

Ans: Cache memory: - The cache memory is employed in computer systems to compensate for the speed differential between main memory access time and processor 10gic.CPU logic is usually faster than main memory access time, with the result that processing speed is limited primarily by the speed of main memory.
Mapping of memory:- The basic characters tics of cache memory is its fast access time. Therefore, very little or no time must be wasted when searching for words in the cache. The transformation of data from main memory to cache memory is referred as a mapping process. Three types of mapping procedures are of practical interest when considering the organization of cache memory: Associative mapping, Direct mapping and Set-associative mapping.

Q.162 What is arithmetic and logic unit? Explain various functions performed by this? **(8)**

Ans: The arithmetic and logic unit is the core of any processor, it is the unit that performs simple arithmetic-logic calculations and shift operations. A typical ALU will have two input ports (A and B) and a result port (Y). It will also have a control input telling it which operation (add, subtract, and, or, etc) to perform and additional outputs for condition codes like carry, overflow, negative, zero result.
Operations: This is used for most of logical processing, for example, for calculations or comparisons. The arithmetic operations like + , – , * , and / are performed here. The logical operations like < , >, = , <= , >= and < > are also performed here.
In most of the arithmetical operations the result is in numerical form while in the case of logical operations the result can be YES/NO or TRUE/FALSE.

Q.163 What are cursor control devices? Explain working of any one such Device. **(8)**

Ans: These are the devices that are used for control of the cursor. They allow the position of control on the computer and the functions of the software to be executed. Example: mouse, trackball, light pen etc.

Mouse:- It is a pointing device having a small box with a round ball on the bottom and three buttons on the top. The mouse is attached on the terminal by cables. It allows the user to manipulate the cursor on the screen. The mouse captures the cursor and the movements of mouse control the operations on the computer.

Mouse could be optical, offering quite and reliable operation, or mechanical which is cheaper but noisier. User can move the mouse, stop it at a point where the pointer is to be located and with the help of buttons, make selection of choices.

Q.164 Explain the difference between an impact and non-impact printer? Which one is advantageous over the other and why? **(8)**

Ans: Impact printers use variations of standards typewriter printing mechanism where a hammer strikes paper through inked ribbon.

Non-Impact printer uses chemical, heat or electrical signals etch or induce symbols on paper. Many of these require special coated or treated paper.
Advantages due to applications :- Impact printers have the ability to produce multiple copies of documents as these printers are relatively cheaper compared to other technologies. So, these are good options if it is to be used for high volume work because of great speeds and less cost.

Non-impact printers provide good quality print at speeds unapproachable by other technologies like in Laser printer. These printers cannot produce multiple copies at one running.

Q.165 Write various specific features of Windows operating system. **(8)**

Ans: It is a multi-tasking operating system with graphic user interface Main features are:
(i) Program Manager: When the start button is pressed, a program manager is available by the name 'program'. This program manager stores the menu for an opening installed program of the window system.
(ii) File manager: A file manager called window explorer appears in
(iii)Print Manager: This is used to set up a printer.
(iv) Task Switching: This allows switch over between open windows.
(v) Close, Minimize, Maximize Button: This is used to open/close the running program.
(vi) My computer: This shows the complete details of the content of the drive and directories.
(vii) Network Neighbourhood: It connects the network to available resources.
(viii) Recycle Bin: This is a temporary storage for deleted files.

Q.166 What do you understand of multiprogramming and time sharing of system? **(8)**

Ans: Multiprogramming
In this many jobs of different users are performed in the memory at a time. The processor executes a portion of one program and then a portion of another. This way it continues the execution of the program in a sequence, until it comes to the end of the program. There are three different states of multiprogramming:
1. Ready: the program is able to use the process or when it is assigned to it.
2. Blocked: the program is waiting for the input and output operations to complete and is able to utilize the processor till then.
3. Running: the program is under processing by CPU.
Timesharing
It is most desirable for an individual user to minimize turn around time. This system allows several users to use the system simultaneously. There are several terminals connected to the system, which operate simultaneously. The CPU allots a fixed time period to each user and serves them in turn. This time sharing system is called interactive processing system.

Q.167 What types of softwares are used for specific application? Explain any one example. **(8)**

Ans: Application softwares are used and developed for the specific area of application of the user. These are customer oriented softwares suitable for specific area of application.
Example: Business Application
Micro electronic technology is enabling offices to function more efficiently nowadays, by various concerns:
a. In business forecasting.
b. To keep records up-to-date.
c. To carry out automatic checks on the stock of a particular item.
d. To prepare pay bills and personnel records.
e. In accounting, invoicing and billing.
f. In banking operations and data storage.
g. In business correspondence and communications.
h. In functions of various types in Life Insurance Business.
i. As an aid to management etc.

Q.168 What is the purpose of CONFIG.SYS and AUTOEXEC.BAT file. **(8)**

Ans: The file Config.sys contains special commands that configure hardware components and application programs. This helps to organize the storage devices and to use operating system to format label, copy and organise hard and floppy disc. It helps to organize the data stored in files. It provides an easy to use interface that let to communicate with the computer. Autoexec.bat is executed immediately after the Config.sys file and contains users own customized startup procedures to load programs automatically, display messages and specifies the software to be loaded. The path of the program is indicated by following example:
Drive:\directory\file
c:\tc\bin.

Q.169 What is mail merge? Write the steps to be followed in mail merge? **(8)**

Ans: In any working environment, there are times when a similar type of letter to be sent to many persons. In such case, general contents of each letter remain same, only name, address and other few information varies. So Mail Merger tool of word-processor is used to avoid same letter again and again.
It requires two different files which contain the following:
1. General body of letter
2. Data which varies from letter to letter
Steps followed in mail merge:
1. Press "N" at Opening menu to get non-document fi1e. Name the file with extension .DAT then key in various data of each client as records. Each field in the record should be separated by "," (comma)
Then save the non-document file. .
2. Press "D" at Opening Menu to open a document fi1e. Name the file and then type the letter.
.DF - This command informs wordprocessor about the name of data file to be used for mail merge.
.RV - This dot command is used in Mail Merge to specify the names of the variables which are used in document file. They are defined in order in which fields are stored ego NAME, ADDRESS1, ADDRESS2
All these places where these variables are used, the names would be preceded and succeeded by a "&" . This will communicate to word processor that following word is not a text but variable.

Q.170 Write applications and advantages of Spread Sheet? **(8)**

Ans: Applications: Spread sheets perform calculations, recalculates the result if any data stored in them changes. It is helpful for creating financial reports, comparing reports etc. It has a strong feature for creating groups which allow to illustrate relationship between two or more sets of data and understands the trends of data changes more easily.
Example: budgets, annual report of firm, payroll, bills, mark sheets, banking, inventory etc.
Advantages:

- Several mathematical, trigonometric, financial and statistical functions are built in. All sorts of complicated calculations can be performed very easily using these functions facilitating rapid operation.
- The results are accurate.
- Worksheet can be very large and any part can be viewed or edited.
- Data can also be viewed in graphs.
- Part or complete worksheet (WS) can be printed.
- WS can be stored and when required can be retrieved and edited.
- Any existing WS can be merged with any existing/new WS.
- Reports can be printed

Q.171 Write down the steps of inserting a formula in an EXCEL spread sheet. Which are the mathematical operators, used in Excel's formula? **(8)**

Ans:A formula is a numeric expression containing mathematical operation that result in a single value. In this we can perform addition, subtraction, multiplication, division or exponentiation.
Move the cell pointer to destined cell and click on the function option (From toolbar OR Menu Bar). Choose the function accordingly.
<some text is missing>
Mathematical Operator:

1	^	Exponentiation
2	+	Addition
3	–	Subtraction
4	*	Multiplication
5	/	Division

Any data can be changed without changing sheet and effect can be seen. This feature is one of the most useful features of spread sheet.

Q.172 For the purpose of deleting the file and directory which commands are used in a DOS system? Explain by suitably taking one example. **(8)**

Ans: For deleting a files and directories, the following commands are used:

Command	:	del (delete) eg.del XX.c (xx.c file name)
Command	:	rd (remove directory) eg. rd cc (cc is directory and should be empty)
Command	:	del tree (delete tree) eg. del tree yy (yy is directory which may not necessarialy be empty ,this command deletes all files in directory yy) .

Q.173 Write a note on capabilities and limitations of a computer. **(4)**

Ans: Capabilities of a Computer:- Computers are capable of having great processing speeds, huge memory, accuracy and versatility. The can be put to use in various areas like:

- For calculations in engineering and scientific research
- In data processing jobs.
- Commercial and financial applications
- Office automation.
- In CAD CAM
- Robotics
- Weather forecasting

Limitations of a computer:

- They need to be first programmed with specific instructions
- They cannot decide how are to be programmed or provide their own input.
- They can't interpret the data they generate.
- They can't implement any decisions they suggest.
- They can keep track of scientific data, but they can't conceive or express the ideas for continued research.

Q.174 What is a Super Computer? Give an example? **(4)**

Ans:A super computer is generally characterized as being the fastest, most powerful and most expensive computer.
One of the most powerful supercomputer today, The Cray-2 is set up in a Cshape, and is small enough to fit in a space of a large business desk.
Supercomputers recognize the largest word lengths of 64 bits or more. They calculate at rates upto 1.2 billion instructions per second. They can take input
from over 10,000 individual workstations.
Supercomputers are widely used in scientific applications such as aerodynamic design and simulation, processing of geological data, processing, of data regarding genetic coding and collecting and processing weather data.

Q.175 Explain the term single address, two address, three address and four address instructions. **(8)**

Ans:Single address instruction:- This type of instruction uses accumulator (AC) register for all data manipulation.
Eg: LOAD A, ADD B, STORE T etc.
Two address instruction:- In this type of instruction two memory address are provided.
Eg. MOV R1, A, ADD RI, B
Three Address instruction:- Three address instruction uses three operands in the instruction.
Eg. ADD R1, A, B

MULX, R1, R2

Q.176 Write a small machine language program for a two address computer to add two numbers and output the sum. **(8)**

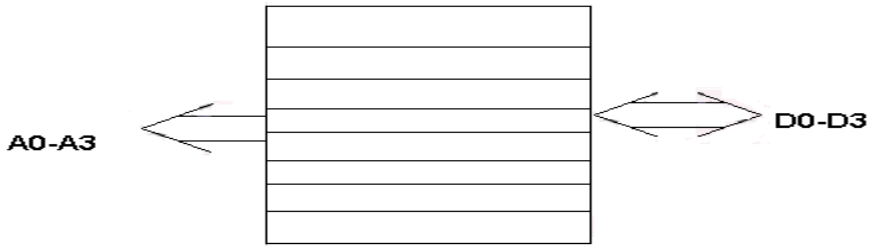
Ans: Let
MOV R1, A
ADD R1, B
MOV C, R1

Q.177 What is the size of MAR of a 16KB memory? Assume that the word size is 1 byte. **(4)**

Ans:MAR size is 16 bits.
As 16KB = 16 * 1024 = 2^{16}.

Q.178 Draw a block diagram of a memory which has 8 words of 4 bits each. **(8)**

Ans:



Q.179 How is data recorded on a CDROM? How is it read? **(8)**

Ans:CD-ROM (an abbreviation of "Compact Disc read-only memory") is a Compact Disc that contains data accessible by a computer. While the Compact Disc format was originally designed for music storage and playback, the format was later adapted to hold any form of binary data. CD-ROMs are popularly used to distribute computer software, including games and multimedia applications, though any data can be stored (up to the capacity limit of a disc). Some CDs hold both computer data and audio with the latter capable of being played on a CD player, whilst data (such as software or digital video) is only usable on a computer. These are called Enhanced CDs. Data is stored on the disc as a series of microscopic indentations ("pits", with the gaps between them referred to as "lands"). A laser is shone onto the reflective surface of the disc to read the pattern of pits and lands. Because the depth of the pits is approximately one-quarter to one-sixth of the wavelength of the laser light used to read the disc, the reflected beam's phase is shifted in relation to the incoming beam, causing destructive interference and reducing the reflected beam's intensity. This pattern of changing intensity of the reflected beam is converted into binary data.

Q.180 If a hard disk rotates at 3600 rpm and surface recording density on it is 1 Mbpi, what is the data transfer rate of the disk? **(4)**

Ans: Data Transfer Rate = (Spindle Speed /60 * recording density)
= 3600/60 * 1 = 60 MBPS

Q.181 Define the terms "Seek time" and "Latency time" of a magnetic disk. **(4)**

Ans: Seek time - The seek time of a hard disk measures the amount of time required for the read/write heads to move between tracks over the surfaces of the platters. Seek time is one of the most commonly discussed metrics for hard disks, and it is one of the most important positioning performance specifications.
Latency Time - The hard disk platters are spinning around at high speed, and the spin speed is not synchronized to the process that moves the read/write heads to the correct cylinder on a random access on the hard disk. Therefore, at the time that the heads arrive at the correct cylinder, the actual sector that is needed may be anywhere. After the actuator assembly has completed its seek to the correct track, the drive must wait for the correct sector to come around to where the read/write heads are located. This time is called latency. Latency is directly related to the spindle speed of the drive.

Q.182 Explain the working of voice recognition devices? What are its applications? **(8)**

Ans: Speech recognition (in many contexts also known as automatic speech recognition, computer speech recognition or erroneously as voice recognition) is the process of converting a speech signal to a sequence of words, by means of an algorithm implemented as a computer program.
Speech recognition applications that have emerged over the last few years include voice dialing (e.g., "Call home"), call routing (e.g., "I would like to make a collect call"), simple

data entry (e.g., entering a credit card number), preparation of structured documents (e.g., a radiology report), domestic applications control and content-based spoken audio search (e.g. find a podcast where particular words were spoken).
Voice recognition or speaker reorganization is a related process that attempts to identify the person speaking, as opposed to what is being said.
Speaker-dependent dictation systems requiring a short period of training can capture continuous speech with a large vocabulary at normal pace with a very high accuracy. Most commercial companies claim that recognition software can achieve between 98% to 99% accuracy (getting one to two words out of one hundred wrong) if operated under optimal conditions.
This explains why some users, especially those whose speech is heavily accented, might actually perceive the recognition rate to be much lower than the expected 98% to 99%. Speech recognition in video has become a popular search technology used by several video search companies.

Q.183 Explain the working of Thermal and Crystal based inkjet printers. **(8)**

Ans:Thermal Ink Jet

Most consumer ink jet printers work by having a print cartridge with a series of tiny electrically heated chambers constructed by photolithography. To produce an image, the printer runs a pulse of current through the heating elements. A steam explosion in the chamber forms a bubble, which propels a droplet of ink onto the paper (hence Canon's tradename for its inkjets, *Bubblejet).* The ink's surface tension as well as the condensing and thus contraction of the vapour-bubble, pulls another charge of ink into the chamber through a narrow channel attached to an ink reservoir.
The ink used is usually water-soluble pigment or dye-based but the print head is produced usually at less cost than other ink jet technologies.
This is not the same thing as a thermal printer, which produce images by heating thermal paper, as seen on some fax machines, cash register and ATM receipts, and lottery ticket printers.
Crystal based Ink Jet
All Epson printers and most commercial and industrial ink jet printers use a piezoelectric material in an ink-filled chamber behind each nozzle instead of a heating element. When a voltage is applied, the crystal changes shape or size, which generates a pressure pulse in the fluid forcing a droplet of ink from the nozzle. This is essentially the same mechanism as the thermal inkjet but generates the pressure pulse using a different physical principle. Piezoelectric ink jet allows a wider variety of inks than thermal or continuous ink jet but is more expensive.
Ink Jet with Piezoelectric is very fast and cost effective. When the Piezo crystal has an applied voltage, the crystal will shake the ink stream, causing it to break off in very small, fine droplets as the ink leaves the orifice plate hole. This droplet of ink can then be either charged or not charged, depending on whether the droplet of ink is to be printed or not.
If the droplet is to be printed onto the paper, the ink droplet is not charged. However, if the droplet is not required to be printed to the paper, it is charged with a positive bias, this way the ink droplet is then attracted to the negatively biased charge plate, the ink will hit the plate and will be vacuumed away by an ink recycle system. (This is used during the printer's

automatic head-cleaning procedure, albeit consuming usable ink in the process.)

Q.184 Differentiate between single-user and multiuser operating system? **(5)**

Ans:A Single user operating system is mainly DOS here one program runs at a time and it is a character user interface, whereas in multiuser more then one program runs at a time and it works in a time sharing mode. The CPU time is shared with different users in a round robin fashion.

Q.185 Briefly define Multiprocessing and Multiprogramming. **(5)**

Ans: Multiprocessing is a method of processing using more than one central processing unit where more than one set of instruction can be executed at a time.
Multiprogramming is the simultaneous handling of multiple independent programs by interleaving or overlapping their execution.

Q.186 Compare XCOPY, DISKCOPY and copy commands in DOS. **(6)**

Ans:XCOPY command in DOS helps in copying a directory structure. DISKCOPY command helps in making a copy of a Disk while copy command is used to copy a single or a group of files from one location to another.

Q.187 What is the advantage of mail merge. **(5)**

Ans: Mail merge is a utility provided by Ms-Word which helps in sending the same message or letter to many people. Fields from a data source are added to a main document containing the required text and can be sent to the printer or an email. Mail merge can also be used to make envelopes, mailing labels from a database.

Q.188 Write the steps for creation of bar charts of annual sale of a company saving five products. **(5)**

Ans: The steps for creating bar charts of annual sale of a company having five products are:-

- First select the data of worksheet from which you want to create chart.
- Choose the Chart menu from Insert menu
- A chart wizard dialog box appears on the screen.
- In Step 1 of 4 we have to select the chart type.
- In Step 2 of 4 we have to mention the data range.(sale of products)
- In Step 3 of 4 we have to select the chart options Title, legends, axis, data table, gridlines, data labels etc.
- In Step 4 of 4 we have to select the chart location i.e the same sheet or new sheet.

Q.189 Give the full form of
i) MSI
ii) EDSAC
iii) RAID **(3)**

Ans:
MSI → Medium Scale Integration
EDSAC → Electronic Delay Storage Automatic Calculator
RAID → Redundant Array of Inexpensive Disks

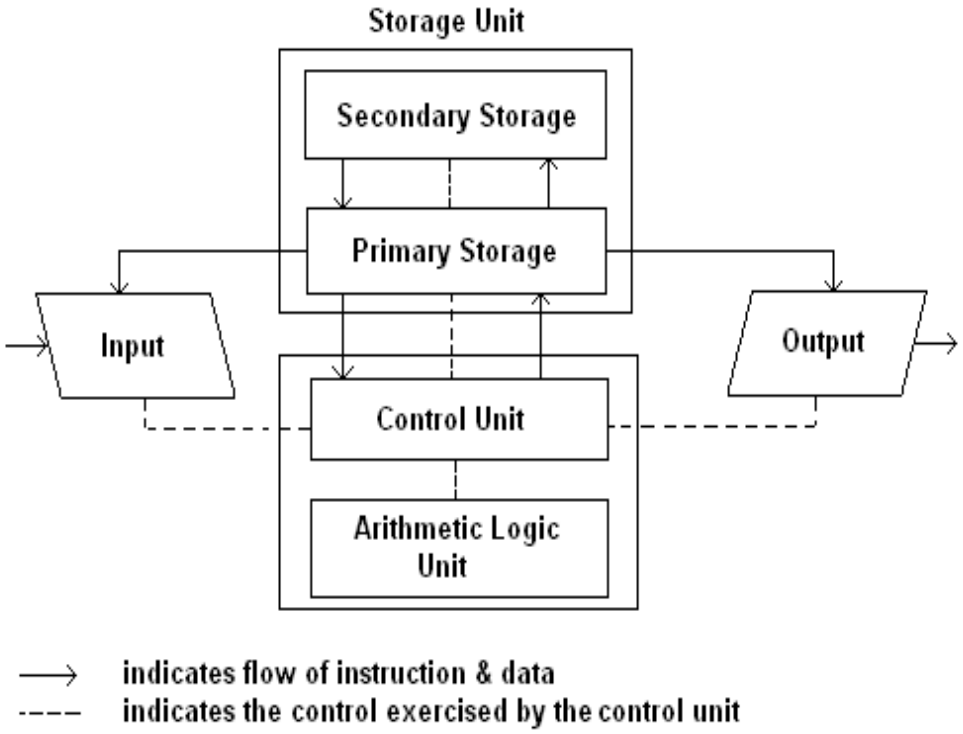
Q.190 Write a short note on super computers and their applications. **(5)**

Ans: Supercomputers are the most powerful and most expensive computers available at a given time. They are mainly used for processing complex scientific applications that require large processing power Super Computers are multi-processing and parallel processing technologies to solve complex problems faster, and hence they are parallel computers or parallel processing systems. Modern Super Computers employ hundreds of processors and are known as massively parallel processors.

Applications
(i) Super Computers are used to analyse large volumes of seismic data during oil-seeking explorations to identify the possible areas of getting petroleum products.
(ii) Super Computers are used to stimulate airflow around a aircraft at different speeds at altitudes for processing an effective aerodynamic design to develop aircrafts with super performance.

Q.191 Describe the basic functions performed by a computer system, with the help of a suitable diagram. **(8)**

Ans: The five functional units of a digital computer system are input unit, output unit, storage unit, arithmetic logic unit, control unit.
INPUT UNIT:- Input device is a device, which accepts (read) the list of instructions and data from the outside world, converts it into machine readable form and transmits it, to the memory unit of the computer.
Ex. Keyboard, Mouse, Scanners, OMR, OCR etc
OUTPUT UNIT :- Output unit is a device, which translates information regarding the result of data processing into a human acceptable form and is supplied to the outside world.
Ex. Monitor, Printer, Plotter etc.
STORAGE UNIT:- Storage unit holds the instructions and data to be processed, and the intermediate and final results of processing. It is of two types, Primary memory (main memory) and secondary memory (auxiliary memory). Primary memory is the built in memory, which holds the data and instructions between the processing steps. CPU directly addresses the main memory.
Ex. RAMs and Semiconductor memories.
Secondary memory stores data file, compilers, application programs etc. The CPU does not directly read from the secondary memory. The information is first transferred to the primary memory and then accessed by the CPU.
Ex Hard Disk, CDs etc.



ALU (Arithmetic Logic Unit) The actual execution of the instructions is carried out in this unit during the processing operations.
Control unit (CU):- It manages and co-ordinates the operations of all the other components of the computer system. Both ALU & CU together form the Central processing unit of the computer system.

Q.192 Find the data transfer rate (in bytes/sec) for a disk pack, with a disk pack capacity of 20 Mbytes, 19 storage surfaces and 700 tracks/surface. **(4)**

Ans: Given Disc capacity = 200 Megabytes = 200 x 1024 kbytes.
No. of cylinders = No. of tracks per surface = 700
No. of tracks/cylinders = No. of used surface of the disk pack
= 19 – 2 = 17
Data Storage/Track = (200x1024)/(700x17) = 17.21 kbytes

Q.193 Give the limitations and uses of a magnetic disk. **(3+3)**

Ans: The uses of magnetic disks

- Used in application based on random data processing
- Used as a shared online secondary storage devices ex. Winchested disks
- Used as backup device for offline storage of data for later retrieval when required. Ex. Floppy disks, Tape disks, Disk packs

Limitations of magnetic disks

- Different to maintain the security and information stored on magnetic disks which are used as online secondary storage devices.
- They must be stored in a dust-free environment
- Costly compared to magnetic tapes
- Disk crash or drive failure often results in loss of entire data stored on it. It is not easy to recover the lost data.

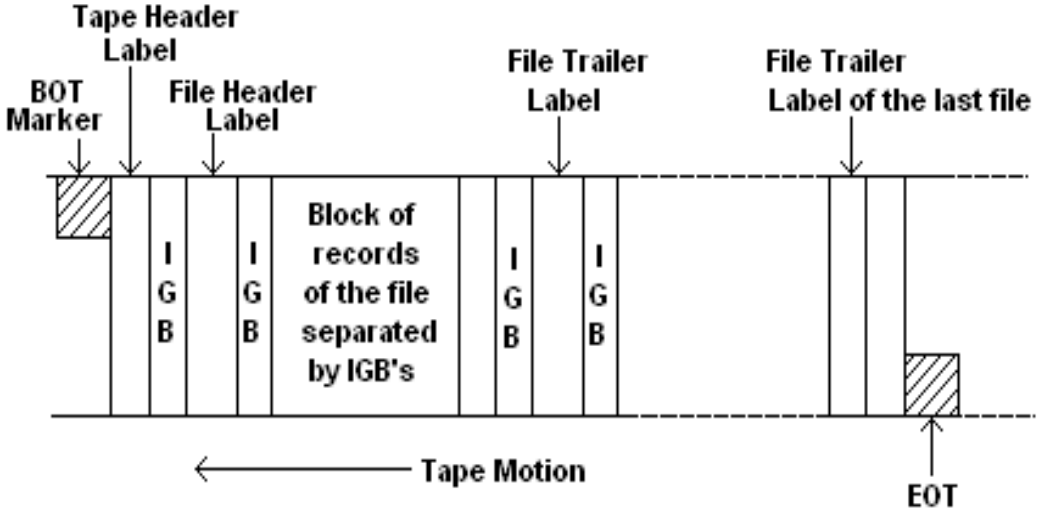
Q.194 Explain how information is recorded on a magnetic tape. **(6)**

Ans: The data on the tape is organized in the form of records separated by gaps. A gap between two consecutive records is the inter-block gap. A gap of records from a block, and a set of blocks constituted a file. A single tape can store many files. Many files are stored one after, the other sequentially.

To identify the files, the computer adds a file header label identifying the beginning of the file and file trailer label to signify the end of file. The file header label contains the label attribute, i.e. name of the file data when created etc.

A tape is wound on the spool & during I/O operation it moves from a supply spool to take-up spool. The first and last several feet of he tape are unused to allow the threading on take-up and supply spools. Two markers BOT and EOT are placed on the opposite edges of tapes for machine identification process. Beginning of tape (BOT) a reflective metal foil indicates the beginning of the tape to the equipment. End of tape (EOT), a reflective metal foil indicates the end of the unusable tape to the equipment.

Tape header label is placed between the BOT and the first file header file. It contains the tapes attributes such as the tape identifier, the number of files it contains, the data it was last used, and other control information that helps to prevent an important tape from accidentally being erased.



Data Organization on a Magnetic Tape

Q.195 Explain the working of an ink-jet printer. What are the two different types of ink-jet printers? **(4+2)**

Ans: Inkjet printers that form character and all kinds of images by spraying small drops of ink on the paper. The printer head of an inkjet printer contain upto 64 tiny nozzles, which can be

selectively heated up in a few microseconds by an integrated circuit register. When the register heats up, the ink near it vapourizes and is ejected through the nozzle and makes a dot on the paper placed in front of the head moves horizontally.
Inkjet printers can be both monochrome and colour. A monochrome inkjet printer uses a single print head, whereas a color inkjet printer uses multiple print heads, one per colour. Colour inkjet printers can produce multiple colour output.

Q.196 Explain the following terms with reference to a CRT monitor:
(i) Screen size
(ii) Resolution
(iii) Refresh rate **(3x2)**

Ans:
(i) SCREEN SIZE:- Display screens of monitor vary in size from 5 to 25 inches (diagonal dimension). Monitor with 17 to 21 inch screen sizes are common today with personal computers and other desktop computers. With users increasingly viewing multiple windows, video clips, animated objects, and complex graphics, monitors with larger screens are preferred. However, the larger the screen the costlier is the monitor,.

(ii) RESOLUTIONS:- Resolution refers to the number of point to which electron beam can be directly (pixels) on the screen. As the pixels are arranged in scan lines, the resolution of a monitor depends on its total number of scan lines and total number or pixels per scan line. The total no. of scan lines is called a monitor's VERTICAL RESOLUTION and the total number of pixels per san line is called its HORIZONAL RESOLUTION. The overall resolution of a monitor is expressed as a multiple of its horizontal and vertical resolutions. The higher the resolution of a monitor, the clearer will be its displayed images. Very high resolution of monitors, project extremely clear images that look almost like photographs.

(iii) REFRESH RATE:- The actual number of times that the electron beam scans the entire screen per second is called the REFRESH RATE of the monitor. The higher the refresh rate, the better will be the display quality of a monitor because it will be less strenuous for the eyes to continually view the screen. Today most monitors operates at 60Hz. That is, they refresh the screen 60 times per second. Better quality monitors operate at 70 to 90 Hz, refreshing the screen 70 to 90 times per second.

Q.197 Explain the function of light pen and joystick. **(4)**

Ans:LIGHT PEN Alight pen is an input device that uses a light sensitive stylus connected by a wire to a video terminal. The user brings the light pen to the desired point on the screen surface and presses a button, causing it to identify the location on the screen. Light pens are used to select the options from a menu displayed on screen or to draw images in graphics system by "dragging" the cursor around the screen. The pixels (dots) on a display screen are constantly being refreshed (re-illuminated) over again. When the user presses the light pen button, allowing the pen to sense light, the pixels being illuminated at the movement identifies on the screen.

JOYSTICK A joystick is an omni-directional level that is used to move the cursor on screen more rapidly then it can be moved with the directional arrow keys. It has a spherical ball at its lower end as well as its upper end. The lower spherical ball moves in a socket. Thus, a joystick can be moved right or left, forward or backward. The electronic circuitry inside the joystick detects and measures the displacement of the joystick from its central position. This information is sent to CPU. The pointer on the CRT screen moves according to the position of the joystick. The joystick is used extensively in video games, but is also used as an input device in computer aided design (CAD) systems.

Q.198 How does a bar code reader read the data? **(4)**

Ans:A bar code reader is a device used for reading (decoding) bar-coded data. It may be a hand-held scanner, or may be embedded in a stationary scanner. It scans a bar-code image and converts into and converts in into a alphanumeric value, which is then fed to the computer to which the bar-code reader is connected. Just as though the alphanumerical value had been typed on a keyboard.
A bar code reader uses a laser-beam scanning technology. The laser beam is stroked across the pattern of bars of a bar-code. Different bar codes having different patterns of bars reflects the beam in different ways, which is sensed by a light-sensitive detector. Reflected them transmitted to recognition logic circuits which convert in into an alphanumeric value.

Q.199 Explain how magnetic ink character reader is used in banks?
(4)

Ans: Magnetic-Ink Character Reader (MICR) is used by the banking industry for faster processing of the large volume of cheques being handled everyday by this industry. Banks that employ, MICR technology use a special type of cheque. The bank's identification code (name, branch etc), account number, and the cheque number are pre-printed (encoded) using characters from a special character set on all these cheques with a special ink that contains magnetizable particles of iron oxide before the cheques are given to the customers for use. When a customer presents a filled-in cheque at a bank, a bank employee manually enters (keys in) that amount written on the cheque in the lower right corner of the cheque using an MICR inscribers, which prints the amount with the magnetic ink. The data of the transaction is automatically recorded for all cheques processed that day. The cheque is then processed using an MICR reader sorter, which can recognize magnetic ink characters. The MICR reader-sorter reads the data on the cheques for distribution to other banks or for further processing. As the cheques enter the reading unit, they pass through a magnetic field, which causes the particles in the ink to become magnetized. Read heads then interpret these characters by examining their shapes. t\he sorter is basically used to sort the cheques into different pockets according to their identification code numbers.

Q.200 What is system software? Mention the most commonly used system software. Explain utility programs? **(2+2+4)**

Ans: System software is a set o one or more programs designed to control the operation and extend the processing capability of a computer system. Example: Operating System.

Some of the most commonly known types of system software are operating systems, programming languages translators, utility programs, performance monitoring software and communications software.
The various tasks performed by the utility programs are

- Hard disks or floppy disks are formatted.
- The files on a hard disk are recognized to conserve storage space.
- The backup of files stored in the hard disk is taken on to a tape or floppy disk.
- A particular file is located from a directory of hundreds of files.
- The amount of available storage space in the hard disk is found.
- The system is scanned for computer viruses.
- The stored records are sorted in a particular order based on some key fields.

Q.201 Write the steps in the execution of an instruction by the CPU. **(6)**

Ans:

- The address of next instruction is read from the program counter by the control unit. The instruction is read into the instruction register of the control unit.
- The operation part is sent to the decode and the address part is sent to the memory address register by the control unit.
- The instruction is interpreted and the control unit sends signal to the appropriate unit i.e., involved to carry out the task specified in the instruction. Ex. For any arithmetic or logic operation, the signal is sent to the CPU. The control unit ensures that the data corresponding to the address part is loaded in the suitable register in ALU, before the signal is sent to the ALU.
- When each instruction is executed, the address of the next instruction is loaded into the program counter and the above steps are repeated.

Q.202 Name the technique to reduce the speed mismatch between the slow I/O devices and the CPU. Explain how it is achieved? **(6)**

Ans: The speed mismatch between slow I/O device such as a printer and the CPU is normally of the order of few thousand. Thus, while a slow I/O device is inputting/outputting a few bytes to/from the main memory, the CPU can perform several thousands of operations. As a result, during reading/writing of data from/to data, from/to memory, the CPU is idle for a large percentage of I/O time. Spooling reduces the idle time of the CPU by placing all data that comes from the input device or goes to an output device on a magnetic disk. The primary reason for doing this to keep the program and data readily available to the fast and expensive CPU on a high speed I/O medium such as a disk. In most computer systems, special low-cost I/O processors are used for spooling the input data from a slow input device on to the disk or for outputting the spooled output data from the disk or for slow output device. These I/O processors function independent of the main processor (CPU). This enables the main high speed, expensive CPU to the fully devoted to main computing jobs. The process of spooling is transparent to the user programs. In general, spooling makes better use of both main memory and the CPU.

Q.203 Give the main features of Windows NT operating system. **(4)**

Ans: The main features of Windows NT operating system are:
(i) Unlike UNIX, its native interface is a GUI. The look and feel of Microsoft Windows NT's GUI is similar to that of Microsoft Windows GUI.
(ii) It supports multiprogramming and is also designed to take advantage of multiprocessing on systems having multiple processors.
(iii) It has in-built networking and communications feature so that any computer with Microsoft Windows NT can be (able) to work as a network client or server.
(iv) It provides strict system security.
(v) It has a rich set of tools for software development and system administration.
(vi) It can run Microsoft windows and many UNIX application directly.
(vii) It design is based on a microkernel so that it can be easily ported to many different types of machines, and its features can be easily enhanced by users.
(viii) It is a true 32-bit operating system in the sense that it can make full use of the 32-bit architectures of the processors an the memory and I/O bust to provide fast processing capability.

Q.204 Explain the difference between volatile and non volatile memory. Give an example of each type of memory. **(4)**

Ans: If the storage unit can retain the data store in it even when the power is turned off or interrupted, it is called NON-VOLATILE storage.
On the other hand, if the data stored are lost when the power is turned off or interrupted, it is called VOLATILE storage.
A non-volatile storage is desirable. The primary storage units are volatile and the secondary units are non-volatile.

Q.205 How does the bus width affect the overall speed of the computer system? Name and explain the three types of I/O buses. **(2+1+3)**

Ans: The bus width of a data bus is an important parameter that affects the overall speed of a computer system. This is because each wire of a bus can transfer one bit at a time. Hence, 8-bit bus (one having 8 parallel wires) can move 8 bits (one byte) at a time, a 16-bit bus can transfer two bytes and a 32-bit bus can transfer four bytes at a time. A wider data bus enables more bits of data to travel simultaneously resulting in faster exchange of data.

(i) DATA BUS: The data bus is used to transfer data between the CPU and I/O devices. A wider data bus will enable faster exchange of data between the CPU and I/O devices.
(ii) ADDRESS BUS: A computer system normally has multiple I/O devices like disk, tape, network etc simultaneously connected to it. Each I/O device has a unique identifier (or address) associated with it. The address bus is used to carry the address of the I/O device to be accessed by the CPU.

(ii) CONTROL BUS: The control bus is used to carry commands such as START, READ, WRITE, REWIND, TAPE, etc., from the CPU to I/O devices. It is also used to carry the status information of the I/O devices of the CPU.

Q.206 Explain the terms data cache and instruction cache. Name and explain the cache replacement policy. **(4+2)**

Ans: The instruction cache is used for storing program instructions and the Data cache is used for storing data. This allows faster identification of availability of accessed word in the cache memory and helps in further improving the processor speed.
REPLACEMENT POLICY: When a new block is to be fetched into the cache, another may have to be replaced to make room for the new block. The replacement policy decides which block to replace in such situation. It will be best to replace a block that is least likely to be needed again in the near future. Although it is impossible to identify such a block, a reasonably effective strategy is to replace the block that has been in the cache longest with no reference to it. This policy is referred to as the least recently used (LRU) algorithm. Hardware mechanisms are needed to identify the least recently used block.

Q.207 Differentiate between:
(i) CPU bound jobs and I/O bound jobs.
(ii) Tightly coupled and loosely coupled multiprocessing systems.
(iii) Internal fragmentation and External fragmentation.
(iv) Multiprocessing and Multiprogramming. **(4x4)**

Ans:
(i) **CPU bound jobs**
These jobs mostly perform numerical calculation with little I/O operations. They are so called because the heavily utilize the CPU during the course of their processing. Programs used for scientific and engineering computations usually fall in this category of jobs.
I/O bound jobs
These jobs normally input vast amount of data, perform very little computation, and output large amount of information. They are so called because during he course of their processing, the CPU utilization is very low and most of the time they perform I/O operations. Programs used for commercial data processing applications usually fall in this category of jobs.

(ii) **Tightly coupled multiprocessing systems**
In this system there is a single system wide primary memory that is shared by all the processors.
Loosely coupled multiprocessing systems
In this system, the processors do not share memory, and each process has its own local memory. In contrast to the tight coupled multiprocessing systems, the processors of loosely coupled systems can be located far from each other to cover a wider geographical area.

(iii) **Internal Fragmentation**
It is a situation when a process allocated more memory than its actual memory requirement and than its actual memory requirement and the additional memory allocated to the process remains

unutilized because it is neither used by the process to which it is allocated nor it can be allocated to any other process for use. Internal fragmentation may occur in the following situations:
(a) When the system uses fixed numbers of fixed-sized memory portions, any space in a partition that is in excess of the actual memory requirement of the process loaded into it is an internally fragmented memory space.
(b) When the system uses variable number of variable sized memory partitions, it may happen that there is a free block of size 200K and a new process arrives whose memory requirement is 199.5K. If we allocate exactly the requested size of memory, we will be left with a free-block of size 0.5K. Thus, the allocated memory may be slightly larger than the requested memory, and the additional memory that gets allocated to a process in this way is an internally fragmented memory space.

External Fragmentation

IT is a situation when enough total free memory space exists to satisfy the memory need of a process, but still the process cannot be loaded because the available free memory need of a process, but still the process cannot be loaded because the available free memory is continuous. For example, if the sizes of the free blocks FREE1, FREE2 and FREE3 and 400K, 100K and 100K respectively, then the total available free memory is 600K. Now if a new process arrives whose memory requirement is 500K, it cannot be loaded because 500K of contiguous free memory space is not available. Thus it is an example of external fragmentation. The amount of unusable memory area due to external fragmentation depends on the total size of the main memory and the average memory requirement of the process. However, statistical analysis indicates that as much as one-third of memory may be unusable due to external fragmentation.

(iv) **Multiprocessing**

Multiprocessing is the simultaneous execution of two or more processes by a computer system having more than one CPU. Multiprocessing makes it possible for the system to simultaneous work on several program segments of one or more programs.

Multiprogramming

Multiprogramming is the interleaved execution of two or more processes by a single CPU computer system. Multiprogramming involves executing a portion of one program, then a segment of another, etc., in brief consecutive time periods.

