

## Assignment 1: Playing with LinkedList

You have been provided two different linked list implementations: **LinkedList.c**, **LinkedListDoubly.c**. The files implement following versions of linked list:

- **LinkedList.c** is a singly linked list implementation with a head pointer.
- **LinkedListDoubly.c** is a doubly linked list with head pointer.

For this assignment, you are required to add different functionalities to the provided implementation.

### Task 1 – 5 are to be done in LinkedList.c file

The file implements a singly linked list using a head pointer that points to the first node of the list. Your job is to add following features.

**Task 1:** Add **getListSize** function. This function will return the current size of the list.

**Task 2:** Add **insertLast** function. Add a new function **insertLast(int item)** to the list. This function will insert a new item at the end of the list.

**Task 3:** Add **insertBefore** function. Add a new function **insertBefore(int oldItem, int newItem)** to the list. This function will insert a new item before an existing item in the list. The function will first search for oldItem in the list. Then it will insert the newItem immediately before the oldItem in the list. If the oldItem is not present in the list, then no insertion should be done.

**Task 4:** Add **deleteAfter** function. Add a new function **deleteAfter( int oldItem)** to the list. This function will delete the item immediately after the given parameter item. The function will first search for input item in the list. Then it will delete the item immediately after the item in the list. If the item is not present in the list, then no deletion should be done.

**Task 5:** Add **deleteLast** function. This function will delete the last element of the list. You must ensure that memory of the deleted item is correctly released.

### **Task 6 – 8 is to be done in LinkedListDoubly.c file**

The file implements a doubly linked list. In this type of lists, each node has two pointers: next and prev. The next pointer points to the next node in the list while the prev pointer points to the previous node in the list. A function **insertFirst** is already implemented in the file. This function inserts an item at the beginning of the linked list. A print function **printListForward** is also implemented. Your job is to add following features.

**Task 6:** Make **printListBackward** function: Write a function **printListBackward** that will print the whole list in reverse order starting from the last node of the list.

**Task 7:** Make **insertBefore** function efficient: You already implemented **insertBefore** function as part of Task 3. Now, you will implement an improved version of the same function for the doubly linked list

**Task 8:** Modify **deleteAfter** function: Add **deleteAfter** function to the doubly linked list. You can do this easily by modifying the implementation you did in Task 4.

### **Task 9-10 can be done with any implementation of Linked list**

Here are two problems on the application of Linked list data structure. You can use any of the above two implementations to solve them.

**Task 9:** Write a function to remove duplicate entries in a Linked list. Also print the number of duplicates you found.

**Task 10:** Find the maximum and the minimum values in a Linked list. Replace the max value node with the first node and min value node with the last node in that Linked list.

### **Points:**

| Task | Points |
|------|--------|
| 1-5  | 25     |
| 6-8  | 15     |
| 9-10 | 10     |

### **Important Instructions:**

1. You **must** use the code base provided with the assignment. Do necessary modification/extension on them.
2. **Only** C programming Language is allowed for this assignment.
3. Try to perform as much as you can. There is a plenty of scope for partial marking.
4. There will be a **viva evaluation** for this assignment. **Remember, your final marking of this assignment highly depends on the viva.**
5. Submit only the given two files which contain your necessary modifications. Any extra files apart from them will be discarded.
6. Submission deadline: August 7, 11:59 am  
Evaluation date & time: August 7, 2:30 pm