COSC2430 lab2: Linked List Template

1. Introduction

You will implement a C++ program to add and remove objects in a linked list. The purpose of this lab is to get students familiar with Linked List operations and type template. Please name the folder on the server as "lab2". Refer to the cpp file on blackboard lab section for linked list template.

2. Input files

- The first line of the input will contain a string, this string will either be "string", "int", or "char" this indicate the data type of the data that will be added to the linked list, each input file only contains a single data type.
- The rest of the input file will contain a list of linked list operation and the data to be operate on.
- There will be two types of operation: `Add` and `Remove`.
    - `Add (index) [data]`
        - The `Add` command will be followed by an integer inside parenthesis (represent the index to be added) and the data will be in between brackets.
        - If the index = 0, meaning the data should be added at the beginning of the linked list.
        - If the index = size of the linked list, meaning the data should be added at the end of the linked list.
        - If the index > size of the linked list, meaning index out of bound, in this case the data should not be added to the linked list.
        - Ex: `Add (0) [abc] //since the data is of type string and the given index is 0, it should be added to the beginning of a string linked list.`
        - Ex: `Add (3) [10] //since the data is of type integer, it should be added to a linked list of type int if the given index is not greater than the size.`
    - `Remove [data]`
        - The `Remove` command will be followed by some data inside brackets, this data will be removed from the linked list if exist.
        - Ex: `Remove [abc] //remove every string that match exactly with "abc" from the list.`
        - Ex: `Remove [10] //remove every integer that equal to 10.`

- Input might contain empty lines, in this case empty line should be ignored.
- Input will not be empty.

3. Output files

- The output file should display every element of your linked list.
- Each element will be separated by a space.

4. Examples

- input1.txt

```
string
Add (0) [hello]
Add (1) [world]
Add (0) [print]
Remove [tomorrow]
```

- ans1.txt

```
print hello world
```

- input2.txt

```
int
Remove [11]
Add (0) [20]
Add (2) [47]
Add (0) [62]
Add (1) [5]
```

- ans2.txt

```
62 5 20
```

- input3.txt

```
int
Add (1) [9]
Add (0) [3]
Add (1) [7]
Add (2) [11]
Add (4) [5]
Remove [7]
Remove [3]
Add (1) [32]
```

- ans3.txt

```
11 32
```

5. Turn in your lab assignment

lab2 needs to be turned in to our Linux server, follow the link here
https://rizk.netlify.app/courses/cosc2430/2_resources/

Make sure to create a folder under your root directory, name it “lab2” (case sensitive), copy all your .cpp and .h file to this folder, “argumentmanager.h” need to be included as well.

PS: This document may have typos, if you think something illogical, please email TAs for confirmation.