

# LINKED LISTS

## Problem 1. Operations on singly linked list.

Each node in a list consists of at least two parts:

1. Data
2. Pointer to the next node

Contents of NumberLinkedList.h:

```
// Specification file for the NumberLinkedList class
#ifndef NUMBERLINKEDLIST_H
#define NUMBERLINKEDLIST_H

class NumberLinkedList {
private:
    // Declare a structure for the list
    struct ListNode
    {
        int value;                // The value in this node
        struct ListNode *next;    // To point to the next node
    };

    ListNode *head;              // List head pointer
public:
    // Constructor
    NumberLinkedList()
        { head = nullptr; }

    // Destructor
    ~NumberLinkedList();

    // Linked list operations
    void insertFirst(int);
    void insertLast(int);
    void deleteFirst();
    void deleteLast();
    int length();
    void delete(int);
    void displayList() const;
};
```

Contents of NumberLinkedList.cpp:

```
// Implementation file for the NumberLinkedList class
#include <iostream>
#include "NumberLinkedList.h"
Using namespace std;

NumberLinkedList::NumberLinkedList()
{
```

```

        // Write your code here
    }

NumberLinkedList::~~NumberLinkedList()
{
    // Write your code here
}

void NumberLinkedList::insertFirst(int item)
{
    // Write your code here
}

void NumberLinkedList::insertLast(int item)
{
    // Write your code here
}

void NumberLinkedList::deleteFirst()
{
    // Write your code here
}

void NumberLinkedList::deleteLast()
{
    // Write your code here
}

int NumberLinkedList::length()
{
    // Write your code here
}

void NumberLinkedList::delete(int i)
{
    // Write your code here
}

void NumberLinkedList::displayList() const
{
    // Write your code here
}

```

Write the following functions manipulating on linked lists which are defined in `NumberLinkedList.cpp`:

- a) `insertFirst(int item)`: inserting an item into the linked list at the beginning of the list.
- b) `insertLast(int item)`: inserting an item into the linked list at the end of the list.
- c) `deleteFirst()`: deleting the item at the beginning of the list.
- d) `deleteLast()`: deleting the item at the end of the list.
- e) `length()`: computing the number of elements of the list.
- f) `delete(int i)`: deleting an i-th node from a linked list.
- g) `displayList()`: displaying all the values stored in the linked list.

## Problem 2.

Write the following functions:

- a): merging two ordered singly linked lists of integers into one ordered list.
- b): concatenating two lists p1 and p2.
- c): checking whether two singly linked lists have the same contents.

**Suggestion:** You can choose 1 between 2 ways

- 1) Write 3 functions:

```
NumberLinkedList merge(NumberLinkedList p1, NumberLinkedList p2)
NumberLinkedList concatenate2Lists(NumberLinkedList p1,
                                   NumberLinkedList p2)
bool equal (NumberLinkedList p1, NumberLinkedList p2)
```

- 2) Write operations for NumberLinkedList class

```
void merge(NumberLinkedList);
void concatenate(NumberLinkedList);
bool equal(NumberLinkedList);
```

## Problem 3.

Suppose a singly linked list is sorted in ascending order. Write the function to insert a new node so that the list is always sorted.

**Suggestion:** Write 2 operations void sort() and void insertSorted(int) for NumberLinkedList class.

## Problem 4. Set representation by using linked list.

Set is a list of elements, in which the order of elements does not matter. We want to define the data structure for set of integers and also the operations on it. Suppose that each set element appears exactly once, it means that all the elements of set are different.

You have to use linked list to define set of integers and write the following functions:

- a) Checks if a linked list is a set
- b) Checks if a set is a subset of another set.
- c) Union of two sets.
- d) Intersection of two sets.

**Suggestion:** Extending the NumberLinkedList class