Section - A

1) How ADT helps in programming?
2) Briefly explain the functions of DMA(Dynamic Memory Allocation).
3) Differentiate between array and linked list.
4) What do you mean by asymptotic notation?
5) Enlist real world examples using asymptotic notation.
6) Explain the time complexity of linear and binary search in the best,worst and average case?
7) What the key condition for applying binary search?
8) Prove that ADT enhance programming efficiency and development?
9) Discuss linear and non-linear data structures with example.
10) Describe various form of data structure.
11) Define data structure and also describe the difference between primitive and non primitive data structure.
12) List the applications using stack data structure.
13) Explain following mathematical notation:
    (i) Infix (ii) Prefix (iii) Postfix
14) Write an algorithm to convert prefix expression to postfix expression. Carefully state any assumption you make regarding the input.
15) Define priority queue. How does it differ from a regular queue?
16) Explain the linked list implementations of enqueue and dequeue operations in queue using C program.
17) List the basic operations that can be performed on a queue?
18) How do you create a queue in C? Provide an example.
19) What is a circular queue, and how does it differ from a linear queue?
20) Explain the enqueue and dequeue operations in a circular queue.
21) Difference between linear search and binary search algorithm.
22) Write most commonly used operations in data structure.
23) What do you mean by worst case time complexity of an algorithm. Explain clearly.
24) How does a **Dequeue** differ from a normal queue?
25) What are advantages and disadvantages of doubly linked list over singly linked list?
26) Explain the divide-and-conquer strategy?
27) Difference between B tree and B + tree.
28) Discuss the property of AVL tree in the context with binary search trees.
29) Explain two traversal strategies used in traversing a graph?
30) Discuss an undirected acyclic graph?
31) Define height and depth of tree?
32) Define the properties of Graph?
33) Define indegree and out degree of a graph?
34) Difference between a Binary Tree and a Binary Search Tree.
35) What is a circular linked list?
36) List two applications of queues.
37) What is a binary search tree?
38) List any two applications of stacks.

39) Explain Warshall's algorithm for transitive closure of a graph with a step-by-step example.
40) Differentiate between a full binary tree and a complete binary tree.

Section -B

1) Consider a two-dimensional array Marks [10][5] having its base address as 2000 and the number of bytes per element of the array is 2. Now, compute the address of the element, Marks [8][5], assuming that the elements are stored in column major order
2) Explain sparse matrix and the ways in which it is represented in the memory?
3) Explain doubly linked list . Write the algorithm for insertion operation in doubly Linked list
4) Define stack and it's implementaion in C.
5) Explain different operations on stack with algorithm & program.(insertion ,deletion and searching)
6) Demonstrate and convert the infix expression **(((A+B)*(C-D))/E)-(F*(G+H)/I+J*K)** to postfix expression using stack.
7) Write a C program to implement insertion of element at end node circular linked list for following functions.
8) Explain with necessary algorithm the representation of stack using linked list and array.
9) Write algorithm to convert a postfix expression into an infix expression. Consider the following arithmetic expression in postfix notation: **752+* 4 1 5- /-**
(i) Find the value of the expression.
(ii) Find the equivalent prefix form of the above expression.
9) Convert infix to postfix ***(a+b)*(c−d)−(g+j)*(h−i)/(m+n)*** using stacks and write the algorithm.
10) Write a C program to implement a queue using a linked list.
11)What is a **Priority Queue**? How does it differ from a regular queue?Write a C program to implement a priority queue using an array.
12)Give the algorithm for binary search. What is its time complexity? Compare its time complexity with that of linear search.
13)Explain how stacks are used to convert an infix expression to a **prefix (Polish notation)** and **postfix (Reverse Polish notation) expression.** Provide a step-by-step manual conversion of the infix expression:
(A+B)∗ (C−D)/E(A + B) * (C - D) / E(A+B)∗ (C−D)/E

14)Explain the step-by-step evaluation process for the postfix expression:

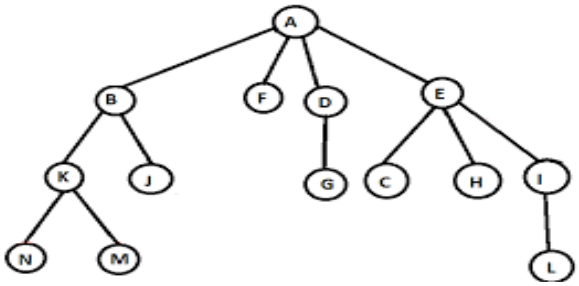
63+2∗ 4/6 3 + 2 * 4 /63+2∗ 4/

15)Discuss the conditions that lead to **stack overflow** and **stack underflow.** Provide C code snippets to demonstrate the push and pop operations.
16)Provide a C program that implements these operations (**Push, Pop, Peek, and IsEmpty**.)using an **array-based stack**.
17)Illustrate different types of arrays with syntax and its operations.
18)Write the condition of overflow and underflow in singly linked list
19)Write an algorithm to convert a valid arithmetic infix expression into its equivalent postfix expression.
20) Why binary search is preferred over sequential search for large datasets.

