Program 1: Array Operations

Array: 10 20 30 40 50
Enter element to insert: 25
Array after insertion: 10 20 25 30 40 50
Enter position to delete: 3
Array after deletion: 10 20 25 40 50

Program 2: Stack Operations

STACK OPERATIONS
1.PUSH
2.POP
3.VIEW
4.QUIT
Enter Choice : 1
Enter Stack element : 12
Enter Choice : 1
Enter Stack element : 23
Enter Choice : 1
Enter Stack element : 34
Enter Choice : 1
Enter Stack element : 45
Enter Choice : 3
Top--> 45 34 23 12
Enter Choice : 2
Popped element is 45
Enter Choice : 3
Top--> 34 23 12
Enter Choice : 4

Program 3: Queue Operations

QUEUE OPERATION
1.INSERT
2.DELETE
3.VIEW
4.QUIT
Enter Choice : 1
Enter element to be inserted : 12
Enter Choice : 1
Enter element to be inserted : 23
Enter Choice : 1

Enter element to be inserted : 34
Enter Choice : 1
Enter element to be inserted : 45
Enter Choice : 1
Enter element to be inserted : 56
Enter Choice : 1
Queue Full
Enter Choice : 3
Front--> 12 23 34 45 56 <--Rear
Enter Choice : 2
Enter position to delete (0 to 4) : 0
Element deleted : 12
Enter Choice : 3
Front--> 23 34 45 56 <--Rear
Enter Choice : 2
Enter position to delete (0 to 3) : 2
Element deleted : 45
Enter Choice : 3
Front--> 23 34 56 <--Rear
Enter Choice : 4
Exiting program.

Program 4: Infix to Postfix Conversion

Enter infix expression: (A+B)*C
Postfix expression: AB+C*

Program 5: Postfix Evaluation

Enter a postfix expression: 23*54*+9-
Evaluated result: 17

Program 6: Singly Linked List Operations

Enter value to insert at beginning: 10
Enter value to insert at end: 20
Enter value to insert at end: 30
Enter key after which to insert: 20
Enter value to insert: 25
Linked List: 10 -> 20 -> 25 -> 30 -> NULL
Enter value to delete: 25

Node with value 25 deleted.
Linked List: 10 -> 20 -> 30 -> NULL

Program 7: Binary Tree Traversals

Inorder Traversal: 4 2 5 1 3
Preorder Traversal: 1 2 4 5 3
Postorder Traversal: 4 5 2 3 1

Program 8: Graph DFS Traversal

Adjacency list of vertex 0
2 -> 1 ->
Adjacency list of vertex 1
2 -> 0 ->
Adjacency list of vertex 2
3 -> 1 -> 0 ->
Adjacency list of vertex 3
2 ->
Visited 2
Visited 3
Visited 1
Visited 0

Program 9: Graph BFS Traversal

BFS Traversal starting from vertex 0: 0 2 1 5 4 3

Program 10: Linear Search

Enter number of elements in array: 5
Enter 5 elements: 10 20 30 40 50
Enter the element to search: 30
Element found at position 3 (index 2)

Program 11: Binary Search

Enter number of elements: 5
Enter 5 integers in ascending order: 23 45 55 67 78
Enter value to find: 78
78 found at location 5

Program 12: Quick Sort

Enter the number of elements: 5
Enter the elements: 12 2 9 3 4
Sorted array: 2 3 4 9 12

Program 13: Merge Sort

Enter number of elements (max 50): 5
Enter 5 elements: 54 76 86 23 11
Sorted array: 11 23 54 76 86

Program 14: Activity Selection

Enter number of activities: 6
Enter start times: 1 3 0 5 8 5
Enter finish times: 2 4 6 7 9 9
Selected activities (0-based index): 0 1 3 4

Program 15: Knapsack Problem

Enter number of items: 4
Enter weights of items: 2 3 4 5
Enter values of items: 3 4 5 6
Enter maximum capacity of knapsack: 5
Maximum value in Knapsack = 7
Items included:
Item 2 (Weight = 3, Value = 4)
Item 1 (Weight = 2, Value = 3)