Q.208 Explain how the following features are carried out in the word processing Packages
(i) Entering mathematical equations.
(ii) Spell-checking.
(iii) Printing a document. **(6)**

Ans:

(i) On the insert menu, click object, and then click on the create New tab.
In the object type dialog box, click Microsoft Equation 3.
Click OK

(ii) Checking Spelling
On the tools menu, click options and then click the spelling and grammar tab.
Clear the check grammar with spelling check box

Click OK

(iii) Printing a document
On the File Menu, click print
In the print dialog box, enter the name of the printer paper range, no. of copies
Click OK

Q.209 Give the steps to insert a series of values in continuous locations like odd numbers 1,3,5,…

Ans:
- On the Edit Menu, click Fill and then click on series
- In the series dialog box, fill in the required data
- Click OK

Q.210 Give the two ways of renaming a work sheet in Microsoft Excel. **(4)**

Ans:
(i) - Right click on the sheet tab
- Select the Rename option

(ii) - Select the sheet option from the Format Menu
- From the sheet option select the Rename option

Q.211 Give the full form of the following abbreviations: **(3)**
i) RISC
ii) UNIVAC
iii) BASIC

Ans:
i) RISC : Reduced Instruction Set Computer.
ii) UNIVAC : Universal Automated Computer.
iii) BASIC : Beginners All purpose Symbolic Instruction Code

Q.212 Write about the hardware components used in the various generations of the computers. Give two examples of different computers used in different generations. **(5)**

Ans:

GENERATIONS	**HARDWARE COMPONENTS**	**EXAMPLES OF DIFFERENT COMPUTERS**
1ST Generation	Vacuum tubes, electromagnetic relay memories and punched cards for secondary storage were used.	ENIAC, EDVAC, ADSAC, UNIVAC I IBM 701
2nd Generation	Transistors, magnetic core	Honeywell 400, IBM

	memory, magnetic tapes and disks for secondary storage were used.	7030, CDC1604, UNIVAC LARC
3rd Generation	IC's with Large Scale Integration technologies, large magnetic core memory, more capacity magnetic tapes and disks for secondary storage were used.	IBM 360/370, PDP-8, PDP-11, CDC 6600
4th Generation	IC's with Very Large Scale Integration chips, microprocessors, semiconductor memories, large capacity hard disks as inbuilt storage and magnetic tape and floppy disks as portable storage media were used.	IBM OC and its clone, Apple II, TRS-80, VAX 9000, CRAY-1, CRAY-2, CRAY-X/MP
5th Generation	Ultra Large Scale Integration Chips Large capacity main memory, large capacity hard disks with RAID support, optical disks as portable read-only storage media were used.	IBM notebooks, Pentium PCs, SUN workstations, IBM SP/2, SGI Origin 2000, PARAM 10000

Q.213 A 6 platter hard disk has 600 tracks per surface. There are 10 sectors per track and 512 bytes per sector. What is the storage capacity of the disk? How many cylinders does the disk pack have? How many tracks are there per cylinder? **(4)**

Ans: Storage capacity of one surface = No. of tracks x No. of sectors x No. of per sector
No. of tracks = 600
No. of sectors =10
No. of bytes per sector = 512
No. of surfaces = 10 (upper and lower surfaces are not used)
Storage capacity of one surface = 600 x 10 x 512 = 3072000 bytes.
Storage capacity of disk pack = Storage capacity of one surface x No. of surfaces
= 3072000 x 10
= 30720000 bytes
= 29.29 Mb
= 30 MB
No. of cylinders = No. of tracks on each disk = 600
No. of tracks per cylinder = No. of usable surfaces on disk = 10

Q.214 Define the following terms with reference to a magnetic disk. **(4x1.5=6)**
i) Access time. ii) Seek Time.
iii) Latency Time. iv) Transfer rate.

Ans:i) Access Time: Access time is the interval between the instant a computer makes a request for transfer of data from a disk system to the primary storage and the instant this operation is completed. Access time depends on seek time, latency time and transfer rate.

ii) Seek Time: The time required to position the read/write head over the desired track is called the seek time. It depends upon the position of the access arms assembly when the read/write command is received.

iii) Latency Time: The time required to spin the desired sector under the read/write head is called the latency time. It is also known as the rotational delay time and depends on the distance of the desired sector from the initial position of the head on the specified track. It also depends on the rotational speed of the disk.
iv) Transfer Rate: Transfer rate is the rate at which data are read from or written to the disk. Transfer rate depends on the density of the stored data and rotational speed of the disk.

Q.215 Mention two advantages and two limitations of magnetic tapes for storage of digital information. **(6)**

Ans: The advantages of magnetic tapes for storage of digital information are

- Unlimited storage: The storage capacity of a magnetic tape is virtually unlimited because we can use as many tapes as required for recording our data.
- Ease of handling: Since tape reels are compact in size and lightweight, they are easily portable from one place to another and are much easier to handle and store.

The limitations of magnetic tapes for storage of digital information are:

- NO direct access: Magnetic tape is a sequential access media and hence data recorded on tape cannot be accessed directly. It is only retrieved serially that is if a data item is at the end of the tape, all the earlier part have to be read before accessing the required information. If access is required frequently, then magnetic tape is not a suitable storage media for such type of data. Too much machine time would be wasted in retrieving the data that is requested.
- Indirect interpretation: Data stored on a magnetic tape is in the form of tiny invisible magnetized and non-magnetized spots. Hence the contents of a tape cannot be interpreted and verified directly. Instead a print run must be made if the accuracy of tape data is questioned. This needs machines interpretation of the stored data.

Q.216 What is the voice reproduction system? How does it function? Give two applications. **(6)**

Ans:A Voice reproduction system produces an audio output from a set of pre recorded audio responses. The pre-recorded sounds are first converted into digital data and permanently stored in the computers memory. When the audio output has to be 'produced, the appropriate sound is selected from the pre-recorded sounds and the selected sound is converted back into analog form and is then sent to the speaker to produce the audio output.
Applications:

- Voice reproduction systems are used in automatic teller machines to provide step-by-step guidance to customers on how to use the ATM.
- Voice reproduction systems are used in talking alarm clocks and home appliances.

- Voice reproduction systems are used in automatic answering machines to give the vacancy status in a particular flight or train.

Q.217 Name the display technologies used by the flat panel monitors. Which is the most commonly used in flat panel monitors and why? **(4)**

Ans: Liquid-crystal display (LCD), electro-luminescent display (ELO) and gas-plasma display (GPO) are the display technologies used in flat panel monitors. Liquid-crystal display. (LCD) monitors are most commonly used. LCD monitors use a special kind of liquid crystals to display images on the screen which are normally transparent, but opaque when charged with electricity.

Q.218 Give one example of the following printers: **(4)**

i) Impact printers. ii) Non impact printers.
iii) Character printers. iv) Page printers.

Ans:
i) Impact printers: Dot matrix printers, chain printers, band printers and drum printers.
ii) Non-impact printers: Ink jet printers, laser printers.
iii) Character printers: Dot matrix printers. ink jet printers,
iv) Page printers: Laser printers.

Q.219 Give full form of the following: **(2)**

i) OCR
ii) MICR

Ans:
i) OCR: Optical Character Recognition.
ii) MICR: Magnetic-Ink Character Recognition.

Q.220 What is an image scanner? Explain the two types of image scanners. Give the limitations of image scanners. **(2+4+2)**

Ans: Image scanner is an input device, which translates paper documents into an electronic format that can be stored in a computer. The input documents can be typed text, pictures, graphics, or even hand written material. They are also known as optical scanners because optical technology is used for converting an image into electronic form.
The two types of image scanners are flat-bed scanners and hand-held scanners.
Flat-Bed Scanner: A flat-bed scanner consists of a box having a glass plate on its top and a lid that covers the glass plate. The document to be scanned is placed upside down on the glass plate. When the scanner is activated the light source which is situated below the glass plate moves horizontally from left to right. After scanning one line the light beam moves up a little and scans the next line. The process is repeated until the document is scanned completely.
Hand-Held Scanner: A hand-held scanner consists of a set of light emitting diodes encased in a small case that is convenient to hold in hand during operation. During scanning the

scanner is dragged slowly from one end of the document to the other with its light on. Dragging should be very steady over the document, otherwise the conversion into its bit map will not be correct. They are mostly used where high accuracy is not important and the volume of the scanned document is low. They are cheaper than flatbed scanners. .
The limitations of image scanners are:.

- Word processing cannot be done on the input documents, as the documents are stored as images instead of text.
- Storing the document as an image requires lot of storage space than compared to storing the document as a text.

Q.221 Explain with the help two examples, how software can be used as teaching and learning tool? **(4)**

Ans:

- Education softwares make the learning process interesting by incorporating audio-visual effects in various applications and by giving encouraging comments to the students.
- Education softwares help in conducting on line examinations, where the students can answer on line and know the results simultaneously. They also change the difficulty level of the questions based on the previous performance of the students.
- Education softwares help the individuals to learn foreign languages easily by incorporating both text and sound. The learner has the flexibility to hear the computer pronounce the word by selecting ttie word on the screen and also know the meaning of the phrase or the word.
- Education softwares act like a perfect tutor. They analyze the mindset of the user and guide them in answering the questions in the right manner. They help them tackle tough questions by giving them more exercises on the related areas.
- Reading material from encyclopedias and dictionaries is available on the CD- ROM. They help in searching for interesting facts also available in the Internet.

Q.222 What are the functions of a communication software? **(4)**

Ans: The functions of communication software are

- To ensure that connectivity is properly established between the source and the destination computer systems.
- The data is encoded in the right format at the source computer.
- Transfer of data from the source computer to the destination computer.
- The received data is decoded at the destination computer.

Q.223 Distinguish between single use and multi-user operating system with an example. **(4)**

Ans: Single-user operating system is an operating system, where only one user can use the computer at a time. They do not maintain log of the system usage. They also do not support security aspects, such as login and password procedures. Ex: MS-DOS operating system is a user friendly, hierarchical directory structure for organizing and storing files.

Multi-user operating system is an operating system, where more than one user can use the computer simultaneously. They maintain an extensive log of the system usage. They also support login and password procedures to restrict the access and to provide security to the system. Ex: UNIX operating system is a complex operating system containing more than 200 utility programs that can be used by the programmers to perform useful functions.

Q.224 What are the functions of the following? **(3)**

(i) Program control Register.
(ii) Instruction Register.
(iii) Decoder.

Ans:i) Program Control Register: Program control Register holds the address of the next instruction to be executed by the CPU. Once the CPU has taken the current instruction, the program control register is automatically incremented to point to the next instruction.

ii) Instruction Register: Instruction register holds the current instruction that is being executed. The address part of the instruction is separated and sent to the memory address register and the operation part is send to the control section.

iii) Decoder: The decoder has the necessary control circuitry to decode and interpret the meaning of each and every instruction supported by the CPU.

Q.225 Explain the operation of a cache memory. **(3)**

Ans: Cache memory is a small, high speed buffer between the processor and the main memory. It temporarily stores the active data and instructions during processing. During processing, the CPU attempts to read from the cache memory and if found the data/instruction. IS- transferred to the CPU. If not found, a block of main memory containing the requested word is transferred to the cache memory and then to the CPU.

Q.226 Name and explain the different types of ROM's. **(6)**

Ans: There are two types of ROMs - manufacture-programmed and user-programmed.
PROM (Programmable Read-Only Memory): User - programmed ROM is also known as PROM as the user can program it. PROM chips cannot be reprogrammed.
EPROM (Erasable Programmable Read-Only Memory): The data on EPROM chips can be erased and reprogrammed to store new information. EPROM chips are of two types - UVEPROM and EEPROM.
UVEPROM (Ultra Violet Erasable Programmable Read-Only Memory): The stored information is erased by exposing the chip to ultraviolet light. .
EEPROM (Electrically Erasable Programmable Read-Only Memory): The stored information is erased by using high voltage electric pulses. It is also known as flash memory.

Q.227 In DOS, what do these commands do? **(4)**

i) COPY ii) CHKDSK iii) PROMPT iv) DISKCOMP

Ans:
i) COPY: Copy command is used to make a duplicate copy of the original file, It is an internal command.

ii) CHKDSK: CHKDSK command is used to check the specified drive for error. It is an external command.

iii) PROMPT: PROMPT command is used to set a new DOS prompt instead of the usual C> or A>.
iv) DISKCOMP: DISKCOMP command compares contents of the two disks, sector by sector. It locates errors and displays them. It is an external command.

Q.228 Explain about the different types of memory buses interconnecting the memory and the CPU. **(6)**

Ans: The three types of memory buses interconnecting the memory and the CPU are the data bus, address bus and the control bus.
i) Data bus: Data bus is used to transfer data between the CPU and the memory. It is a bi-directional bus. Bus width affects the overall speed of the computer. A wider data bus enables more bits of data transfer.

ii) Address bus: Address bus is used to carry the address of the memory location whenever data is to be transferred to or from memory. It is a unidirectional bus. The width of the address bus must be equal to the number of bits in the memory address register (MAR). The width of. address bus also determines the maximum number of memory locations a computer can have.

iii) Control bus: The CPU sends the control signals to the memory that specify whether the' data is to be read from or written to the specified location. It is a bi-directional bus.

Q.229 Give the major features of windows 98. **(6)**

Ans: The major features of windows 98 are:

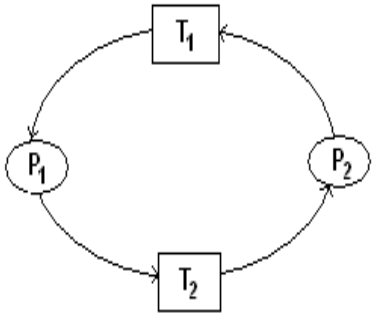
- Faster than Windows 95: The maintenance wizard of windows 98 checks the hard disk for problems and frees its space so that the programs run faster without any additional hardware.
- Web Integration: The interactive content of the Internet is combined with the power of computer and improved web features are provided.
- Improved Reliability: New utilities, wizards and resources are introduced which help the computer in running smoothly and efficiently. System file checker of Windows 98 restores the critical files if they are changed. Scan disk runs automatically if the computer is not shut down properly.
- Multiple Display: Several monitors can be used to run different programs on separate monitors.
- Power management: The computer can be started in few seconds i.e. the start up
- time is improved. The programs can be restored in their last saved positions.

- More Entertaining: New features like enhanced television, video playback and web TV for windows makes computer more entertaining.

Q.230 When are the two processes said to be in a deadlocked state. Explain with the help of an example. **(8)**

Ans: When competing processes prevent their mutual progress even though no single one requests more sources than are available, then the situation is called deadlock and the processes involved are said to be in a deadlock state. It may happen that some of the processes that entered the waiting state (because the requested' resources were not available at the time of request) will never again change state; because the resources they have requested were held up by other waiting processes.
Consider a system having two tape-drives T1 and T2 and the resource allocation strategy is such that a requested resource is immediately allocated to the requesting process if the resource is free. Consider two concurrent processes P1 and P2 requesting the tape drives in the following order:

1. P1 requests for one tape drive and the system allocates T1 to it.
2. P2 requests for one tape drive and the system allocates T2 to it.
3. P1 requests for one more tape drive and enters a waiting state because no tape drive is available.
4. P2 requests for one more tape drive and enters a waiting state because no tape drive is available.



Example of a deadlock situation involving processes P1 and P2 and resources T1 and T2
From now on, P1 and P2 will wait for each other indefinitely, since P1 will not release T1 until it gets T2 to carry out its designated task, that is, not until P2 has released T2, whereas P2 will not release T2 until it gets T1. Therefore, the two processes are in a state of deadlock. The requests made by the two processes are legal because each is requesting for only two tape drives, which is equal to the total number of tape drives available in the system. However, the deadlock problem occurs because the total request of both processes exceeds the total number of units for the tape drive and the resource allocation policy is such that it immediately allocates a resource on request if the source is free.

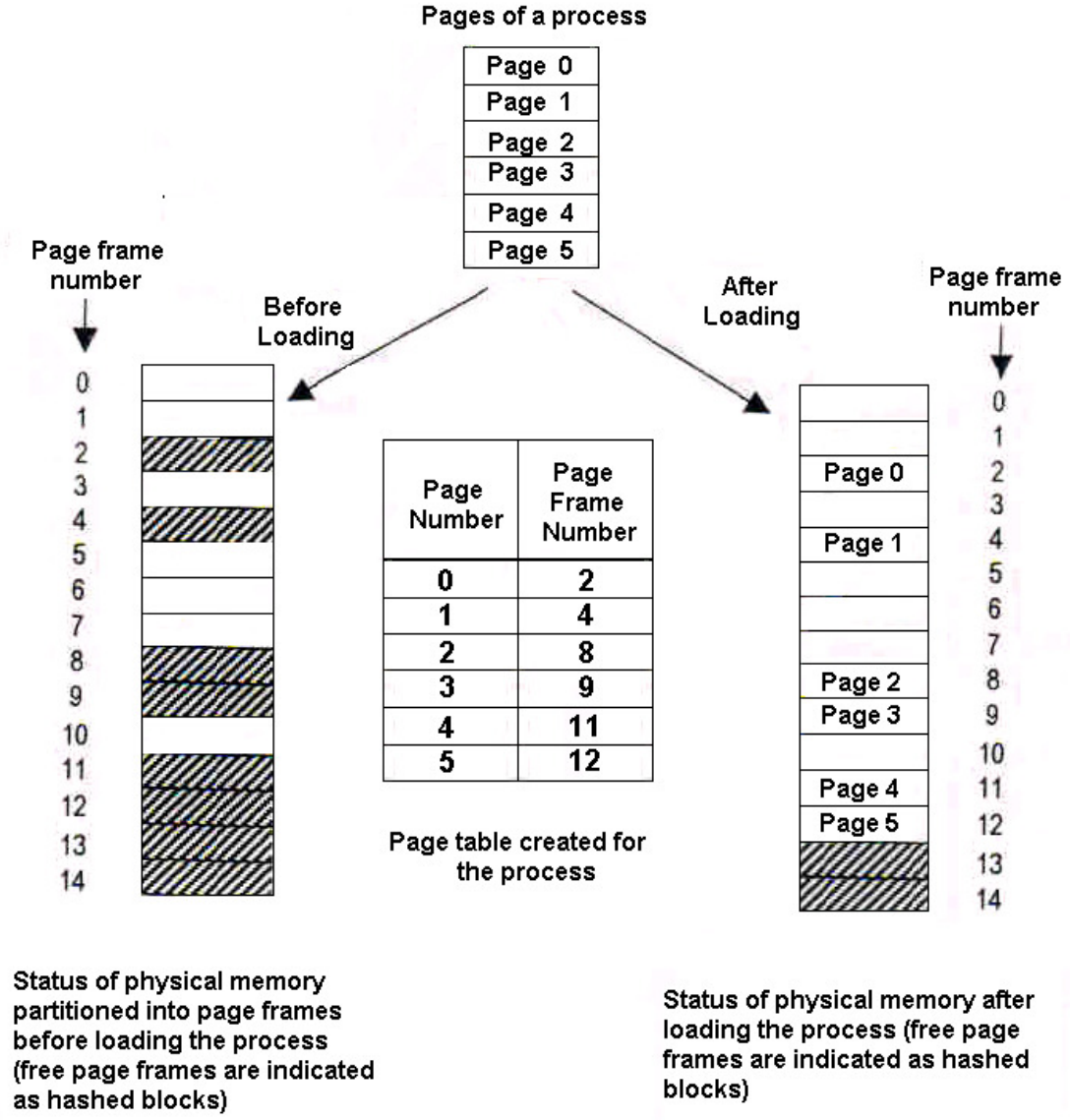
Q.231 What is paging? Explain paging mechanism with the help of an example. **(8)**

Ans: Paging is a memory management scheme, which allows the process's memory to be non-contiguous, thus allowing process to be allocated physical memory wherever free memory blocks are available.

In paging the physical memory of the computer system is partitioned into fixed-sized blocks called frames. The total memory requirement of a process including the data and instructions is partitioned into blocks of same size called pages. When a process is to be loaded into memory, the pages are loaded into the free page frames wherever the page frames are available. A mapping table called the page table is maintained by the operating system to keep track of which page of the process is loaded into which page frame of the memory.

Consider a process, which is partitioned into 6 pages, Page 0 to page 5. The physical memory is partitioned into 15 page frames, 0-14 whose size is the same as the size of the pages of the process. The paging mechanism is illustrated above.

During paging, every address generated by the CPU is divided into two parts, page, number and page offset. The corresponding page frame number from the page number is obtained from the page table. To define the physical memory address that is sent to the memory unit, the page offset is added to the base address of the page frame number.



Q.232 It is required to print the current date at the right-bottom and a heading at the top-center of the page. Give steps to achieve this. **(4)**

Ans: To print a date at the right-bottom, the steps are:

- On the insert menu, click Page Number option.
- In the Position box, specify the print the page numbers in the footer at the bottom of the page and alignment to the right.

To insert a header at the top-center the steps are:

- On the View menu, click Header and Footer.
- Insert the heading text and center-align the text.

Q.233 Explain the word wrap feature in MSWORD. **(4)**

Ans: The word-wrap feature automatically moves the last typed word to the next line if it goes past the right margin. That is if the last word cannot fit completely within the defined margins, the word is moved to the next line.

Q.234 A worksheet contains the following details. Give the steps to insert a bar chart of the Amount in all years. **(4)**

	A	B	C	D	E	F
Sales	1981	1982	1983	1984	1985	1986
Amount In crores	50	54	62	78	75	79

Ans: Enter the data into the worksheet.
Select the **Chart option** from the **Insert menu**
Select the **data range** on the worksheet to plot.
Select the **chart option** regarding the type and the placement.
Select the **location** where to place the chart.

Q.235 Explain the functions in a spreadsheet package. **(4)**
i) MIN ii) SUM

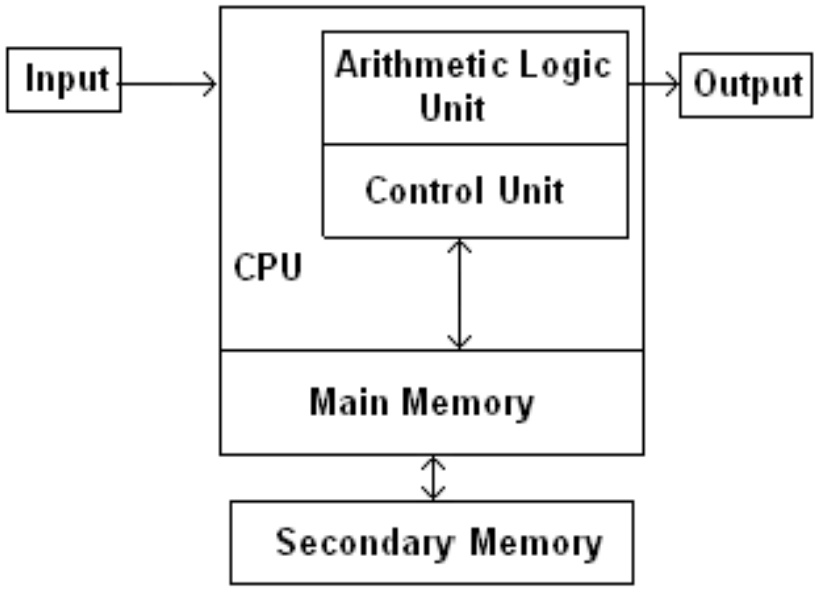
Ans:

i) MIN: This function returns the minimum value among the argument passed. It cannot take more than 30 arguments. Ex. MIN(B2..B9) – finds the minimum of the 8 number, B2 to B9 cells.

ii) SUM: This function calculates the sum of the numbers in a list. Ex. SUM(A1..A4, B1..B4) – sums the values in the ranges, A1..A4 and B1..B4.

Q.236 Draw a block diagram to illustrate the basic organisation of a computer system and explain the functions of the various units. (7)

Ans:



CPU: the main processing unit with abilities to perform arithmetic and logic operations + controlling various tasks and resources.
Main memory: temporarily storing data required by CPU to perform required operations at a given point of time, volatile.
Secondary memory: storing and achieving data, non-volatile
I/O: Managing input/output

Q.237 What do you understand by memory hierarchy? Name the general classes of storage media that might make up a memory hierarchy. **(7)**

Ans:
There is a trade-off to be made while designing the systems architecture in the context of computer memory, among the three key characteristics of memory, namely cost, capacity and access time. The plan is not to rely on a single memory component to technology, but to employ a memory hierarchy, as shown below. Going down the hierarchy, the following thing occurs:

- Decreasing cost per bit
- Increasing Capacity
- Increasing access time i.e. slower memory
- Decreasing frequency of access of the memory by the CPU

Following is a traditional memory hierarchy.

REGISTERS
CACHE
MAIN MEMORY
MAGNETIC DISKS
MAGNETIC TAPE

Q.238 Draw the truth table for the following Boolean expression $F(A,B,C) = A + B \cdot C + A \cdot B$ **(7)**

Ans:

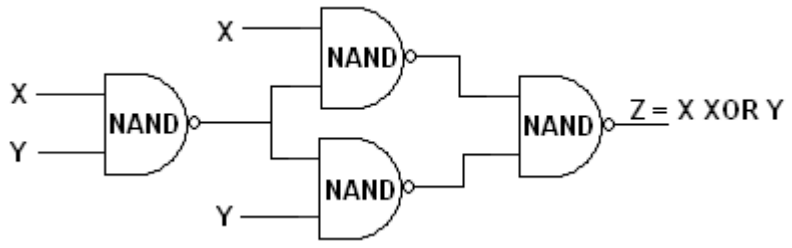
A	B	C	A'	B.C	A.B'	F(A,B,C)
0	0	0	1	0	0	1
0	0	1	1	0	0	1
0	1	0	1	0	0	1
0	1	1	1	1	0	1
1	0	0	0	0	1	1
1	0	1	0	0	1	1
1	1	0	0	0	0	0
1	1	1	0	1	0	1

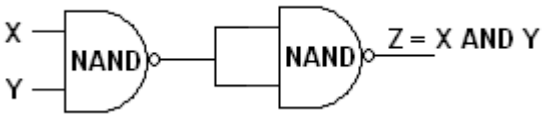
Q.239 What is the difference between a sequential circuit and a combinatorial circuit? What is meant by state of a circuit? **(7)**

Ans :
Logic circuits without feedback from output to the input, constructed from a functionally complete gate set, are combinatorial. Logic circuits that contain no memory (ability to store information) are combinatorial. Those, which contain memory, including flip-flops, are said to be sequential.
The logical structure/situation of a circuit at any given point of time is known as the state of the circuit.

Q.240 What are universal gates? Realize the following gates with the help of universal gates.
(i) XOR (ii) AND

Ans:
NAND and NOR are universal gates because all other gates can be constructed with these two gates.





Q.241 What are the advantages of EBCDIC code?

Ans :EXTENDED BINARY CODED DECIMAL INTERCHANGE CODE (EBCDIC)
Using an 8-bit code, it is possible to represent 256 different characters or bit combinations. This provides a unique code for each decimal value 0 through 9 (for a total of 10), each uppercase and lowercase letter (for a total of 52), and for a variety of special characters. In addition to four numeric bits, four zone bit positions are used in 8-bit code. Each group of the eight bits makes up one alphabetic, numeric, or special character and is called a byte.
In EBCDIC, the bit pattern 1100 is the zone combination used for the alphabetic characters A through I, 11 0 1 is used for the characters J through R, and 111 0 is the zone combination used for characters S through Z. The bit pattern 1111 is the zone combination used when representing decimal digits. For example, the code 11000001 is equivalent to the letter A; the code 11110001 is equivalent to the decimal digit 1. Other zone combinations are used when forming special characters. Not all of the 256 combinations of 8-bit code have been assigned characters.
Since one numeric character can be represented and stored using only four bits (8-4-2-1), using an 8-bit code allows the representation of two numeric characters (decimal digits). Representing two numeric characters in one byte (eight bits) is referred to as packing or packed data. By packing data (numeric
characters only) in this way, it allows us to conserve the amount of storage space required, and at the same time, increases processing speed.

Q.242 Explain the rules of BCD addition.

Ans : The rules for BCD addition: First, the BCD digits are added as if they were two 4-bit binary numbers. When the binary sum is less than or equal to 1001 (decimal 9), the corresponding BCD digit sum is correct. However, when the binary sum is greater than 1001, we obtain an invalid BCD result. The addition of binary 0110 (decimal 6) to the binary sum converts it to the correct BCD representation and also produces an output carry as required.

Q.243 What do you understand by direct and indirect addressing? Explain with the help of examples. **(7)**

Ans : In direct addressing, the address field contains the effective address of the operand: EA = A. So it requires only one memory reference and no calculations. The address space is very limited.
In indirect addressing, the address field refer to the address of a word in the memory which in turn contains a full-length address of the operand.
EA = (A) ( ): contents of

Q.244 What are I/O devices necessary for a computer system. Why are I/O devices very slow compared to the speed of primary storage and CPU. **(7)**

Ans : I/O devices are the interfaces of the computer / CPU to the outside world, be it other systems or human users.

I/O devices are very slow compared to the primary memory or CPU because the data transfer rate of these peripherals is often much slower than that of the memory or CPU, so it is impractical to use the high-speed system bus to communicate directly to a peripheral.
SRAM is static RAM made of semi conductor devices i.e. traditional flip-flop logic-gate configurations, as there is no loss/gradual decay of charge, they do not need any refreshing circuits. DRAMs are dynamic RAMs made of capacitors, which have a tendency to gradual discharge, so they need a refreshing circuit for periodic charge refreshing to maintain data storage.

Q.245 What are the differences between SRAM and DRAM? Explain in brief why a refreshing circuit is needed for DRAM. **(5)**

Ans : SRAM is static RAM made of semi conductor devices i.e. traditional flip-flop logic-gate configurations, as there is no loss/gradual decay of charge, they do not need any refreshing circuits. DRAMs are dynamic RAMs made of capacitors, which have a tendency to gradual discharge, so they need a refreshing circuit for periodic charge refreshing to maintain data storage.

Q.246 Explain the principle of duality in Boolean Algebra. How is it useful? **(5)**

Ans : Duals are opposites or mirror images of original operators or constants. Switching logic using binary operators exhibits duality between the AND and OR operators. The dual of AND is OR and vice versa.
The principles of duality and complementing provide features that are used in the development of DeMorgan's theorems.

Q.247 Give the dual of the following Boolean expressions:
(i) A + B
(ii) A. B + A . B **(4)**

Ans : Use the principles as given above i.e. The dual of AND is OR and vice versa.

Q.248 Compare the characteristics. of impact and non-impact printers with examples. What are digitizers?

Ans : Impact printers print the document character by character, with a writing head striking on an inkcoated ribbon to get the character printed. Non-impact prints do not print by any such physical impact, rather it prints block by block or line by line with ink or laser jet. Examples of impact printers are daisy wheel and dot matrix printers, whereas non-impact printers are laser or ink-jet printers.
A digitizer is an input device used for drawing free-hand images and graphics. This is a pad with a grid of sensor's wire. A pen is moved on to the grid and every movement of the pen on the gird is captured. This pen is also known as stylus.

Q.249 What are the basic functions of an operating system?
Ans : Functions include:

- controlling the user interface
- controlling tasks in progress
- controlling access to data
- allocating resources
- memory management
- process scheduling

Q.250 How does DOS manage the file system?

Ans : DOS manages the file system with FAT or File Allocation Table. It is a table that the operating system uses to locate files on a disk. Due to fragmentation, a file may be divided into many sections that are scattered around the disk. The FAT keeps track of all these pieces.
In DOS systems, FATs are stored just after the boot sector.

Q.251 What are serial and parallel devices? Explain with examples

Ans : Serial devices take data bit by bit, parallel devices can take a number of bits at a time through a parallel port.

Q.252 Explain how registers are organized in a processor. Explain the functions of the following registers
(i) Memory Address Register (MAR)
(ii) Memory Buffer Register (MBR)
(iii) Program Counter (PC)
(iv) Accumulator (A)
(v) Instruction Register (IR)
(vi) Input Output Register (I/O - R)

Ans :
(i) Memory Address Register (MAR) specifies the address in memory of the word to be written from or read into MBR.
(ii) Memory Buffer Register (MBR) contains a word to be stored in memory, or is used to receive a word from memory
(iii) Program Counter (PC) contains the address of the next instruction-pair to be fetched from memory
(iv) Accumulator (A) temporarily hold operands and results of ALU operations
(v) Instruction Register (IR) contains the op code instruction being executed
(vi) Input Output Register (I/O – R) temporarily hold I/O buffer data

Q.253 Explain the following terms:
(i) Virtual memory
(ii) Paging
(iii) Plotter

Ans :
(i) Virtual Memory: This is system memory that is simulated by the hard drive. When all the RAM is being used (for example if there are many programs open at the same time) the computer will swap data to the hard drive and back to give the impression that there is slightly more memory.

(ii) Paging: A method of managing virtual memory. When a requested page is not found in main memory, an interrupt occurs. The paging device machine then transfers the requested inactive page to memory. High rates of page swapping can degrade performance.
(iii) Plotter: A mechanical device which produces printout using vector or co-ordinate graphics often by using a pen moved about on rails.

Q.254 Explain the type of files the following extensions depict-.COM, .BAT, .DAT, .EXE, .SYS. In what situations does one need to edit the CONFIG.SYS file?

Ans:
.COM: command files e.g. command.com
.BAT : Batch files
.DAT : data files
.EXE : executable files
.SYS : system files e.g. config.sys
when a systems configuration details need to be changed e.g. the user wants a program to start off automatically when the system starts, or a software installation needs special configuration settings to run properly, config.sys
needs to be changed. .

Q.255 What is meant by booting? Explain the basic booting process.

Ans : During this process the computer will perform a self-diagnostic, reporting any errors it may encounter. Much can be learned about the state of the computer by paying close attention to the series of messages passing across the display during startup.
The basic booting process has the following sequence of operations:
1. Power-On Self Test (POST)
2. System Initialization, CMOS and BIOS Check
   After reading in the CMOS settings the boot sequence will continue with:
   PCI Initialization - If you have a PCI bus the system will now initialize the cards on the bus.
   Configuration Display - The BIOS now tests and displays the system configuration, including the status of:
   CPU
   Floppy drives
   Mouse and keyboard ports
   Serial ports
   Parallel ports
   Cache memory information: common sizes for cache memory are 64 KB, 128 KB, 256 KB, or 512 KB.

3. Loading the Disk Operating System (DOS)
4. Configuring the System with CONFIG.SYS
5. Running Programs at Startup with AUTOEXEC.BAT
6. Running Windows

Q.256 What is the difference between multi-tasking and multi-programming?

Ans : Multiprogramming is a method of running several different programs in a computer apparently at the same time.
Usually on a mainframe - the computer has a number of programs loaded into memory and the operating system switches quickly between them, processing a little bit of each one in turn. The high speed of the processor makes it seem like more than one program is being run at the same time.
Multi-tasking: Two or more programs actually run at the same time. Windows 95 and OS/2 support multitasking. Windows 3.1 supports "task switching" but not multi-tasking. Extended memory is divided up into "virtual machines" that share time on a single processor. With multi-tasking, a computer could be receiving communication via modem in the background while running Excel in the foreground.

Q.257 Explain the following terms with respect to a memory
(i) Access time
(ii) Storage capacity
(iii) Cost per bit of storage
(iv) Word length.

Ans :
(i) Access time: for random-access memory, this is the time it takes to perform a read or write operation, that is, the time from the instant that an address is presented to the memory to the instant that data have been stored or made available for use. For non-random access memory, it is the time it takes to position the read-write mechanism at the desired location.

(ii) Storage capacity: the total number of bits/bytes/words that a memory component can maximally hold, i.e. 16 MB RAM means the RAM chip can hold at the most 16 megabytes of data.

(iii) Cost per bit of storage: the cost of storing one bit of data in a particular type of memory component! technology, i.e. in pure semi-conductor based cache memory, the price of such memory component is very high, so the cost per bit of storage is also very high.

(iv) Word length: the 'natural' unit of memory organisation, can be 16/32/64 bits, length of the word is typically equal to the number of bits used to represent a number and to the instruction length.

Q.258 Explain how cache memory is used to increase the speed of processing of a computer.

Ans : Cache is the fastest memory component available providing a large memory size at the

price of less expensive types of semi conductor memories. It actually speeds up the processing of a computer by acting as a buffer reducing the delay in transfer of data in between the CPU and the slower and larger main memory.

Q.259 Write short notes on the following:
(i) Utility and availability of the chart facilities in Spread sheets.
(ii) Image editing software packages.
(iii) Features of groupware.
(iv) Salient features of Word Processing package.

Ans :(i) Most spreadsheet packages have chart drawing facilities with various analytical options for better visual representation and easier interpretation of spreadsheet data.
A chart consists of two axes. The horizontal (X) axis is called the category axis. The vertical (Y) axis is called the value axis. The chart contains one or more values. Each value is a list of number to be charted. A value entry contains two lists:

- the categories list, represented in the X axis.
- the series list, represented on the Y axis.

The following features are supported: Vertical Bar Chart

- Line Chart
- Pie Chart
- Area Chart
- Stacked Vertical Bar Chart Vertical Bar 3D Chart
- Customizable size, colors, title and legend
- Output HTML image map and tool tips

(ii) Image editing software packages:
There is a huge variety of software for manipulating images in various ways. Much of this software can be grouped under the heading image processing software. Another very important category is what we call image editing software. This group includes painting programs, graphic art packages and so on. They are often useful in conjunction with image processing software packages, insituations where direct immediate interaction with an image is the easiest way of achieving something. For instance, if a region of an image is to be masked out for subsequent image processing, it may be easiest to create the mask using an art package by directly drawing on top of the original image. Art packages also often allow the user to move sections of the images around and brighten or darken selected regions interactively. Few dedicated image processing packages offer the same flexibility and ease of use in this respect.

(iii) Features of groupware:
Groupwares are programs that help people work together collectively while located remotely from each other. Groupware services can include the sharing of calendars, collective writing, e-mail handling, shared database access, electronic meetings with each person able to see and display information to others, and other activities.
It was a term coined by marketers around 1995 to mean "software that facilitates group work," never emerged as a well-defined software category. Today the term is used less and tends to be narrowly identified with three products: Lotus Notes, Microsoft Exchange and

Novel GroupWise.

(iv) Parallel processing:
A computing method that can only be performed on systems containing two or more processors operating simultaneously. Parallel processing uses several processors, all working on different aspects of the same program at the same time, in order to share the computational load.

Q.260 Differentiate between interpreter and compiler. **(3)**

Ans: Interpreter: The language processor that translates (converts) each statement of source program into machine code and executes it immediately before to translate the next statement is called Interpreter. If there is an error in the statement the interpreter terminates its translating process at that statement and displays an error message. The GWBASIC is an example of interpreter.
Compiler: The language processor that translates the complete source program as a whole in machine code before execution is called compiler. The C and C++ compilers are best examples of compilers.
The program translated into machine code is called the object program. The source code is translated to object code successfully if it is free of errors. If there are any errors in the source code, the compiler specifies the errors at the end of compilation. The errors must be removed before the compiler can successfully compile the source code. The object program can be executed a number of times without translating it again.
The Interpreter differs from compiler that translates the entire source program into object program before execution.
The main advantage of Interpreter is that it makes easy to trace out and correct errors in the source program.

Q.261 Discuss how a laser printer works. **(4)**

Ans: Laser printer works as follows:
Charging the drum: When the printer receives a command from the computer's operating system to begin the print process, the photosensitive drum is negatively charged.
Exposing the drum: After the drum is charged, the laser flashes like a strobe light, the beam reflecting from the mirror to the negatively charged drum. As the drum spins, the laser creates the outline of the image on the drum by building a pattern of strongly charged and less-than-strongly-charged areas of negative voltage.
Developing the image: After the image is electro magnetically "written" to the drum, the toner must be applied to the print drum so the image can be transferred to paper. Adjacent to the print drum and the reservoir of toner is a smaller drum called the *developer.* The developer attracts toner powder to itself as it rotates between the toner and the print drum. When the print drum has toner on only the areas of slight negative charge, then the image is ready for transfer to paper.
Transferring the image: Now the registration rollers feed the paper into the printer and over the transfer corona wire, which gives the paper a strong positive charge. As the paper moves beneath the print drum, the weakly charged negative toner particles are strongly

attracted to the positively charged paper. So the image moves onto the paper. The paper continues to move through the assembly, passing over a static eliminator strip that removes all electromagnetic charge from the paper.
Fusing the image: As the paper leaves the print drum, all that holds the toner on the paper is a slight positive charge and a bit of gravity. The *juser* assembly finishes the printing by melting the toner into the paper as the sheet moves between a heated, Teflon coated roller and a rubber roller. The paper is then ejected from the print device.

Q 262 Write the trouble shooting steps if the computer is unable to load DOS. **(2)**

Ans.
1. If the computer is not coming to a MS-DOS prompt, reboot the computer and as the computer is booting, press the F5 key when you see the message "Starting MS-DOS" or the MS-DOS version. This will load the default standard MS-DOS.
2. If you successfully get to a MS-DOS prompt and would like to prevent the computer from loading the program that is preventing you from getting to a MS-DOS prompt, or if you would like to fix possible error messages you may be receiving when booting the computer, edit the autoexec.bat and/or the config.sys files.

