Lab 4: Recursion

1. Introduction

We saw several examples of recursion in the lecture today. In this lab, you will gain experience using recursion in your program. More specifically, you will re-implement some of the functionalities of LinkedList class using recursion, instead of using loops.

The code skeleton is under: **/comp/15/files/l4**

1. Log in to the CS homework server.
2. Move to your **comp15** directory that you created in Lab 1.
3. Create a directory named **lab4** under the comp15 directory.
4. Move to the lab directory.
5. Copy the code skeleton to the current working directory.
6. Leverage the features of Git to manage your progress toward writing the program.

The code skeleton provides you with five files: **test.cpp**, **Node.hpp**, **Node.cpp**, **LinkedList.hpp** and **LinkedList.cpp**. First, take a look at the files to find the internal structure of the **LinkedList** and **Node** classes. Node class has already been implemented for you. Note that the LinkedList class supports only selected operations, and the copy constructor and assignment operator are commented out so that this lab can be completed within our lab time, but you are encouraged to work on them later on. **test.cpp** provides you with one test case that your LinkedList implementation is supposed to pass.

2. Requirements

1. Implement the **LinkedList** class.
    a. The constructor and destructor: You can reuse the ones you implemented in lab2.
    b. The addToBack(), removeBack(), getSize(), and toStringInReverseOrder() methods: Implements these methods using recursion. You can add any necessary methods in the private section.
    c. Format of toString(): Find and follow the format of expected outputs in **test.cpp**.
2. Pass the test case written in **test.cpp** with no memory leaks and no memory errors.

3. README

Create the README file that includes the following categories with appropriate section headers.

1. **Name**: Your name.

2. **Date**: The last updated date.
3. **Summary**: A brief summary of the lab.
4. **Files**: A list of files that are necessary to build and test the program.
5. **Instructions**: A sequence of commands to compile and test the program. Note that you are expected to report procedures without using the make command.
6. **References**: A list of citations to information used to complete the lab.

4. Submission

Submit your files listed below using Gradescope.
Files: **Node.hpp Node.cpp LinkedList.hpp LinkedList.cpp test.cpp README**