21) Illustrate how searching in an unordered list differ from searching in an ordered list with example.
22)Explain the role of searching? List the applications using searching techniques in data structure.
23) Illustrate linear and binary search alorithm with example.
23)Discuss the implementation of single-linked list. Write C function to implement following operations on singly-linked list.

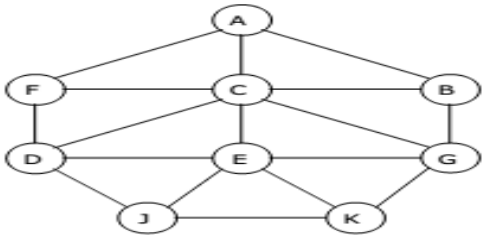
a. To count number of nodes
b. To reverse the direction of links.
c. To delete alternate nodes that is first, third, fifth and so on.

24) Given a sequence of numbers: 32,1,5,10,23,20,41,55,2,9,44,30,25,17,28. Apply Merge sort to sort the array.
25) Elaborate fundamental operations of a binary search tree, including insertion, deletion, and searching.
26) Explain the quick sort algorithm. Discuss its partitioning process, pivot selection strategies, and apply quick sort to sort the given array: 25, 84, 10, 57, 117, 156, 48, 73, 36, 105, 122, 92, 149, 69, 134.
27) Write and explain Selection sort algorithm. Discuss the strategy used for selecting the key and sort the given array: 47, 82, 15, 63, 29, 54, 91, 38, 72, 26, 59, 10, 42, 77, 93 using selection sort.
28) Perform the following traversals on the tree:
In-order, Pre-order, Post-order.



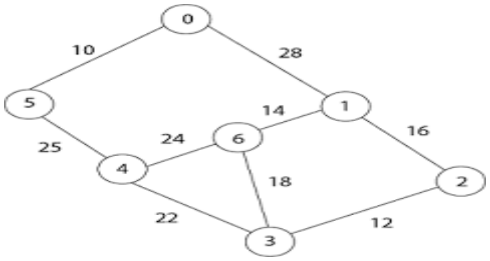
29) Construct a binary tree from preorder to inorder.
Postorder - **9,1,2,12,7,5,3,11,4,8**
Inorder- **9,5,1,7,2,12,8,4,3,11**

30) Describe the Depth-first search algorithm for traversing graphs.
Implement the same for the given graph: Consider J as The source Node.



A Graph G

31) Describe Kruskal's algorithm and find the cost of minimum spanning tree using Prim`s Algorithm.



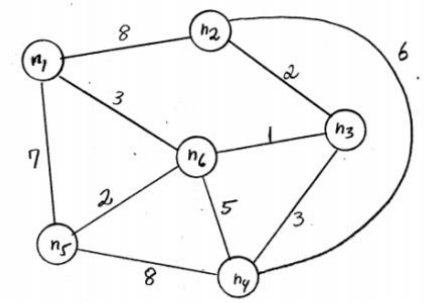
32) Given a sequence of numbers

12	**3**	**45**	**1**	**56**	**4**	**89**

Using the Heap Sort algorithm, sort the numbers in ascending order.

33) Given a sequence of numbers: **34,23,67,45,12,54,87,43,98,75,84,93,31.** Construct Binary search tree and delete 45 and 93.
34) Describe the difference between BFS and DFS with an example.
35) Write and explain Bubble sort algorithm. Discuss the strategy used for sort the given array: **47, 82, 15, 63, 29, 54, 91, 38, 72, 26, 59, 10, 42, 77, 93** using bubble sort.
36) Differentiate binary search and linear search? Apply binary search to find item 40 in the sorted array: **11, 22, 30,33,40,44, 55, 60,66, 77, 80,88,99.** Also discuss the complexity of binary search.
37) Construct a binary search tree using the elements: 30, 20, 40, 10, 25, 35, 50. Show in order, preorder, and post order traversals.
38) Construct a binary tree using
Preorder - 1,2,4,8,9,10,11,5,3,6,7
Inorder- 8,4,10,9,11,2,5,1,6,3,7

39) Write the Floyd Warshall algorithm to compute the all-pair shortest path. Apply the algorithm on following graph.



40) Construct a binary tree using
Preorder - D, B, E, A, F, C
Inorder- D, E, B, F, C, A