Q.263 Differentiate between warm boot and cold boot. **(3)**

Ans: Cold Boot: It means starting from 'off state' or 'power off and than on by using the power button. You perform a cold boot every time you turn on the power switch of your computer. To "boot" the computer means to start it up and reset the memory and BIOS.
Warm Boot: It means restarting the computer using Alt+Ctrl+Del combination or restart command from the startup menu.

Q.264 Give the Principal working of Plotter

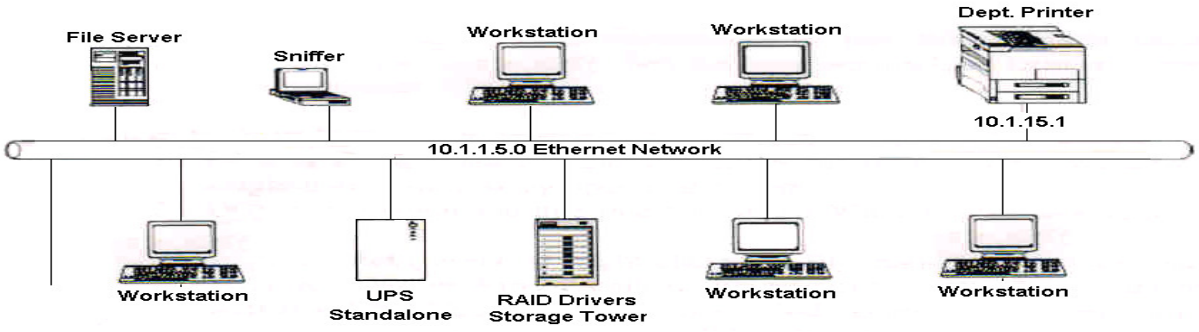
Ans: Plotter: A plotter is an output peripheral device used with a computer, which can be linkened to a printer. However instead of printing text or images, a plotter is more usually used to draw up technical plans and blueprints. Instead of the print cartridge found in a printer, a plotter commonly uses a pen.
With a plotter, a method of moving the paper and penholder are used to allow the pen to be in precisely the right point at the right time. However the order of print is different from a printer. The pen does not start at the top right moving to the left and then down a line. The plotter takes the whole image data, and then calculates a path for the pen. While the path for the pen may look complicated while the pen is in motion, In fact the plotter has calculated the shortest route for the pen, which involves the least amount of crossed lines as possible. Plotters are used to produce designs and graphs.

There are basically two types:
a) Drum Plotters: It is having a drum in which the paper is fixed and it is moved to and fro which makes it in vertical motions. The pens are clamped in the holder with different set of

colors and they are moved in horizontal directions. When the both move simultaneously the graphs or design is produced on the paper.
b) Flatbed Plotters: In this type of plotters the paper is not fixed and set of colored pens are in motion. The computer controls these.

Q.265 Write a short note on computer networking concepts. Show with the help of a diagram how will you connect five computers on a LAN. **(8)**

Ans: By computer network we mean an interconnected set of autonomous computers. The term autonomous implies that the computers can function independent of others. However, these computers can exchange information with each other through the communication network system.
Computer Network can be classified on the basis of their scale as: Local Area (LAN), Metropolitan Area Network (MAN) and Wide Area Networks (WAN).
Local Area Network (LAN)
LAN is usually privately owned and links the devices in a single office, building or campus of up to few kilometers in size. These are used to share resources (may be hardware or software resources) and to exchange information. LANs are distinguished from other kinds of networks by three categories: their size, transmission technology and topology.
LAN typically used transmission technology consisting of single cable to which all machines are connected. Traditional LANs run at speeds of 10 to 100 Mbps (but now much higher speeds can be achieved). The most common LAN topologies are bus, ring and star.
Metropolitan Area Networks (MAN)
MAN is designed to extend over the entire city. It may be a single network as a cable TV network or it may be means of connecting a number of LANs into a larger network so that resources may be shared. For example, a company can use a MAN to connect the
LANs in all its offices in a city. MAN is wholly owned and operated by a private company or may be a service provided by a public company.
Wide Area Network (WAN)
WAN provides long-distance transmission of data, voice, image and information over large geographical areas that may comprise a country, continent or even the whole world.
In contrast to LANs, WANs may utilize public, leased or private communication devices, usually in combinations, and can therefore span an unlimited number of miles. A WAN that is wholly owned and used by a single company is often referred to as *enterprise network.*
We can connect five computers on a LAN as shown below:



Q.266 What are the characteristics of following types of operating systems?

(i) Time-sharing
(ii) Client-server
(iii) Multi-tasking
(iv) Multi-programming **(8)**

Ans: (i) **Time sharing:** In time sharing several systems (called as dumb servers having only computer peripherals) are attached to a single dedicated server having own CPU. Dumb servers share the CPU of dedicated server, as they don't have their own CPUs. Every action or command in Time-sharing operating systems is so short that very short span of CPU time is assigned for each user and thus the users at dumb systems have an impression that they have their own CPU though the fact is they share the CPU of dedicated server. Such short periods of time is called as time-slots or time-slices or time-quantum.

(ii) **Client-server**: The operating system has a multi-user session manager to enable multiple client-server sessions on the server and a multi-user stack protocol manager to manage one or more protocol stacks used in communicating with the clients. When a user connects to the server via a first client, the stack protocol manager assigns a first protocol stack to this first client-server connection and the session manager creates a first session for the user. When the user subsequently reconnects to the server using a second client that is different from the first client, the stack manager assigns a second protocol stack to a second client-server connection and the session begins creating a second session for the user. During this latter process, however, the session manager recognizes that the user is affiliated with the first session. The session manager adapts the first session to conform to the system configuration of the second client. The session manager then re-associates the second protocol stack with the reconfigured first session so that the user is returned to his/her original session, even though they logged on from a different client. .

(iii) **Multi-tasking**: Multitasking is a method of running several jobs at a time, now jobs can be either in the form of programs, processes, threads, user performing multi tasks at a time in single pc only. The main idea behind to do so is better CPU utilization. Several jobs are kept in the memory at a time such that when CPU is busy in execution of one job or task then as switches over to other job and make it ready to get its next turn for execution by CPU.

(iv) **Multi-programming**: When multitasking is just talking about executing multiple programs concurrently then the term multitasking term is referred as multiprogramming.

Q.267 Compare DOS and Windows as two operating systems. **(8)**

Ans: DOS Vs Windows:

- DOS is a single-user, single task operating system while Windows is a single-user, multitasking operating system.
- DOS has a command-line interface while Windows native interface is a GUI.
- DOS restricts users to eight-character file name with three-character extensions. While Windows allows file names to contain 255 characters as well as some punctuation marks like periods, commas, and semicolon.

- DOS is a simple operating system while Windows is a complete operating environment. All programs of Windows confirm to a standard way of working.
- DOS supports 2 GB of maximum partition size, while Windows supports 2 TB or more.
- DOS uses FAT 16 file system, while Windows uses FAT 32.
- Server administration is not possible in DOS.
- DOS does not support networking, Windows does.

Q.268 What are real-time systems? What are the differences between application software and system software? **(7)**

Ans: A real-time system (RT) is a system that satisfies the requirement of producing the desired results before a specified deadline. For a Real time system, the correct functioning of the system depends on the results produced and the time at which they are produced. A real-time operating system has well-defined, fixed time constraints. Processing must be done within the defined constraints, or the system will fail. A real time system is considered to function correctly only if it returns the correct result within any time constraints. So the following features are desirable in a real-time operating system:

- Multi-tasking within an operation
- Ability to define the priorities of tasks
- Priority driven or deadline oriented scheduling
- Programmer defined interrupts

There are two categories of Real time systems:
i) Soft RT system: A system whose operation is degraded if results are not preceded according to specified timing requirements.
ii) Hard RT system: A system whose operation is incorrect if results are not produced according to the timing constraints. Catastrophic results will happen then.
Application software and system software:
Application software are designed to perform specific data processing or computational tasks for the user. These programs are specifically designed to meet end-user requirements. For example: spreadsheets, word processors, media players and database applications etc.
While a **System software** is an essential part of computer operations. A System software is any computer software which manages and controls computer hardware so that application software can perform a task. Operating systems, such as Microsoft Windows, Mac OS X or Linux, are prominent examples of system software. The function of the systems software is to manage the resources of the computer, automate its operation and facilitate program development.

Q.269 Differentiate between an Optical Mouse and Mechanical Mouse. **(4)**

Ans: The mechanical mouse is the one that has a trackball on its lower side. The sensors inside this mouse are mechanical. On the contrary, in an optical mouse the movements of the mouse are detected through laser and there are no mechanical movements in it. It is also much quicker than the mechanical one.

Q.270 What is a digitizer? Explain the working of a digitizer. **(7)**

Ans: A Digitizer is a device, which converts analog information into a digital form. You can easily do your signatures using this pen or styles or pen on the digitizer and it is the job of digitizer or change your signatures in bit map and send them to computer for storage. From there, whenever needed the same shape can be regenerated. Digitizers are very accurate devices. They are available in many sizes. The most common sizes are 6 into 8 inch and 12 into 18 inch. The cost of digitizer increases with increase in size. Thus bigger size digitizers are very costly.

The styles used with digitizer looks like an ordinary pen. You can hold it in the same way and can make artistic strokes on the digitizer using it. Whatever strokes you will apply on the digitizer, the same can be seen on the screen.

Working of an Digitizer:

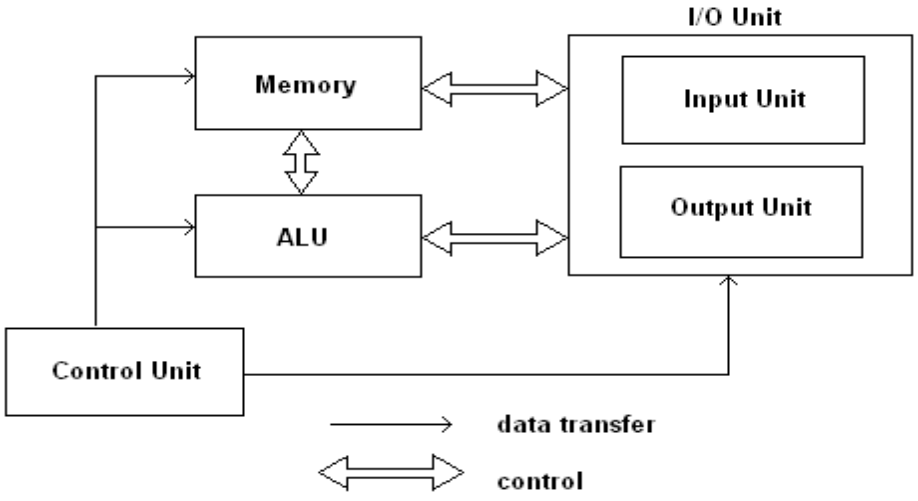
In case of a digitizer each position on the tablet relates to a specific position on the screen. So it traces the existing drawing more accurately and it can easily create original drawing such as architectural drawing with precise dimensions. The styles draws directly on the tablet and its movements are captured and translated into a corresponding drawing on the computer.

A puck or cursor can also be used instead of styles. In both the cases the exact positions of drawing device is detected by the tablet in terms of coordinates and is sent to the computer.

Q.271 With the help of a block diagram, explain the basic organization of a computer system. **(8)**

Ans: Basic Computer Organization: In Computer System, different parts of a computer are organized in such a way that, it helps to perform various operations to complete the given specific task. There are five basic operations. They are:

1. Inputting
2. Outputting
3. Control unit
4. Storage unit
5. Processing



1. INPUT: This operation is used to feed the information in the computer. The standard devices are keyboard, Mouse-a pointing device, and card redirect. The input devices must accept the data from the outside world and the computer to process it must accept the same data. The data or information feuded through the keyboard are stored in the storage device.
2. OUTPUT: This operation is used to display the fed data or the processed data. Some standard output devices are monitor or screen, printer, etc. These output devices must accept the data, which was processed by the processor. The processing is done binary format and it must be converted to understandable form.
3. CONTROL UNIT: This unit is used to control all the devices, which is helpful for processing. It controls the inflow and outflow of data. It works like a traffic cop, which controls the movement of data from memory to processing unit. We can also say it as central nervous system of the computer.
4. STORAGE UNIT: Storage unit is to store any kind of information. Whatever the data inserted or feeded through keyboard is first stored in the memory for further processing. It must store the intermediate results and also final result. The memory in the storage unit is divided in the form of cells. Each and every cell has its address. Memory is divided in two types: RAM and ROM.

Q.272 Give three advantages of windows over DOS operating environment. **(6)**

Ans: There are numerous advantages for windows users and programmers alike over the older DOS text based environment. They are:
1. Standardized graphical user interface
2. Multitasking capability
3. OOP approach in programming
4. Memory control
5. Hardware Independence
6. Use of dynamic link libraries (DLL)

Biggest Different is Graphical user interface and Multimedia feature both are support by Window, not MS-DOS. <Also see Ans7a, June 2004>

Q.273 What is an operating system? Explain how throughput, turnaround time and response time are used to measure the overall performance of a computer system. **(6)**

Ans: Operating system is an integrated set of programs that
a. Controls the resources of a computer system.
b. Provides its user an interface that is easier to use as compared to a bare machine.
There are several possible purposes of an operating system: To provide an environment for a computer user to execute programs on computer hardware in a convenient and efficient manner. To allocate the separate resources of the computer as needed to solve the problem given. The allocation process should be as fair and efficient as possible.
Throughput means number of processes completed / time unit.
Turnaround Time is the average time elapsed from when process is submitted to when it has completed.
Response Time - Average time elapsed from when process is submitted until useful output is obtained.

Typically, Utilization and Throughput are traded off for better Response Time. Response time is important for OS's that aim to be user-friendly. In general, we would like to optimize the average measure. In some cases, minimum and maximum values are optimized, e.g. it might be a good idea to minimize the maximum response time.

Q.274 What is an MICR device? Where is it primarily used? **(2)**

Ans:MICR refers to Magnetic Ink Character Recognition, and is a special type of printing used at the bottom of cheques, to be read by computers. It contains the cheque number, bank routing code, and account number.

Q.275 What are the main functions provided by most Operating Systems? **(6)**

Ans: Main functions of an operating system are:
1. Process management: A process is a program in execution. It is the job which is currently being executed by the processor. During its execution a process would require certain system resources such as processor, time, main memory, files etc. OS supports multiple processes simultaneously. The process management module of the as takes care of the creation and termination of the processes, assigning resources to the processes, scheduling processor time to different processes and communication among processes.
2. Memory management: It takes care of the allocation and deallocation of the main memory to the various processes. It allocates main and secondary memory to the system/user program and data. To execute a program, its binary image must be loaded into the main memory.
OS decides: which part of memory are being currently used and by whom, which process to be allocated memory, Allocation and deallocation of memory space.
3. I/0 management: This module of the as co-ordinates and assigns different I/O devices namely terminals, printers, disk drives, tape drives etc. It controls all I/O devices, keeps track of I/O request, issues command to these devices.
4. File management: Data is stored in a computer system as files. The file management module of the as would manage files held on various storage devices and transfer of files from one device to another. This module takes care of creation, organization, storage, naming, sharing, backup and protection of different files.
5.Scheduling: The as also establishes and enforces process priority. That is, it determines and maintains the order in which the jobs are to be executed by the computer system. This is so because the most important job must be executed first followed by less important jobs.
6.Security management: This module of the OS ensures data security and integrity. That is, it protects data and program from destruction and unauthorized access. It keeps different programs and data which are executing concurrently in the memory in such a manner that they do not interfere with each other.
7. Processor management: as assigns processor to the different task that must be performed by the computer system. If the computer has more than one processor idle, one of the process waiting to be executed is assigned to the idle processor.

COMPUTER NETWORK
QUESTION BANK

UNIT-1

1. a. Compare Connection oriented and connectionless service. 6M
   b. Explain the functions and protocols and services of each layer? 6M

2. What are the major components of an optical communication system? 12M

3. a) Explain about the Guided transmission Medias in computer networks? 6M
   b) Briefly explain about the public switched telephone networks 6M

4. Write about network software & hardware 12M

5. Discuss about a) Computer applications 6M

   b) Features of LAN, MAN, WAN 6M

6. With a neat diagram describe in detail about the Network architecture. 12M

7. Explain in detail about a) Data Link Layer and Network Layer 6M

   b) Physical Layer 6M

8. Distinguish between point to point links and multi point links. Give relevant diagrams 12M

9. Discuss various types of networks topologies in computer network. 12M

10. Explain the ISO-OSI model of computer network with a neat diagram 12M

UNIT-2

1. Explain the operation of the bit-oriented protocol HDLC with the required frames	12M
2. Explain the various error detection and correction Mechanisms used in computer network	12M
3. Discuss about a) GO BACK NARQ	6M
b) Selective repeat ARQ	6M
4. Explain about the Elementary data link protocols?.	12M
5. a. Discuss stop and wait protocol	7M
b. Explain about the Carrier Sense Multiple Access Protocols?	5M
6. Briefly discuss about data link layer design issues?	12M
7. Explain about the data Link Layer switching	12M
8. Explain MAC sub layer protocol and frame structure of IEEE 802.11.	12M
9. Describe sliding window protocol using Go back n.	12
10. Compare and distinguish in detail the concept of Virtual-Circuit and Data gram Networks.	12M

UNIT-3

1. a.)State the major difference between distance vector routing and link state routing.	7M
b) How these routing techniques work	5M
2. Explain about the Routing algorithms.	12M
3. Discuss about a) concept of leaky bucket algorithm?	7M
b) token bucket algorithm with neat diagram	5M
4. Explain in detail congestion control algorithms	12M
5. Explain CSMA/CD protocol	12M
6. Classify the functions of Bridges & Switches in brief.	12M
7. Explain the function of ARP & RARP	12M
8. a. Explain distance vector routing in detail.	7M
b. Discuss about Inter-network Routing?	5M
9. Write a detail note on Internet protocols	12M

10. a) Explain about the Shortest Path Algorithm? 7M

b) Explain packet switching in detail. 5M

UNIT-4

1. a) Explain the duties of transport layer. 6M

b) Write short notes on performance issues of transport layer 6M

2. Show the different approaches in Packet Switching. Explain them in detail. 12M

3. Enumerate the mechanism of three way handshake protocol for TCP 12M

4. Explain in detailed about the Transport Layer. 12M

5. Describe about a) TCP connection management. 5M

b) Avoidance of congestion in TCP 7M

6. Explain the function of TCP/IP protocol. 12M

7. a) Explain about elements of transport protocols 7M

b) Briefly explain the internet transport protocols 5M

8. a) Discuss TCP and its various operations 12M

b) Write short notes on User Datagram Protocol (UDP). 12M

9. Explain the operation of TCP with neat sketch. 12M

10. Discuss the various timers used by TCP to perform its various operations. 12M

UNIT-5

1. Explain how security is provided in interact operations in detail 12M

2. a) Explain the working of Electronic mail. How SMTP used in Email applications 6M

b) List and discuss the types of DNS records. 6M

3. Discuss in detail about world wide web 12M

4. a) Explain the encryption and decryption methods. 6M

b) Discuss Application layer in details 6M

5. Explain in detail about function and structure of e-mail protocol. 12M

6. a) Discuss the File transfer Protocol (FTP)with a neat diagram. 6M

b) Explain briefly simple network management protocol 6M

7. Write short notes on IMAP and MIME with an example 12M

8. a)Discuss the features of HTTP and also discuss how HTTP works. 6M

b) Explain about Application layer and its services in detail? 6M

9. Describe the role of a DNS on a computer network with reference to its components. 12M

10. a) Write briefly about World wide web 6M
b) Describe about High speed LANs 6M

Raajdhani Engineering College, Bhubaneswar

Question Bank

MODULE-1

Sub: DBE **BTech. 4th Sem** **Faculty: Prof. Suren Kumar Sahu**

Short Questions

1. What is candidate key ?
2. What is primary key ?
3. What is database management system?
4. What ate different types of database users?
5. What is data dictionary and what are its contents?
6. What are the different database languages explain with example ?
7. Explain meta data?
8. What is derived attribute?. Give an example.
9. What is weak entity. Give an example
10. What is recursive relationship? Give an example
11. What is the job of query processor?
12. What is an alternate key.
13. What is difference between super key and candidate key.
14. What is the function of data manager.
15. What is an entity, entity set and entity instance.

Long Questions

1. What is distinction among primary key, candidate key and super key
2. Describe three level architecture of DBMS.
3. What are the disadvantage of file processing system.
4. State the advantage and disadvantage of database management system.
5. What are the function of DBA ?
6. What are the different attributes used in ER-Diagram.
7. Differentiate between physical data independence and logical data independence
8. Explain the characteristics and requirement of ER-Model.
9. Draw an ER-Diagram for a Bank Enterprise.
10. Draw an diagram Hospital management system
11. Write the steps of conversion of ER-Diagram to Relational database ..
12. What is generalization, specialization and aggregation. Explain with example.
13. Draw an ER-Diagram for Hostel Management system
14. Describe the components of DBMS.
15. Describe the different symbols use for ER-Diagram.
16. What is schema and sub schema , explain with example.
17. Explain total and partial participation with example
18. Explain any five relational algebra operation with example
19. **Dept(Dno,Dnm)**
    Emp(eno,enm.city,sal,dno)
    Project(pcode,pname,duration)
    Work_assign(pcode,eno,dno,work_assign_dt,duedt)

Solve the given queries using relational algebra

A. Find all the emp names who are working for CSE department
B. Display all empno,ename who are working for project code='P3'
C. Find total number of employees working in banking project

20. Write short notes on:
    a. Foreign Key
    b. Relational constraint
    c. Data Model

DBE QUESTION BANK
MODULE-2

1.

a. **Explain functional dependence.**
b. **What is update anomalies?**
c. **What is MVD?**
d. **What are free and bound variables in TRC?**
e. **Let R=(A,B,C,D) and functional dependencies A→C, AB→D. What is the closure of {AB}**
f. **What do you mean by ACID properties of a transaction?**
g. **Find the tuple calculus representation for the following SQL query: select R1.a,R2.b from R1 and R2 where R1.b=R2.a**
h. **What is trivial MVD?**
i. **What is meant by concurrency?**
j. **What is transitivity dependency?**
k. **What is metadata?**
l. **What is inner join and outer join.**
m. **What is difference between procedural and non procedural language.**
n. **What is 4NF?**
o. **What are problems of concurrent transaction?**

2.

a. **What is normalization? What is a key attribute in a relation .Explain the difference between first, second and third normal forms.**
b. **State Armstrong's axiom?**
c. **Suppose relation R(A,B,C,D,E) has functional dependencies**
   AB->C
   D->A
   AE->B
   CD->E
   BE->D
   Find F^+.
d. **Consider the set of relations:**
   Student(name,roll,mark)
   Score(roll,grade)
   Details(name,address)

   Represent the following query:
   "Find name and address of students scoring grade A"
   in relational algebra, tuple relational calculus, domain relational calculus & SQL
e. **Justify "BCNF is more stronger then 3NF" . Explain with example.**
f. **What is a relational constraint? Describe its types with examples**
g. **Define primary key, candidate key and foreign key .**
h. **What is minimal cover set. What are the constraints to find it.**
i. **Suppose relation R(A,B,C,D,E) has functional dependencies**
   F={AB->C, D->A, AE->B, CD->E,B E->D}.
   Find F^c set.
j. **Define the anomalies of a relation in DBMS. Explain with example**
k. **Suppose relation R(A,B,C,D,E) has functional dependencies**

Module-III

Short Questions

2008

1. What are the two techniques to prevent deadlock? **[2][Concurrency control]**
2. What is meant by Concurrency? **[2][Concurrency control]**

2009

1. What do you mean by atomicity? Explain with example. **[2][Transaction processing]**
2. What is meant by two phase commit protocol? **[2][Concurrency control]**
3. Explain one of the pessimistic concurrency control scheme with example. [**2][Concurrency control]**
4. What is the difference between REDO and UNDO operation? [**2][Database recovery]**
5. What is mean by concurrency? **[2][Concurrency control]**
6. What do you mean by serializability in transaction processing? **[2][Concurrency control]**

2010

1. What do you mean by ACID properties ofa transaction? **[2][Transaction processing]**
2. Forthefollowing operations:

   T1 : read (x) ;

   x=x+10;

   write(x);

   T2: read(x);

   read(x);

   For simultaneous execution Which problem can happen? **[2][Transaction processing]**

3. What is a timestamp? If TS (Ti)>TS (Tj)then which transaction is younger? Justify. Consider TS (Ti) is the timestamp of transaction Ti. [**2][Transaction processing]**

2011

1. What is serial schedule? [**2][Transaction processing]**
2. What is the difference between concurrency control and crash control? **[2][Concurrency control]**

2012

1. What is transaction atomicity? [**2][Transaction processing]**
2. What is two phase locking? **[2][Concurrency control]**

Question Bank
For
Object Oriented programming using Java

UNIT I
PART A

1. What are the OOP Principles?
2. What is Encapsulation?
3. What is Polymorphism?
4. What is Inheritance?
5. What are the features of Java Language?
6. What is the need for Java Language?
7. What is platform independency?
8. What is Architecture Neutral?
9. How Java supports platform independency?
10. Why Java is important to Internet?
11. What are the types of programs Java can handle?
12. What is an applet program?
13. Compare Application and Applet.
14. What are the advantages of Java Language?
15. Give the contents of Java Environment (JDK).
16. Give any 4 differences between C and Java.
17. Give any 4 differences between C++ and Java.
18. What are the different types of comment symbols in Java?
19. What are the data types supported in Java?
20. What is the difference between a char in C/C++ and char in Java?
21. How is a constant defined in Java?
22. What is the use of final keyword?
23. What are the different types of operators used in Java?
24. What is short-Circuit operator?
25. What is labeled break?
26. What is the use of for each control structure?
27. What is the need for static variables?
28. What is the need for static methods?
29. Compare static constants and final constants.
30. Why is main method assigned as public?
31. Why is main method assigned as static?
32. What are the types of variables Java handles?
33. What are the relationships between classes?
34. What is the general form of a class?
35. What is the use of new keyword?
36. If ObjA1 is an object of class A created using new keyword, What does the statement A ObjA2=ObjA1; mean?
37. What is a constructor?

38. What is the difference between a constructor and a method?
39. What is the use of this keyword?
40. What are destructors?
41. How is object destruction done in Java?
42. What is Garbage collection?
43. What is the use of finalize method?
44. Compare Garbage collection and finalize method?
45. How is it guaranteed that finalize methods are called?
46. What is method overloading?
47. What is a String in Java?
48. What is the difference between a String in Java and String in C/C++?
49. Name a few String methods.
50. What is the difference between Concat method and + operator to join strings?
51. What is String Buffer?
52. How does String class differ from the String Buffer class?
53. Name some methods available under String Buffer class.
54. Output of some expressions using String methods.
55. How will you initialize arrays?
56. What is arraycopy method? Explain with syntax.
57. What are the methods under Util.Arrays?
58. Use the array sort method to sort the given array.
59. Give the syntax for array fill operation.
60. What is vector? How is it different from an array?
61. What is the constraint for using vectors?
62. What is wrapper class?
63. What are the different access specifiers available in Java?
64. What is the default access specifier in Java?
65. What is a package in Java?
66. Name some Java API Packages.
67. Name some JavaDoc Comments.
68. What is CommandLine Arguments.

Part B

1. Explain OOP Principles.
2. Explain the features of Java Language.
3. Compare and Contrast Java with C.
4. Compare and Contrast Java with C++.
5. Explain Constructors with examples.
6. Explain the methods available under String and String Buffer Class.
7. Explain the Date Class methods with examples.
8. Discuss in detail the access specifiers available in Java.
9. Explain the different visibility controls and also compare with each of them.

10. Explain the different methods in java.Util.Arrays class with example.
11. Explain Packages in detail.
12. Discuss the methods under Array Class.
13. Discuss some of the classes available under Lang package.
14. Illustrate with examples: static and final.
15. Explain method overriding with example program.
16. What is javaDoc? Explain the comments for classes, methods, fields and link.
17. Application Programs in Java.

UNIT II
PART A

1. Define Inheritance
2. What are the types of inheritance?
3. How is multiple inheritance achieved in java?
4. What is the use of super keyword?
5. What are object wrappers? Give example.
6. What is Inheritance Hierarchy?
7. Differentiate overloading and overriding.
8. Define polymorphism.
9. Differentiate static binding and dynamic binding.
10. When will a class be declared as final?
11. When will a method be declared final?
12. What is an abstract class?
13. What is the need for abstract classes?
14. Explain about protected visibility control.
15. What are the methods under "object" class / java.lang.Object.
16. Explain toString method of object class.
17. What is reflection?
18. What are the uses of reflection in Java.
19. How will you create an instance of Class.
20. What are the methods under reflection used to analyze the capabilities of classes?
21. How to create arrays dynamically using reflection package.
22. Define an interface.
23. What is the need for an interface?
24. What are the properties of an interface?
25. Differentiate Abstract classes and interface.
26. What is object cloning?
27. Differentiate cloning and copying.
28. Differentiate shallow copy and deep copy in cloning.
29. Does Inheritance removes any fields/or methods of super class?
30. Mention the use of final keyword.
31. What is nested class? Mention its types.
32. What is inner class?

33. What is the need for inner classes?
34. What are the rules for inner class?
35. What is local inner class and anonymous inner class? Give their advantages.
36. Write the advantages and disadvantages of static nested class.
37. Define proxies.
38. Write the application of proxies.
39. What are the properties of proxy classes?

PART B

1. Explain the concept of inheritance and its types.
2. Explain the concept of overriding with examples.
3. What is dynamic binding? Explain with example.
4. Explain the uses of reflection with examples.
5. Define an interface. Explain with example.
6. Explain the methods under “object” class and “class” class.
7. What is object cloning? Explain deep copy and shallow copy with examples.
8. Explain static nested class and inner class with examples.
9. With an example explain proxies.
10. Develop a message abstract class which contains playMessage abstract method. Write a different sub-classes like TextMessage, VoiceMessage and FaxMessage classes for to implementing the playMessage method.
11. Develop a abstract Reservation class which has Reserve abstract method. Implement the sub-classes like ReserveTrain and ReserveBus classes and implement the same.
12. Develop an Interest interface which contains simpleInterest and compInterest methods and static final field of Rate 25%. Write a class to implement those methods.
13. Develop a Library interface which has drawbook(), returnbook() (with fine), checkstatus() and reservebook() methods. All the methods tagged with public.
14. Develop an Employee class which implements the Comparable and Cloneable interfaces. Implement the sorting of persons (based on name in alphabetical). Also implement the shallow copy (for name and age) and deep copy (for DateOfJoining).
15. Explain the different methods supported in Object class with example.
16. Explain the methods supported in Class class.
17. Explain the Methods supported in reflect package. Also write a program to implement the reflection of a particular class details like constructors, methods and fields with its modifiers.
18. Develop a static Inner class called Pair which has MinMax method for finding min and max values from the array.

19. What is proxy class? Develop a code for constructing a proxy objects to trace a binary search method with explanations.

UNIT III
PART A

1. Draw the inheritance hierarchy for the frame and component classes in AWT and Swing.
2. What are the advantages of using swing over awt?
3. How do achieve special fonts for your text? Give example.
4. Give the syntax of drawImage() and copyArea() methods.
5. What is Adapter class?
6. Draw the AWT event Hierarchy.
7. What are the swing components?
8. What are the methods under Action Interface.
9. What are the methods under WindowListener Interface.
10. What is the difference between Swing and AWT?

PART B

1. Explain the classes under 2D shapes.
2. Explain event handling with examples.
3. Explain action event with an example.
4. What are the swing components. Explain.
5. Describe the AWT event hierarchy.

UNIT IV
PART A

1. What is generic programming?
2. What are Checked and UnChecked Exception?
3. What are checked exceptions?
4. What are runtime exceptions?
5. What is the difference between error and an exception?
6. What classes of exceptions may be caught by a catch clause?.
7. If I want an object of my class to be thrown as an exception object, what should I do?
8. How to create custom exceptions?
9. What are the different ways to handle exceptions?
10. What is the purpose of the finally clause of a try-catch-finally statement?
11. What is the basic difference between the 2 approaches to exception handling.
12. Is it necessary that each try block must be followed by a catch block?

13. How does Java handle integer overflows and underflows?

PART B

1. Explain generic classes and methods.
2. Explain exception hierarchy.
3. What are the advantages of Generic Programming?
4. Explain the different ways to handle exceptions.
5. How Java handle overflows and underflows?

UNIT V
PART A

1. Describe synchronization in respect to multithreading.
2. Explain different way of using thread?
3. What is synchronization and why is it important?
4. When a thread is created and started, what is its initial state?
5. What are synchronized methods and synchronized statements?
6. What is daemon thread and which method is used to create the daemon thread?
7. What method must be implemented by all threads?
8. What kind of thread is the Garbage collector thread?
9. What is a daemon thread?
10. What is a thread?
11. What is the algorithm used in Thread scheduling?
12. What are the different level lockings using the synchronization keyword?
13. What are the ways in which you can instantiate a thread?
14. What are the states of a thread?
15. What are the threads will start, when you start the java program?
16. What are the different identifier states of a Thread?
17. Why do threads block on I/O?
18. What is synchronization and why is it important?

PART B

1. Explain the different states of a thread.
2. Explain thread synchronization with examples.
3. Explain the algorithm used for thread scheduling.
4. Describe multi threading.
5. Explain Deadlocks.

CS8391-DATA STRUCTURES QUESTION BANK

UNIT I

2MARKS

1.Define data structure.

The data structure can be defined as the collection of elements and all the possible operations which are required for those set of elements. Formally data structure can be defined as a data structure is a set of domains D, a set of domains F and a set of axioms A. this triple (D,F,A) denotes the data structure d.

2. What do you mean by non-linear data structure? Give example.

The non-linear data structure is the kind of data structure in which the data may be arranged in hierarchical fashion. For example- Trees and graphs.

3. What do you linear data structure? Give example.

The linear data structure is the kind of data structure in which the data is linearly arranged. For example- stacks, queues, linked list.

4. List the various operations that can be performed on data structure.

Various operations that can be performed on the data structure are

- Create
- Insertion of element
- Deletion of element
- Searching for the desired element
- Sorting the elements in the data structure
- Reversing the list of elements.

5. What is abstract data type? What are all not concerned in an ADT?

The abstract data type is a triple of D i.e. set of axioms, F-set of functions and A-Axioms in which only what is to be done is mentioned but how is to be done is not mentioned. Thus ADT is not concerned with implementation details.

6. List out the areas in which data structures are applied extensively.

Following are the areas in which data structures are applied extensively.

- Operating system- the data structures like priority queues are used for scheduling the jobs in the operating system.
- Compiler design- the tree data structure is used in parsing the source program. Stack data structure is used in handling recursive calls.

- Database management system- The file data structure is used in database management systems. Sorting and searching techniques can be applied on these data in the file.
- Numerical analysis package- the array is used to perform the numerical analysis on the given set of data.
- Graphics- the array and the linked list are useful in graphics applications.

- Artificial intelligence- the graph and trees are used for the applications like building expression trees, game playing.

7. What is a linked list?

A linked list is a set of nodes where each node has two fields 'data' and 'link'. The data field is used to store actual piece of information and link field is used to store address of next node.

8. What are the pitfall encountered in singly linked list?

Following are the pitfall encountered in singly linked list

- The singly linked list has only forward pointer and no backward link is provided. Hence the traversing of the list is possible only in one direction. Backward traversing is not possible.
- Insertion and deletion operations are less efficient because for inserting the element at desired position the list needs to be traversed. Similarly, traversing of the list is required for locating the element which needs to be deleted.

9. Define doubly linked list.

Doubly linked list is a kind of linked list in which each node has two link fields. One link field stores the address of previous node and the other link field stores the address of the next node.

10. Write down the steps to modify a node in linked lists.

- ➢ Enter the position of the node which is to be modified.
- ➢ Enter the new value for the node to be modified.
- ➢ Search the corresponding node in the linked list.
- ➢ Replace the original value of that node by a new value.
- ➢ **Display the messages as "The node is modified".**

11. Difference between arrays and lists.

In arrays any element can be accessed randomly with the help of index of array, whereas in lists any element can be accessed by sequential access only.

Insertion and deletion of data is difficult in arrays on the other hand insertion and deletion of data is easy in lists.

12. State the properties of LIST abstract data type with suitable example.

Various properties of LIST abstract data type are

(i) It is linear data structure in which the elements are arranged adjacent to each other.

(ii) It allows to store single variable polynomial.

(iii)If the LIST is implemented using dynamic memory then it is called linked list.

Example of LIST are- stacks, queues, linked list.

13. State the advantages of circular lists over doubly linked list.

In circular list the next pointer of last node points to head node, whereas in doubly linked list each node has two pointers: one previous pointer and another is next pointer. The main advantage of circular list over doubly linked list is that with the help of single pointer field we can access head node quickly. Hence some amount of memory get saved because in circular list only one pointer is reserved.

14. What are the advantages of doubly linked list over singly linked list?

The doubly linked list has two pointer fields. One field is previous link field and another is next link field. Because of these two pointer fields we can access any node efficiently whereas in singly linked list only one pointer field is there which stores forward pointer.

15. Why is the linked list used for polynomial arithmetic?

We can have separate coefficient and exponent fields for representing each term of polynomial. Hence there is no limit for exponent. We can have any number as an exponent.

16. What is the advantage of linked list over arrays?

The linked list makes use of the dynamic memory allocation. Hence the user can allocate or de allocate the memory as per his requirements. On the other hand, the array makes use of the static memory location. Hence there are chances of wastage of the memory or shortage of memory for allocation.

17. What is the circular linked list?

The circular linked list is a kind of linked list in which the last node is connected to the first node or head node of the linked list.

18. What is the basic purpose of header of the linked list?

The header node is the very first node of the linked list. Sometimes a dummy value such - 999 is stored in the data field of header node.

This node is useful for getting the starting address of the linked list.

19. What is the advantage of an ADT?

- **Change:** the implementation of the ADT can be changed without making changes in theclient program that uses the ADT.
- **Understandability:** ADT specifies what is to be done and does not specify theimplementation details. Hence code becomes easy to understand due to ADT.
- **Reusability:** the ADT can be reused by some program in future.

20. What is static linked list? State any two applications of it.

- The linked list structure which can be represented using arrays is called static linked list.
- It is easy to implement, hence for creation of small databases, it is useful
- The searching of any record is efficient, hence the applications in which the record need to be searched quickly, the static linked list are used.

16 MARKS

1. Explain the insertion operation in linked list. How nodes are inserted after a specified node.

2. Write an algorithm to insert a node at the beginning of list?

3. Discuss the merge operation in circular linked lists.

4. What are the applications of linked list in dynamic storage management?

5. How polynomial expression can be represented using linked list?

6. What are the benefit and limitations of linked list?

7. Define the deletion operation from a linked list.

8. What are the different types of data structure?

9. Explain the operation of traversing linked list. Write the algorithm and give an

example.

\

UNIT II

2MARKS

1. Define Stack

A Stack is an ordered list in which all insertions (Push operation) and deletion (Pop operation) are made at one end, called the top. The topmost element is pointed by top. The top is initialized to -1 when the stack is created that is when the stack is empty. In a stack S = (a1,an), a1 is the bottom most element and element a is on top of element ai-1. Stack is also referred as Last In First Out (LIFO) list.

2. What are the various Operations performed on the Stack?

The various operations that are performed on the stack are

CREATE(S) – Creates S as an empty stack.

PUSH(S,X) – Adds the element X to the top of the stack.

POP(S) – Deletes the top most elements from the stack.

TOP(S) – returns the value of top element from the stack.

ISEMTPTY(S) – returns true if Stack is empty else false.

ISFULL(S) - returns true if Stack is full else false.

3.Write the postfix form for the expression -A+B-C+D?

A-B+C-D+

4.What are the postfix and prefix forms of the expression?

A+B*(C-D)/(P-R)

Postfix form: ABCD-*PR-/+

Prefix form: +A/*B-CD-PR

5.**Explain the usage of stack in recursive algorithm implementation?**

In recursive algorithms, stack data structures is used to store the return address when a recursive call is encountered and also to store the values of all the parameters essential to the current state of the function.

