DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
NITK – Surathkal

CS203: Data Structures and Algorithms Lab
Assignment - 3

Instructions:

1. Implement the following exercise using C.
2. You are required to complete this exercise on or before 02/09/2024.
3. Submit all the programs in a single *.zip* file.

Exercise:

1. Create a singly linked list and implement the following operations:
    a. Insert node at the front.
    b. Insert node at the end.
    c. Insert node at a specified position
    d. Delete node at the front.
    e. Delete node at the end.
    f. Delete node at a specified position
    g. Traverse the list and print the elements of the list.

2. Create a doubly linked list and implement the following operations:
    a. Insert node at the front.
    b. Insert node at the end.
    c. Insert node at a specified position
    d. Delete node at the front.
    e. Delete node at the end.
    f. Delete node at a specified position.
    g. Traverse the list and print the elements of the list.

3. Create a circular linked list and implement the following operations:
    a. Insert node at the front.
    b. Insert node at the end.
    c. Insert node at a specified position.
    d. Delete node at the front.
    e. Delete node at the end.
    f. Delete node at a specified position.
    g. Traverse the linked list and print the elements of the list.

4. Create a circular doubly linked list and implement the following operations:
    a. Insert node at the front.
    b. Insert node at the end.
    c. Insert node at a specified position
    d. Delete node at the front.
    e. Delete node at the end.
    f. Delete node at the a specified position
    g. Traverse the linked list and print the elements of the list.

5. Write a program to add two polynomials using Linked list.

6. Write a program to count the number of nodes in a given linked list.

7. Write a program to merge two sorted linked lists.

8. Write a program to concatenate two linked lists.

9. Write a program to reverse a linked list using:
    a. Iterative approach
    b. Recursive approach

10. Write programs to compare two linked lists:
    a. Check if elements and arrangements are same in both the linked lists
    b. Check if only elements are same.