

**Array, Records and Pointers:**

**1. Traversing:** carry out the following operations on an Array

- a) Maximum Value
- b) Minimum Value
- c) Average Value
- d) Total Value
- e) Sin Value

**2. Inserting: Insert**

- a) 54 to a sorted array.
- b) Rahim to a sorted array,
- c) 99 to position 5
- d) Karim to position 5.

**3. Deleting**

- a) Delete Karim from a sorted array,
- b) Delete an Item from position 2

**4. Sorting:**

- a) Sort integer data using Bubble sort.
- b) Sort string data using Bobble sort.

**5. Searching:**

- a) Search for 77 using Linear Binary Search:
- b) Search for Karim using Linear/Binary Search

**6. Merging:**

- a) Add two integer type arrays.
- b) Add two character type arrays

7. Copy elements of a 2D array into a 1D/linear array and print the elements of group 3 from the 1D array

8. Matrix: Addition Subtraction/Multiplication of two matrices.

9. Sparse Matrix: Store the element of a Triangular matrix A into a 1D array B and locate the elements A<sub>y</sub> in the array B.

**Linked List:**

- 1. Create a Linked List and store the value 5, 3, 9, 42, 0, 1
- 2. Traversing Perform same operations as done on Array
- 3. Inserting: Perform same operations as done on Array.
- 4. Deleting: Perform same operations as done on Array.
- 5. Sorting: Sort the contents of a list
- 6. Searching Perform same operations as done on Array.

**STACKS, QUEUES, RECURSION:**

- 1. Push an Item onto a Stack
- 2. Delete the top elements of Stack
- 3. Find the value of a Arithmetic expression P written in Postfix notation
- 4. Transform an Infix expression into Postfix expression
- 5. Find the value of an Arithmetic expression I written in Infix notation.
- 6. Sort 10 integer data using Quick-Sort algorithm.
- 7. Calculate the factorial of a given number using recursive technique.
- 8. Calculate the Fs of a Fibonacci sequence using recursive technique.
- 9. Solve Towers of Hanoi problem for N disks.

10. Insert an element into a queue
11. Delete an element from a queue.

**TREE:**

1. Write a program to insert an element in a Binary search tree, if the element already inserted before then display the location.
2. Write a program to insert an element into the heap.
3. Write a program to delete an element from the heap.
4. Traverse the tree in preorder.
5. Traverse the tree in inorder.
6. Traverse the tree in postorder.

**GRAPH AND THEIR APPLICATION:**

1. Write a program that takes an Adjacent matrix A with m vertices as input and output the following:  
a) adjacent of  $V_i, V_j$   
b) no. of paths of length 2 from  $V_i$  to  $V_j$   
c) no of paths of length 3 from  $V_i$  to  $V_i$  d) no, of paths of length 4 from  $V_i$  to  $V_j$
2. Take adjacency matrix with m nodes as input and calculate B, and from that calculate Path Matrix and tell whether the matrix is strongly connected or not.
3. Find out the Path Matrix of an adjacent matrix with m nodes using Warshall's Algorithm.
4. Find out the shortest path of a Weighted Graph G with m nodes  $V_i, V_2 \dots V_m$  and weight of each edge is  $w(e)$  using Warshall's Algorithm.
5. Write a program to create a Linked Representation of Graph, enter some data and read those data from the

**Graph**

6. Write a program to traverse a Graph represented in Linked List using Breadth-First Search.

7. Write a program to traverse a Graph represented in Linked List using Depth-First Search.

**SOME ADVANCED PROBLEM WHICH TO ADD  
LATER:**

(Md. Tohidul Islam) Associate Professor

Department of CSE University of Rajshahi