6.**Define Queue.**

A Queue is an ordered list in which all insertions take place at one end called the rear, while all deletions take place at the other end called the front. Rear is initialized to -1 and front is initialized to 0. Queue is also referred as First In First Out (FIFO) list.

7.**What are the various operations performed on the Queue?**

The various operations performed on the queue are

CREATE(Q) **–** Creates Q as an empty Queue.

Enqueue(Q,X) **–** Adds the element X to the Queue.

Dequeue(Q) **–** Deletes a element from the Queue.

ISEMTPTY(Q) **–** returns true if Queue is empty else false.

ISFULL(Q) - returns true if Queue is full else false.

8.How do you test for an empty Queue?

The condition for testing an empty queue is rear=front-1. In linked list implementation of queue the condition for an empty queue is the header node link field is NULL.

9.**Write down the function to insert an element into a queue, in which the queue is implemented as an array. (May 10)**

Q **–** Queue

X **–** element to added to the queue Q IsFull(Q)

– Checks and true if Queue Q is full Q->Size -

Number of elements in the queue Q Q->Rear **–**

Points to last element of the queue Q Q->Array

– array used to store queue elements void

```
enqueue (int X, Queue Q) {
        if(IsFull(Q))
          Error ("Full queue");
        else      {
                Q->Size++;
```

```
            Q->Rear = Q->Rear+1;
            Q->Array[ Q->Rear ]=X;
        }
    }
```

10..Define Dequeue.

Deque stands for Double ended queue. It is a linear list in which insertions and deletion are made from either end of the queue structure.

11.**Define Circular Queue.**

Another representation of a queue, which prevents an excessive use of memory by arranging elements/ nodes Q_1,Q_2,**…Q_n** in a circular fashion. That is, it is the queue, which wraps around upon reaching the end of the queue

12. List any four applications of stack.

- Parsing context free languages
- Evaluating arithmetic expressions
- Function call
- Traversing trees and graph
- Tower of Hanoi

16 MARKS

1. Write an algorithm for Push and Pop operations on Stack using Linked list. (8)
2. Explain the linked list implementation of stack ADT in detail?
3. Define an efficient representation of two stacks in a given area of memory with n words and explain.
4. Explain linear linked implementation of Stack and Queue?
    a. Write an ADT to implement stack of size N using an array. The elements in the stack are to be integers. The operations to be supported are PUSH, POP and DISPLAY. Take into account the exceptions of stack overflow and stack underflow. (8)
    b. A circular queue has a size of 5 and has 3 elements 10,20 and 40 where F=2 and R=4. After inserting 50 and 60, what is the value of F and R. Trying to insert 30 at this stage what happens? Delete 2 elements from the queue and insert 70, 80 & 90. Show the sequence of steps with necessary diagrams with the value of F & R. (8 Marks)
5. Write the algorithm for converting infix expression to postfix (polish) expression?

6. Explain in detail about priority queue ADT in detail?
7. **Write a function called 'push' that takes two parameters: an integer variable and a stack into** which it would push this element and returns a 1 or a 0 to show success of addition or failure.
8. What is a DeQueue? Explain its operation with example?
9. Explain the array implementation of queue ADT in detail?
10. Explain the addition and deletion operations performed on a circular queue with necessary algorithms.(8) (Nov 09)

UNIT III

2MARKS

1.Define tree?

Trees are non-liner data structure, which is used to store data items in a shorted sequence.

It represents any hierarchical relationship between any data Item. It is a collection of nodes, which has a distinguish node called the root and zero or more non-empty sub trees T1, **T2,….Tk. each of which are connected by a directed edge from the root.**

2. Define Height of tree?

The height of n is the length of the longest path from root to a leaf. Thus all leaves have height zero. The height of a tree is equal to a height of a root.

3. Define Depth of tree?

For any node n, the depth of n is the length of the unique path from the root to node n. Thus for a root the depth is always zero.

4. What is the length of the path in a tree?

The length of the path is the number of edges on the path. In a tree there is exactly one path form the root to each node.

5. Define sibling?

Nodes with the same parent are called siblings. The nodes with common parents are called siblings.

6. Define binary tree?

A Binary tree is a finite set of data items which is either empty or consists of a single item called root and two disjoin binary trees called left sub tree max degree of any node is two.

7. What are the two methods of binary tree implementation?

Two methods to implement a binary tree are,

a. Linear representation.

b. Linked representation

8. What are the applications of binary tree?

Binary tree is used in data processing.

a. File index schemes

b. Hierarchical database management system

9. List out few of the Application of tree data-structure?

- Ø The manipulation of Arithmetic expression
- Ø Used for Searching Operation
- Ø Used to implement the file system of several popular operating systems
- Ø Symbol Table construction
- Ø Syntax analysis

10. Define expression tree?

Expression tree is also a binary tree in which the leafs terminal nodes or operands and non-terminal intermediate nodes are operators used for traversal.

11. Define tree– traversal and mention the type of traversals?
Visiting of each and every node in the tree exactly is called as tree traversal.

Three types of tree traversal

1. Inorder traversal
2. Preoder traversal
3. Postorder traversal.

12. Define in -order traversal?

In-order traversal entails the following steps;

a. Traverse the left subtree

b. Visit the root node

c. Traverse the right subtree

13. Define threaded binary tree.

A binary tree is threaded by making all right child pointers that would normally be null point to the in order successor of the node, and all left child pointers that would normally be null point to the in order predecessor of the node.

14. What are the types of threaded binary tree?

i. Right-in threaded binary tree
ii. Left-in threaded binary tree
iii. Fully-in threaded binary tree

15. Define Binary Search Tree.

Binary search tree is a binary tree in which for every node X in the tree, the values of all the keys in its left subtree are smaller than the key value in X and the values of all the keys in its right subtree are larger than the key value in X.

16.What is AVL Tree?

AVL stands for Adelson-Velskii and Landis. An AVL tree is a binary search tree which has the following properties:

1.The sub-trees of every node differ in height by at most one.

2.Every sub-tree is an AVL tree.

Search time is O(logn). Addition and deletion operations also take O(logn) time.

17. List out the steps involved in deleting a node from a binary search tree.

- Deleting a node is a leaf node (ie) No children
- Deleting a node with one child.
- Deleting a node with two Childs.

18. What is 'B' Tree?

A B-tree is a tree data structure that keeps data sorted and allows searches, insertions, and deletions in logarithmic amortized time. Unlike self-balancing binary search trees, it is optimized for systems that read and write large blocks of data. It is most commonly used in database and file systems.

19. Define complete binary tree.

If all its levels, possible except the last, have maximum number of nodes and if all the nodes in the last level appear as far left as possible

16 MARKS

1.Explain the AVL tree insertion and deletion with suitable example.
2.Describe the algorithms used to perform single and double rotation on AVL tree.
3.Explain about B-Tree with suitable example.
4.Explain about B+ trees with suitable algorithm.
5.Write short notes on
   i. Binomial heaps
   ii. Fibonacci heaps
6.Explain the tree traversal techniques with an example.
7.Construct an expression tree for the expression (a+b*c) + ((d*e+f)*g). Give the outputs when you apply inorder, preorder and postorder traversals.
8.How to insert and delete an element into a binary search tree and write down the code for the insertion routine with an example.
9.What are threaded binary tree? Write an algorithm for inserting a node in a threaded binary tree.
10.Create a binary search tree for the following numbers start from an empty binary search tree. 45,26,10,60,70,30,40 Delete keys 10,60 and 45 one after the other and show the trees at each stage.

UNIT- IV

2 MARKS

1. Write the definition of weighted graph?

A graph in which weights are assigned to every edge is called a weighted graph.

2. Define Graph?

A graph G consist of a nonempty set V which is a set of nodes of the graph, a set E which is the set of edges of the graph, and a mapping from the set of edges E to set of pairs of elements of V. It can also be represented as G=(V, E).

3.**Define adjacency matrix?**

The adjacency –matrix is an n x n matrix A whose elements aij are given by
aij = 1 if (vi, vj) Exists =0 otherwise

4.**Define adjacent nodes?**

Any two nodes, which are connected by an edge in a graph, are called adjacent nodes. For example, if an edge x∈E is associated with a pair of nodes

(u,v) where u, v ∈V, then we say that the edge x connects the nodes u and v.

5.**What is a directed graph?**

A graph in which every edge is directed is called a directed graph.

6.**What is an undirected graph?**

A graph in which every edge is undirected is called an undirected graph.

7.**What is a loop?**

An edge of a graph, which connects to itself, is called a loop or sling.

8.**What is a simple graph?**

A simple graph is a graph, which has not more than one edge between a pair of nodes.

9.**What is a weighted graph?**

A graph in which weights are assigned to every edge is called a weighted graph.

10.**Define indegree and out degree of a graph?**

In a directed graph, for any node v, the number of edges, which have v as their initial node, is called the out degree of the node v.

Outdegree: Number of edges having the node v as root node is the outdegree of the node v.

11.Define path in a graph?

The path in a graph is the route taken to reach terminal node from a starting node.

12.**What is a simple path?**

i. A path in a diagram in which the edges are distinct is called a simple path.

ii. It is also called as edge simple.

13.**What is a cycle or a circuit?**

A path which originates and ends in the same node is called a cycle or circuit.

14.**What is an acyclic graph?**

A simple diagram, which does not have any cycles, is called an acyclic graph.

15.**What is meant by strongly connected in a graph?**

An undirected graph is connected, if there is a path from every vertex to every other vertex. A directed graph with this property is called strongly connected.

16.**When a graph said to be weakly connected?**

aij = 1 if (vi, vj) Exists =0 otherwise

When a directed graph is not strongly connected but the underlying graph is connected, then the graph is said to be weakly connected.

17.**Name the different ways of representing a graph? Give examples (Nov 10)**

a. Adjacency matrix

b. Adjacency list

18.**What is an undirected acyclic graph?**

When every edge in an acyclic graph is undirected, it is called an undirected acyclic graph. It is also called as undirected forest.

19.**What is meant by depth?**

The depth of a list is the maximum level attributed to any element with in the list or with in any sub list in the list.

20.**What is the use of BFS?**

BFS can be used to find the shortest distance between some starting node and the remaining nodes of the graph. The shortest distance is the minimum number of edges traversed in order to travel from the start node the specific node being examined.

21.**What is topological sort?**

It is an ordering of the vertices in a directed acyclic graph, such that: If there is a path from u to v, then v appears after u in the ordering.

22.Write BFS algorithm

1. Initialize the first node's dist number and place in queue

2. Repeat until all nodes have been examined

3. Remove current node to be examined from queue

4. Find all unlabeled nodes adjacent to current node

5. If this is an unvisited node label it and add it to the queue

6. Finished.

23.**Define biconnected graph?**

A graph is called biconnected if there is no single node whose removal causes the graph to break into two or more pieces. A node whose removal causes the graph to become disconnected is called a cut vertex.

24.**What are the two traversal strategies used in traversing a graph?**

a. Breadth first search
b. Depth first search

25.**Articulation Points (or Cut Vertices) in a Graph**

A vertex in an undirected connected graph is an articulation point (or cut vertex) if removing it (and edges through it) disconnects the graph. Articulation points represent vulnerabilities in a connected network – single points whose failure would split the network into 2 or more disconnected components. They are useful for designing reliable networks.

16 MARKS

1. Explain the various representation of graph with example in detail?
2. Explain Breadth First Search algorithm with example?
3. Explain Depth first and breadth first traversal?
4. What is topological sort? Write an algorithm to perform topological sort?(8) (Nov 09)
5. (i) write an algorithm to determine the biconnected components in the given graph. (10) (may 10)

   (ii)determine the biconnected components in a graph. (6)

6. Explain the various applications of Graphs.

UNIT – V

2 MARKS

1.What is meant by Sorting?

Sorting is ordering of data in an increasing or decreasing fashion according to some linear relationship among the data items.

2. List the different sorting algorithms.

- Bubble sort
- Selection sort
- Insertion sort
- Shell sort
- Quick sort
- Radix sort
- Heap sort
- Merge sort

3. Why bubble sort is called so?

The bubble sort gets its name because as array elements are sorted they gradually **"bubble" to their proper positions, like bubbles risin**g in a glass of soda.

4. State the logic of bubble sort algorithm.

The bubble sort repeatedly compares adjacent elements of an array. The first and second elements are compared and swapped if out of order. Then the second and third elements are compared and swapped if out of order. This sorting process continues until the last two elements of the array are compared and swapped if out of order.

5. What number is always sorted to the top of the list by each pass of the Bubble sort algorithm?

Each pass through the list places the next largest value in its proper place. In essence, each item **"bubbles" up to the location where it belongs.**

6. When does the Bubble Sort Algorithm stop?

The bubble sort stops when it examines the entire array and finds that no "swaps" are needed. The bubble sort keeps track of the occurring swaps by the use of a flag.

7. State the logic of selection sort algorithm.

It finds the lowest value from the collection and moves it to the left. This is repeated until the complete collection is sorted.

8. What is the output of selection sort after the 2nd iteration given the following sequence?

16 3 46 9 28 14

Ans: 3 9 46 16 28 14

9.How does insertion sort algorithm work?

In every iteration an element is compared with all the elements before it. While comparing if it is found that the element can be inserted at a suitable position, then space is created for it by shifting the other elements one position up and inserts the desired element at the suitable position. This procedure is repeated for all the elements in the list until we get the sorted elements.

10.What operation does the insertion sort use to move numbers from the unsorted section to the sorted section of the list?

The Insertion Sort uses the swap operation since it is ordering numbers within a single list.

11. How many key comparisons and assignments an insertion sort makes in its worst case?

The worst case performance in insertion sort occurs when the elements of the input array are in descending order. In that case, the first pass requires one comparison, the second pass **requires two comparisons, third pass three comparisons,….kth pass requires (k**-1), and finally the last pass requires (n-1) comparisons. Therefore, total numbers of comparisons are:

f(n) = 1+2+3+……+(n-k)+…..+(n-2)+(n-1) = n(n-1)/2 = O(n2)

12. Which sorting algorithm is best if the list is already sorted? Why?

Insertion sort as there is no movement of data if the list is already sorted and

complexity is of the order O(N).

13. Which sorting algorithm is easily adaptable to singly linked lists? Why?

Insertion sort is easily adaptable to singly linked list. In this method there is an array link of pointers, one for each of the original array elements. Thus the array can be thought of as a linear link list pointed to by an external pointer first initialized to 0. To insert the k^{th} element the linked list is traversed until the proper position for x[k] is found, or until the end of the list is

reached. At that point x[k] can be inserted into the list by merely adjusting the pointers without shifting any elements in the array which reduces insertion time.

14. Why Shell Sort is known diminishing increment sort?

The distance between comparisons decreases as the sorting algorithm runs until the last phase in which adjacent elements are compared. In each step, the sortedness of the sequence is increased, until in the last step it is completely sorted.

15. Which of the following sorting methods would be especially suitable to sort alist L consisting of a sorted list followed by a few "random" elements?

Quick sort is suitable to sort a list L consisting of a sorted list followed by a few "random" elements.

16.What is the output of quick sort after the 3rd iteration given the following sequence?
24 56 47 35 10 90 82 31

Pass 1:- (10) 24 (56 47 35 90 82 31)

Pass 2:- 10 24 (56 47 35 90 82 31)

Pass 3:- 10 24 (47 35 31) 56 (90 82)

17.Mention the different ways to select a pivot element.

The different ways to select a pivot element are

- Pick the first element as pivot
- Pick the last element as pivot
- Pick the Middle element as pivot
- Median-of-three elements
- Pick three elements, and find the median x of these elements
- Use that median as the pivot.
- Randomly pick an element as pivot.

18.What is divide-and-conquer strategy?

- Divide a problem into two or more sub problems
- Solve the sub problems recursively
- Obtain solution to original problem by combining these solutions

19. Compare quick sort and merge sort.

Quicksort has a best-case linear performance when the input is sorted, or nearly sorted. It has a worst-case quadratic performance when the input is sorted in reverse, or nearly sorted in reverse.

Merge sort performance is much more constrained and predictable than the performance of quicksort. The price for that reliability is that the average case of merge sort is slower than the average case of quicksort because the constant factor of merge sort is larger.

20.Define Searching.

Searching for data is one of the fundamental fields of computing. Often, the difference between a fast program and a slow one is the use of a good algorithm for the data set. Naturally, the use of a hash table or binary search tree will result in more efficient searching, but more often than not an array or linked list will be used. It is necessary to understand good ways of searching data structures not designed to support efficient search.

21.What is linear search?

In Linear Search the list is searched sequentially and the position is returned if the key element to be searched is available in the list, otherwise -1 is returned. The search in Linear Search starts at the beginning of an array and move to the end, testing for a match at each item.

22.**What is Binary search?**

A binary search, also called a dichotomizing search, is a digital scheme for locating a specific object in a large set. Each object in the set is given a key. The number of keys is always a power of 2. If there are 32 items in a list, for example, they might be numbered 0 through 31 (binary 00000 through 11111). If there are, say, only 29 items, they can be numbered 0 through 28 (binary 00000 through 11100), with the numbers 29 through31 (binary 11101, 11110, and 11111) as dummy keys.

23.Define hash function?

Hash function takes an identifier and computes the address of that identifier in the hash table using some function.

16 MARKS

1. Write an algorithm to implement Bubble sort with suitable example.
2. Explain any two techniques to overcome hash collision.
3. Write an algorithm to implement insertion sort with suitable example.
4. Write an algorithm to implement selection sort with suitable example.
5. Write an algorithm to implement radix sort with suitable example.
6. Write an algorithm for binary search with suitable example.
7. Discuss the common collision resolution strategies used in closed hashing system.
8. Given the input { 4371, 1323, 6173, 4199, 4344, 9679, 1989 } and a hash function of h(X)=X (mod 10) show the resulting:
    a. Separate Chaining hash table
    b. Open addressing hash table using linear probing
9. Explain Re-hashing and Extendible hashing.

10. Show the result of inserting the keys 2,3,5,7,11,13,15,6,4 into an initially empty extendible hashing data structure with M=3. (8) (Nov 10)

11. what are the advantages and disadvantages of various collision resolution strategies? (6)

Question Bank for Operating System

---

16. UNIT-WISE QUESTION BANK

UNIT 1: INTORDUCTION TO OS

8M QUESTIONS

1. Define operating system and list the basic services provided by operating system.
2. Differentiate among the following types of OS by defining their essential properties.
    a) Time sharing system
    b) Parallel system
    c) Distributed system
    d) Real time system
3. Explain the essential properties of
    a) Batch System
    b) Time sharing
    c) Real time
    d) Parallel
    e) Distributed
    f) Handheld
    g) Embedded
    h) Smart Card O.S
4. Differentiate among the following types of OS by defining their essential properties:
    a) Time Sharing System
    b) Parallel System
    c) Simple batch System
    d) Real time System
5. Explain batch system and Multiprogrammed System in detail.
6. Explain the terms :
    a) Real time System
    b) Distributed System
7. Explain the terms :
    a) Parallel System
    b) Batch System
8. Explain O.S as extended machine in detail.
9. Explain OS as resources manager

Question Bank for Operating System

---

10. Explain essential features of following structure of O.S

    a) Monolithic System
    b) Layered Systems
    c) Micro Kernels
    d) Client Server Model
    e) Virtual Machines
    f) Exokernels

2M/SHORT ANSWER QUESTIONS

1) What is the key difference between a trap and an interrupt?
2) What are the types of System calls?
3) List any four process management system call.
4) Define user mode and kernel mode. Why two modes are required?
5) What is the O.S features required for multiprogramming
6) What are the advantage and disadvantage of multiprocessor system?
7) Describe the difference between symmetric and asymmetric multiprocessing?
8) Distinguish between the client-server and peer-to-peer models of distributed system
9) What difference is between loosely coupled and tightly coupled system.
10) What are advantages of distributed System?
11) What are the requirements of hard real time and soft real time system?
12) What are the drawbacks of monolithic system?
13) What are the advantages of layered structure over monolithic structure?
14) Give examples of microkernel
15) What are differences between macro kernel and micro kernel?
16) Justify whether following statements are true or false
    a) The user application interacts directly with O.S.
    b) Shell is part of operating System

Raajdhani Engineering College,Bhubaneswar
Department of Computer Science and Engineering

Question Bank for Operating System

UNIT 2: PROCESS MANAGEMENT

8M QUESTION

1) Define process and Explain process states in details with diagram
2) Explain process states and process control block in details
3) Explain the process state transition diagram used in multiprogramming environment. Describe the fields in a process control block (PCB).What is switching overhead?
4) What is thread? Explain classical thread model OR Explain threads in detail
5) Explain and differentiate between user level and kernel level thread.
6) List the main difference and similarities between threads and process.
7) What are various criteria for a good process scheduling algorithm? Explain any two preemptive scheduling algorithms in brief.
8) Explain the following process scheduling algorithm
    a) Priority scheduling
    b) Shortest job first scheduling
9) Explain the effect of increasing the time quantum to an arbitrary large Number and decreasing the time quantum to an arbitrary small number for round robin scheduling algorithm with suitable example?
10) Consider following processes with length of CPU burst time in milliseconds

Process	Burst time
P1	5
P2	10
P3	2
P4	1

All process arrived in order p1, p2, p3, p4 all time zero

a) Draw Gantt charts illustrating execution of these processes for SJF and round robin (quantum=1)
b) Calculate waiting time for each process for each scheduling algorithm
c) Calculate average waiting time for each scheduling algorithm

37. Consider following processes with length of CPU brust time in millisecond

Process	Brust time	Priority
P1	10	3

Question Bank for Operating System

P2	1	1
P3	2	3
P4	1	4
P5	5	2

All processes arrived in order p1, p2, p3, p4, p5 all at time zero.

1) Draw Gant charts illustrating execution of these processes for SJF, non preemptive priority (smaller priority number implies a higher priority) & round robin(quantum=1)

2) Calculate turnaround time for each process for scheduling algorithm in part (1)

3) Calculate waiting time for each scheduling algorithm in part (1)

38. Explain the following term related to IPC: a) Race condition b) critical region

39. What are critical sections? Why mutual exclusion required? Explain any 2 methods of achieving mutual exclusion in detail.

40. Explain the terms related to IPC –a) Race condition b) critical section c)Mutual exclusion d)Semaphores

41. Explain in detail the following solutions for achieving mutual exclusion a)look variable b)TSL instruction

42. Explain Peterson's solution for achieving mutual exclusion

43. Discuss in detail following solution for achieving mutual exclusion a) Disabling interrupts b) Strict alteration

44. Explain semaphore in detail

45. What is semaphore? Discuss product-consumer problem with semaphore.

46. Write short note on: a) Dining philosopher problem b) System calls c) Monitors

d) Peterson's solution for achieving mutual exclusion e) Semaphores

f) Readers & writers problem.

47. Explain the terms: a) time sharing b) mutual exclusion

48. What is monitor? Explain solution for producer-consumer problem using monitor. Explain monitors in detail.

49. Write short on: a) message passing b) shell

50. How message passing is used in IPC.

51. What is message passing? Discuss procedure consumer problem with message passing.

52. Explain use of message passing & semaphore for inter process communication?

53. Explain dinning philosopher problem & its solution.

54. What is dinning philosopher problem? Explain its solution with monitor.

Question Bank for Operating System

---

55. What is dinning philosopher problem? Explain its solution with semaphore.
56. Explain readers & writers problem? Give its solution with semaphore.
57. Write short notes on: a) Process states b) Critical section c) Race condition d) Starvation e) PCB f) Two level scheduling g) Round robin scheduling

Question Bank for Operating System

UNIT 03: FILE MANAGEMENT

8 MARKS

58. What are the objectives and minimal set of requirement for the file management system?
59. What criteria are important in choosing a file organization?
60. Explain briefly file system architecture & file management function.
61. List& briefly explain 5 file organization.
62. Compare file organization methods
63. Which are the typical information elements of a file directory?
64. Which are the typical operations performed on directory?
65. What are the typical access rights that may be granted or denied to a particular user for a particular file?
66. What are methods of free space management of Disk?
67. Explain linked list allocation using index in details.
68. Explain file system consistency in detail.
69. Explain file system reliability & performance in detail.
70. What is directory? Explain directory operation in details.
71. Explain linked list allocation of file in detail.
72. Explain file system performance in detail.
73. Explain the following techniques to improve file system performance.
    a) block read ahead and
    b) Reducing disk arm motion
74. Explain file system implementation using linked list with index and i-node in detail?
75. Explain the following file allocation methods
    a) Contiguous allocation b) i-node
76. What are points to be consider in file system design? Explain linked list allocation & index allocation in detail.
77. Differentiate between windows and unix file system.

2 MARKS/SHORT ANSWER QUSTIONS

78. What is the difference between field and record?
79. What is the difference between file and database?
80. What is file management system?
81. What is relation between pathname & a working directory?

Question Bank for Operating System

UNIT 4: MEMORY MANAGEMENT

82. What are the memory management requirements?
83. Explain multiprogramming with fixed partition.
84. Explain multiprogramming with dynamic partition.
85. Write short note on: Relocation problem for multiprogramming with fixed partitions.
86. Explain static partitioned allocation with partition sizes 300,150, 100, 200, 20. Assuming first fit method indicate the memory status after memory request for sizes 80, 180, 280, 380, 30.
87. Discuss in details memory management with buddy system.
88. A 1MB block of memory is allocated using the buddy system.
    i. Show the results of the following sequence in a figure: Request 70; Request 35; Request 80; Return A; Request 60; Return B; Return D; Return C.
    ii. Show the binary tree representation following Return B.
89. Explain memory management with bit maps in detail.
90. Explain memory management with linked list in details.
91. What are the differences of internal and external memory Fragmentation?
92. Explain following allocation algorithm.
    a. First fit
    b. Best fit
    c. Worst fit
    d. Next fit
93. Explain the difference between logical and physical addresses?
94. What is paging? Discuss basic paging technique in details.

    OR

    Explain paging in detail.
95. Explain hierarchical page table and inverted page table.
96. Explain Segmentation in detail.

    OR

    What is segmentation? Explain the basic segmentation method.
97. What is demand paging? Explain it with address translation mechanism used. What are its specific advantages? How a page table is implemented?
98. What is virtual memory? How it is implemented.
99. Write short on:
    a. multiprogramming with fixed & variable partition.

b. Relocation problem for multiprogramming with fixed partition.
c. Use of multiprogramming in memory management.
d. TLB.
e. Paging.
f. Design issues of paging system.
g. Relocation and protection.
h. policy driven scheduling.

100. Write short note on:
a. Segmentation
b. Page table
c. Compaction
d. Working set model
e. fragmentation

101. Write short note on:
a) Not-recently used page replacement algorithm.
b) Optimal page replacement algorithm.
c) Swapping.
d) Relocation and protection.

102. Explain following page replacement algorithm in detail.
i. LRU ii. FIFO

103. Explain the following page replacement algorithm.
a) Optimal page replacement
b) Least recently used page replacement.

104. Explain difference between internal external fragmentations in detail.

105. Consider the following page reference string.

1,2,3,4,5,3,4,1,6,7,8,7,8,9,7,8,9,5,4,5,4,2How many page faults would occur for the following replacement algorithm, assuming four and six frames respectively? a. page replacement. b. FIFO page replacement.

106. Describe the term page fault frequency. What is thrashing? How is it controlled by OS?

107. Free memory holes of sizes 15K, 10K, 5K, 25K, 30K, 40K are available. The processes of size 12K, 2K,

25K, 20K is to be allocated. How processes are placed in first fit, best fit, worst fit. Calculate internal as well as external fragmentation.

Question Bank for Operating System

---

108. On a simple paging system with 2^24 bytes of physical memory, 256 pages of logical address Space, and a, page size of 2^10 bytes, how many bits are in logical address?

109. A certain computer provides its user with a virtual memory space of 2^32 bytes. The computer has 2^35 bytes of physical memory. The virtual memory is implemented by paging the page size is 4096 bytes. A user process generates the virtual address 11123456. Explain how the system establishes the corresponding physical location.

110. Calculate page faults for (LRU, FIFO, OPT) for following sequences where page frame is three.

0,1,2,1,4,2,3,7,2,1,3,5,1,2,5.

Question Bank for Operating System

UNIT 5: DEVICE MANAGEMENT

111. Discuss briefly the following issues related to device independent i/o software.
    a. Uniform interfacing for device drivers.
    b. Buffering.
112. Discuss in details devices drivers.
113. Write short notes on:
    a. Devices independent I/O software
    b. Goals of I/O software
    c. Interrupt handler
    d. I/O Devices.
    e. Device drivers
    f. Device controllers
    g. Disk space management
    h. Disk arm scheduling algorithm
114. Discuss the following:
    a) Magnetic disk
    b) CDs
    c) RAID
    d) DVDs
    e) Formatting Disk
115. Discuss the following related to disk space management
    a) Block size
    b) Keeping track of free blocks.

116. Suppose a disk drive has 400 cylinders , numbered 0 to 399.The driver is currently serving a request at cylinder 143 and previous request was at cylinder 125 .The queue of pending request in FIFO order is: 86,147,312,91,177,48,309,222,175,130.

Starting from the current head position what is the total distance in cylinders that the disk to satisfy all the pending request for each of the following disk scheduling algorithms?

1] SSTS 2] SCAN 3] C-SCAN

Question Bank for Operating System

UNIT 6: DEADLOCK AND CASE STUDY

117. What are the conditions for deadlock? Explain deadlock detection and recovery in detail. **A/M 2011 8M.**

118. Explain deadlock prevention in detail. **N/D 2011 8M M/A 2009.**

119. Write short notes on:

a. Deadlock modeling **(DEC 2008 O/N2010) 6M(N/D2008).**

b. Bankers algorithm**. A/M 2011 6M.**

120. Explain deadlock avoidance using banker's algorithm in details. **O/N2010 8M M/J 2009 8M.**

121. What is deadlock? Explain deadlock detection with multiple resources of each type. **M/A 2010 8M.**

122. Explain bankers algorithm for multiple resources to avoid deadlock. **M/A 2010 8M.**

123. Explain various methods for recovery from deadlock.**DEC 2010 8M.**

124. Discuss deadlock detection with one resource of each type.**DEC 2009 8M.**

125. Write short notes on-

a) Bankers algorithm for single resources.

b) Ostrich algorithm. **M/J 2010 6M M/A 2010.**

126. Explain deadlock avoidance with suitable example using banker's algorithm. **M/J 2012 8M.**

127. Consider the following snapshot-

	Allocated				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	0	0	1	2	0	0	1	2	1	5	2	0
P1	1	0	0	0	1	7	5	0				
P2	1	3	5	4	2	3	5	6				
P3	0	6	3	2	0	6	5	2				
P4	0	0	1	4	0	6	5	6				

Answer the following questions using banker's algorithm:

a) What are contents of matrix end?

b) Is the system in safe state?

c) If request for process p1 arrives for (0,4,2,0) .Can the request be granted immediately? **N/D 2008 8M.**

Question Bank for Operating System

---

128. What are deadlock? Explain its model with example. Explain any three methods of dealing with deadlock. **N/D 2008 8M.**

129. A system has three types of resources R1 R2 R3 and their number of units are 3, 2, 2 respectively. Four processes P1 P2 P3 p4 are currently competing for these resources in following number.

1. P1 is holding one unit of R1 and is requesting for one unit of R2.
2. P2 is holding two units of R2 and is requesting for one unit each of R1 and R3.
3. P3 is holding one unit of R1 and is requesting for one unit of R2.
4. P4 is holding two units of R3 and requesting for one unit of R1.

Determine which if any of the processes are deadlock in this state**. M/J 2012 8M.**

130. Explain swap space management methods of disk in detail. **N/D 2008 8M.**

131. Consider system with total of 150 minutes of memory allocated to three processes as shown. Apply banker's algorithm to determine whether it would be safe to grant each of following request. If yes-Indicate sequence of termination that could be possible. If no- Show reduction of resulting allocation table.

1. A 4^{th} process is arrived with maximum need of 60 and initial need of 25 units.
2. A 4^{th} process is arrived with maximum need of 60 and initial need of 35 units.

Process	Max	Hold
P1	70	45
P2	60	10
P3	60	15

132. Explain history of windows operating system.

133. Write short notes on-

a) Features of windows- 7.
b) WINDOWS -7 architecture
c) WINDOWS -7 Registry

134. Explain architectural features of WINDOWS-7.

Question Bank for Operating System

---

135. Explain system structure of WINDOWS- 7.
136. Explain process and thread management WINDOWS- 7 in detail.
137. Explain in brief concurrency control supported by WINDOWS -7.
138. Briefly explain security features of WINDOWS7.
139. Explain memory management and I/O management.

TYPICAL QUESTIONS & ANSWERS

PART I

OBJECTIVE TYPE QUESTIONS

Each Question carries 2 marks.

Choose the correct or best alternative in the following:

Q.1 The most important feature of spiral model is
- **(A)** requirement analysis.
- **(B)** risk management.
- **(C)** quality management.
- **(D)** configuration management.

Ans: B

Q.2 The worst type of coupling is
- **(A)** Data coupling.
- **(B)** control coupling.
- **(C)** stamp coupling.
- **(D)** content coupling.

Ans: D

Q.3 IEEE 830-1993 is a IEEE recommended standard for
- **(A)** Software requirement specification.
- **(B)** Software design.
- **(C)** Testing.
- **(D)** Both **(A)** and **(B)**

Ans: A

Q.4 One of the fault base testing techniques is
- **(A)** unit testing.
- **(B)** beta testing.
- **(C)** Stress testing.
- **(D)** mutation testing.

Ans: D

Q.5 Changes made to an information system to add the desired but not necessarily the required features is called
- **(A)** Preventative maintenance.
- **(B)** Adaptive maintenance.
- **(C)** Corrective maintenance.
- **(D)** Perfective maintenance.

Ans: D

Q.6 All the modules of the system are integrated and tested as complete system in the case of
- **(A)** Bottom up testing
- **(B)** Top-down testing
- **(C)** Sandwich testing
- **(D)** Big-Bang testing

Ans: D

Q.7 If every requirement stated in the Software Requirement Specification (SRS) has only one interpretation, SRS is said to be

(A) correct. **(B)** unambiguous.
(C) consistent. **(D)** verifiable.

Ans: B

Q.8 A fault simulation testing technique is

(A) Mutation testing **(B)** Stress testing
(C) Black box testing **(D)** White box testing

Ans: A

Q.9 Modules X and Y operate on the same input and output data, then the cohesion is

(A) Sequential **(B)** Communicational
(C) Procedural **(D)** Logical

Ans: B

Q.10 If the objects focus on the problem domain, then we are concerned with

(A) Object Oriented Analysis.
(B) Object Oriented Design
(C) Object Oriented Analysis & Design
(D) None of the above

Ans: A

Q.11 SRS is also known as specification of

(A) White box testing **(B)** Stress testing
(C) Integrated testing **(D)** Black box testing

Ans: D

Q.12 The model in which the requirements are implemented by category is

(A) Evolutionary Development Model
(B) Waterfall Model
(C) Prototyping
(D) Iterative Enhancement Model

Ans: A

Q.13 SRD stands for

(A) Software requirements definition
(B) Structured requirements definition
(C) Software requirements diagram
(D) Structured requirements diagram

Ans: B

Q.14 A COCOMO model is

(A) Common Cost Estimation Model.
(B) Constructive Cost Estimation Model.
(C) Complete Cost Estimation Model.
(D) Comprehensive Cost Estimation Model.

Ans: B

Q.15 Which of the following statements is true
(A) Abstract data types are the same as classes
(B) Abstract data types do not allow inheritance
(C) Classes cannot inherit from the same base class
(D) Object have state and behavior

Ans: B

Q.16 The desired level of coupling is
(A) No coupling **(B)** Control coupling
(C) Common coupling **(D)** Data coupling

Ans: D

Q.17 In the spiral model 'risk analysis' is performed
(A) In the first loop **(B)** in the first and second loop
(C) In every loop **(D)** before using spiral model

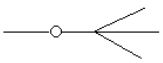
Ans: C

Q.18 For a well understood data processing application it is best to use
(A) The waterfall model **(B)** prototyping model
(C) the evolutionary model **(D)** the spiral model

Ans: A

Q.19 Coupling and cohesion can be represented using a
(A) cause-effect graph **(B)** dependence matrix
(C) Structure chart **(D)** SRS

Ans: B

Q.20 The symbol  represents

(A) mandatory 1 cardinality **(B)** mandatory many cardinality
(C) optional 0 or 1 cardinality **(D)** optional zero-many cardinality

Ans: D

Q.21 Each time a defect gets detected and fixed, the reliability of a software product
(A) increases. **(B)** decreases.
(C) remains constant. **(D)** cannot say anything.

Ans: A

Q.22 Output comparators are used in
- **(A)** static testing of single module
- **(B)** dynamic testing of single module
- **(C)** static testing of single and multiple module
- **(D)** dynamic testing of single and multiple module

Ans: D

Q.23 The feature of the object oriented paradigm which helps code reuse is
- **(A)** object.
- **(B)** class.
- **(C)** inheritance.
- **(D)** aggregation.

Ans: C

Q.24 The level at which the software uses scarce resources is
- **(A)** reliability
- **(B)** efficiency
- **(C)** portability
- **(D)** all of the above

Ans: B

Q.25 If every requirement can be checked by a cost-effective process, then the SRS is
- **(A)** verifiable
- **(B)** traceable
- **(C)** modifiable
- **(D)** complete

Ans: A

Q.26 Modifying the software to match changes in the ever changing environment is called
- **(A)** adaptive maintenance
- **(B)** corrective maintenance
- **(C)** perfective maintenance
- **(D)** preventive maintenance

Ans: A

Q.27 All activities lying on critical path have slack time equal to
- **(A)** 0
- **(B)** 1
- **(C)** 2
- **(D)** None of above

Ans: A

Q.28 Alpha and Beta Testing are forms of
- **(A)** Acceptance testing
- **(B)** Integration testing
- **(C)** System Testing
- **(D)** Unit testing

Ans: A

Q.29 An object encapsulates
- **(A)** Data
- **(B)** Behaviour
- **(C)** State
- **(D)** Both Data and behaviour

Ans: D

Q.30 In function point analysis, number of general system characteristics used to rate the system are

(A) 10 **(B)** 14
(C) 20 **(D)** 12

Ans: B

Q.31 Aggregation represents

(A) is_a relationship **(B)** part_of relationship
(C) composed_of relationship **(D)** none of above

Ans: C

Q.32 If P is risk probability, L is loss, then Risk Exposure (RE) is computed as

(A) RE = P/L **(B)** RE = P + L
(C) RE = P*L **(D)** RE = 2* P *L

Ans: C

Q.33 Number of clauses used in ISO 9001 to specify quality system requirements are:

(A) 15 **(B)** 20
(C) 25 **(D)** 28

Ans: B

Q.34 ER model shows the

(A) Static view. **(B)** Functional view.
(C) Dynamic view. **(D)** All the above.

Ans: A

Q.35 The tools that support different stages of software development life cycle are called:

(A) CASE Tools **(B)** CAME tools
(C) CAQE tools **(D)** CARE tools

Ans: A

Q.36 Changes made to the system to reduce the future system failure chances is called

(A) Preventive Maintenance **(B)** Adaptive Maintenance
(C) Corrective Maintenance **(D)** Perfective Maintenance

Ans: A

Q.37 Requirements can be refined using

(A) The waterfall model **(B)** prototyping model
(C) the evolutionary model **(D)** the spiral model

Ans: B

Q.38 The model that assumes that effort and development time are functions of product size alone is

(A) Basic COCOMO model **(B)** Intermediate COCOMO model
(C) Detailed COCOMO model **(D)** All the three COCOMO models

Ans: A

Q.39 Structured charts are a product of

(A) requirements gathering **(B)** requirements analysis
(C) design **(D)** coding

Ans: C

Q.40 The problem that threatens the success of a project but which has not yet happened is a

(A) bug **(B)** error
(C) risk **(D)** failure

Ans: C

Q.41 The main purpose of integration testing is to find

(A) design errors **(B)** analysis errors
(C) procedure errors **(D)** interface errors

Ans: D

Q.42 Pseudocode can replace

(A) flowcharts **(B)** structure charts
(C) decision tables **(D)** cause-effect graphs

Ans: A

Q.43 If a program in its functioning has not met user requirements is some way, then it is

(A) an error. **(B)** a failure.
(C) a fault. **(D)** a defect.

