

Minimum Set of Lab Programs for Data Structures

(MCA-2T1-C)

Module I

Write a Program in C/C++ to:

1. Create One dimensional array of Integers and then perform the following operations on it:

- a) Read the Elements in an Array
- b) Display the Elements in Input Order
- c) Display the Elements in Reverse Order
- d) Search a Particular Element
- e) Replace a Particular Element
- f) Insert a Missed Element
- g) Sort the Array Elements in any Order
- h) Insert a Missed Element (after sorting)
- i) Delete a Particular Element
- j) Frequency of a Particular Element
- k) Display the Distinct Elements
- l) Frequency of Distinct Elements
- m) Display the Duplicate Elements
- n) Remove the Duplicate Elements
- o) Display the sum of all Elements
- p) Get the Average of all Elements
- q) Display the Maximum Value Element
- r) Display the Minimum Value Element

2. Create a Two dimensional array of Integers and then perform the following operations on it:

- a) Read the elements of the 2D array
- b) Display the elements
- c) Get Transpose of Matrix (2D Array)

- d) Perform the addition & subtraction of two matrices(2D arrays)
- e) Perform the Multiplication of Two Matrices(2D arrays)
- 3. **Generate the information matrix from a Sparse Matrix and then regenerate the Original Matrix.**
- 4. **Perform a Linear Search as well as the Binary Search on an array of Integers**
- 5. **Perform Bubble Sort & Insertion Sort on an array of Integers**
- 6. **Create a Stack and perform all the permissible operations of Stack**
- 7. **Create a Simple Queue and perform the permissible operations of Simple Queue**
- 8. **Create a Circular Queue and perform the permissible operations of Circular Queue**
- 9. **Convert an Infix expression to its corresponding postfix expression and then evaluate it as an application of Stack.**
- 10. **Get the Factorial of a number using Recursion.**

Module II

Write a Program in C/C++ to:

- 11. **Create a Singly Linked List and perform the following operations on it:**
 - s) Display the Node Information in Input Order
 - t) Search a Particular Node
 - u) Replace a Particular Node Information
 - v) Insert a Missed Node Information (at the Beginning, at Specific Location or at the end)
 - w) Sort the Node Information in any Order (Prefer Insertion Sort)
 - x) Sort the Nodes based on Address exchange
 - y) Insert a Missed Node (after sorting)
 - z) Delete a Particular Node (at the Beginning, at Specific Location or at the end)
 - aa) Find Frequency of a Particular Node Information
 - bb) Reverse the List
- 12. **Create a Doubly Linked List and perform the following operations on it:**
 - a) Display the Node Information in Input Order
 - b) Search a Particular Node
 - c) Replace a Particular Node Information
 - d) Insert a Missed Node Information (at the Beginning, at Specific Location or at the end)
 - e) Sort the Node Information in any Order (Prefer Insertion Sort)
 - f) Sort the Nodes based on Address exchange

- g) Insert a Missed Node (after sorting)
- h) Delete a Particular Node (at the Beginning, at Specific Location or at the end)
- i) Find Frequency of a Particular Node Information
- j) Display the Node Information in Reverse Input Order

13. Create a Circular Linked List and perform the following operations on it:

- 11. Display the Node Information
- 12. Search a Particular Node
- 13. Replace a Particular Node Information
- 14. Insert a Missed Node Information
- 15. Sort the Node Information in any Order
- 16. Sort the Nodes based on Address exchange
- 17. Insert a Missed Node (after sorting)
- 18. Delete a Particular Node
- 19. Find Frequency of a Particular Node Information
- 20. Display the Node Information in Reverse Order

14. Create a Linked Stack and perform all the permissible operations of Stack

15. Create a Linked Queue and perform all the permissible operations of Queue

16. Create a Binary Tree and use the traversal Techniques to display the nodes/vertices of Binary Tree?

17. Create a Binary Search Tree (BST) and then perform the following operations on it:

- a) Visit the nodes of BST (InOrder, PreOrder & PostOrder)
- b) Delete any particular leaf node
- c) Delete a node with only One SubTree (Left or Right)
- d) Delete a node with both SubTrees

18. Perform Selection Sort, Quick Sort & Merge Sort on an array of Integers.