

ASSIGNMENTS **(Theory)**

Program : BCA Semester : 3rd
Course : Data Structure

Instructions :

1. All the Questions Carry 8 Marks Each (Total Marks 200)
2. All the Questions are Mandatory

Module I: Introduction and Arrays

1. Define data structures. Classify various types with examples.
2. What are the advantages and disadvantages of arrays?
3. Describe the structure of a one-dimensional array. How are elements accessed and stored?
4. How is an array initialized and accessed in memory? Explain the concept of index calculation.
5. Write a program to perform matrix addition using 2D arrays.

Module II: Stacks and Queues

1. Define a stack. Explain push and pop operations with an example.
2. Differentiate between stack and queue in terms of structure and usage.
3. What are the different types of queues? Explain with diagrams.
4. Explain the role of stacks in expression evaluation and parsing.
5. Write a program to implement a circular queue using arrays.

Module III: Linked Lists

1. Define a singly linked list. Write an algorithm to insert a node at the end.
2. Explain the differences between singly, doubly, and circular linked lists with diagrams.
3. What are the advantages of using a linked list over an array?
4. Compare arrays and linked lists in terms of memory usage, performance, and flexibility.
5. Discuss the traversal and search operations in a singly linked list.

Module IV: Trees

1. Define a binary tree. Explain its types and applications.
2. Compare binary tree and binary search tree.
3. What are tree traversal techniques? Explain inorder, preorder, and postorder traversal with an example.
4. Write an algorithm for inorder traversal using recursion.
5. Describe the use of trees in hierarchical data representation.

Module V: Graphs and Sorting

1. Explain the adjacency matrix and adjacency list representations of graphs.
2. Compare BFS and DFS with respect to data structure usage and traversal order.
3. What is hashing? Explain different collision resolution techniques.
4. Compare and contrast bubble sort and selection sort in terms of time complexity and usage.
5. Write an algorithm for merge sort and explain its divide-and-conquer approach.