Ans: D

Q.44 The testing that focuses on the variables is called

(A) black box testing **(B)** white box testing
(C) data variable testing **(D)** data flow testing

Ans: A

Q.45 CASE Tool is

(A) Computer Aided Software Engineering
(B) Component Aided Software Engineering
(C) Constructive Aided Software Engineering
(D) Computer Analysis Software Engineering

Ans: A

Q.46 Software consists of

(A) Set of instructions + operating procedures
(B) Programs + documentation + operating procedures
(C) Programs + hardware manuals
(D) Set of programs

Ans: B

Q.47 Which is the most important feature of spiral model?

(A) Quality management **(B)** Risk management
(C) Performance management **(D)** Efficiency management

Ans: B

Q.48 Which phase is not available in software life cycle?

(A) Coding **(B)** Testing
(C) Maintenance **(D)** Abstraction

Ans: D

Q.49 Which is not a step of requirement engineering?

(A) Requirements elicitation **(B)** Requirements analysis
(C) Requirements design **(D)** Requirements documentation

Ans: C

Q.50 FAST stands for

(A) Functional Application Specification Technique
(B) Fast Application Specification Technique
(C) Facilitated Application Specification Technique
(D) None of the above

Ans: C

Q.51 For a function of two variables, boundary value analysis yields

(A) 4n + 3 test cases **(B)** 4n + 1 test cases
(C) n + 4 **(D)** None of the above

Ans: B

Q.52 Site for Alpha Testing is

(A) Software Company **(B)** Installation place
(C) Any where **(D)** None of the above

Ans: A

Q.53 Which is not a size metric?

(A) LOC **(B)** Function count

(C) Program length **(D)** Cyclomatic complexity

Ans: D

Q.54 As the reliability increases, failure intensity
(A) decreases **(B)** increases
(C) no effect **(D)** none of the above

Ans: A

Q.55 Software deteriorates rather than wears out because
(A) software suffers from exposure to hostile environments.
(B) defects are more likely to arise after software has been used often.
(C) multiple change requests introduce errors in component interactions.
(D) software spare parts become harder to order.

Ans: B

Q.56 What are the three generic phases of software engineering?
(A) Definition, development, support
(B) What, how, where
(C) Programming, debugging, maintenance
(D) Analysis, design, testing

Ans: A

Q.57 The spiral model of software development
(A) Ends with the delivery of the software product
(B) Is more chaotic than the incremental model
(C) Includes project risks evaluation during each iteration
(D) All of the above

Ans: C

Q.58 Which of these terms is a level name in the Capability Maturity Model?
(A) Ad hoc **(B)** Repeatable
(C) Reusable **(D)** Organized

Ans: C

Q.59 Which of the items listed below is not one of the software engineering layers?
(A) Process **(B)** Manufacturing
(C) Methods **(D)** Tools

Ans: B

Q.60 Which of the following are advantages of using LOC (lines of code) as a size-oriented metric?
(A) LOC is easily computed.
(B) LOC is a language dependent measure.

(C) LOC is a language independent measure.
(D) LOC can be computed before a design is completed.

Ans: A

Q.61 Top down approach is used for
(A) development. **(B)** identification of faults.
(C) testing and validation. **(D)** reverse engineering.
Ans: A

Q.62 Which of the following is <u>not</u> an attribute of software engineering
(A) Efficiency. **(B)** Scalability.
(C) Dependability. **(D)** Usability.

Ans: C

Q.63 A key concept of quality control is that all work products
(A) are delivered on time and under budget
(B) have complete documentation
(C) have measurable specification for process outputs
(D) are thoroughly tested before delivery to the customer

Ans: C

Q.64 The ISO quality assurance standard that applies to software engineering is
(A) ISO 9000 **(B)** ISO 9001
(C) ISO 9002 **(D)** ISO 9003

Ans: B

Q.65 What types of models are created during software requirements analysis?
(A) Functional and behavioral **(B)** Algorithmic and data structure
(C) Architectural and structural **(D)** Usability and reliability

Ans: A

Q.66 What is the normal order of activities in which software testing is organized?
(A) unit, integration, system, validation
(B) system, integration, unit, validation
(C) unit, integration, validation, system
(D) none of the above

Ans: A

Q.67 Software feasibility is based on which of the following
(A) business and marketing concerns
(B) scope, constraints, market
(C) technology, finance, time, resources
(D) technical prowess of the developers

Ans: C

Q.68 FP-based estimation techniques require problem decomposition based on
(A) information domain values **(B)** project schedule
(C) software functions **(D)** process activities

Ans: C

Q.69 The software metrics chosen by an organization are driven by the business or technical goals an organization wishes to accomplish.
(A) True **(B)** False

Ans: A

Q.70 The goal of quality assurance is to provide management with the data needed to determine which software engineers are producing the most defects.
(A) True **(B)** False

Ans: B

Q.71 In the context of requirements analysis, partitioning results in the elaboration of data, function, or behavior.
(A) True **(B)** False

Ans: A

Q.72 Units and stubs are not needed for unit testing because the modules are tested independently of one another
(A) True **(B)** False

Ans: A

PART II

DESCRIPTIVES

Q 1 Define a software process. How do software myths affect a software process ?

Ans: Software process is a Coherent set of activities for specifying, designing, implementing and testing **software** systems.
Software myths propagates misinformation and confusion. Software myths had a no of attributes. For instance, they appeared to be reasonable statements of fact, they had an intuitive feel , and they were often promulgated by experienced practitioners who "know the score" .

Q 2 What is the advantage of using prototype software development model instead of waterfall model? Also explain the effect of defining a prototype on the overall cost of the software project?

Ans:The waterfall model: This is the classic SDLC model, with a linear and sequential method that has goals for each development phase. The waterfall model simplifies task scheduling, because there are no iterative or overlapping steps. One drawback of the waterfall is that it does not allow for much revision.
The prototyping model: In this model, a prototype (an early approximation of a final system or product) is built, tested, and then reworked as necessary until an acceptable prototype is finally achieved from which the complete system or product can now be developed.

Q 3 How does the risk factor affect the spiral model of software development?

Ans: Risk Analysis phase is the most important part of "Spiral Model". In this phase all possible (and available) alternatives, which can help in developing a cost effective project are analyzed and strategies are decided to use them. This phase has been added specially in order to identify and resolve all the possible risks in the project development. If risks indicate any kind of uncertainty in requirements, prototyping may be used to proceed with the available data and find out possible solution in order to deal with the potential changes in the requirements.

Q 4 Define software reliability. What is the difference between hardware & software reliability?

Ans:Software reliability is the probability that software will provide failure-free operation in a fixed environment for a fixed interval of time. Probability of failure is the probability that the software will fail on the next input selected. **Software reliability** is typically measured per some unit of time, whereas probability of failure is generally time independent. These two measures can be easily related if you know the frequency with which inputs are executed per unit of time. Mean-time-to-failure is the average interval of time between failures; this is also sometimes referred to as Mean-time-before-failure.
Software reliability tends to change continually during test periods. While hardware reliability may change during certain periods such as initial burn in or the end of useful life however it has a much greater tendency then software value.

Hardware faults are not physical faults whereas software faults are design faults that are harder to visualise, classify, detect and correct. In reality, the division between hardware and software reliability is somewhat artificial. Both may be defined in the same way therefore one may combine hardware and software reliability to get system reliability. The source of failure in hardware has generally being physically deterioration.

Q 5 Explain the following reliability model:
(i) Basic model (ii) Logarithmic (iii) Poisson Model

Ans: **(i) Basic Execution Time Model**
The model was developed by J.D. MUSA (MUSA79) in 1979 and is based on execution time. It is assumed that failures may occur according to a non-homogeneous Poisson Process (NHPP) Real world events may be described using Poisson processes. Example of Poisson processes are:

- Number of telephone calls expected in a given period of time
- Expected number of road accidents in a given period of time
- Expected number of persons visiting in a shopping mall in a given period of time.

In this model, the decrease in failure intensity, as a function of the number of failures observed, is constant and is given as:

$$\lambda(\mu) = \lambda_0 \left(1 - \frac{\mu}{V_0}\right)$$

Where: Initial failure intensity at the start of execution.
V_0: Number of failures experienced, if program is executed for
Infinite time period.
μ : Average or expected number of failures experienced at a given point in time.
This model implies a uniform operational profile. If all input classes are selected equally often, the various faults have an equal probability of manifesting themselves. The correction of any of those faults then contributes an equal decrease in the failure intensity. The negative sign shown that there is a negative slope meaning thereby a decrementing trend in failure intensity.

(ii) & (iii) Logarithmic Poisson Execution Time Model
This model is also developed by Musa et. Al. (MUSA79). The failure intensity function is different here as compared to Bias model. In this case, failure intensity function (decrement per failure) decreases exponentially whereas it is constant for basic model.
This failure intensity function is given as:
$\lambda(\mu) = \lambda_0 \exp(-\theta\mu)$
Where θ is called the failure intensity decay parameter.
The θ represents the relative change of allure intensity per failure experienced. Failure intensity of the logarithmic Poisson execution time model drops more rapidly initially than that of the basic model and later it drops more slowly.
The expression for failure intensity is given as:

$$\lambda(\iota)=\lambda_0/(\lambda_0\theta\iota+1)$$

The relations for the additional number of failures and additional execution time in this model are:
Hence, at larger values of execution time, the logarithmic Poisson model will have larger values of failure intensity than the basic model.

Q 6 Distinguish software faults and software failures

Ans: In case of a failure, the software does not do what the user expects. A fault is a programming error that may or may not actually manifest as a failure. A fault can also be described as an error in the correctness of the semantic of a computer program. A fault will become a failure if the exact computation conditions are met, one of them being that the faulty portion of computer software executes on the CPU. A fault can also turn into a failure when the software is ported to a different hardware platform or a different compiler, or when the software gets extended.

Q 7 Why is SRS also known as the blackbox specification of system ?

Ans: SRS document is a contract between the development team and the customer.
Once the SRS document is approved by the customer, any subsequent controversies are settled by referring the SRS document. The SRS document is known as black-box specification. Since the system is considered as a black box whose internal details are not known and only its visible external (i.e. input/output) behaviour is documented. SRS document concentrates on what needs to be done and carefully avoids the solution ("how to do") aspects. The SRS document serves as a contract between development team and the customer. SRS should be carefully written. The requirements at the SRS stage are written using end-user terminology. If necessary later a formal requirement specification may be developed from it.

Q 8 Write short notes on
(i) Configuration Management
(ii) Key process areas of Capability Maturity model(CMM)

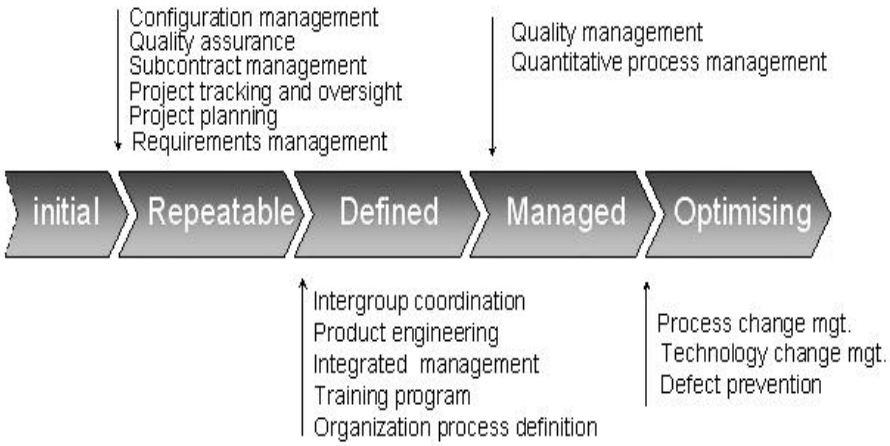
Ans: (i) **Software configuration** management is a set of tracking and control activities that begin when a software project begins and terminate only when the software is taken out of operation. **Configuration control** has the following meanings:

- The management of security features and assurances through control of changes made to hardware, software, firmware, documentation, test, test fixtures and test documentation of an automated information system, throughout the development and operational life of a system. *Source Code Management* or *revision control* is part of this.
- The control of changes--including the recording thereof--that are made to the hardware, software, firmware, and documentation throughout the system lifecycle.
- The control and adaption of the evolution of complex systems. It is the discipline of keeping evolving software products under control, and thus contributes to satisfying quality and delay constraints. Software configuration management (or SCM) can be divided into two areas. The first (and older) area of SCM concerns the storage of the entities produced during the software development project, sometimes referred to as component repository management. The second area concerns the activities performed for the production and/or change of these entities; the term engineering support is often used to refer this second area.
- After establishing a configuration, such as that of a telecommunications or computer system, the evaluating and approving changes to the configuration and to the interrelationships among system components.
- In distributed-queue dual-bus (DQDB) networks, the function that ensures the resources of all nodes of a DQDB network are configured into a correct

dual-bus topology. The functions that are managed include the head of bus, external timing source, and default slot generator functions.

(ii) Key process areas of Capability Maturity model (CMM)

Predictability, effectiveness, and control of an organization's software processes are believed to improve as the organization moves up these five levels. While not rigorous, the empirical evidence to date supports this belief.



Except for Level 1, each maturity level is decomposed into several key process areas that indicate the areas an organization should focus on to improve its software process.

The key process areas at Level 2 focus on the project's concerns related to establishing basic project management controls. They are Requirements Management, Project Planning, Project Tracking and Oversight, Subcontract Management, Quality Assurance, and Configuration Management.

The key process areas at Level 3 address both project and organizational issues, as the organization establishes an infrastructure that institutionalizes effective engineering and management processes across all projects. They are Organization Process Focus, Organization Process Definition, Training Program, Integrated Management, Product Engineering, Intergroup Coordination, and Peer Reviews.

The key process areas at Level 4 focus on establishing a quantitative understanding of both the process and the work products being built. They are Quantitative Process Management and Quality Management.

The key process areas at Level 5 cover the issues that both the organization and the projects must address to implement continual, measurable software process improvement. They are Defect Prevention, Technology Change Management, and Process Change Management.

Each key process area is described in terms of the key practices that contribute to satisfying its goals. The key practices describe the infrastructure and activities that contribute most to the effective implementation and institutionalization of the key process area.

Q 9 Explain cause effect graphing .

Ans: *Cause-effect graphing* is a test case design technique that provides a concise representation of logical conditions and corresponding actions.

There are four steps:

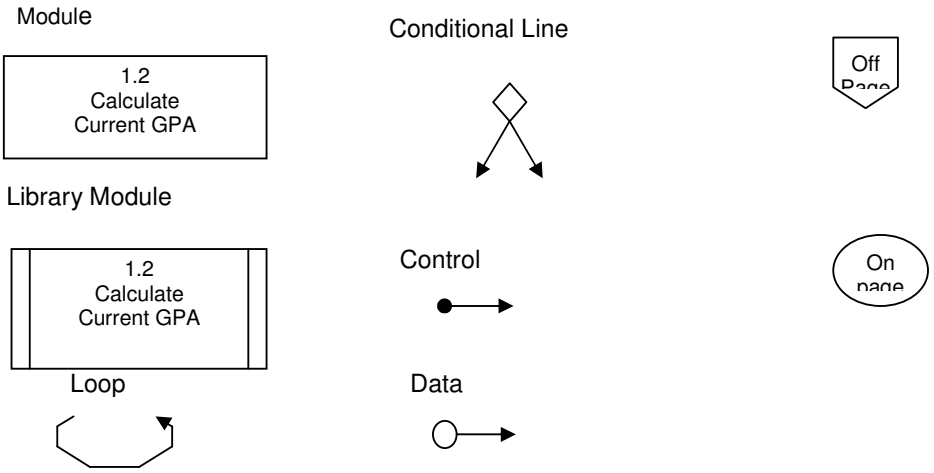
1. *Causes* (input conditions) and *effects* (actions) are listed for a module and an identifier is assigned to each.

2. A cause-effect graph is developed.
3. The graph is converted to a decision table.
4. Decision table rules are converted to test cases.

Q10 Write a short note on structure chart. **(6)**

Ans: Structure Chart is an important program design technique and shows all components of code in a hierarchical format.

Structure Chart Elements

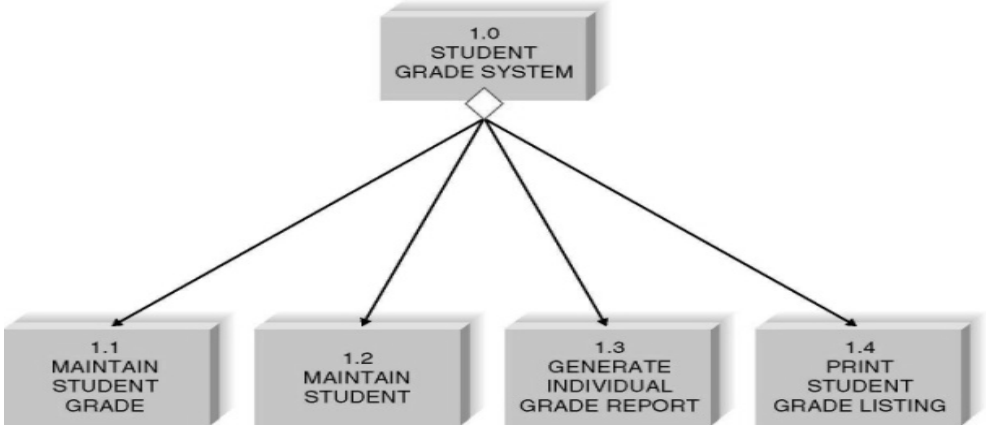


Building the Structure Chart

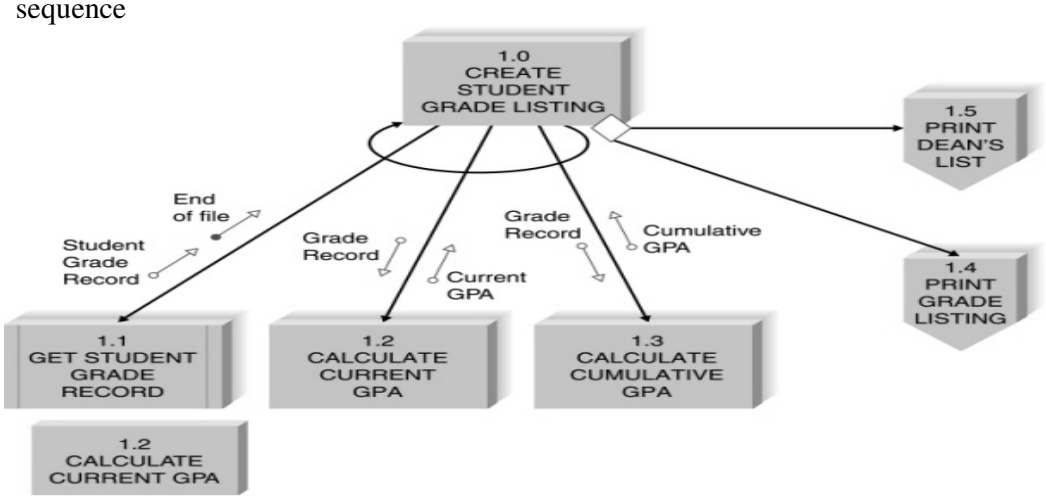
- Processes in the DFD tend to represent one module on the structure chart
  - Afferent processes – provide inputs to system
  - Central processes – perform critical system operations
  - Efferent processes – handle system outputs
- The DFD leveling can correspond to the structure chart hierarchy

Types of Structure Charts

- Transaction structure – control module calls subordinate modules, each of which handles a particular transaction
  - Many afferent processes
  - Few efferent processes
  - Higher up levels of structure chart
  - Using inputs to create a new output



- Transform structure – control module calls several subordinate modules in sequence



- Each subordinate performs a step in a process that transforms an input into an output
  - Few afferent processes
  - Many efferent processes
  - Lower levels of structure chart
  - Coordinates the movement of outputs

Steps in Building the Structure Chart

1. Identify top level modules and decompose them into lower levels
2. Add control connections
3. Add couples
4. Review and revise again and again until complete

Design Guidelines

- High quality structure charts result in programs that are modular, reusable and easy to implement.
- Measures include:
  - Cohesion
  - Coupling
  - Appropriate levels of fan-in and fan-out

Q 11 For the DFD given in Fig.2 design the structure chart. **(8)**

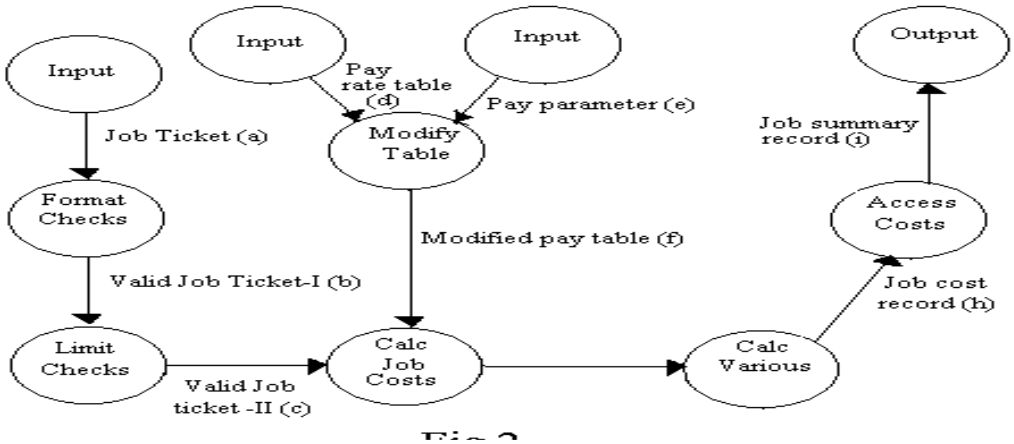
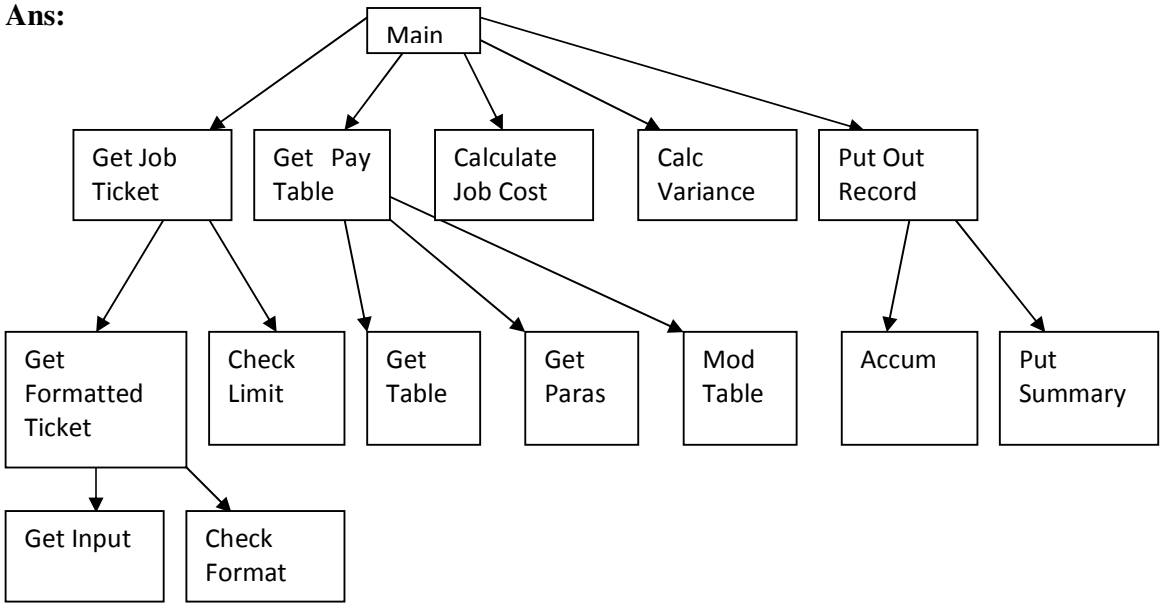


Fig.2

Ans:



Q12 Compare the basic COCOMO model with the detailed COCOMO model .

Ans. COCOMO consists of a hierarchy of three increasingly detailed and accurate forms.

- Basic COCOMO - is a static single-valued model that computes software development effort (and cost) as a function of program size expressed in estimated lines of code.
- Intermediate COCOMO - computes software development effort as function of program size and a set of "cost drivers" that include subjective assessment of product, hardware, personnel and project attributes.
- Embedded COCOMO - incorporates all characteristics of the intermediate version with an assessment of the cost driver's impact on each step (analysis, design, etc.) of the software engineering process.

Q 13 Define software testing. Explain various level of testing .

Ans: Software testing is a process used to help identify the correctness, completeness and quality of developed computer software. With that in mind, testing can never completely establish the correctness of computer software. Only the process of formal verification can prove that there are no defects.

"Testing is the process of demonstrating that defects are not present in the application that was developed."

"Testing is the activity or process which shows or demonstrates that a program or system performs all intended functions correctly."

"Testing is the activity of establishing the necessary "confidence" that a program or system does what it is supposed to do, based on the set of requirements that the user has specified."

Testing is a process of executing a program with the intent of finding an error. various level of testing are :

Unit Testing

Integrating testing

Validation testing
System testing

Q 14 Differentiate between functional testing and structural testing.

Ans: Functional testing means behavioural testing or **Black box** testing. In this techniques, tester design test case with the behaviour of the modules.
Structural testing means **White Box** testing. In this testing, tester design test cases from the structure of the module**.**

Q 15 Explain some of the limitations of testing.

Ans: Though testing is an important part of system development and leads to a valid, verified and efficient system, it also faces some limitation in its scope. **Following are some of such limitations.**

- It is very difficult to trace out logical errors through Testing.
- Stress testing or load tests are not the realistic options and hence, it cannot be defined that how application or module will be reacting at heavy data loads.
- In Integration testing, skeletons of different modules are used, which cannot describe the full functioning and in-turn the complete behavior of module they are representing.
- Being performed at later stages, testing may lead to a complete re-development of the module under testing and hence putting all effects in vain.

Q 16 Why is maintenance of a software important? Discuss some of the problems that are faced during maintenance of software.

Ans: The modification of a software product, after delivery, to correct faults, to improve performance or other attributes, or to adapt the product to a changed environment.
Maintenance is an important part of the software life-cycle. It is expensive in manpower and resources, and one of the aims of software engineering is to reduce its cost.
The most important problem during maintenance is the before correcting or modifying a program, the programmer must first understand it.
The problems are:

- Often another person or group of persons working over the years in isolation from each other writes the program.
- Often the program is changed by person who did not understand it clearly, resulting in a deterioration of the program's original organization.
- There is a high staff turnover within information technology industry. Due to this persons who are not the original authors maintain many systems. These persons may not have adequate knowledge about the system.
- Some problems only become clearer when a system is in use. Many users know what they want but lack the ability to express it in a form understandable to programmers. This is primarily due to information gap.

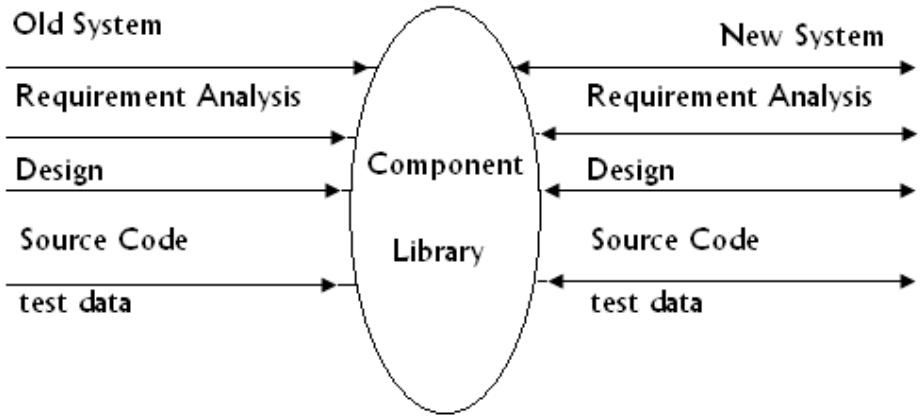
Q 17 Explain the reuse maintenance model with the help of a diagram.

Ans: Reuse maintenance model: This model is based on the principal that maintenance could be viewed as an activity involving the reuse. of existing program components.

The reuse model has four main steps:

1. Identification of the parts of the old system that are candidates for reuse.
2. Understanding these system parts.
3. Modification of the Old system parts appropriate to the new requirements.
4. Integration of the modified parts into the new system.

With the five reuse model, the starting point may be any phase of the life cycle. The requirements, the design, the code or the test data :- unlike other models.



Q 18 Differentiate between object oriented and function oriented design .

Ans:• Function-oriented design relies on identifying functions which transform their inputs to create outputs. In most systems, functions share some global system state.

• The functional design process involves identifying data transformations in the system, decomposing functions into a hierarchy of sub-functions, describing the operation and interface of each system entity and documenting the flow of control in the system.

• Data-flow diagrams are a means of documenting end-to-end data flow through a system. They do not include control information. Structure charts are a way of representing the hierarchical organization of a system. Control may be documented using a program description language (PDL).

• Data-flow diagrams can be implemented directly as a set of cooperating sequential processes. Each transform in the data-flow diagram is implemented as a separate process. Alternatively, they can be realized as a number of procedures in a sequential program.

• Functional design and object-oriented design usually result in totally different system decompositions. However, the most appropriate design strategy is often a heterogeneous one where both functional and object-oriented approaches are used.

Q 19 How is software design different from coding ?

Ans: Points of difference between software design and coding can be laid down as under:

Design :

1. Design is most crucial and time-consuming activity
2. Screen of the system depends on the correct design specifications which is a key activity of the process.
3. Software design is based on the findings collected in the initial investigation phase.
4. Design includes the following:
   (i) User interface design
   (ii) Process Design
   (iii) Database design
5. Designs are transformed into actual code or program during the implementation phase.
6. it is more feasible to rectify design as different users may have conflicting user requirements and only the final and valid design goes for next phase.

Coding:-

1. Involves conversion of detailed design specification laid out by designers into actual code, files or database.
2. Less time consuming then the design phase and performed by programmers or coders.
3. More concerned with technical aspect of the software rather than its functional aspect.
4. Different software such as programming languages front-end tools, database management system, utilities etc are used to facilitate the coding process.

Q 20 What problems arise if two modules have high coupling?

Ans: Coupling means the interconnection of different modules with each other or we can say, it tells about the interrelationship of different modules of a system. A system with high coupling means there are strong interconnections between its modules. If two modules are involved in high coupling, it means their interdependence will be very high. Any changes applied to one module will affect the functionality of the other module. Greater the degree of change, greater will be its effect on the other. As the dependence is higher, such change will affect modules in a negative manner and in-turn, the maintainability of the project is reduced. This will further reduce the reusability factor of individual modules and hence lead to unsophisticated software. So, it is always desirable to have inter-connection & interdependence between modules.

Q 21 **Explain**

a. the activities of software maintenance. (2)
b. key process areas of CMM. (6)

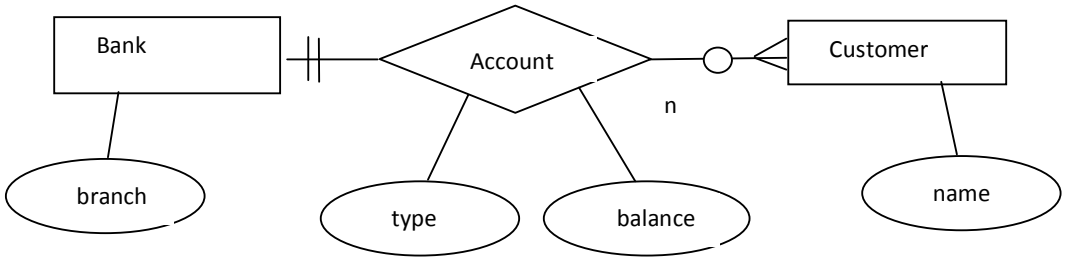
Ans:

a) Software maintenance is a broad activity that includes error correction, enhancement of capabilities, deletion of obsolete capabilities and optimization.
b) Key process areas of CMM are
   1. Requirements management which establishes a common relationship between the customer and developer
   2. Software project planning where reasonable plans for managing the software project are established

3. Software project tracing and oversight which establishes adequate visibility into actual progress to help management track deviations from plans
4. Software subcontract management for selecting efficient subcontractors and managing them
5. Software quality assurance which provides visibility into the software process and product
6. Software configuration management which establishes and maintains the integrity of the products throughout the life cycle.

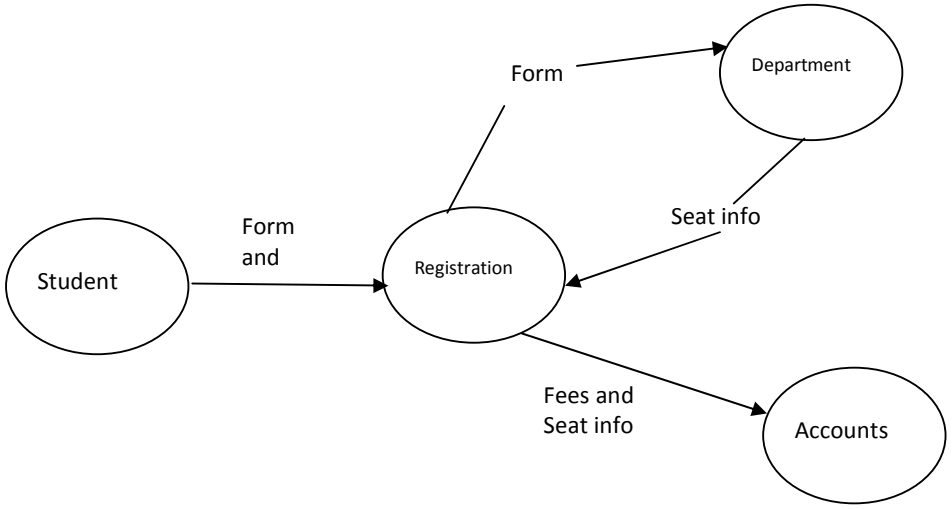
Q 22 **Draw E-R diagram for the following situation**
An account is a relationship between customer and bank. A customer has a name. A bank has a branch. A customer may have several accounts of different type and balance. **(8)**

Ans:



Q 23 **Consider a program which registers students for different programs. The students fill a form and submit it. This is sent to the departments for confirmation. Once it is confirmed, the form and the fees is sent to the account section. Draw a data flow diagram using SRD technique.** **(8)**

Ans:



Q 24 What are the three activities of risk assessment? (4)

Ans: The three activities are identifying, analyzing and giving priorities. Risks can be identified by a check list or looking at previously identified risks. Risk analysis involves examining how project outcomes may change with modification of risk input variables. Risk prioritization helps in focus on most severe risks.

Q 25 What is a modular system? List the important properties of a modular system. (6)

Ans: A modular system consists of well defined manageable units to well defined interfaces among them. Desirable properties are

- Each module is a well defined subsystem useful to others
- Each module has a well defined single purpose
- Modules can be separately compiled and stored in library
- Modules can use other modules
- Modules should be easier to use than build
- Modules should have a simple interface

Q 26 Define the following:

a. Aggregation among objects.
b. Class.
c. Repeated inheritance.
d. Encapsulation.
e. Scenario. (2 x 5)

Ans:

a).Aggregation among objects

Aggregation among objects represents a relationship. It is a whole/part relationship. Aggregation may imply containment.

b).Class

A class is a template that specifies the properties of objects. Classes have an interface which consists of the operation, a body which implements the operations and instance variable which contain the state of an object.

c).Repeated inheritance

If a class inherits more than once from the same class then it is referred to as repeated inheritance.

d).Encapsulation

An object encapsulates the data and information it contains and supports a well defined abstraction. Encapsulation leads to the separation of the interface and implementation.

e).Scenario

A scenario is a sequence of events that occur in a particular execution of the system. A scenario can be described textually by enumerating the sequence of events or it can be shown as an event trace diagram in which events between objects are shown.

Q 27 Explain the following:

(i) Equivalence class testing. (6)
(ii) User and System documentation with examples. (6)

(iii) Core dumps. (4)

Ans:

(i) Equivalence class testing is based on partitioning the input domain of a program into a number of equivalence classes so that the test of a representative value of each class is equivalent to testing any other value. Two steps for this method are
Identify equivalence class by taking each input condition and partition into valid and invalid classes.
Generate the test cases using the equivalence class of the previous step. The test cases are generated for the valid and the invalid classes.

(ii) User documentation contains descriptions of the functions of a system without reference to how these functions are implemented. Examples are installation guide and reference guide. System documents contain all the facets of the system, including analysis, specification design, implementation, testing, security error diagnosis and recovery. Examples are SRS and system test plan.

(iii) Core dumps are a debugging technique. A printout of all relevant memory locations is obtained and studied. All dumps should be well documented and retained for possible use on subsequent problems. Its advantages are that a complete dump of memory at a crucial time is obtained Require CPU and I/O time and can get expensive if used indiscriminately. Sometimes it is difficult to interpret the dump which is represented using hexadecimal numbers.

Q 28 Define capability. What are the quantities that determine the degree of capability of a software reliability model? (6)

Ans: Capability refers to the ability of the model to estimate with satisfactory accuracy quantities needed by managers; engineers and users in planning and managing software development or running operational systems. The quantities are MTTF, expected date of reaching reliability, resource and cost requirements to reach the objectives.

Q.29 Explain the waterfall model. Explain why it is more advantageous than adhoc methods.

Ans Waterfall Model:

1.The **waterfall model** is a sequential software development process, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Initiation, analysis, design (validation), Construction, testing and maintenance.

2. The first formal description of the waterfall model is given by Winston W. Royce in 1950, however, popular in 1970, and further refined by Barry Boehm.

3. To follow the *waterfall model*, one proceeds from one phase to the next in a purely sequential manner. For example, one first completes requirements specifications, which are set in stone. When the requirements are fully completed, one proceeds to design.

4. Process structured as a cascade of phases where output of one is input of next.

5. Many variants of model depending on organization and specific project. However underlying phases are same for all.

Why Waterfall model is advantageous than Adhoc Methods

Ad-hoc Process Models-"Process capability is unpredictable because the software process is constantly changed or modified as the work progresses. Schedules, budgets, functionality, and product quality are generally inconsistent. Performance depends on the capabilities of individuals and varies with their innate skills, knowledge, and motivations. There are few stable software processes in evidence, and performance can be predicted only by individual rather than organizational capability." So to overcome this problem waterfall model provides the following advantages:

a. Waterfall model is simple to follow, however real projects rarely follows this approach.
b. Iteration is not required.
c. It is widely use because it is easy.

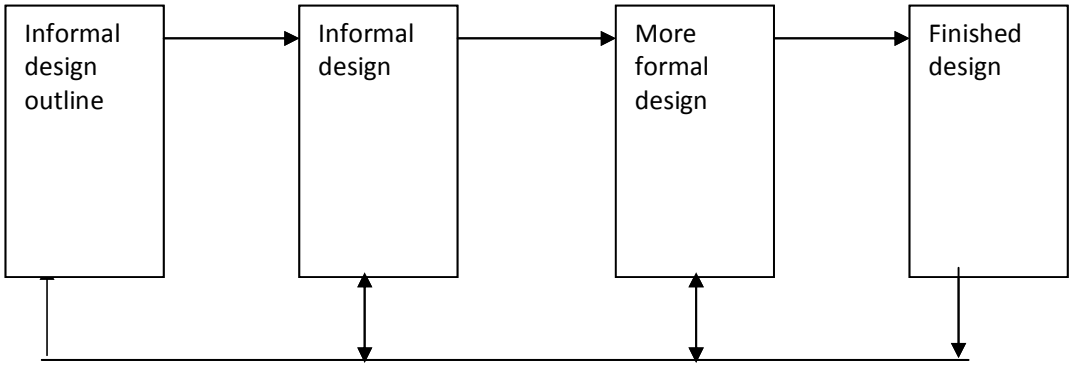
Q.30 What are the objectives of software design? How do we transform an informal design to a detailed design?

