Data Structures

Lab Exercise # 2: Singly Linked Lists and File I/O

In the lab exercise, you will complete a C++ program that enables users to perform various operations on a singly linked list. The starter code on Brightspace includes the user interface (UI) and a singly linked list class named **MyLinkedList**. The MyLinkedList class comprises several member methods that are currently incomplete. Your task for this lab is to finish these methods, which are also listed below:

1. **bool empty() const** **(0.05 Point)**
   This method should return true if the Linked List is empty, and false otherwise.
2. **void addFront(int elem)** **(0.1 Point)**
   This method should add a new node to the linked list. This method takes as an int argument which should be saved as the element of the new node.

```
>addFront 20
Head->20->10->NULL
>addFront 30
Head->30->20->10->NULL
```

3. **void removeFront()** **(0.1 Point)**
   This method should remove the first element of the linked list. This method should print an error message if the list is already empty.

```
Head->30->20->10->NULL
>removeFront
Head->20->10->NULL
```

4. **void remove(int elem)** **(0.1 Point)**
   This method should remove the first occurrence of an element of the linked list. This method should print an error message if no such element is found.

```
Head->30->10->20->10->NULL
>remove 10
First occurrence of 10 has been successfully deleted from the the list.
Head->30->20->10->NULL
```

5. **int countElem(int elem) const** **(0.1 Point)**
   This method should count the frequency (occurrence) of the nodes having a specific value (elem) in the linked list. The method should return 0 if no such node is found.

```
>display
Head->2->3->2->1->2->4->5->NULL
>count 2
2 occurs 3 time(s) in the list
```

6. **int getIndexOf(int elem) const** **(0.1 Point)**

This method should return the first index of a node in the linked list with a specific element. This method should return -1 if no such node is found in the List.

```
Head->2->3->2->1->2->4->5->NULL
>indexOf 4
First index of 4 in the list is: 5
```

7. **void display() const** **(0.1 Point)**

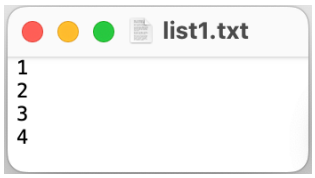
This method should print the entire linked list in the same manner as depicted in the following screenshot.

```
>display
Head->2->3->2->1->2->4->5->NULL
```

8. **void readData(string file)** **(0.1 Point)**

This method should read a text file that contains several integer numbers. Each line of the text file contains a number representing the value/element of a node of the linked list. The ***readData*** method should create new nodes for each number in the file and add it to the linked list. The existing elements in the linked list should not be deleted. You can assume that the input-file is in the correct format.

```
>read list1.txt
Head->4->3->2->1->NULL
```



9. **void writeData(string file) const** **(0.1 Point)**

This method should write the contents of the linked list onto a given file. Each line in the generated file should contain the value/element of a node in the linked list. The file should be overwritten/truncated if it is already present. You will observe that if you write a linked list to a file and then read it again, the order of the elements in the linked list will be reversed.

```
>write list2.txt
5 records have been exported to list2.txt
```

10. **~MyLinkedList ()** **(0.05 Point)**

This method should delete all nodes of the linked list by making free the memory reserved for the linked list.

Comments: **(0.1 Point)**

Comments are an essential part of any program. You should always write comments in your code.

Note:

Please upload a single cpp file named "***lab2.cpp***" to Brightspace.

Code of Conduct

All assignments are graded, meaning we expect you to adhere to the academic integrity standards of NYU Abu Dhabi. To avoid any confusion regarding this, we will briefly state what is and isn't allowed when working on an assignment/lab-task.

Any documents and program code that you submit must be fully written by yourself. You can, of course, discuss your ideas with fellow students, as long as these discussions are restricted to general solution techniques. Put differently, these discussions should not be about concrete code you are writing, nor about specific results you wish to submit. When discussing an assignment with others, this should never lead to you possessing the complete or partial solution of others, regardless of whether the solution is in paper or digital form, and independent of who made the solution, meaning you are also not allowed to possess solutions by someone from a different year or course, by someone from another university, or code from the Internet, etc. This also implies that there is never a valid reason to share your code with fellow students, and that there is no valid reason to publish your code online in any form.

Every student is responsible for the work they submit. If there is any doubt during the grading about whether a student created the assignment themselves (e.g. if the solution matches that of others), we reserve the option to let the student explain why this is the case. In case doubts remain, or we decide to directly escalate the issue, the suspected violations will be reported to the academic administration according to the policies of NYU Abu Dhabi.

(https://students.nyuad.nyu.edu/campus-life/community-standards/policies/academic-integrity/)