

COSC 2436 lab2: Linked List and Template

Create a C++ program to insert and remove elements in a linked list. The purpose of this lab is to get students familiar with Linked List operations and type template.

1. Input files

- The first input line will contain a string indicating the data type of the linked list
 - Options will either be 'string', 'int' or 'char'
 - Each input file will only contain a single data type
- The second input line will indicate how to insert new data into the linked list.
 - Options will be either 'head' or 'tail'.
 - If it says 'head', add elements only to the beginning of the linked list for this input file. If it says 'tail', add elements only to the end of the linked list for this input file.
 - Each input file will only contain a single insert option
- Each element will be on its own line.
- There are three additional operations: 'remove head', 'remove tail', and 'remove [index]'
 - If the line says 'remove head', remove the first element in the linked list
 - If the line says 'remove tail', remove the last element in the linked list.
 - If the line says 'remove [index]', remove the element at [index] in the linked list
 - If the index is 0, remove the head of the linkedlist
 - If the index is equal to or greater than the amount of elements in the linked list, remove the tail of the linkedlist
 - Otherwise, remove the element at the specified index
 - Example: linked list currently is hello, computer, science, world
 - The operation 'remove [1]' would remove computer since its index is 1
- When reading the input, \n and \r should be removed before processing the string.

2. Assumptions

- The keywords 'head', 'tail', and 'remove' will never appear as data.
- No empty lines, but there can have empty files; empty input files should be empty output files.

3. Output files

- The output file should display every element in the linked list on a single line, separated by a space.
- Empty input files should result in an empty output file.

4. Reminder

- Turn in your lab assignment to our Linux server, follow the link [here](#) for more instructions.
- Make sure to only have **1** .cpp file with the main() function in your working directory, otherwise your program will fail the grading script, resulting in a 0.
 - Create a folder under your root directory, name the folder *lab2* (case sensitive), copy all .cpp and .h files to the folder (ArgumentManager.h is also needed)
 - Only include the necessary files in your working directory in your final submission
 - To test your program, copy the input files into the server and run your program. After verifying that they pass, delete the .txt files.

Please reach out to the TAs via email or teams for any clarifications or typos.