Ans Objectives of software design

The purpose of the design phase is to plan a solution of the problem specified by the requirements document. This phase is the first step in moving from the problem domain to the solution domain. In other words, starting with what is needed; design takes us toward how to satisfy the needs, so the basic objectives are:

- Identify different types of software, based on the usage.
- Show differences between design and coding.
- Define concepts of structured programming.
- Illustrate some basic design concepts.
- See how to design for testability and maintainability.

Non-formal methods of specification can lead to problems during coding, particularly if the coder is a different person from the designer that is often the case. Software designers do not arrive at a finished design document immediately but develop the design iteratively through a number of different phases. The design process involves adding details as the design is developed with constant backtracking to correct earlier, less formal, designs. The transformation is done as per the following diagram.



Q.31. Discuss the important issues that a SRS must address.

Ans. SRS is the process of establishing the services, the system should provide of the system in new, it is difficult for software engineer to understand the nature of problem constraints under which it must operates, so software engineer take help of a step called requirement capture and analysis. It is first formal document

produce in the software development process and it serves as a basis for the contract between a producer and a software developer/supplier.

The important issues that a SRS must address are:

(a) System goals and requirements are different: Goal is a more general characteristics." e. g. Whole system should be designed in a user friendly manner or more robust system".
Requests are more testable in nature. For example all users command selection should be only using pop up menus.

(b) Request Definition: A statement in a natural language stating what services the system should expected to provide. It should be understandable by clients, contractors, management & users.

(c) Request Specification: A structured document maintaining the services in more details than definition, and precise enough to act as a contract. It should be understandable by technical staff both at a developer and producer's place.

(d) Software Specification (Design Specification): It is an abstract design description of the software which is the basis for design and implementation. There should be a clear relationship between this documents and the software request specification. The reader of this document is software engineer, system analyst and project leaders.

(e) Requirement Capture and Analysis: It is the process of designing the system request captures through: -
(i) Observation of existing system.
(ii) Discussion with potential users, producers.
(iii) Personal interviews and task analysis.
(iv) Standard published documents/reports from the user.

(f) Feasibility Study:
(i) It is estimate made weather the identified user needs can be satisfied using the current technology.
(ii) Weather the proposed system is cost effective.
(iii) Weather it is possible to develop the system within the budgetary and time constraints.

(g) Suggestions for preparing an SRS Document:
(i) Only specify external system behavior.
(ii)Specifying the constraints on implementation.
(iii)It should record for thought about the life cycle of the system.
(iv) It should characterize acceptable responses to the undesired events.

Q.32 Explain throw-away prototyping and evolutionary prototyping. Discuss the differences between the two.

Ans Throw-Away Prototyping: Also called **close ended prototyping**. Throwaway or Rapid Prototyping refers to the creation of a model that will eventually be discarded rather than becoming part of the final delivered software. Rapid Prototyping involved creating a working model of various parts of the system at a very early stage, after a relatively short investigation. The method used in building it is usually quite informal, the most important factor being the speed with which the model is provided. The model then becomes the starting point from which users can re-examine their expectations and clarify their requirements. When

this has been achieved, the prototype model is 'thrown away', and the system is formally developed based on the identified requirements.

The most obvious reason for using Throwaway Prototyping is that it can be done quickly. If the users can get quick feedback on their requirements, they may be able to refine them early in the development of the software. Speed is crucial in implementing a throwaway prototype, since with a limited budget of time and money little can be expended on a prototype that will be discarded. Strength of throwaway prototyping is its ability to construct interfaces that the users can test. The user interface is what the user sees as the system, and by seeing it in front of them, it is much easier to grasp how the system will work.

Evolutionary prototyping: Evolutionary Prototyping (also known as breadboard prototyping) is quite different from Throwaway Prototyping. The main goal when using Evolutionary Prototyping is to build a very robust prototype in a structured manner and constantly refine it. The reason for this is that the Evolutionary prototype, when built, forms the heart of the new system, and the improvements and further requirements will be built. When developing a system using Evolutionary Prototyping, the system is continually refined and rebuilt. "Evolutionary prototyping acknowledges that we do not understand all the requirements and builds only those that are well understood."

Evolutionary Prototypes have an advantage over Throwaway Prototypes in that they are functional systems. Although they may not have all the features the users have planned, they may be used on an interim basis until the final system is delivered. In Evolutionary Prototyping, developers can focus themselves to develop parts of the system that they understand instead of working on developing a whole system.

Q.33 Explain the cost drivers and EAF of the intermediate COCOMO model.

Ans. There are 15 different attributes, called cost drivers attributes that determine the multiplying factors. These factors depend on product, computer, personnel, and technology attributes also know as project attributes. Of the attributes are required software reliability (RELY), product complexity (CPLX), analyst capability (ACAP), application experience (AEXP), use of modern tools (TOOL), and required development schedule (SCHD). Each cost driver has a rating scale, and for each rating, there is multiplying factor is provided. For eg. for the product attributes RELY, the rating scale is very low, low, nominal, high, and very high. The multiplying factor for these ratings is .75, .88, .1.00, .1.15, and 1.40, respectively. So, if the reliability requirement for the project is judged to be low then the multiplying factor is .75, while if it is judged to be very high the factor is 1.40. The attributes and their multiplying factors for different ratings are shown in the table below

	Ratings					
Cost Drivers	Very Low	Low	Nominal	High	Very High	Extra High
Product attributes						
Required software reliability	0.75	0.88	1.00	1.15	1.40	
Size of application database		0.94	1.00	1.08	1.16	
Complexity of the product	0.70	0.85	1.00	1.15	1.30	1.65
Hardware attributes						

Run-time performance constraints			1.00	1.11	1.30	1.66
Memory constraints			1.00	1.06	1.21	1.56
Volatility of the virtual machine environment		0.87	1.00	1.15	1.30	
Required turnabout time		0.87	1.00	1.07	1.15	
Personnel attributes						
Analyst capability	1.46	1.19	1.00	0.86	0.71	
Applications experience	1.29	1.13	1.00	0.91	0.82	
Software engineer capability	1.42	1.17	1.00	0.86	0.70	
Virtual machine experience	1.21	1.10	1.00	0.90		
Programming language experience	1.14	1.07	1.00	0.95		
Project attributes						
Use of software tools	1.24	1.10	1.00	0.91	0.82	
Application of software engineering methods	1.24	1.10	1.00	0.91	0.83	
Required development schedule	1.23	1.08	1.00	1.04	1.10	

Q.34 Compare the following

(i) Productivity and difficulty
(ii) Manpower and development time
(iii) Static single variable model and static multivariable model
(iv)Intermediate and Detailed COCOMO model

Ans.

(i) Productivity and difficulty

Productivity refers to metrics and measures of output from production processes, per unit of input. Productivity P may be conceived of as a metrics of the technical or engineering efficiency of production. In software project planning , productivity is defined as the number of lines of code developed per person-month

Difficulty The ratio (K/t_d^2), where K is software development cost and t_d is peak development time, is called difficulty and denoted by D, which is measured in person/year.

$D= (K/t_d^2)$

The relationship shows that a project is more difficult to develop when the manpower demand is high or when the time schedule is short.

Putnam has observed that productivity is proportional to the difficulty

$$P\alpha D^{\beta}$$

(ii) Manpower and development time

Manpower may refer to labour Manpower, either an abstract term for human labour effort (as opposed to machines, animals etc.) or the number of human productive units available/needed for professional or other tasks, also used when referring to such personnel as a resource (e.g. "a manpower shortage") and development time is the time required to develop a project.

The **Norden/ Reyleigh** equation represents manpower, measured in person per unit time as a function of time. It is usually expressed in person-year/year(PY/YR).

(iii) Static single variable model and static multivariable model

Static single variable model: Methods using this model use an equation to estimate the desired value such as cost, time, effort etc. They all depend on same variable used as predictor (say, size). An example of the most common equation is

$$C = a L^b$$

Where C is the cost (effort expressed in the unit of manpower, for e.g. persons-months) and L is the size generally given in the line of code (LOC). The constants a and b are derived from the historical data of the organization. Since a and b depends on the local development environment, these models are not transportable to different organizations.

Static multivariable model: They depend on several variables representing various aspects of the software development environment, for e.g. methods used, user participation, customer oriented changes, memory constraints, etc. The model provides relationship between delivered line of source code (L in thousand lines) and effort (E in person-months) and is given by the following equation:

$$E = 5.2 L^{0.91}$$

In the same fashion, duration of development (D in months) is given by

$$D = 4.1 L^{0.36}$$

(iv) Intermediate and Detailed COCOMO model

Intermediate COCOMO computes software development effort as function of program size and a set of "cost drivers" that include subjective assessment of product, hardware, personnel and project attributes. This extension considers a set of four "cost drivers", each with a number of subsidiary attributes:-

- Product attributes
  - Required software reliability
    - Size of application database
    - Complexity of the product
- Hardware attributes
  - Run-time performance constraints
  - Memory constraints
  - Volatility of the virtual machine environment
  - Required turnabout time
- Personnel attributes
  - Analyst capability
  - Software engineering capability
  - Applications experience
  - Virtual machine experience
  - Programming language experience
- Project attributes
  - Use of software tools
  - Application of software engineering methods
  - Required development schedule

Each of the 15 attributes receives a rating on a six-point scale that ranges from "very low" to "extra high" (in importance or value). An effort multiplier from the table below applies to the rating. The product of all effort multipliers results in an *effort adjustment factor (EAF)*. Typical values for EAF range from 0.9 to 1.4.

	Ratings					
Cost Drivers	Very Low	Low	Nominal	High	Very High	Extra High

Product attributes						
Required software reliability	0.75	0.88	1.00	1.15	1.40	
Size of application database		0.94	1.00	1.08	1.16	
Complexity of the product	0.70	0.85	1.00	1.15	1.30	1.65
Hardware attributes						
Run-time performance constraints			1.00	1.11	1.30	1.66
Memory constraints			1.00	1.06	1.21	1.56
Volatility of the virtual machine environment		0.87	1.00	1.15	1.30	
Required turnabout time		0.87	1.00	1.07	1.15	
Personnel attributes						
Analyst capability	1.46	1.19	1.00	0.86	0.71	
Applications experience	1.29	1.13	1.00	0.91	0.82	
Software engineer capability	1.42	1.17	1.00	0.86	0.70	
Virtual machine experience	1.21	1.10	1.00	0.90		
Programming language experience	1.14	1.07	1.00	0.95		
Project attributes						
Use of software tools	1.24	1.10	1.00	0.91	0.82	
Application of software engineering methods	1.24	1.10	1.00	0.91	0.83	
Required development schedule	1.23	1.08	1.00	1.04	1.10	

The Intermediate Cocomo formula now takes the form:

$E=a_i(KLOC)^{(b_i)}.EAF$

where E is the effort applied in person-months, **KLOC** is the estimated number of thousands of delivered lines of code for the project, and **EAF** is the factor calculated above. The coefficient $\mathbf{a_i}$ and the exponent $\mathbf{b_i}$ are given in the next table.

Software project	a_i	b_i
Organic	3.2	1.05
Semi-detached	3.0	1.12
Embedded	2.8	1.20

The Development time **D** calculation uses **E** in the same way as in the Basic COCOMO.

Detailed COCOMO

Detailed COCOMO - incorporates all characteristics of the intermediate version with an assessment of the cost driver's impact on each step (analysis, design, etc.) of the software engineering process.

Q.35 List the points of a simplified design process.

Ans. A design process may include a series of steps followed by designers. Depending on the product or service, some of these stages may be irrelevant, ignored in real-world situations in order to save time, reduce cost, or because they may be redundant in the situation. Typical stages of the simplified design process include:

- **Pre-production design**
  - Design brief - an early often the beginning statement of design goals
  - Analysis- analysis of current design goals
  - Research-investigating similar design solutions in the field or related topics
  - Specification - specifying requirements of a design solution for a product (product design specification) or service.
  - Problem solving - conceptualizing and documenting design solutions
  - Presentation - presenting design solutions
- **Design during production**
  - Development - continuation and improvement of a designed solution
  - Testing – on-site testing a designed solution
- **Post-production design feedback for future designs**
  - Implementation - introducing the designed solution into the environment
  - Evaluation and conclusion - summary of process and results, including constructive criticism and suggestions for future improvements
- **Redesign** - any or all stages in the design process repeated (with corrections made) at any time before, during, or after production.

Q.36 Explain the following with the help of an example

(i) Common coupling
(ii) Communicational cohesion
(iii) Class diagram
(iv) Structure chart

Ans.

(i) Common coupling: Common coupling is when two modules share the same global data (e.g. a global variable). Changing the shared resource implies changing all the modules using it. Diagnosing problems in structures with considerable common coupling is time consuming and difficult. However, this does not mean that the use of global data is necessarily ”bad”. It does not mean that a software designer must be aware of potential consequences of common coupling and take special care to guard against them.

(ii)Communicational cohesion: Communicational cohesion is when parts of a module are grouped because they operate on the same data (e.g. a module which operates on the same record of information). In this all of the elements of a component operate on the same input data or produce the same output data. So we can say if a module performs a series of actions related be a sequence of steps to be followed by the product and all actions to be performed on the same data.

(iii)Class diagram: A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes. The UML specifies **two types of scope for members: instance and classifier.** In the case of instance members, the scope is a specific instance. For attributes, it means that its value can vary between instances. For methods, it means that its invocation affects the instance state, in other words, affects the instance attributes. Otherwise, in the classifier member, the scope is the class. For attributes, it means that its value is equal for all instances. For methods, it means that its invocation do not affect the instance state. Classifier members are commonly recognized as "static" in many programming languages.

(iv) Structure chart: A structure chart is a top-down modular design tool, constructed of squares representing the different modules in the system, and lines that connect them. The lines represent the connection and or ownership between activities and sub activities as they are used in organization charts.
In structured analysis structure charts are used to specify the high-level design, or architecture, of a computer program. As a design tool, they aid the programmer in dividing and conquering a large software problem, that is, recursively breaking a problem down into parts that are small enough to be understood by a human brain. A structure chart depicts

- the size and complexity of the system, and
- number of readily identifiable functions and modules within each function and
- Whether each identifiable function is a manageable entity or should be broken down into smaller components.

Q.37. Explain Equivalence Class Partitioning and Boundary value analysis. Compare the two.

Ans. Equivalence Class Partitioning: It is a technique in black box testing. It is designed to minimize the number of test cases by dividing tests in such a way that the system is expected to act the same way for all tests of each equivalence partition. Test inputs are selected from each class. Every possible input belongs to one and only one equivalence partition." In another words it is a method that can help you derive test cases. You identify classes of input or output conditions. The rule is that each member in the class causes the same kind of behaviour of the system. In other words, the "Equivalence Class Partitioning" method creates sets of inputs or outputs that are handled in the same way by the application. **E.g.:** If you are testing for an input box accepting numbers from 1 to 1000 then there is no use in writing thousand test cases for all 1000 valid input numbers plus other test cases for invalid data. Using equivalence partitioning method above test cases can be divided into three sets of input data called as classes. Each test case is a representative of respective class.
Boundary Value Analysis: It's widely recognized that input values at the extreme ends of input domain cause more errors in system. More application **errors occur at the boundaries** of input domain. 'Boundary value analysis' testing technique is used to identify errors at boundaries rather than finding those exist in centre of input domain. Boundary value analysis is a next part of Equivalence partitioning for designing test cases where test cases are selected at the edges of the equivalence classes. **E.g.** if you divided 1 to 1000 input values in valid data equivalence class, then you can select test case values like: 1, 11, 100, 950 etc.

Q.38 Explain the following terms with reference to software reliability models.
(i) Predictive validity
(ii) Resource usage
(iii) Applicability

Ans. (i) **Predictive validity:** It is the capability of the model to predict future failure behaviour from present and past failure behaviour. This capability is significant only when failure behaviour is changing. There are two general

ways of viewing predictive validity based on the two equivalent approaches to characterizing failure random process, namely;

1. the number of failure approach and
2. the failure time approach.

We can visually check the predictive validity by plotting the relative error against the normalised test time.

(ii) Resource usage: It is linearly proportional to execution time τ and mean failures experienced μ. Let X_r be the usage of resource r. Then

$X_r=\theta_r\tau+\mu_r\mu$

Where θ_r is the resource usage per CPU hr. It is nonzero for failure identification personnel (θ_i) and computer time (θ_c). The quantity μ_r is the resource usage per failure. It is nonzero for failure identification personnel (μ_i), failure correction personnel (μ_f) and computer time (μ_c).

(iii) Applicability: It is another important characteristic of a model. The model should be judged by its degree of applicability across software products that vary in size, structure and function. The model should be usable in different development environments, different operational environments, and different life cycle phases. It is desirable that a model should be robust with respect to deviations from its assumptions, errors in the data or parameters it employs, and unusual conditions.

Q.39 What are the assumptions of the execution-time component model? Compare the execution-time component for basic model and the logarithmic Poisson model.

Ans. The execution-time component is based on the following assumptions:

(1) Tests represent the environment in which the program will be used.
(2) All failures are observed.
(3) Failure intervals are s-independent of each other.
(4) Hazard rate is a constant that changes at each fault correction.
(5) The failure rate is proportional to the s-expected value of the number of faults remaining.
(6) The fault-correction occurrence rate is proportional to the failure-occurrence rate. Both rates are with respect to execution time.

The two models-basic execution time model and logarithmic Poisson execution time model- have failure intensity functions that differ as functions of execution time. However, the difference between them is best defined in terms of slope or decrement per failure experienced. The decrement in the failure intensity function remains constant for basic model while the decrement per failure decreases exponentially as shown in the following graph.

Q.40. Describe the various types of restructuring techniques. How does restructuring help in maintaining a program?

Ans. Software restructuring modifies source code and / or data an effort to make it amenable to future changes. In general, restructuring does not modify the overall program architecture. It tends to focus on the design details of individual modules and on local data structures define within the module. There are following types of restructuring techniques:

(a) **Code Restructuring**: It is performed to yield a design that produces the same function but with higher quality than the original program. In general, code-restructuring techniques model program logic using Boolean algebra and then apply a series of transformation rules that yield restructured logic. The objective is to take "spaghetti-bowl" code and derive a procedural design that conforms to the structured programming philosophy.

(b)**Data Restructuring**: Before data restructuring begins, a reverse engineering activity called analysis of source code must be conducted. The intent is to extract data items and objects, to get information on data flow, and to understand the existing data structures that have been implemented. This activity is sometimes called data analysis. Once it has been completed, data redesign commences. Another form of redesign, called data rationalization, which ensures that all data naming conventions conform to local standards and that aliases are eliminated as data flow through the system.

The restructuring helps in maintaining a program in the following ways:

(a) Programs have higher quality, better documentation, less complexity, and conformance to modern software engineering practices and standards.

(b) Frustration among software engineers who must work on the program is reduced, thereby improving productivity and making learning easier.

(c) Effort requires performing maintenance activities is reduced.

(d) Software is easier to test and debug.

Q.41. Describe the various steps of the reuse-oriented model.

Ans. The reuse-oriented model, also called reuse-oriented development (ROD), is a method of software development in which a program is refined by producing a sequence of prototypes called models, each of which is automatically derived from the preceding one according to a sequence of defined rules.

The reuse-oriented model can reduce the overall cost of software development compared with more tedious manual methods. It can also save time because each phase of the process builds on the previous phase that has already been refined. When carefully carried out, ROD can minimize the likelihood of errors or bugs making their way into the final product.

The reuse-oriented model is not always practical in its pure form because a full repertoire of reusable components may not be available. In such instances, some new program components must be designed. If not thoughtfully done, ROD can lead to compromises in perceived requirements, resulting in a product that does not fully meet the needs of its intended users.

Q.42 What is ripple effect? How does it affect the stability of a program?

Ans. The **ripple effect** is a term used to describe a situation where, like the ever expanding ripples across water when an object is dropped into it, an effect from an initial state can be followed outwards incrementally. Examples can be found in economics where an individual's reduction in spending reduces the incomes of others and their ability to spend. In sociology it can be observed how social interactions can affect situations not directly related to the initial interaction. and in charitable activities where information can be disseminated and passed from community to community to broaden it's impact.

In software, the effect of a modification may not be local to the modification, but may also affect other portions of the program. There is a ripple effect from the location of the modification to the other parts of the programs that are affected by the modification. One aspect of the ripple effect concerns the performance of the program. The primary attribute affecting the ripple effect as a consequence of a program modification is the stability of the program. Program stability is defined as the resistance to the amplification of changes in the program.

Q.43 List four reasons why it is difficult to improve software process.

Ans It is difficult to improve software process due to following reasons:

1. Lack of knowledge-Many software developers are not aware of best practices of industry. In fact best practices available in literature are not being used widespread in software development.

2. Not enough time-There is always a shortage of time because upper management are always demanding more software of higher quality in minimum possible time. Unrealistic schedule sometimes leave insufficient time to do the essential project work.

3. Wrong motivations-The process improvement initiatives are taken for wrong reasons like sometimes contractor is demanding achievement of CMM or sometimes senior management is directing the organization to achieve CMM without a clear explanations why improvement was needed and its benefits.

4. Insufficient commitments-The software improvement fails due to lack of true commitment. Management sets no expectations from the development community regarding process improvement.

Q.44 Discuss the various strategies of design. Which design strategy is most popular and practical?

Ans The most commonly used software design strategy involved decomposing the design into functional components with system state information held in a shared data area.

The design strategies are:

1. **Functional design** The system is designed from a functional viewpoint, starting from with a high level view and progressively refining this into a more detailed design. The system state is centralised and shared between the functions operating on that state.
2. **Object-oriented design:** The system is viewed as a collection of objects rather than as functions. Object-oriented design is based on the idea of information hiding. In a object-oriented design, the system state is decentralized and each object manages is own state information

Q.45 Differentiate between structured analysis and structured design.

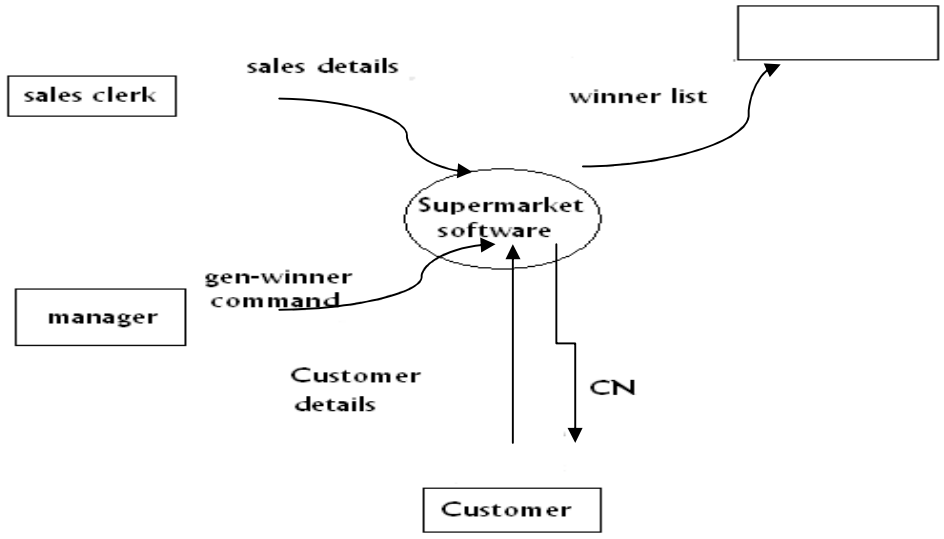
Ans. The aim of structured analysis is to transform the textual description of a problem into a graphic model. During structured analysis, functional decomposition of the system takes place, i.e. each function that the system performs is analyzed and hierarchically decomposed into more detailed functions. On the other hand, during structured design all functions identified during structured analysis are mapped to a module structure. This module structures is also

called software architecture for the given problem and it can be directly implemented using a conventional programming language.
The purpose of structured analysis is to capture the detailed structure of the system as perceived by the user, whereas the purpose of structured design is to define the structure of the solution that is suitable for implementation in some programming language.

Q.46 A supermarket needs to develop software to encourage regular customers. For this, the customer needs to supply his name, address, telephone number and driving license number. A customer is assigned a unique customer number (CN) by the computer. When a customer makes a purchase, the value of the purchase is credited against his CN. At the end of each year, surprise gifts to 10 customers who have made the highest purchase is given. In addition, a 22 carat gold coin is given to every customer who has made a purchase over Rs.10,000/-. The entries are reset on the last day of the year.

(i) Draw the context diagram
(ii) Give data dictionary entries for
- **address**
- **CN**
- **gold-coin-winner-list**
- **totalsales**

Ans.



Data dictionary entries for:
address: name + house# + street# + city + pin
name: string
house#: string
street#: string
city: string
pin: integer
CN: {integer|character|spl. chararacter}*
Gold-coin-winner-list: {address}*
Total sales: integer

Q.47 Explain the following:

(i) Information flow index.

(ii) Span and average span size for a program.

(iv) Function points.

Ans (i)Information Flow index Information flow metrics are applied to the components of a system design. For any component 'A', We can define three measures:

1: 'FAN IN' is simply a count of the number of other components that can all, or pass control, to component A.

2: 'FAN OUT' is the number of components that are called by component A.

3: using the following formula derives this. We will call this measure the INFORMATION FLOW index of component A , abbreviated as IF(A).

$IF(A) = \{FAN\ IN(A) * FAN\ OUT\ (A)\}^2$

(ii) Span and average span size of a program

Span size indicates the number of statements that pass between successive uses of a variable. For example in the following sample of code, variable 'x' has 4 spans of 10,12,7, and 6 statements while 'y' has 2 spans of 23 and 14 statements. This shows x is being used more than y.

```
...
21 scanf(x,y);
...
32 a=x;
...
45 b=x-y;
...
53 c=x;
...
60 printf(x,y);
```

Average span size=(Σ span size of a variable in a program)/no. of span

For example

Average span size of variable x in the above sample=(10+12+7+6)/4=8.75

Average span size of variable y=(23+14)/2=18.5

(iii)Function points: Function point measures the functionality from the user point of view, that is, on the basis of what the user request and receives in return. Therefore, it deals with the functionality being delivered, and not with the lines of code, source modules, files etc. Measuring size in this way has the advantage that size measure is independent of the technology used to deliver the function.

Importance of function point:

- This is independent of the languages tools, or methodology used for implementation.
- They can be estimated from requirement specification or design specification.
- They are directly linked to the statement of request.

Q.48 Explain the development phases of the detailed COCOMO model.

Ans A software development is carried out in four successive phases, which are as follows:

1**: plan/requirements:** this is the first phase of the development cycle. The requirement is analyzed, the product plan is set up and a full product specification is generated. This phase consumes from 6% to 8% of the effort and 10% to 40% of the development time.

2: product design:The second phase of the COCOMO development cycle is concerned with the determination of the product architecture and the specification of the subsystems. This requires 16% to 18% of the normal effort and 19% to 38% of the development time.

3: programming: This is the third phase and is divided into sub phases : detailed phase and code/unit phase. This requires 48% to 68% of the effort and 24% to 64% of the development time.

4: integration/test : This phase of COCOMO occurs before delivery. This mainly consists of putting the tested parts together and then testing the final product. This requires 16% to 34% of normal effort and 18% to 34% of the development time.

Q.49 Why does the software design improve when we use object-oriented concepts?

Ans. The software design improves when we use object-oriented concepts because

1.Object-orientation works at higher level of abstraction. The development can proceed at the object level and ignore the rest of the system for as long as necessary.

2.The data on which a system is based tends to be more stable than the functionality it supports.

3.Object-orientated programming encourages and supports good programming techniques.

4.Object-oriented programming design and programming promote code re-use.

Q.50 Describe system testing.

Ans. System testing:

System test are designed to validate a fully developed systems with a view to assuring that it meets its requirements. There are three types of system testing:

Alpha testing:

This refers to system testing that is carried out by the test team within the organisation.

- **Beta testing:**

This is the system testing performed by a select group of friendly customers.

- **Acceptance testing:**

this is performed by the customer to determine whether or not to accept the delivery of the system.

Q.51 Define graph matrix and connection matrix.

Ans: Graph matrix: A graph matrix is a square matrix whose size(i.e., number of rows and columns) is equal to the number of nodes on the flow graph. Each row and column corresponds to an identified node, and matrix entries correspond to connections (an edge) between nodes.

Connection matrix: In connection matrix, each letter has been replaced with a 1, indicating that a connection exists (zeros have been excluded for clarity).

Q.52 What are dynamic testing tools? Explain the functions that they must support.

Ans. Dynamic testing tools:

1: **coverage analyzers (execution verifiers):** A coverage analyzer is the most common and important tool for testing. It is often relatively simple. One of the common strategies for testing involves declaring the minimum level of coverage, ordinarily expressed as a percentage of the elemental segments that are exercised in aggregate during the testing process.

2: **output comparators:** these are used in dynamic testing-both single-module and multiple- module(system level) varieties to check that predicted and actual outputs are equivalent. This is also done during regression testing.

3: **Test data generators:**this is a difficult one, and at lease for the present is one for which no general solution exists. one of the practical difficulties with test data generation of sets of inequalities that represent the condition along a chosen path.

4: **Test file generators:** this creates a file of information that is used as the program and does so based on comments given by the user and/or from the data descriptions program's data definition section.

5: **test harness systems:**this is one that is bound around the test object and that permits the easy modification and control of test inputs and output's and provides for online measurement of CI coverage values.

6: **Test archiving systems:** the goal is to keep track of series of tests ant to act as the basis for documenting that the tests have been done and that no defects were found during the process.

Functions that dynamic testing tools supports:

1: input setting: selecting of the test data that the test object reads when called.

2: stub processing: handling outputs and selecting inputs when a stub is called.

3: results displays: providing the tester with the values that the test object produces so that they can be validated.

4: test coverage measurement: determining the test effectiveness in terms of the structure of the program.

5: test planning: helping the tester to plan tests so they are both efficient and also effective at forcing discovery of defects.

Q.53 What is reverse engineering?

Ans. It is a process of analysing software with a view to understanding its design and specification.

- In reverse engineering, source code and executable code are the input.
- It may be part of a re-engineering process but may also be used to re-specify a system for re-implementation.
- Reverse engineering often proceeds re-engineering but is sometimes worth wise in its own right.
- Builds a program database and generates information from this.

- Program understanding tools (browsers, cross reference etc.) may also be used in this process.
- Design and specification may be reverse engineer to :
  - Serves as input to SRS for program replacement.
  - Be available to help program maintenance.

Q.54 What is structured programming and why is it important?

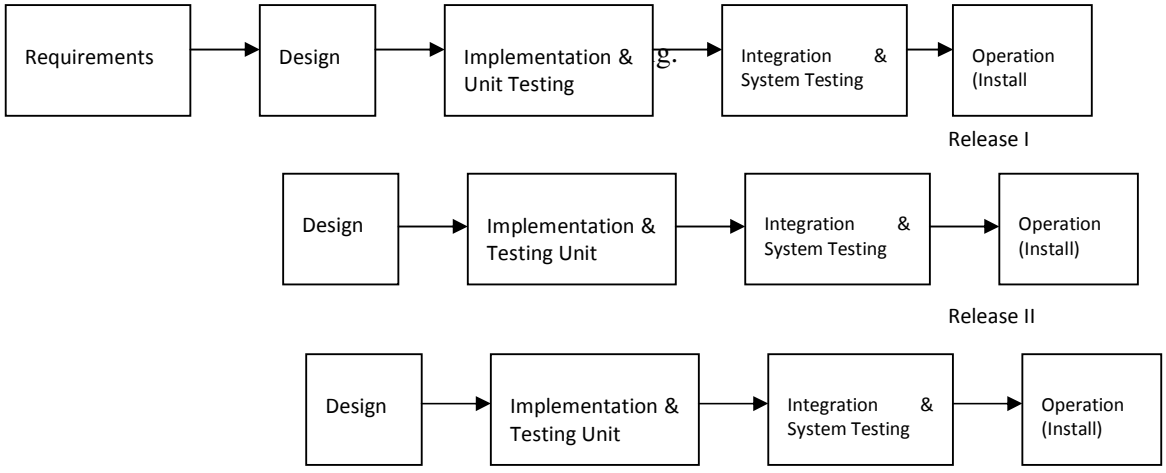
Ans. Structured programming is a term which was coined in the late 1960's to mean programming without using go to statements, programming using only while loops and if statements as control constructs and designing using a top-down approach. The adoption of structured programming was an important milestone in the development of software engineering because it was the first away from an undisciplined approach to software development.
Structured programming means programs can be read sequentially and are therefore easier to understand and inspect.

Q.55 Discuss any two software characteristics. (5)

Ans.**Software is not manufactured**: The life of a software is from concept exploration to the retirement of the software product. It is one time development effort and continuous maintenance effort in order to keep it operational. However, making 1000 copies of software is not an issue and it does not involve any cost. In case of hardware product, every product costs us due to raw material and other processing expenses. We do not have assembly line in software development. Hence it is not manufactured in the classical sense.
Software is flexible: We all feel that software is flexible. A program can be developed to do almost anything. Sometimes this characteristic may be the best and may help us to accommodate any kind of change.

Q.56 Differentiate between iterative Enhancement Model and Evolutionary Development model. (5)

Ans. Iterative Enhancement Model: This model has the same phases as the waterfall model, but with fewer restrictions. Generally the phases occur in the same order as in the waterfall model, but these may be conducted in several cycles. A useable product is released at the end of the each cycle, with each release providing additional functionality.



g.

Evolutionary Development Model: Evolutionary development model resembles iterative enhancement model. The same phases as defined for the waterfall model occur here in a cyclical fashion. This model differs from iterative enhancement model in the sense that this does not require a useable product at the end of each cycle. In evolutionary development, requirements are implemented by category rather than by priority.

(6)

Q.57 Explain the software life cycle model that incorporates risk factor.

Ans. The problem with traditional software process models is that they do not deal sufficiently with the uncertainty, which is inherent to software projects. Important software projects failed because project risks were neglected and nobody was prepared when something unforeseen happened. Barry Boehm recognized this and tried to incorporate the " project risk" factor into a life cycle model. The results is the spiral model, which was presented in 1986 BOEH86
Each loop of the spiral from X-axis clockwise through 3600 represents one phase. One phase is split roughly into four sectors of major activities.

- Planning : Determination of objectives, alternatives and constraints
- Risk Analysis : Analyze alternatives and attempts to identify and resolve the risks involved
- Development : Product development and testing product
- Assessment : Customer evaluation

During the first phase, planning is performed, risks are analyze, prototypes are built, and customers evaluate the prototype. During the second phase, a more refined prototype is built, requirements are documents and validated, and customers are involved in assessing the new prototype. By the time third phase begins, risks are known, and a somewhat more traditional development approach
The advantage of this model is the wide range of options to accommodate the good features of outer life cycle models. It becomes equivalent to another life cycle model in appropriate situations. It also incorporates software quality objectives into software development. The risk analysis and validation steps eliminate errors in the early phases of development

Q.58 Explain any two requirement elicitation methods.

Ans. Interviews:
After receiving the problem statement from the customer, the first step is to arrange a meeting with the customer. During the meeting or interview, both the parties would like to understand each other. Normally specialized developers often called 'requirement engineers' interact with the customer. The objective of conducting an interview is to understand the customer's expectation from the software. Both parties have different feelings, goals, opinions, vocabularies, understanding, but one thing in common, both want the project to be a success. With this is mind, requirement engineers normally arrange interviews. Requirements engineers must be open minded and should not approach the interview with pre-conceived notion about what is required.
Interview may be open-ended or structured. In open ended interview there is no pre-set agenda. Context free questions may be asked to understand the problem

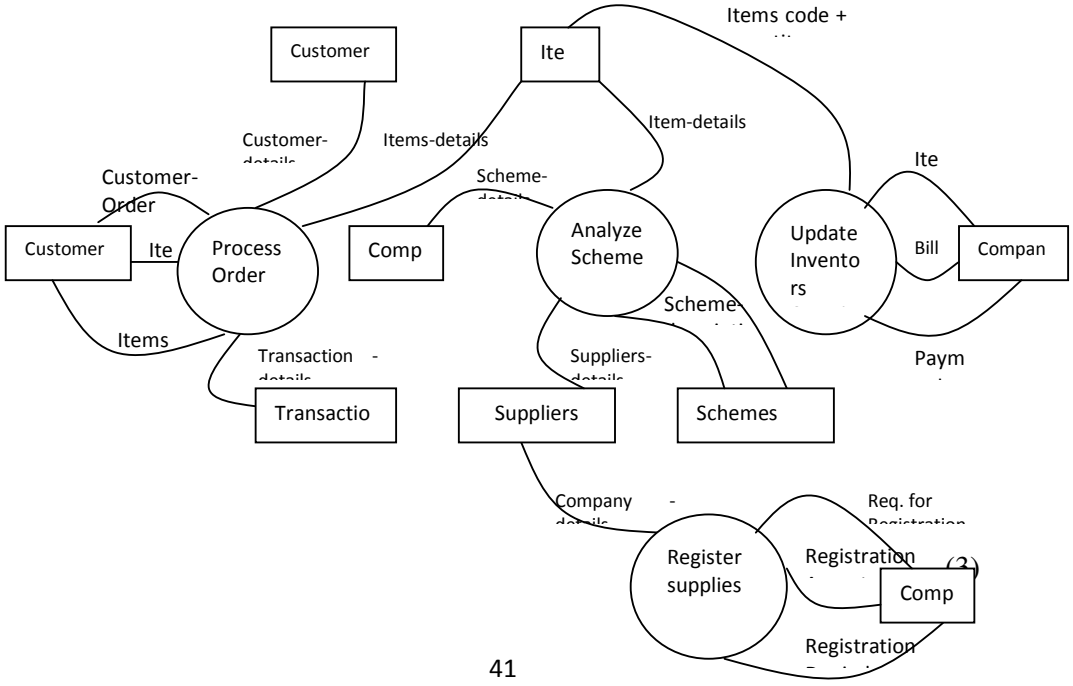
and to have an overview of the situation.
In structured interview, agenda of fairly open questions is prepared. Sometimes a proper questionnaire is designed for the interview. Interview, may be started with simple questions to set people at ease. After making atmosphere comfortable and calm, specific questions may be asked to understand the requirements, The customer may be allowed to voice his or her perception about a possible solution.

Facilitated Application Specification Technique

A team oriented approach is developed for requirements gathering and is called facilitated application specification Techniques (FAST
This approach encourages the creation of a joint team of customers and developers who work together to understand the expectations and propose of set of requirements. The basic guidelines of FAST are given below :

- Arrange a meeting at a neutral site for developers and customers.
- Establishment of rules for preparation and participation.
- Prepare and informal agenda that encourages free flow of ideas.
- Appoint a facilitator to control the meeting. A facilitator may be a developer, a customer, or an outside expert. Prepare a definition mechanism-Board, flip chart, worksheets, wall stickiest, etc.
- Participants should not criticize or debate.

Q.59 A store is in the business of selling paints and hardware items. A number of reputed companies supply items to the store. New suppliers can also register with the store after providing necessary details. The customer can place the order with the shop telephonically. Or personally. In case items are not available customers are informed. The detail of every new customer is stored in the company's database for future reference. Regular customers are offered discounts. Additionally details of daily transactions are also maintained. The suppliers from time to time also come up with attractive schemes for the dealers. In case, scheme is attractive for a particular item, the store places order with the company. Details of past schemes are also maintained by the store. The details of each item i.e. item code, quantity available etc. is also maintained. Draw a level 1 DFD for the above requirement.



Q.60 **List any three characteristics of a good SRS.**

Ans:The SRS should be: Correct , Unambiguous, Complete

1. Correct: An SRS is correct iff every requirement stated therein is one that the software shall meet .
2. Unambiguous: An SRS is unambiguous iff every requirement stated therein has only one interpretation. Each sentence in SRS should have the unique interpretation.
3. Complete: An SRS is complete iff it includes all significant requirements full labels and references to all figures and diagram and definition of all terms and units of measures

Q.61 **Define the following terms:**

(i) Product metrics
(ii) Live variables
(iii) FAN IN **(10)**

Ans: (i) Product metrics: describe the characteristics of the product such as size, complexity, design features, performance, efficiency, reliability, portability, etc.
(ii) A variable is live from its first to its last reference within a procedure.
(iii) FAN IN is simply a count of the number of other Components that can call, or pass control, to Component A.

