

## **Data Structures and Algorithms Lab**

### **Assignment-2**

1. Write a menu-driven program that performs the following on a single linked list. (Write a function for each of the operations)
  - ✓ I. Create a linked list.
  - ✓ II. Print the content of the list.
  - ✓ III. Insert an element at the front of the list
  - ✓ IV. Insert an element at the end of the list
  - ✓ V. Insert a node after the kth node.
  - ✓ VI. Insert a node after the node (first from the start) containing a given value.
  - ✓ VII. Insert a node before the kth node.
  - ✓ VIII. Insert a node before the node (first from the start) containing a given value.
  - ✓ IX. Delete the first node.
  - ✓ X. Delete the last node.
  - ✓ XI. Delete a node after the kth node.
  - ✓ XII. Delete a node before the kth node.
  - ✓ XIII. Delete the kth node.
  - ✓ XIV. Delete the node(first from the start) containing a specified value.
  - ✓ XV. Find the reverse of a list(not just printing in reverse)
  - ✓ XVI. Sort the list
  - ✓ XVII. Search a given element
  - ✓ XVIII. Merge two lists; those are in ascending order.
  - ✓ XIX. Concatenate two list
  - ✓ XX. Find if two lists are equal(Boolean output)
- ✓ 2. Write a menu-driven program representing a polynomial as a data structure using a singly linked list and write functions to add, subtract and multiply two polynomials.
- ✓ 3. Implement Doubly Linked List for the operations (I to XV) of Sl. No. 1.
- ✓ 4. Given a problem, “people are standing in a circle waiting to be executed. Counting begins at a specified point in the circle and proceeds in a specified direction (e.g., clockwise). After a specified number of people is skipped, the next person is executed (i.e., removed). The procedure is repeated with the remaining people, starting with the next person, going in the same direction and skipping the same number of people until only one person remains and is freed”. The solution is to find the person to be freed. Solve it using
  - ✓ i) 2D array
  - ✓ ii) A circular linked list.
  - ✓ iii) A circular doubly linked list.