

# Data Structures using C++

## Assignment I

**Note:** Source code has to be thoroughly documented

1. Given a matrix  $A$  of size  $m \times n$ , write a function that returns the sum of elements below the main diagonal, i.e. those elements  $a_{ij}$  for which  $i > j$
2. Create a linked list ADT with functions for insertion, deletion, traverse forward, traverse reverse, searching for an element, reversing a list and also test the above functionality.
3. Develop a program to add two polynomials using linked list implementation for polynomials
4. Given two sorted lists,  $L1$  and  $L2$ , write a procedure to merge  $L1$  and  $L2$  and generate a third list
5. Develop a self-adjusting list using linked list. A self-adjusting list is like a regular list, except that all insertions are performed at the front, and when an element is accessed by the Find, it is moved to the front of the list
6. Implement doubly linked list ADT
7. Create a Stack ADT using array implementation and solve the following
  - a. Balancing symbols
  - b. Evaluation of postfix expression
  - c. Converting Infix to postfix
8. Create a Stack ADT using linked list implementation
9. Implement Linear Queue ADT using linked list implementation
10. Implement Circular Queue ADT using array implementation
11. Implement Deque ADT using doubly linked list. Deque is a data structure in which insertions and deletions are performed only at the ends