# Programs for DSA Lab, 2023, TIU

**Arrays**

1. WAP to search an element in an array using Linear search and tell whether it is smallest element or not.
2. WAP to read a 2D array marks which stores the marks of 5 students in three subjects. Also, write program to display the highest marks in each subject.
3. WAP to transpose a 3X3 matrix.
4. WAP to fill a square matrix with value 0(zero) on the diagonals, value 1 on the upper right triangle and value -1 on the lower left triangle.

# Linked List

1. WAP to create a Single Linked List and perform the following operations: (Options: ***1:*** create a list containing 08 values/nodes, ***2:*** display the list, ***3:*** add/insert a node at the beginning, ***4:*** add/insert a node at the end, ***5:*** add/insert node at a particular position/location, ***6:*** delete a node from the beginning, ***7:*** delete a node from the end, ***8:*** delete a node from a particular position/location, ***9:*** reverse the list, ***10:*** exit). You can take user input based on the given choices.
2. WAP to create a Circular Linked List and perform the following operations: (Options: ***1:*** create a list containing 08 values/nodes, ***2:*** display the list, ***3:*** add/insert a node at the beginning, ***4:*** add/insert a node at the end, ***5:*** add/insert node at a particular position/location, ***6:*** delete a node from the beginning, ***7:*** delete a node from the end, ***8:*** delete a node from a particular position/location, ***9:*** reverse the list, ***10:*** exit). You can take user input based on the given choices.
3. WAP to create a Double Linked List and perform the following operations: (Options: ***1:*** create a list containing 08 values/nodes, ***2:*** display the list, ***3:*** add/insert a node at the beginning, ***4:*** add/insert a node at the end, ***5:*** add/insert node at a particular position/location, ***6:*** delete a node from the beginning, ***7:*** delete a node from the end, ***8:*** delete a node from a particular position/location, ***9:*** reverse the list, ***10:*** exit). You can take user input based on the given choices.
4. WAP to store polynomial using linked list and perform addition.

# Stacks

1. WAP to create a stack using array and perform push, pop operations.
2. WAP to implement stack using linked list and perform push, pop operations.
3. WAP to convert infix expression to postfix expression.
4. WAP to evaluate a postfix expression.
5. WAP to convert infix expression to prefix expression.
6. WAP to evaluate a prefix expression.

# Queues

1. WAP to implement a linear queue using array and perform the following: ***1:*** insert an element, ***2:*** delete an element, ***3:*** peek, ***4:*** display the queue, ***5:*** exit. You can take user input based on the given choices.
2. WAP to implement a queue using linked list and perform the following: ***1:*** insert an element, ***2:*** delete an element, ***3:*** peek, ***4:*** display the queue, ***5:*** exit. You can take user input based on the given choices.
3. WAP to implement circular queue and perform the following: ***1:*** insert an element, ***2:*** delete an element, ***3:*** peek, ***4:*** display the queue, ***5:*** exit. You can take user input based on the given choices.

# Trees

1. WAP to create a BST (binary search tree) and perform all following operations:
   1. Insert an element
   2. Preorder traversal
   3. Inorder traversal
   4. Postorder traversal
   5. Find smallest element
   6. Find largest element
   7. Delete an element
   8. Count the total number of nodes
   9. Exit

# Graph

1. WAP to create a graph of n vertices using an adjacency list. Also write the code to read and print its information.
2. WAP to implement Breadth First Search.
3. WAP to implement Depth First Search.

# Searching and Sorting

1. WAP to search an element in an array using Binary search.
2. WAP to sort an array of elements using bubble sort.
3. WAP to sort an array using insertion sort.
4. WAP to sort an array using selection sort.
5. WAP to sort an array using merge sort.