Q.62 **Compare the Organic Semi detached and Embedded cocomo modes** **(8)**

Ans:

Mode	Project size	Nature of Project	Innovation	Deadline of the project	Development Environment
Organic	Typically 2-50 KLOC	Small size project, experienced developers in the familiar environment. For example, pay roll, inventory projects etc.	Little	Not tight	Familiar & In house
Semi detached ty	Typically 50-300 KLOC	Medium size project, Medium size team, Average previous experience on similar projects. For example:	Medium	Medium	Medium

		Utility systems like compilers, database systems, editors etc.			
Embedded	Typically over 300 KLOC	Large project, Real time systems, Complex interfaces, very little previous experience. For Example: ATMs, Air Traffic Control etc.	Significant	Tight	Complex Hardware/customer interfaces required

Q.63 **Why is good design important for a product?** **(3)**

Ans: A good design is the key to successful product. Almost 2000 years ago a Roman Architect recorded the following attributes of a good design:

- Durability
- Utility and
- Charm

A well-designed system is easy to implement, understandable and reliable and allows for smooth evolution.

Q.64 Define coupling. Discuss various types of coupling. **(8)**

Ans: Coupling is the measure of the degree of interdependence between modules.

Type of coupling : Different types of coupling are content, common, external, control, stamp and data. The strength of coupling from lowest coupling (best) to highest coupling (worst) is given in the figure.

Data coupling	Best
Stamp coupling	↑
Control coupling	↑
External coupling	↑
Common coupling	↑
Content coupling	(Worst)

Given two procedures A and B, we can identify a number of ways in which they can be coupled.

Data coupling

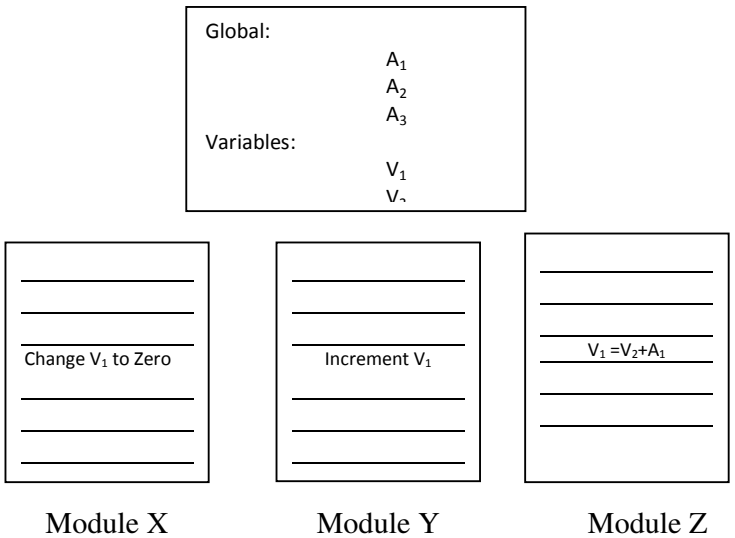
The dependency between module A and B is said to be data coupled if their dependency is based on the fact they communicate by only passing of data. Other than communicating through data, the two modules are independent. A good strategy is to ensure that no module communication contains "tramp data" only the necessary data is passed. Students name, address, course are example of tramp data that are unnecessarily communicated between modules. By ensuring that modules communicate only necessary data, module dependency is minimized.

Stamp coupling: Stamp coupling occurs between module A and B when complete data structure is passed from one module to another. Since not all data making up the structure is usually necessary in communication between the modules, stamp coupling typically involves data. If one procedure only needs a part of a data structure, calling modules pass just that part, not the complete data structure.

Control coupling: Module A and B are said to be control coupled if they communicate by passing of control information. This is usually accomplished by means of flags that are set by one module and reacted upon by the dependent module.

External coupling: A form of coupling in which a module has a dependency to other module, external to the software being developed or to a particular type of hardware. This is basically related to the communication to external tools and devices.

Common coupling: With common coupling, module A and module B have shared data. Global data areas are commonly found in programming languages. Making a change to the common data means tracing back to all the modules which access that data to evaluate the effect of change. With common coupling, it can be difficult to determine which module is responsible for having set a variable to a particular value. Fig. below shows how common coupling works



Content coupling : Content coupling occurs when module A changes data of module B or when control is passed.

Q.65 **Explain the concept of bottom-up, top-down and hybrid design.** **(8)**

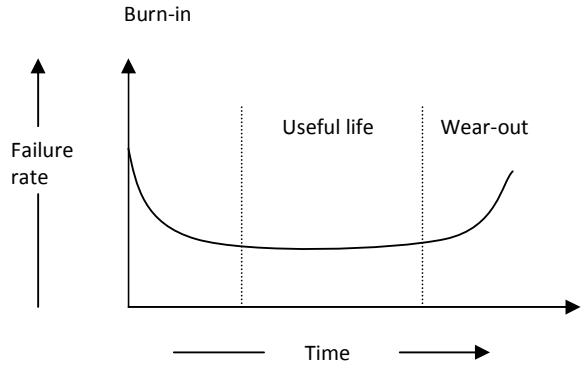
Ans: Bottom up design:This approach leads to a style of design where we decide how to combine these modules to provide larger ones; to combine those to provide even larger ones, and so on, till we arrive at one big module which is the whole of the desired program.

Since the design progressed from bottom layer upwards, the method is called bottom up design. The main argument for this design is that if we start coding a module soon after its design, the chances of recoding is high; but the coded module can be tested and design can be validated sooner than

a module whose sub modules have not yet been designed.
Top down design:A top down design approach starts by identifying the major modules of the system, decomposing them into their lower level modules and iterating until the desired level of details is achieved. This is stepwise refinement; starting from an abstract design, in each step the design is refined to a more concrete level, until we reach a level where no more refinement is needed and the design can be implemented directly.
Hybrid design:Pure top-down or pure bottom up approaches are often not practical therefore hybrid approach which combines the above two approaches is often used.

Q.66 **Explain the bath tub curve of hardware reliability.** **(5)**

Ans: As indicated in the figure below, there are three phase in the life of any hardware component i.e. burn-in, useful life & wear out. In burn-in phase, failure rate is quite high initially, and it starts decreasing gradually as the time progresses. It may be due to initial testing in the premises of the organization. During useful life period, failure rate is approximately constant. Failure rate increases in wear-out phase due to wearing out/aging of components. The best period is useful life period. The shape of this curve is like a "bath tub" and that is why it is known as bath tub curve.



Bath tub curve of hardware reliability

Q.67 **Consider a program for the determination of the nature of roots of a quadratic equation. Its input is a triple of positive integers (say a, b, c) and values in the interval [0,100]. The program output may be one of the following:**
[Not a quadratic equation; Real roots; Imaginary roots; Equal roots]. Design the boundary value test cases and robust test cases for the program. **(16)**

Ans. Quadratic equation will be of type:

$$ax^2 + bx + c = 0$$

Roots are real if $(b^2 - 4ac) > 0$

Roots are imaginary if $(b^2 - 4ac) < 0$

Roots are equal if $(b^2 - 4ac) = 0$

Equation is not quadratic if a=0

The boundary value test cases are:

Test Case	a	b	c	Expected output
1	0	50	50	Not Quadratic
2	1	50	50	Real Roots
3	50	50	50	Imaginary Roots
4	99	50	50	Imaginary Roots
5	100	50	50	Imaginary Roots
6	50	0	50	Imaginary Roots
7	50	1	50	Imaginary Roots
8	50	99	50	Imaginary Roots
9	50	100	50	Equal Roots
10	50	50	0	Real Roots
11	50	50	1	Real Roots
12	50	50	99	Imaginary Roots
13	50	50	100	Imaginary Roots

As we know, robust test cases are 6n+1. Hence, in 3 variable input cases total number of test cases are 19 as given below:

Test Case	A	B	C	Expected Output
1	-1	50	50	Invalid Input
2	0	50	50	Not Quadratic Equation
3	1	50	50	Real Roots
4	50	50	50	Imaginary Roots
5	99	50	50	Imaginary Roots
6	100	50	50	Imaginary Roots
7	101	50	50	Invalid Input
8	50	-1	50	Invalid Input
9	50	0	50	Imaginary Roots
10	50	1	50	Imaginary Roots
11	50	99	50	Imaginary Roots
12	50	100	50	Equal Roots
13	50	101	50	Invalid Input
14	50	50	-1	Invalid Input
15	50	50	0	Real Roots
16	50	50	1	Real Roots
17	50	50	99	Imaginary Roots
18	50	50	100	Imaginary Roots
19	50	50	101	Invalid Input

Q.68 **Discuss the problems faced during software maintenance (7)**

Ans. **Problems During Maintenance** The most important problem during maintenance is that before correcting or modifying a program, the programmer must first understand it. Then, the programmer must understand the impact of the intended change. Few problems are discussed

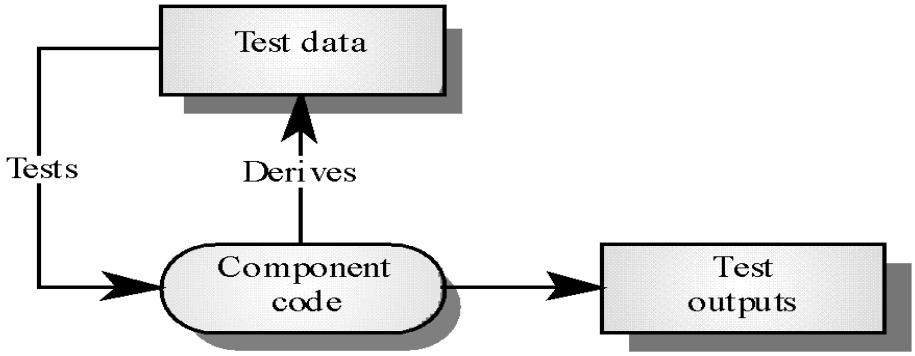
below.
Often the program is written by another person or group of persons working over the years in isolation from each other.
Often the program is changed by person who did not understand it clearly, resulting in a deterioration of the program's original organization.
Program listing, even those that are well organized, are not structured to support reading or comprehension. We normally read article or book straight through, but with listing, programmer rummages back and forth.
There is a high staff turnover within information Technology industry. Due to this many systems are maintained by persons who are not the original authors. These persons may not have adequate knowledge about the system. This may mean that these persons may introduce changes to programs without being aware of their effects on other parts of the system -the ripple effect. This problem may be worsened by the absence of documentation. Even where it exists, it may be out of date or inadequate.
Some problems only become clearer when a system is in use. Many users know that they want but lack the ability to express it in a form understandable to programmers/analysts. This is primarily due to information gap.
Systems are not designed for change. If there is hardly any scope for change, maintenance will be very difficult. Therefore approach of development should be the production of maintainable software.

Q.69 Explain acceptance testing and beta testing. (5)

Ans. Acceptance Testing and Beta testing: System tests are designed to validate a fully developed system to assure that it meets its requirements.
Acceptance and beta testing are form of system testing:
• **Beta testing.** Beta testing is the system testing performed by a select group of friendly customers.
• **Acceptance Testing.** Acceptance testing is the system testing performed by the cutomer to determine whether he should accept the delivery of the system.

Q.70. Define the following
(i) Structural testing
(ii) Special value testing
(iii) Mutation testing (9)

Ans(i) Structural Testing:Sometime called white-box testing. Here derivations of test cases are according to the program structure. Knowledge of the program is used to identify additional test cases. The objective of structural testing is to exercise all program statements (not all path combinations)



(ii)Special Value Testing: It is the form of functional testing. Special value testing occurs when a tester uses his or her domain knowledge, experience with similar program and information about "soft spots" to dense test case. No guidelines are used other than to use "best engineering judgment". As a result special value testing is heavily dependent on the abilities of the testing persons.

(iii)Mutation testing In mutation testing, the software is first tested by using an initial test suite built up from the different white box testing strategies. After the initial testing is complete, mutation testing is taken up. The idea behind mutation testing is to make few arbitrary changes to a program at a time. Each time the program is changed, it is called as a mutated program and the change effected is called as a mutant. A mutated program is tested against the full test suite of the program. If there exists at least one test case in the test suite for which a mutant gives an incorrect result, then the mutant is said to be dead. If a mutant remains alive even after all the test cases have been exhausted, the test data is enhanced to kill the mutant. A major disadvantage of the mutation-based testing approach is that it is computationally very expensive, since a large number of possible mutants can be generated.

Q.71. What is CMM? Describe its levels & compare it with ISO 9001?

Ans. CAPABILITY MATURITY MODEL (CMM): CMM is a strategy for improving the software process, irrespective of the actual life cycle model used. Software Engineering Institute (SEI) of Carnegie-Mellon University developed CMM in 1986. CMM is used to judge the maturity of the software processes of an organization and to identify the key practices that are required to increase the maturity of these processes.

Different levels of CMM:

1) **Initial (maturity level 1)-**At this, the lowest level, there are essentially no sound software engineering management practices in place in the organization. Instead everything is done on adhoc basis. In maturity level 1 organization, the software process is unpredictable.

2) **Repeatable (maturity level 2)-**At this level, policies for managing a software project and procedures to implement those policies are established. Planning and managing new projects is based on experience with similar projects. An objective in achieving level 2 is to institutionalize effective management processes for software projects, which allow organizations to repeat successful practices, developed on other projects. The software process capability at level 4 organizations can be summarized as disciplined.

3) **Defined (maturity level 3)-** At this level, the standard process for developing and maintaining software across the organization is documented, including both software

engineering and management processes. Processes established at level 3 are used to help the software managers and technical staff to perform more effectively. The software process capability at level 4 organizations can be summarized as standard and constituent.

4) **Managed (maturity level 4)-**At this level, the organization sets quantitative quality goals for both software products and processes. Productivity and quality are measured for important software process activities across all projects as part of an organizational measurement program. The software process capability at level 4 organizations can be summarized as "predictable".

5) **Optimizing (maturity level 5)-**At this level, the entire organization s focused on continuous process improvement. The organizations have the means to identify weaknesses and strengthen the process proactively, with the goal of preventing the occurrence of defects. . The software process capability at level 4 organizations can be summarized as continuously improving.

ISO 9001 and CMM:

1: **Management responsibility:** ISO 9001 requires that the quality policy be defined, documented, understood, implemented, and maintained. Whereas in CMM, management responsibility for quality policy and verification activities is primarily addressed in software quality assurance.

2: **Document control:** ISO 9001 requires that the distribution and modification of documents be controlled. In the CMM, the configuration management practices characterizing document control are described in software configuration management.

3: **Purchasing:** ISO 9001 requires that purchased products conform to their specified requirements.

In the CMM, this is addressed in software subcontract management.

4: **Training:** ISO 9001 requires that training needs to be identified and that training be provided. Records of training are maintained.

In CMM, specified trainings are identified in the training and orientation practices in the ability to perform common feature.

5: **Internal quality audit:**ISO 9001 requires that the audits be planned and performed. The results of audits are communicated to management, and any deficiencies found are corrected.

Specific audits in the CMM are called out in the auditing practices of the verifying implementation common feature.

Q.72 Spiral model is a realistic approach to the development of large-scale systems & software. Justify & explain the model?

Ans. There are several advantages of Spiral model that makes it a realistic approach to development of large-scale systems and software, viz:

1) The spiral model promotes quality assurance through prototyping at each stage in system development.
2) The spiral model is a realistic approach to the development of large-scale software products because the software evolves as the process progresses. The developer and client better understand and react to risk at each evolutionary level.
3) The model uses prototyping as a risk reduction mechanism and allows for the development of prototypes at any stage of the evolutionary development.

4) It maintains a systematic stepwise approach like the classic life cycle model but incorporates it into an iterative framework that more reflect the real world.
If employed correctly this model should reduce risk before they become problematic as consideration of technical risk are considered at all stages
Spiral model is also known, as spiral life cycle model is a system development life cycle model used in information technology. This model of development combines the features of prototyping model and waterfall model. The spiral model is used for large, expensive and complicated project.

- Presented by BARRY BOHEM in 1986 incorporating the project risk factor.
- Designed in order to overcome the disadvantages of waterfall model.
- The radial dimension represents the cumulative cost incurred in accomplishing the steps done so far and the angular dimension represents the progress made in completing each cycle of the spiral.
- Each loop is a development stage.
- Balance all the risk elements that is the high-risk element must be lowered.
- The people concerned with the project complete each phase with a review.

Different Phases of spiral model:

1) **Planning:** In this phase the objectives, alternatives and constraints of the project are determined or documented. The objectives and other specifications are fixed in order to decide which strategies or approaches to follow during the project life cycle.

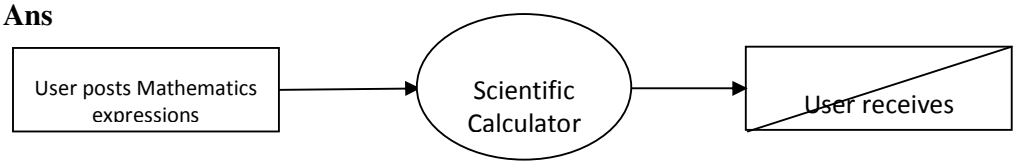
2) **Risk analysis:** In this phase, all possible and available alternatives which can help in developing a cost effective project are analyzed and strategies are decided to use them. This phase has been added specially in order to identify and resolve all the possible risk in the project development.

3) **Development:** In this phase the actual development of the project is carried out. The output of this phase is passed through all the phases iteratively in order to obtain improvements in the same.

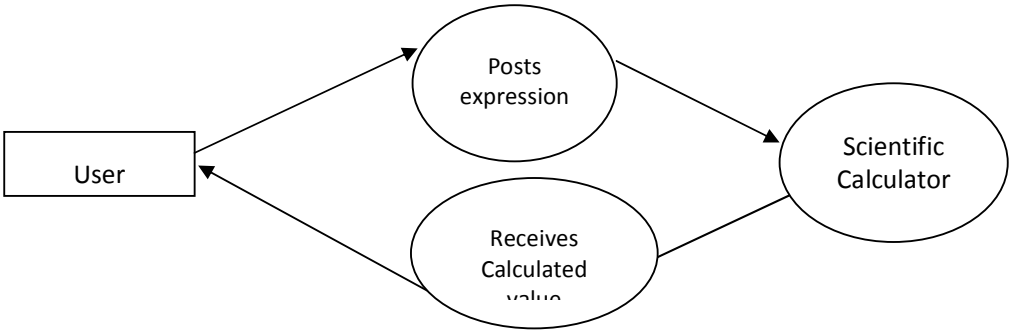
4) **Assessment:** In this phase, developed product is pass on to the customer in order to receive customer comments and suggestions, which can help in identifying and resolving potential problems or errors in the software development.

Q.73 **A program is to be developed to simulate the operations of a scientific calculator. List the facilities to be provided by this calculator. Analyze this using DFD?**

Ans



Context Free Diagram



Level 0 DFD

Q.74 What is function point? Explain its importance. What is function-oriented metrics?

Ans. Function point measures the functionality from the user point of view, that is, on the basis of what the user request and receives in return. Therefore, it deals with the functionality being delivered, and not with the lines of code, source modules, files etc. measuring size in this way has the advantage that size measure is independent of the technology used to deliver the function.

Importance of function point:

- This is independent of the languages tools, or methodology used for implementation.
- They can be estimated from requirement specification or design specification.
- They are directly linked to the statement of request.

Function-oriented metrics: Function-oriented software metrics use a measure of the functionality delivered by the application as a normalization value. function-oriented metrics were first proposed by Albrecht, who suggested a measure called function point. Function points are derived using a empirical relationship based on countable measures of software's information domain and assessments of software complexity.

Q.75 Explain basic information flow model & it's more sophisticated versions?

Ans. Information flow metrics are applied to the components of a system design.
For any component 'A', we can define three measures:

1: 'FAN IN' is simply a count of the number of other components that can all, or pass control, to component A.

2: 'FAN OUT' is the number of components that are called by component A.

3: using the following formula derives this.

We will call this measure the INFORMATION FLOW index of component A, abbreviated as IF(A).

$IF(A) = \{FAN\ IN(A) * FAN\ OUT\ (A)\}^2$

The sophisticated IF model differs from basic model in its definition of FAN IN and FAN OUT

For a component A let:

a= the number of components that call A.

b=the number of parameters passed to A from components higher in the hierarchy

c= the number of parameters passed to A from components lower in the hierarchy

d=the number of data elements read by components A

then

FAN IN(A)=a+b+c+d

Also let:

e= the number of components called by A.

f=the number of parameters passed from A to components higher in the hierarchy

g= the number of parameters passed from A to components lower in the hierarchy

h=the number of data elements written to by A

FAN OUT(A)=e+f+g+h

Other than those changes to the basic definitions, the derivation, analysis and interpretation remain the same.

Q.76 Explain COCOMO model with its relevant equations. Explain various attributes of cost drivers used in COCOMO model.

Ans. COCOMO stands for constructive cost model

- An empirical model based on project experience.
- Well-documented, independent model, which is not tied to a specific software vendor.
- There is a long history of COCOMO model, from initial versions published in 1981(COCOMO-81) through various versions instantiations to COCOMO-2.
- Then COCOMO-2 takes into account different approaches to software development, reuse etc.

Various attributes of cost drivers used in COCOMO model are:

1. **Physical attributes**: These are concerned with required characteristics of the software product being developed.
2. **Computer attributes:** These are constraints imposed on the software by the hardware platform. These affect software productivity because effort must be expended to overcome the hardware limitations.
3**: Personnel attributes**: These are multipliers which takes into account the experience of the people working on the project.
4: **Project attributes**: These are concerned with the use of software tools, the project development schedule and the use of modern programming practices.

Q.77 What is Data Binding?

Ans. The matrix that attempts to capture the module-level concept of coupling is data binding. Data binding are a measure that captures the data interaction across portions of a software system. In other words, data binding try to specify how strongly coupled different modules in a software system are.

Q.78 Define cohesion & coupling? Give suitable examples.

Ans Coupling: Coupling refers to the strength of the relationship between modules in a system. Coupling represents how strongly different modules are interconnected with each other.

Cohesion: Cohesion refers to the strength of relationship among elements within a module. Cohesion represents how strongly the internal elements of a module are bound to each other.

Q.79 Explain various Object Oriented concepts used in Software Engineering. Give an example.

Ans .Various concepts of Object Oriented concepts used in Software Engineering:

- **Object:-** An object is something which is capable of being seen, touched or sensed. Each object has certain distinctions or attributes which

enable you to recognize and classify it. Each object has certain actions or methods associated with it.

- **Class:-**A class encapsulates data and procedural abstractions required to describe the content and behavior of some real world entity. A class is a generalized description that describes a collection of similar objects.
- **Encapsulation:-** An object is said to be encapsulate (hide) data and program. This means that the user cannot see the inside of the object but can use the object by calling the program part of the object. This hiding of details of one object from another object, which uses it, is known as encapsulation.
- **Inheritance:-**Inheritance is defined as the property of objects by which instances of a class can have access to data and programs contained in a previously defined class, without those definitions being restated. Classes are linked together in a hierarchy. They form a tree, whose root is the class of "objects". Each class (except for the root class) will have a super class (a class above it in the hierarchy) and possibly subclasses. A class can inherit (i.e. acquire) methods from its super class and in turn can pass methods on to its subclasses. A subclass must have all the properties of the parent class, and other properties as well.
- **Polymorphism:-**Polymorphism includes the ability to use the same message to objects of different classes and have them behave differently. Thus we could define the message "+" for both the addition of numbers and the concatenation (joining) of characters. Polymorphism provides the ability to use the same word to invoke different methods, according to similarity of meaning.
- **Object/class associations:-**Objects/classes interact with each other. Multiplicity defines how many instances of one object/class can be associated with one instance of another object/class.
- **Messages:-**The interaction or communication between the different objects and classes is done by passing messages. The object, which requires communicating with another object, will send a request message to the latter. A message can be sent between two objects only if they have an association between them.

Q.80 . Compare and contrast reliability and availability?

Ans. Reliability:The probability of failure-free system operation over a specified time in a given environment for a given purpose is called reliability
Availability:The probability that a system, at a point in time, will be operational and able to deliver the requested services is known as availability.
Both of these attributes can be expressed quantitatively. It is sometimes possible to subsume system availability under system reliability. Obviously if a system is unavailable it is not delivering the specified system services. However, it is possible to have systems with low reliability that must be available. So long as system failures can be repaired quickly and do not damage data, low reliability may not be a problem.
Availability takes repair time into account.

Q.81 Design black box test suits for a function that checks whether a character or string upto ten characters in a palindrome?

Ans Following may be the test cases:
- Try with Even number of characters (MALAYALAM)
- Try with Odd number of characters (ADDA)
- Try with maximum length of string

The following are negative test cases: (Warning message should come for the following)
- Try with empty string
- Try with numbers
- Try with special characters

Q.82 How is cyclomatic complexity useful in program test? What is sequence of testing? What is testability?

Ans. Cyclomatic complexity measures the amount of decision logic in a single software module. It is used for two related purposes in the structured testing methodology. First, it gives the number of recommended tests for software. Second, it is used during all phases of the software lifecycle, beginning with design, to keep software reliable, testable, and manageable. Cyclomatic complexity is based entirely on the structure of software's control flow graph.

The sequence of testing is:

Unit testing: Unit testing is undertaken after a module has been coded and successfully reviewed. Unit testing (or module testing) is the testing of different units (or modules) of a system in isolation.

Integration testing: The primary objective of integration testing is to test the module interfaces, i.e. there are no errors in the parameter passing, when one module invokes another module. During integration testing, different modules of a system are integrated in a planned manner using an integration plan. The integration plan specifies the steps and the order in which modules are combined to realize the full system. After each integration step, the partially integrated system is tested.

System testing System tests are designed to validate a fully developed system to assure that it meets its requirements. There are essentially three main kinds of system testing:

• **Alpha Testing-** Alpha testing refers to the system testing carried out by the test team within the developing organization.

• **Beta testing-**Beta testing is the system testing performed by a select group of friendly customers.

• **Acceptance Testing-**Acceptance testing is the system testing performed by the customer to determine whether he should accept the delivery of the system.

Software testability is the degree to which a software artifact (i.e. a software system, software module, requirements- or design document) supports testing in a given test context.

Testability is not an intrinsic property of a software artifact and can not be measured directly. Instead testability is an extrinsic property which results from interdependency of the software to be tested and the test goals, test methods used, and test resources.

A lower degree of testability results in increased test effort. In extreme cases a lack of testability may hinder testing parts of the software or software requirements at all.

Q.83 Explain various types of static and dynamic testing tools.

Ans. Static testing tools:

1. **Static analysers** A static analyser operates from a pre-computed database of descriptive information derived from the source text of the program. The idea of a static analyser is to provide allegations, which are claims about the analysed programs that can be demonstrated by systematic examination of all cases.

2. **Code inspectors** A code inspector does a simple job of enforcing standards in a uniform way for many programs. These can be single statement or multiple statement rules. The AUDIT system is available which imposes some minimum conditions on the programs.

3. **Standard enforces**This tool is like a code inspector; expect that the rules are generally simpler. The main distribution is that a full-blown static analyser looks at whole programs, whereas a standard enforcer looks at only single statements.

Dynamic testing tools

1**. Coverage analyzers (execution verifiers)** A coverage analyzer is the most common and important tool for testing. It is often relatively simple. One of the common strategies for testing involves declaring the minimum level of coverage, ordinarily expressed as a percentage of the elemental segments that are exercised in aggregate during the testing process.

2. **Output comparators** These are used in dynamic testing-both single-module and multiple-module (system level) varieties to check that predicted and actual outputs are equivalent. This is also done during regression testing.

3. **Test data generators**This is a difficult one, and at least for the present is one for which no general solution exists. One of the practical difficulties with test data generation of sets of inequalities that represent the condition along a chosen path.

4. **Test file generators** This creates a file of information that is used as the program and does so based on comments given by the user and/or from the data descriptions program's data definition section.

5. **Test harness systems**This is one that is bound around the test object and that permits the easy modification and control of test inputs and output's and provides for online measurement of CI coverage values.

6. **Test archiving systems:** The goal is to keep track of series of tests ant to act as the basis for documenting that the tests have been done and that no defects were found during the process.

Q. 84 **Explain the following Software Metrics**

(i) Lines of Code **(ii) Function Count**

(iii) Token Count **(iv) Equivalent size measure**

Ans (i) Lines of code (LOC) is a software metric used to measure the size of a software program by counting the number of lines in the text of the program's source code. LOC is typically used to predict the amount of effort that will be required to develop a program, as well as to estimate programming productivity or effort once the software is produced. **Advantages:-**

1. Scope for Automation of Counting: Since Line of Code is a physical entity; manual counting effort can be easily eliminated by automating the counting process. Small utilities may be developed for counting the LOC in a program. However, a code counting utility developed for a specific language cannot be used for other languages due to the syntactical and structural differences among languages.

2. An Intuitive Metric: Line of Code serves as an intuitive metric for measuring the size of software due to the fact that it can be seen and the effect of it can be

visualized. Function Point is more of an objective metric which cannot be imagined as being a physical entity, it exists only in the logical space. This way, LOC comes in handy to express the size of software among programmers with low levels of experience.

Disadvantages

1. Lack of Accountability: Lines of code measure suffers from some fundamental problems. Some think it isn't useful to measure the productivity of a project using only results from the coding phase, which usually accounts for only 30% to 35% of the overall effort.
2. Lack of Cohesion with Functionality: Though experiments have repeatedly confirmed that effort is highly correlated with LOC, functionality is less well correlated with LOC. That is, skilled developers may be able to develop the same functionality with far less code, so one program with less LOC may exhibit more functionality than another similar program. In particular, LOC is a poor productivity measure of individuals, because a developer who develops only a few lines may still be more productive than a developer creating more lines of code.
3. Adverse Impact on Estimation: Because of the fact presented under point (a), estimates based on lines of code can adversely go wrong, in all possibility.
4. Developer's Experience: Implementation of a specific logic differs based on the level of experience of the developer. Hence, number of lines of code differs from person to person. An experienced developer may implement certain functionality in fewer lines of code than another developer of relatively less experience does, though they use the same language.

(ii) Function count:- It measures the functionality from the user point of view, that is, on the basis of what the user requests and receives in return. Therefore it deals with the functionality being delivered, and not with the lines of code, source modules, files etc. measuring size is in this way has the advantage that size measure is independent of the technology used to deliver the function.

Features:-

- Function point approach is independent of the language tools, or methodologies used for implementation.
- They can be estimated from requirement specification or design specification.
- They are directly linked to the statement of requirements.
- They are based on the system user's external view of the system, non-technical users of the software system have a better understanding of what function points are measuring.

(iii) Token count:- A program is considered to be series of tokens and if we count the number of tokens, some interesting results may emerge. Tokens are classified as either operators or operands. All software science measures are functions of the counts of these tokens.

Variables, constants and even labels are operands. Operators consist of arithmetic symbols such as +, -. /, * and command names such as "while", "for", "printf", special symbols such as : =, braces, parentheses, and even function names such as "eof".

The size of the vocabulary of a program, which consists of the number of unique tokens to build a program, is defined as:-

$$n = n_1 + n_2$$

Where, n : vocabulary of a program.

n_1: number of unique operators

n_2: number of unique operands.

(iv) Equivalent size measure:- Models makes no distinction between a new software and the portion being reused. The equivalent size measure Se, is a function of Sn (new developed code) and Su (reused code); means that the efforts required to develop the software with Sn and Su is equivalent to the effort of developing a product with Se "from scratch". Sometimes there may be an undesirable situation in which it actually costs more to put together several existing program segments then it would to construct equivalent software from scratch.

Boehm has proposed a function for equivalent size measure as

$$Se=Sn+(a/100)Su$$

Where the adjustment factor 'a' is determined by the percentage of modification required of design(DM), of the code(CM), and of the efforts necessary for the integration of the modified code(IM).

Q. 85 **Explain incremental model? Define core product and detailed plan.**

Ans. The incremental model is proposed by D.L PARNAS. It is basically implemented by combining elements of linear sequential model and iterative prototyping model. Incremental development refer to the process of developing to completion only a part of the requirement at a time by selectively developing parts of the requirements, designing, coding and testing software that means these require functions, the software product may be built up incrementally to its final configuration. This approach model releases the product partially to the customer in the beginning and slowly at increased functionality to the system. That is why it is called incremental model.

The model prioritizes the system requirements and implements them and implements them in groups. Each new release of the system enhances the functionality of the previously released system thereby reducing the cost of project. In the first release on the functionality of the product is offered to the customer. In the second release, functionality A+ functionality B is offered. Finally, in release 3 functionality A, B and C are offered. Therefore with each release in addition to new functionality in the system, functionality of earlier release may also be enhanced.

Advantages of incremental Model:-

- As product is to be delivered in parts, total cost of the project is distributed.
- Limited number of persons can be put on to the project because work is to be delivering in parts.
- As development activities for next release and use of early version of product is done. Simultaneously, error if found can be corrected.
- Customers or end users get the chance to see the useful functionality early in SDLC.
- As a result, of end users feedback requirements for successive release become clearer.
- Testing also becomes easier.
- Risk of failure of product is decreased as user start using the product early.

Q.86 What is Data Dictionary? Explain each component?

Ans Data dictionary is a storehouse of data giving information about data. It is a list of terms and their definition for all data items and data files of a system. A data dictionary contains descriptions and definitions concerning the data structure, data elements, their interrelationships and other characteristics of a system.

Objectives of Data dictionaries:-

1) A standard definition of all terms in a system, that is each item of data is uniquely identified and defined.
2) Easy cross-referencing between sub-systems, programs and modules.
3) Simpler program maintenance.

Data Items:

There are three classes of data items in a data dictionary:-

1) Data element- It is the smallest unit of data which cannot be meaningfully decomposed further. e.g. Employee number etc.
2) Data structure- A group of data elements forms a data structure.
3) Data Flows and Data Stores- data Flows are data structures in motion whereas data stores are data structures at rest.

Q.87 What specific languages can be used in SRS? What are the advantages of using these specific languages of SRS?

Ans. Requirement specification necessitates the use of some specification language. The language should support the desired qualities of the SRS- modifiability, understandability, unambiguous, and so forth. The language should be easy to learn and use.

For ease of understanding a natural language might be preferable. Though formal notations exist for specifying specific properties of the system, natural languages are now most often used for specifying requirements. The overall SRS is generally in a natural language, and when flexible and desirable, some specifications in the SRS may use formal languages.

The major advantage of using a natural language is that both client and superior understand the language. However, by the very nature of a natural language, it is imprecise and ambiguous. To reduce the drawback of natural language, most often natural language is used in a structured fashion. In structured English, requirements are broken into sections and paragraphs. Each paragraph is then broken into sub paragraphs.

In an SRS, some parts can be specified better using some formal notation, example- to specify formats of inputs or outputs, regular expression can be very useful.

Similarly when discussing system like communication protocols, finite state automata can be used. Decision tables are useful to formally specify the behavior of a system on different combination of input or settings.

Q.88 Draw the flow chart of Risk Management-Activity and explain various Software risks.

Ans "RISK" is a problem that could cause some loss or threatens the success of the project, but which has not happened yet.

Typical Software Risks:-

1. Dependencies: -Many risks arise due to dependencies of project on outside agencies or factors. It is not easy to control these external dependencies. Some typical dependency-related risk factors are:-

- Availability of trained, experienced people.
- Intercomponent or inter-group dependencies.
- Customer-furnished items or information
- Internal and external subcontractor relationships.

2. Requirement Issues:-Many project face uncertainty and turmoil around the product's requirements. If we do not control requirements-related risk factors, we might either build the wrong product, or build the right product badly. Either situation results in unpleasant surprises and unhappy customers. Some typical factors:-

- Lack of clear product vision.
- Lack of agreement on product requirements.
- Unprioritized requirements.
- New market with uncertain needs.
- Rapidly changing requirements.

3. Management Issues:-Project managers usually write the risk management plan, and most people do not wish to air their weaknesses in public. If we do not confront such touchy issues, we should not be surprised if they bite us at some point.

- Inadequate planning and task identification
- Inadequate visibility into actual project status.
- Unclear project ownership and decision making.
- Unrealistic commitments made, sometimes for the wrong reasons.
- Staff personality conflicts.
- Poor communication.

4. lack of Knowledge:-The rapid rate of change of technologies, and the increasing change of skilled staff, means that our project teams may not have the skills we need to be successful. Some of the factors are:-

- Inadequate training.
- Poor understanding of methods, tools, and techniques.
- Inadequate application domain experience.
- New technologies.
- Ineffective, poorly documented, or neglected processes.

Q.89 Software project planning entails what activities? What are the difficulties faced in measuring the Software Costs?

Ans. Software project planning entails the following activities:

- Estimation:
  - –Effort, cost, resource, and project duration
- Project scheduling:
- Staff organization:
  - –staffing plans
- Risk handling:
  - -identification, analysis, and abatement procedures
- Miscellaneous plans:
  - –quality assurance plan, configuration management plan, etc.

Software costs are due to the requirement for software, hardware and human resources. One can perform cost estimation at any point in the software life cycle.

As the cost of software depends on the nature and characteristics of the project, the accuracy of estimate will depend on the amount of reliable information we have about the final product. So when the product is delivered, the costs can be actually determined as everything spend is known by then. However when the software is being initiated or during feasible study, we have only some idea about the functionality of software. There is very high uncertainty about the actual specifications of the system hence cost estimations based on uncertain information cannot be accurate.

Q.90 Explain the types of COCOMO Models and give phase-wise distribution of effort.

Ans. COCOMO model stand for Constructive Cost Model. It is an empirical model based on project experience. It is well-documented, independent model, which is not tied to a specific software vendor. Long history from initial version published in 1981(COCOMO-81) through various instantiations to COCOMO 2. COCOMO 2 takes into account different approaches to software development, reuse etc.
This model gives 3 levels of estimation namely basic, intermediate and detail.
1) Basic COCOMO model:- It gives an order of magnitude of cost. This model uses estimated size of software project and the type of software being developed.
The estimation varies for various types of projects and these various kinds are:-

- Organic-mode project:- These project include relatively small teams working in a familiar environment, developing a well understood application, the feature of such project are-

1) The communication overheads are low.
2) The team members know what they can achieve.
3) This project is much common in nature.

- Semi-detached model: A project consists of mixed project team of experienced and fresh engineers. The team has limited experience of related system development and some of them are unfamiliar with output and also some aspects of system being developed.
- Embedded model:- There will be very strong coupled hardware, software regulations and operational procedures. Validation costs are very high. For e.g. System program and development of OCR for English.

2) Intermediate COCOMO model:-
The intermediate COCOMO model estimates the software development effort by using 15 cost drivers' variables besides the size variable used in basic COCOMO.
3) Detailed COCOMO model:-
The detailed COCOMO model can estimate the staffing cost and duration of each of the development phases, subsystem, and modules. It allows you to experiment with different development strategies, to find the plan that best suits your needs and resources.

Q.91 Differentiate between function oriented design and object oriented design.

Ans. Function oriented design:- Function oriented design strategy relies on decomposing the system into a set of interacting functions with a centralized system state shared by these functions. Functions may also maintain local state information but only for the duration of their execution. Function oriented design

conceals the details of an algorithm in a function but system state information is not hidden.

Object oriented design:-Object oriented design transforms the analysis model created using object-oriented analysis into a design model that serves as a blueprint for software construction. It is a design strategy based on information hiding. Object oriented design is concerned with developing an object-oriented model of a software system to implement the identified requirements. Object oriented design establishes a design blueprint that enables a software engineer to define object oriented architecture in a manner that maximizes reuse, thereby improving development speed and end product quality.

Object oriented design Vs function oriented design-

- Unlike function oriented design methods, in OOD, the basic abstractions are not real world function such as sort, display, track etc. but real world entities such as employee, picture, machine etc.
- In OOD, state information is not represented in a centralized shared memory but is distributed among the objects of the system.

Q.92 What is stepwise refinement? Discuss partitioning & abstraction.

Ans. Stepwise Refinement:-Stepwise Refinement is a top-down design strategy originally proposed by Niklaus Wirth. A program is developed by successively refining levels of procedural detail. A hierarchy is developed by decomposing a macroscopic statement of function in a stepwise fashion until programming language statements are reached.

Refinement is actually a process of elaboration. We begin with a statement of function that is defined at a high level of abstraction. That is, the statement describes function or information conceptually but provides no information about the internal workings of the function or the internal structure of the information. Refinement causes the designer to elaborate on the original statement, providing more and more detail as each successive refinement occurs.

