Experiment No-03: Advanced Linked List.

Objectives

- Insert and delete at a particular position from a single linked list (SLL).
- Reverse a single linked list.
- Create a doubly linked list (DLL).

Prerequisite: [Function], [Pointer], and [Structure].

Example 1: Delete element from a particular position of the SLL.

```
#include<iostream>
#include<bits/stdc++.h>
using namespace std;

// Create a Node Data Type
struct Node
{
    int data;
    Node *next;
    // Initialization
    Node(int x)
    {
        data = x;
        next = NULL;
    }
};
            // This program only includes the Function
// k is the position of the node in the linked list
Node* DeleteKthNode(Node *head, int k)
{
    Node *temp = head, *prev = NULL, *fr = NULL;
    int cnt = 0;
    while (temp!=NULL)
    {
        cnt++;
        if (cnt == k)
        {
            break;
        }
        prev = temp; // previous element of the kth node
        temp = temp->next; // kth node
    }
    fr = temp->next; // front element of the kth node
    prev->next = fr; // set the prev next pointer to kth node front node
    delete temp; // delete the node
    return head;
    }
```

Example 2: Reverse a SLL and return the new head.

```
#include<bits/stdc++.h>
using namespace std;

       // This program only includes the Function

Node* ReverseList(Node *head)
{
    Node *p = NULL,*c = NULL;

    while(head != NULL)
    {
        c = head->next;
        head->next = p;
        p = head;
        head = c;
    }

    head = p; // new head of the list
    return head;

}
```

Example 3: Create a doubly linked list from an array of values.

```
#include<bits/stdc++.h>
using namespace std;

//Create a Node Data Type for DLL
struct Node
{
    int data;
    Node *next;
    Node *bak;

    Node (int x)
    {
        data = x;
        next = NULL;
        bak = NULL;
    }
};

              // This program only included the Function
Node* CreateDLL(int arr[], int arrsize)
{
    Node *head = NULL, *temp = NULL, *prev = NULL;

    head = new Node(arr[0]); // set the head pointer
    prev = head;
```

```
    for (int i = 1; i<arrsize; i++)
    {
        temp = new Node(arr[i], nullptr, prev); // insert new node
        prev->next = temp;
        prev = temp;
    }
    return head;
}
```

Practice Exercise

0. Illustrate the process of reversing SLL by hand.

1. Write a C++ program to find the position of an element from a Singly Linked List [Linear Search].

2. Write a C++ program to insert an element at k^{th} position in a singly linked list. [Consider possible edge cases]

3. Write a C++ program to insert a node at the beginning of a DLL. [Consider possible edge cases]

4. Write a C++ program to insert a node at the end of a DLL. [Consider possible edge cases]

5. Write a C++ program to delete the first node of a DLL. [Consider possible edge cases]

6. Write a C++ program to delete the last node of a DLL. [Consider possible edge cases]

Resources (Link)

Try to solve similar problems at an online Judge.

1. Search in a SLL

2. Reverse a SLL

3. Construct a DLL

4. Insert a node in DLL

5. Delete a node in DLL