Partitioning:-Problems are often too large and complex to be understood as a whole. For this reason, we tend to partition such problems into parts that can be easily understood and establish interfaces between the parts so that overall function can be accomplished.

Partitioning decomposes a problem into its constituent parts. We establish a hierarchical representation of function or information and then partition the uppermost element by:-

1) Exposing increasing detail by moving vertically in the hierarchy or
2) Functionality decomposing the problem by moving horizontally in the hierarchy.

Abstraction:- Abstraction permits one to concentrate on a problem at some level of generalization without regard to irrelevant low level details; use of abstraction also permits one to work with concepts and terms that are familiar in the problem environment without having to transform them to an unfamiliar structure.

- It allows considering the modules at the abstract level without worrying about its details.
- It provides external behavior of the modules.
- It is used for existing modules as well as for modules that are being design.
- It is essential for the problem partitioning.

Q.93 Differentiate between failures and faults.

Ans. Failure:-Failure is the departure of external results of program operation from requirements. So failure is something dynamic. Failure can also be defined as deficiency in performance, attributes and excessive response time.

There are four general ways of characterizing failure occurrence in time:-

1) Time of failure.
2) Time interval between failures.
3) Cumulative failures experienced up to a given time.
4) Failure experienced in a time interval.

Various failure classes are transient, permanent, recoverable, unrecoverable, non-corrupting, and corrupting.

Fault:- fault is the defect in the program that, when executed under particular condition, causes of failure. A fault is a property of the program rather than a property of its execution or behavior.

Various fault classes are data faults, control faults, input/output faults, interface faults, storage management faults, and exception management faults.

Q.94 What do you understand by black box testing? Explain:
(i) Equivalence class
(ii) Equivalence partitioning

Ans. Black Box Testing:-Black Box Testing is also called behavioral testing, focuses on the functional requirements of the software. It enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. It is a complementary approach that is likely to uncover a different class of errors than white box testing.

Black Box Testing attempts to find errors in the following categories:-

1) Incorrect or missing functions.
2) Interface errors.
3) Errors in data structures or external database access.
4) Behavior or performance errors.
5) Initialization and termination errors.

Black Box Testing tends to be applied during later stages of testing because black box testing purposely disregards control structures; attention is focused on the information domain.

Equivalence class:-It represents a set of valid or invalid states for input conditions. An input condition is either a specific numeric value, a range of values, a set of related values, or a Boolean condition. Equivalence class may be defined according to the following guidelines:-

1) If an input condition specifies a range, one valid and two invalid equivalence classes are defined.
2) If an input condition requires a specific value, one valid and two invalid equivalence classes are defined.
3) If an input condition specifies a member of a set, one valid and one invalid equivalence class are defined.
4) If an input condition is Boolean, one valid and one invalid class are defined.

Equivalence Partitioning:-Equivalence partitioning is black box testing method that divides the input domain of a program into classes of data from which test cases can

be derived. An ideal test case single-handedly uncovers a class of errors that might otherwise require many cases to be executed before the general error is observed. Equivalence partitioning strives to define test case that uncovers classes of errors, thereby reducing the total number of test case that must be developed.

Q.95 Explain

(i) Reverse Engineering

(ii) Re-Engineering

Ans i) REVERSE ENGINEERING:-It is a process of analyzing software with a view to understanding its design and specification.

- In this, source code and executable code are the input.
- It may be part of a re-engineering process but may also be used to re-specify a system for re-implementation.
- Builds a program data base and generates information from this.
- Program understanding tools (browsers, cross reference generates, etc.) may be used in this process.
- Design and specification may be reverse re-engineer to:-
    a) Serve as input to SRS for program replacement.
    b) Be available to help program maintenance.

Reverse Engineering often precedes Re-Engineering but is sometimes worthwhile in its own right. The design and specification of a system may be reverse engineered so that they can be an input to the requirements specification process for the system replacement. The design and specification may be reverse engineered to support program maintenance.

ii) RE-ENGINEERING**:-** It is re-organizing and modifying existing system to make them more maintainable. It involves:-

- Source code translation.
- Reverse engineering.
- Program structure development.
- Program modularization.
- Data re-engineering.

Restructuring or re-writing part or all of the legacy system without hanging its functionality. Legacy system is a system that is hard to maintain. So it involves:-

1) Re-documenting the system.
2) Organizing and re-structuring the system.
3) Modifying and upgrading structure and value of the system data.
4) Input to a re-engineering process is a legacy system and output is a structure modularized version of the same program. So re-engineering involves adding effort to make them easier to maintain. The system may be restructured or redocumented.

When to Re-Engineer?

- When the system changes are mostly confined to part of the system then re-engineer that part.
- When hardware or software support becomes obsolete.
- When tools to support re-structuring are available.

Advantages of Re-Engineering:-

1) Reduced risk – there is a high risk in new software development. There may be development problems, staffing problems and specification problems.
2) Reduced cost – the cost of re-engineering is often significantly less than the cost of developing new software.

Re-Engineering cost factors:-

1) The quality of the software to be re-engineered.
2) The tool support available for re-engineering.
3) The extent of the data conversion, which is required.
4) The availability of expert staff for re-engineering.

Q.96 Explain various types of debugging techniques used in Software testing.

Ans. Debugging is the activity of locating and correcting errors. Various debugging techniques are:-

1) Core dumps:-A printout of all registers and relevant memory location is obtained and studies. All dumps should be well documented and retained for possible use on subsequent problems.

Advantages:-

1. The complete contents of a memory at a crucial instant of time are obtained for study.
2. Can be cost effective if used to explore validity of a well formulated error hypothesis.

Disadvantages:-

1. Require some CPU time, significant input time, and much analysis time.
2. Wasteful if used indiscriminately.
3. Hexadecimal numbers are cumbersome to interpret and it is difficult to determine the address of source language variables.

2) Traces:-Printout contains only certain memory and register contents and printing is conditional on some event occurring. Typical conditionings are entry, exit, or use of-

1) A particular subroutine, statement, macro, or database;
2) Communication with a terminal, printer, disk, or other peripheral;
3) The value of a variable or expression; and
4) Timed actuations in certain real time system.

A special problem with trace programs is that the conditions are entered in the source language and any changes require a recompilation.

3) Print statements:-The standard print statement in the language being used is sprinkled throughout the program to output values of key variables.

Advantages:-

1) This is a simple way to test whether a particular variable changes, as it should after a particular event.
2) A sequence of print statements portrays the dynamics of variable changes.

Disadvantages:-

1) They are cumbersome to use on large programs.
2) If used indiscriminately they can produce copious data to be analyzed much of which are superfluous.

4) Debugging programs:-A program which runs concurrently with the program under test and provides commands to examine memory and registers, stop execution of a program at a particular point, search for references to particular constants, variables, registers.

Advantages:-

1) Terminal oriented real time program.
2) Considerable flexibility to examine dynamics of operation.

Disadvantages:-

1) Generally works on a machine language program.
2) Higher-level language versions must work with interpreters.
3) More commonly used on microcomputers than large computers.

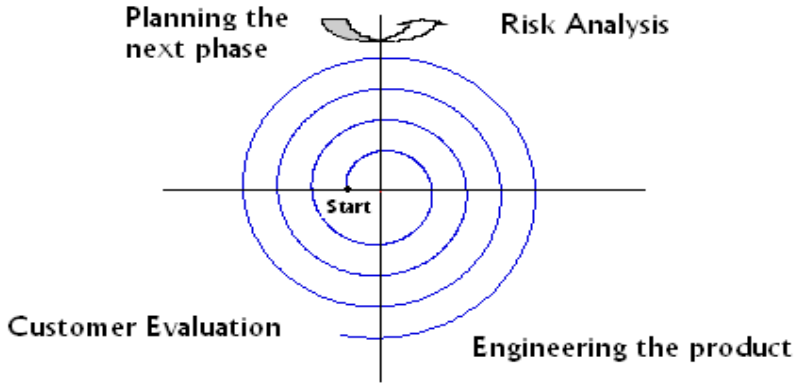
(ii) **Spiral Model:** The Spiral model is one of the popular model used for large projects. This model was proposed by Boehm in 1988 and it focuses on minimizing the risk through the use of prototype. We can view the Spiral Model as a waterfall model with each stage preceeded by Risk analysis stage. The model is divided into four quadrants, each with a specific purpose as shown in the fig. Each spiral represents the progress made in the project. In the first quadrant objectives, alternative means to develop product and constraints imposed on the products are identified. The next quadrant deals with identification of risk and strategies to resolve the risks. The third bottom right quadrant follows the waterfall model. In the bottom left quadrant customer evaluates the product requirements are further refined. If at some stage during the project risk cannot be resolved, project is terminated. The model is used if the requirements are very complex or some new technology is being introduced by the company.

Advantages:

1. The model tries to resolve all possible risks involved in the project.
2. Each phase of the model improves the quality of the product.

Disadvantages:

1. The model is suitable only for large size projects because in some cases the cost of risk analysis may exceed the actual cost of the project.
2. Expertise in risk management and project management is essential.



The Spiral Model

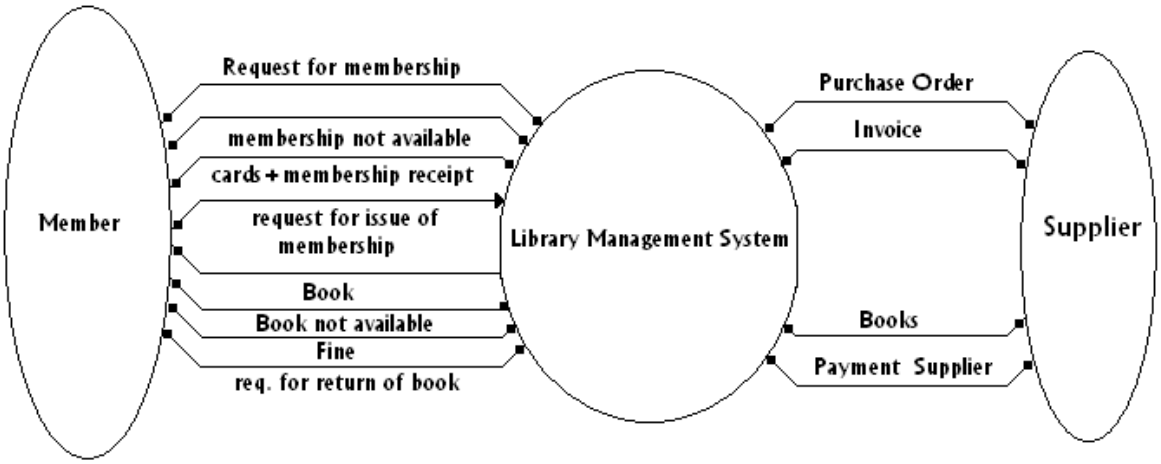
Optimizing: This is the highest level of process maturity in CMM. This level focuses on continuous process improvement in the organization using quantitative feedback, latest tools and technology. The important KPAs at this level are:

- Defect Prevention
- Technology Change Management

- Process Change Management

Q.97 What is the use of a data flow diagram? Explain the important concepts of data flow diagram.

Ans. A data flow diagram is used to show the functional view of an application domain. It shows all the important business processes and the flow of data between those processes. The main concepts used are:
(i) A process represents some kind of transformation on data. It takes as input one or more inputs and after doing necessary processing generates the output. It is represented by a circle with the name written inside as shown
(ii) Data Flow: A data flow represents data in motion and is represented by an arrow. The data flows represent the flow of data among processes, stores and external agents.
(iii) Data Store: A data store represents the data at rest. At the time of implementation it is represented by data base or files. Graphically I is shown as in figure.
(iv) External Agent: An external agent represents a person, a system or any other software which interacts with the system by providing necessary inputs and outputs. Graphically it is represented by a rectangle.
(b) Draw a context diagram for a Library management system.



Q.98 What is Software requirement Specification (SRS)? Why is it important? List the characteristic of a good quality SRS?

Ans: Software Requirement Specification Document is the output of requirement analysis stage o the software development life cycle. It documents all types of requirements and constraints imposed on the end product. This document is important because it is used in all the successive stages of SDLC. Any error introduced here will result in to incomplete and bad quality product. The characteristics of a good quality SRS are:
(i) Correctness
(ii) Completeness
(iii) Consistency
(iv) Unambiguousness
(v) Ranking for importance and/ or stability
(vi) Modifiability
(vii) Verifiability

(viii) Traceability
(ix) Design Independent
(x) Understandable by customer

Q.99 What is the difference between module coupling and module collection? List different types of coupling and cohesion.

Ans: Cohesion is the property of a single module and can be described as a glue that keeps the data elements within a single module together. While defining, we must aim for high cohesion. Different types of cohesion are:

- Coincidental Cohesion
- Logical Cohesion
- Temporal Cohesion
- Communicational Cohesion
- Sequential Cohesion
- Functional Cohesion
- Procedural Cohesion

Coupling on the other hand is the measure of dependence among modules. A designer must try for minimum coupling. Different types of coupling are:

- Content Coupling
- Common Coupling
- Control Coupling
- Stamp Coupling
- Data Coupling

Q 100 What are the different levels of testing? Explain in details.

Ans: The different levels of testing are:

- Unit Testing
- Integration Testing
- System Testing

Unit Testing: The unit testing is done to test the individual module of the software. Test cases are designed to test the program logic, functionality, interfaces etc. in a module. Since individual module is being tested, white box testing techniques are used here. Wherever required, stubs and drivers are also written to test the module, which increase the cost of the testing.
Integration Testing: In this the module are systematically integrated and tested to find interface problems, protocol design error, errors due to global values, input/ output format errors etc. Different strategies can be:

- Bottom up Integration: In this type of testing the modules at the leaf level are first tested and then we move up in the hierarchy. Drivers are used in this type of testing at different levels of hierarchy. Driver is a program which accepts the test case data to be inputted and printed by the module to be tested.
- Top Down Integration: In this case we start the top most module in the hierarchy and move down till the leaf modules are tested. Where ever required necessary stubs/ drivers are used. A stub is a program which simulates the module called by the module to be tested.

- Big-Bang Testing: In this all the modules after unit testing are combined and tested in one go. The problem with this kind of testing is debugging of errors.
- Sandwich Testing: This technique makes use of combination of top down and Bottom up testing.

System Testing: This testing focuses on validating the product with respect to software specification Document. Techniques like function testing (using Black Box testing), Performance testing and acceptance testing (to be done by end user) is done here to test the product.

Q.101 What are the advantages of using testing tools? Explain in detail different type of testing tools.

Ans: The advantages of testing tools are:

- They improve the productivity and quality of software development.
- Help in identification of errors which are difficult and time consuming to find manually.
- Reduce the testing time.
- Help in running large volumes of test unattended for 24 hours.
- Automatic generation of test cases.
- Automated the regression testing.
- Improve the productivity of the testers.

Some of the important testing tools are:

(a) Test Case Generators: These tools generate test cases from SRS, program or test design languages. They use certain rules called test design techniques to generate rest cases.

(b) Capture/ Playback and Test harness tools: These tools automate the re running of manual test by recording and replaying the test scripts. The recorded test scripts can also be edited as per need. They can be either intrusive or non intrusive type. Intrusive tools along with software under test reside on the same machine.

(c) Coverage Analysis Tools: These tools ensure that the software is tested and helps the tester to find the parts which are not covered.

(d) Test Comparators: These tools compare the results of software under test with the expected results and generate the report.

(e) Memory Testing Tools: These tools are used to test memory related problems. such as using uninitialized memory location, accessing memory locations which are out of range etc.

(f) Simulators: These tools are used to simulate the hardware/ software with which software under test is going to interact.

(g) Test database: It is a sample of database which is being manipulated by software under test.

Q102 Define Reverse Engineering? What are the main objectives of reverse engineering?

Ans: The reverse engineering is the process of generating representations that are implementation independent, starting from code. It is opposite of normal forward engineering process. The main objectives of the reverse Engineering process are:

(i) It helps the companies to understand the complexities of the system

(ii) Helps the analyst to generate useful lost information about legacy systems

(iii) Can be used to identify reusable components for analysis and future use
(iv) Helps in generating graphical representation of the system from different perspectives e.g. ER diagram, DFD, class diagram etc.
(v) Can be used as a part of Reengineering process.
(vi) Over a period of time modifications made to the software also result into unexpected problems. The anomalies can be detected using reverse engineering techniques.

Q.103 Write short notes on following maintenance models – Quick-fix Model and Iterative Enhancement model.

Ans: Quick-fix Model: This is the simplest model used for the maintenance of the software. In this model changes at the code level are made as early as possible without anticipating future maintenance problems. This model is not suitable for large and complex software systems. It should be used if the size of the software is small and is developed and maintained by one person. Also if the customers want fixing of bugs to be done immediately one can use this model. In that bugs can be fixed temporarily and parallely, proper correction can be done.

Iterative Enhancement model: This model incorporates changes in the software based on the analysis of the existing system. Also the complete documentation of the system is available before doing any changes. In case of any change made, all the documents at different stages i.e. SRS, Design document, testing document etc. are also modified so as to support the next change successfully.

Q.104 What is verification? (2)

Ans. Verification is the process of determining whether the output of one phase of software development conforms to that of its previous phase, whereas validation is the process of determining whether a fully developed system conforms to its requirements specification. Thus while verification is concerned with phase containment of errors, the aim of validation is that the final product be error free.

Q.105 Describe design walk throughs and critical design review. (8)

Ans.A design walkthrough is a quality practice that allows designers to obtain an early validation of design decisions related to the development and treatment of content, design of the graphical user interface, and the elements of product functionality. Design walkthroughs provide designers with a way to identify and assess early on whether the proposed design meets the requirements and addresses the project's goal.

For a design walkthrough to be effective, it needs to include specific components. The following guidelines highlight these key components. Use these guidelines to plan, conduct, and participate in design walkthroughs and increase their effectiveness.

1. Plan for a Design Walkthrough
2. Get the Right Participants
3. Understand Key Roles and Responsibilities
4. Prepare for a Design Walkthrough
5. Use a Well-Structured Process
6. Review and Critique the Product, not the Designer

7. Review, do not Solve Problems

The purpose of critical design review is to ensure that the detailed design satisfies the specifications laid down during system design. The critical design review process is same as the inspection process in which a group of people gets together to discuss the design with the aim of revealing design errors or undesirable properties.

Q.106 Explain the relationship between

(i) Productivity and difficulty

(ii) **Time and cost.** **(6)**

Ans

(i) Productivity and difficulty

Productivity refers to metrics and measures of output from production processes, per unit of input. Productivity P may be conceived of as a metrics of the technical or engineering efficiency of production. In software project planning , productivity is defined as the number of lines of code developed per person-month

Difficulty The ratio (K/t_d^2), where K is software development cost and t_d is peak development time, is called difficulty and denoted by D, which is measured in person/year.

$$\mathbf{D= (K/t_d^2)}$$

The relationship shows that a project is more difficult to develop when the manpower demand is high or when the time schedule is short.

Putnam has observed that productivity is proportional to the difficulty

$$P\alpha\ D^{\beta}$$

The average productivity may be defined as

P=Lines of code produced/Cumulative manpower used to produce code

=S/E

Where S is the lines of code produced and E is cumulative manpower used from t=0 to t_d (inception of the project to the delivery time)

(ii) Time and cost

In software projects, time cannot be freely exchanged against cost. Such a trade off is limited by the nature of the software development. For a given organization, developing software of size S, the quantity obtained is constant. We know

$K^{1/3} t_d^{4/3} = S/C$

If we raise power by 3, then Kt_d^4 is constant for constant size software. A compression of the development time t_d will produce an increase of manpower cost.. If compression is excessive, not only would the software cost much more, but also the development would become so difficult that it would increase the risk of being unmanageable.

Q 107 Write short notes on

(i) Configuration Management

(ii) Decision Table

Ans: (i) Configuration Management:

Due to several reasons, software changes during its life cycle. As a result of the changes made, multiple versions of the software exist at one time. These changes must be managed, controlled and documented properly in order to have reliable systems. Configuration management helps the developers to manage these changes systematically by applying procedures and standards and by using automated tools.

Though multiple versions of the software exist in the tool repository, only one official version of set of project components exist called baseline. The different components of the baseline are called configuration items. Some examples of configuration items are project plan, SRS, Design document, test plans, user manuals etc. A group of people constitute Configuration Control Board (CCB) which controls the changes to be made to the software. Whenever changes are to be made the following steps are followed:

(i) Submit the change request along with details to CCB

(ii) CCB accesses the change request after proper evaluation.

(iii) Depending upon the results, the request is either accepted or rejected or can be deferred for the future assessment.

(iv) If accepted, proper plan is prepared to implement the change.

(v) Once changes are made, after validating by the quality personnel, all configuration items are updated.

Some popular configuration management tools are Clear CASE, Visual Source Safe etc.

(ii) Decision Table: When the process logic for a process involves multiple conditions and is very complicated, it is not advisable to use structured English. Instead decision tables are used. Main parts of the table are:

1. Condition Stubs
2. Action Stubs
3. Roles

Condition stubs list all the conditions relevant to the decision. Action part lists all the actions that will take place for a valid set of conditions. Finally rules part of the table specify the set of conditions that will trigger a particular action. A sample decision table is shown below:

	Rule 1	Rule 2	Rule 3
Condition 1	X	X	
Condition 2		X	
Condition 3	X		X
Condition 4			X
Action 1	X		
Action 2		X	
Action 3			X

From this table it is clear that if conditions 1 and 3 are satisfied, Action 1 will be triggered.

PART III

NUMERICALS

Q 1 For the flow graph shown in fig ,

(i) compute the McCobe's cyclomatic complexity

(ii) Find out independent path

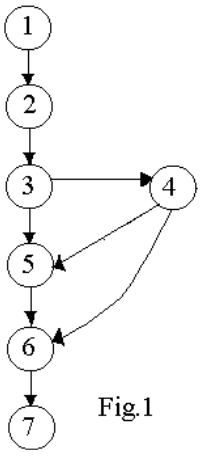


Fig.1

Ans

(i) Cyclomate complexity, V (G)

V(G) = E-N+2

Where E is the no of flow graph edges and N is the no of nodes .

So we have E = 8 N = 7 and V(G) = 8-7+2 = 3

(ii) independent path

(i) 1,2,3,5,6,7

(ii) 1,2,3,4,5,6,7

(iii)1,2,3,4,6,7

Q 2 Drive an expression for peak manning of a project using Nordes / Rayleigh equation .

Ans Norden Rayleigh curve used to approximate software project .

M(t) =Man power utilization

$= dy/dt = 2kate^{-at2}$ ----------(1)

K = Area under the curve t=time

Integrating eg . (1) with respect to t

$Y(t) = k[1-e^{-at2}]$ ----------(2)

Differentiating eq (1) with respect to t

$M'(t) = d^2y/dt^2 = 2kae^{-at2}(1-2^{at2})$ ---------(3)

Put $a=1/2t_d^2$ and $t=t_d$ in eq (1)

$M(t) = 2kate^{-at2} = 2k*1/2t^2_d *t^d e^{-1/2t_d^2 *t_d}$

$= k/t_de^{-1/2} = k/t_de^{1/2}$

$M0 = k/t_de^{1/2}$

M0 = peak manning of a project .

Q 3 Assuming the Putnam model and given S = 100,000 C = 5000 and D_0 = 15 calculate the development time t_d. (4)

Ans: In Putnam model
$(S/C)^3 = D_0 t_d^7$
lhs = $20^3 = 8000$
$t_d^7 = 8000/ 15$
t_d is seventh root of 533.33

Q 4. Define the failure intensity of the Basic model. (4)

Ans: The failure intensity of the Basic model is

$$\lambda(\mu) = \lambda_0 [1 - \mu / v_0]$$

where λ_0 is the initial failure intensity and μ is the average number of failures expected at any given point in time. The quantity v_0 is the total number of failures that would occur at infinite time.

Q 5. Assume a program will experience a total of 200 failures. Initial failure intensity is 16 failure/ CPU hr. It has now experienced 50 failures. Determine the following after specifying the formula

a. Current failure intensity
b. Decrement of failure intensity
c. Failure intensity at 100 CPU hr. (4 x 3)

Ans:

i. Current failure intensity
$\lambda(\mu) = \lambda_0 [1 - \mu / v_0] = 16 [1- 50 / 200] = 12$ failure / CPU hr

ii. Decrement of failure intensity
$d\lambda / d\mu = -\lambda_0 / v_0 = 16 / 200 = -0.08$ CPU hr

iii. Failure intensity at 100 CPU hr
$\lambda(\tau) = \lambda_0 \exp(- \lambda_0 \tau / v_0)$
$= 16 \exp (-16 * 100 / 200) = 16 \exp(-8)$

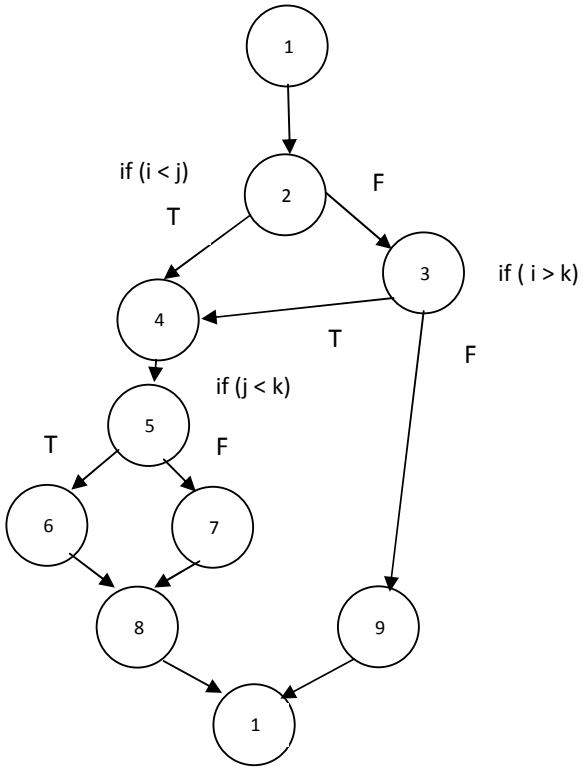
Q 6 Consider the program given below

```
void main()
        {
            int i,j,k;
          readln (i,j,k);
            if( (i < j) || ( i > k) )
            {
                writeln("then part");
                if (j < k)
                        writeln ("j less then k");
                            else writeln ( " j not less then k");
            }
          else writeln( "else Part");        }
```

(i) Draw the flow graph. (4)
(ii) Determine the cyclomatic complexity. (4)
(iii) Arrive at all the independent paths. (8)

Ans:

```
void main()
{
1    int i,j,k;
2   readln (i,j,k);
3      if( (i < j) || ( i > k) )
       {
4           writeln("then part");
5           if (j < k)
6                   writeln ("j less then k");
7                       else writeln ( " j not less then k");
8    }
9   else writeln( "else Part");
}
```



(ii) Cyclomatic complexity = E - N + 2 = 12 – 10 + 2 = 4
(iii) The four independent paths are
Path1 : 1 2 3 9 10
Path2 : 1 2 4 5 7 8 10
Path3 : 1 2 4 5 6 8 10
Path4 : 1 2 3 4 5 7 8 10

Q 7 . a) Define Annual Change Traffic (ACT) and Annual Maintenance Effort (AME) of Boehm model. (4)

b) For a software system of 90 KLOC, 5KLOC of code was added. The initial development cost was Rupees 8 lakhs with a total KLOC of 50. Total

lifetime for the software is 10 years. Compute ACT and the total cost of the system. (6)

Ans: a) Boehm proposed a formula for estimating maintenance costs as part of his COCOMO model. He used a quantity called Annual Change Traffic (ACT) which is defined as:

" The fraction of a software product's source instructions which undergo change during a year either through addition, deletion or modification

ACT is defined as

$$ACT = (KLOC_{added} + KLOC_{deleted}) / KLOC_{total}$$

(Annual Maintenance Effort)

AME is defined as ACT X SDE

where SDE is the software development effort in man-months]

ACT: Annual Change Traffic.

b) ACT = (5) / 50 = 0.10

Development cost = Rs. 8 lakhs.

Life time = 10 years

Total cost = development cost + 10 * (development cost * 0.10)

= 8 + 10 * .8 = 16 lakhs

Q 8. Consider a project for which the manpower requirement is 200 PY and the development time is 2 years.
calculate the peak manning time.

Ans Software Project cost K=200 PY

Peak development time t_d =2 year =2 years

Peak manning $m_0=(k/ t_d * sqrt(e))=200/2*1.648=164.8 \approx 164$ persons

Q.9 For the program given below, calculate the values of software science measures like η, N, V,E and λ.

1.	**int. sort (int x[], int n)**
2.	**{**
3.	**int i, j, save, im1;**
4.	**/*This function sorts array x in ascending order*/**
5.	**If (n<2) return 1;**
6.	**for (i =2; i<=n;i++)**
7.	**{**
8.	**im1=i-1;**
9.	**for (j=1;j<=im;j++)**
10.	**if (x[i]<x[j])**
11.	**{**
12.	**Save = x[i];**
13.	**x [i]=x[j];**
14.	**x [j]=save;**
15.	**}**
16.	**}**
17.	**return 0;**
18.	**}**

Ans:

This list of operators and operands is given in Table.

Operators	Occurrences	Operands	Occurrences
int	4	SORT	1
( )	5	x	7
,	4	n	3
[ ]	7	i	8
if	2	j	7
<	2	save	3
;	11	im1	3
for	2	2	2
=	6	1	3
-	1	0	1
<=	2	-	-
++	2	-	-
return	2	-	-
{ }	3	-	-
$\eta_{1=14}$	N_1=53	$\eta_{2=10}$	N_2=38

Here N1=53 and N2=38. The program length N =N_1+N_2=91
Vocabulary of the program $\eta = \eta_{1+} \eta_2$=14+10=24
Volume V=Nx$\log_2 \eta$
=91x $\log_2$ 24=417 bits.
The estimated program length N of the program
=14 $\log_2$14+10 $\log_2$10
=14*3.81+10*3.32
=53.34+33.2=86.45
Conceptually unique input and output parameters are represented by η^*_2.
η^*_2=3 {x:array holding the integer to be sorted. This is used both as input and output}.
{N: the size of the array to be sorted}.
The potential volume V*=5 $\log_2$ 5=11.6
Since L=V*/V
=11.6/417=0.027
D=I/L
=1/0.027=37.03
Estimated program level

$$\hat{L} = \frac{2}{\eta_1} x \frac{\eta_2}{N_2} = \frac{2}{14} x \frac{10}{38} = 0.038$$

We may use another formula

$$\hat{V}^* = V x \hat{L} = 417 x 0.038 = 15.67$$

The discrepancy between V* and $\hat{V}^*$ does not inspire confidence in the application of this portion of software science theory to more complicated programs.

E=V/ $\hat{L}$= $\hat{D}$ x V
= 417/0.038=10973.68
Therefore, 10974 elementary mental discriminations are required to construct the program.

$$T = E/\beta = \frac{10974}{18} = 610 \text{ seconds } \approx 10 \text{ minutes}$$

Q.10 **A software project is planned to cost 95PY in a period of 1 year and 9 months. Calculate the peak manning and average rate of software them build up.** **(5)**

Ans. Software project cost K=95PY
Peak development time t_d=1.75years

$$\text{Peak manning} = m_o = \frac{K}{t_d\sqrt{e}}$$

$$\frac{95}{1.75x1.648} = 32.94 = 33 \text{ persons}$$

Average rate of software team build up
$=m_o/t_d=33/1.75=18.8$ person/ year or 1.56 person / month.

Q.11 **Assume that a program will experience 200 failures in infinite time. It has now experienced 100. The initial failure intensity was 20 failures/CPU hr.** **(6)**

(i) Determine the current failure intensity

(ii) Find the decrement of failure intensity per failure.

Ans: Here

$$V_o = 200 \text{ failure}$$
$$\mu = 100 \text{ failure}$$
$$\lambda_o = 20 \text{ failures / CPUhr.}$$

(i) Current failure intensity:

$$\lambda(\mu) = \lambda_o\left(1 - \frac{\mu}{V_o}\right)$$

$$= 20\left(1 - \frac{100}{200}\right) = 20(1 - 0.5) = 10 \text{ failures / CPU hr.}$$

(ii) Decrement of failure intensity per failure can be calculated as:

$$\frac{d\lambda}{d\mu} = \frac{-\lambda_o}{V_o} = -\frac{20}{200} = -0.1 \text{ / CPU hr.}$$

Q.12 **Compute function point value for a project with the following domain characteristics:**

No. of I/P = 30
No. of O/P = 62
No. of user Inquiries = 24
No. of files = 8
No. of external interfaces = 2

Assume that all the complexity adjustment values are average. Assume that 14 algorithms have been counted.

Ans. We know

UFP=$\Sigma W_{ij} Z_{ij}$ where j=2 because all weighting factors are average.

=30*4+62*5+24*4+8*10+2*7

=120+310+96+80+14

=620

CAF=$(0.65+0.01\Sigma F_i)$

=0.65+0.01(14*3)

=0.65+0.42

=1.07

nFP=UFP*CAF

=620*1.07

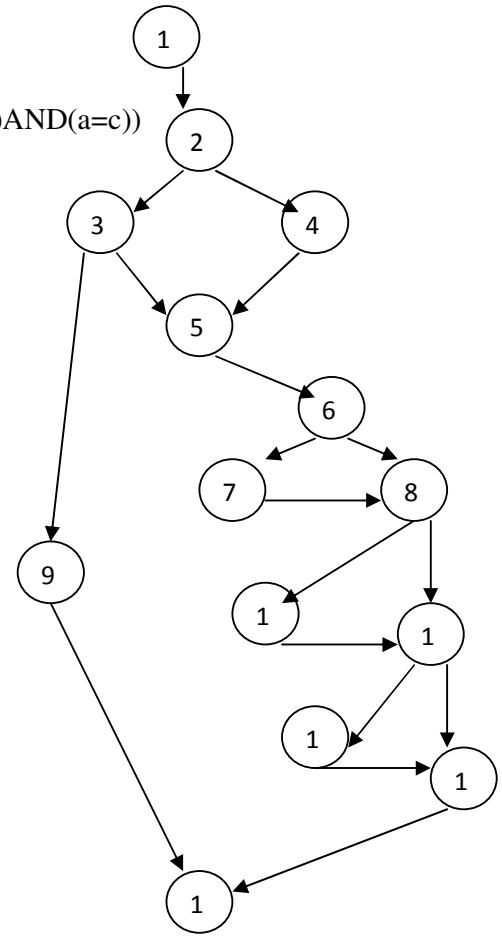
=663.4≈663

Q.13 Consider a program that reads a set of Data for 'n' no. of triangles. The program reads three integer values as representing the sides of triangles. The program prints for each triangle whether the triangle is isosceles or equilateral or a simple. Develop logic and do the following:
(i) Compute cyclomatic complexity?
(ii) Design test cases for loop testing?

Ans. The program logic will be as follows:

Enter three sides of a triangle.
Read a, b and c
If(a<b+c)AND(b<a+c)AND(c<a+b)
  Then is_a_triangle=TRUE
Else is_a_triangle=FALSE;
IF is_a_triangle
Then
  If(a=b)XOR(a=c)XOR(b=c) AND NOT ((a=b)AND(a=c))
    Then print "Triangle is Isosceles"
  If(a=b)AND(b=c)
    Then print "Triangle is Equilateral"
  If(a<>b)AND (a<>c) AND(b<>c)
    Then print "Triangle is scalene"
Else
  Print "Not a triangle"

The flow graph of the problem is given as in the fig:



(i) There are 6 independent paths i.e.
1,2,3,9,14
1,2,4,5,6,8,11,13,14
1,2,3,5,6,8,11,13,14
1,2,4,5,6,7,8,11,13,14
1,2,4,5,6,8,10,11,13,14
1,2,4,5,6,8,11,12,13,14
Cyclomatic complexity is 6

(ii) Few test cases are:

Test #	Description	Test Data
1	A valid scalene triangle.	a = 2, b = 3, c = 4
2	An illegal triangle with a zero length side.	a = 0, b = 1, c = 1
3	An illegal triangle with one short side.	a = 1, b = 4, c = 2
4	A valid equilateral triangle.	a = 1, b = 1, c = 1
5	A valid isosceles triangle.	a = 1, b = 2, c = 2

Q.14 Consider a program which computes the square root of an input integer between 0 and 5000. Determine the equivalence class test cases. Determine the test cases using boundary value analysis also.

Ans. For a program that computes the square root of an input integer which can assume values in the range of 0 to 5000, there are three equivalence classes: The set of negative integers, the set of integers in the range of 0 and 5000, and the integers larger than 5000. Therefore, the test cases must include representatives for each of the three equivalence classes and a possible test set can be: {-5,500,6000}.
Boundary value analysis leads to selection of test cases at the boundaries of the different equivalence classes. For a function that computes the square root of integer values in the range of 0 and 5000, the test cases must include the following values: {0, -1,5000,5001}.

Q.15 For the structural chart given in fig.1 calculate the Information Flow index of individual modules as well as whole software.

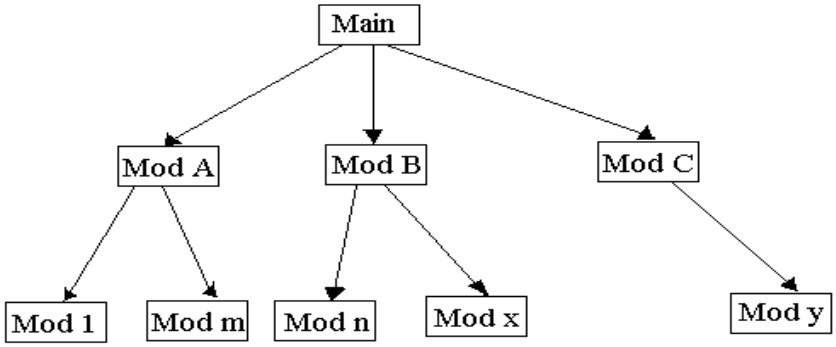


Fig.1

Ans:

Module	Fan-in	Fan-out	Information Flow index of module(Fan-in*Fan-out)^2
main	2	2	16
A	2	2	16
B	2	2	16
C	2	1	4
l	0	1	0
m	1	1	1
n	0	1	0
x	1	1	1

y	1	0	0

Total information flow = IF (main) + IF (A) + IF (B) + IF (C) + IF (I) + IF (m)
+ IF (n) + IF (x) + IF (y)
= 16 + 16 + 16 + 4 + 0 + 1 + 0 + 1 + 0
= 54

Q.16 For the flow graph shown in Fig2, compute McCabe's Cyclomatic Complexity.

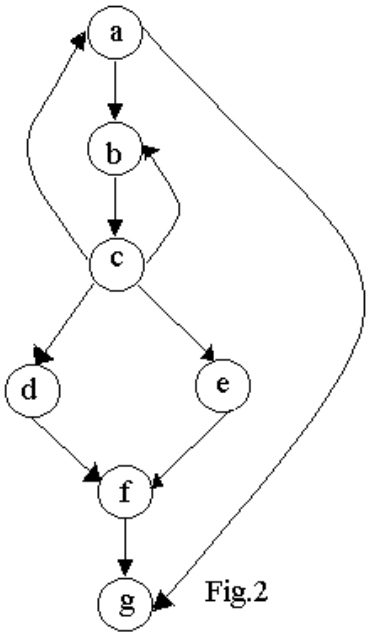


Fig.2

Ans:In this flow graph, Number of regions are 5, hence the Cyclomatic complexity is = 5
OR
No of edges = 10
No of nodes (N) = 7
Hence Cyclomatic complexity + E-N+2 = 10-7+2 = 5

Q 17 Write a program in C to add two numbers and then calculate the values of the following software science metrics :

(i) n(vocabulary of a program)
(ii) N (program length)
(iii)V(Volume)
(iv)L(program level)

Ans

```
Void Main()
{ int i,j,sum
 scanf("%d%d",i,j);
sum = i+j;
printf("%d" ,sum);}
```

operators	occurrences	operands	occurrences
main()	1		
		i	2

		j	2
scanf	1		
		sum	2
printf	1		
n1=3	N1=3	n2=3	N2=6

(i) n(vocabulary of a program) = n1 + n2 = 6

(ii) N (program length) = N1 + N2 = 9

(iii) V(Volume) = N $\log_2$ n

= 9 $\log_2$ 6

(iv) L(program level) = 2/n1 * n2/N2

= 2/3 * 3/6

= 1/3

= 0.33