COSC2430 Hw2: Linked List

1. **Introduction**

   You will create a C++ program to add, remove and sort objects in a linked list. The purpose of this homework is to get students familiar with Linked List and simple sorting techniques. Please name the folder on the server as “hw2”.

2. **Input, Command, Output and example files**
   a. Input file
      - The first line of the input file will tell you how to add data to your linked list
        a. Alphabetically
           - A-Z-a-z (capital letters count first this is how c++ handles comparing strings anyway)
        b. Length
           - Shortest length - longest length of string
        c. Beginning
           - Add current string to the front of your list
        d. End
           - Add current string to the end of your list
      - The rest of the file will contain a list of sentences. Don’t add empty lines or duplicate sentences (case and space sensitive)

   b. Command file
      - Command file will be filled with 3 types commands that will be run in order they appear
        a. Add (int location) [string sentence]
           - The add command will be followed by an integer inside parenthesis and then followed by a sentence inside brackets
           - If the location is bigger than the size of the list then don’t add it
           - If the sentence is a duplicate don’t add it
           - EX: Add (0) [this would add to the beginning of your list]
           - EX: Add (3) [example sentence]
             - This will be the added to the 3rd index meaning it will be the 4th thing in your list
             - Won’t get added if your list is size 2 or less
        b. Remove [string sentence segment]
           - The remove function will be followed by a string inside of brackets
           - Every sentence that contains that string in any part of the sentence will get removed
           - EX: Remove [a]

        - This will remove every single sentence with the letter a in it

    c. Sort (alphabetically/length)
        - The sort command will be followed by the word alphabetically or length inside of parenthesis
        - This will sort the entire list either alphabetically or by length
        - EX: Sort (alphabetically)

c. Output file
    - The output file will contain every element of your list after it’s been modified by the command file.
    - Each element will be on its own line

d. Examples
    i. Ex1:

    input21.txt:

```
Alphabetically
for this example the command file will be empty
that means you just need to sort this file alphabetically


don't add any empty lines
test lines
duplicate line there should only be one of this in the output
duplicate line there should only be one of this in the output

extra sentence
```

    command21.txt: (Empty)

    ans21.txt:

```
don't add any empty lines
duplicate line there should only be one of this in the output
extra sentence
for this example the command file will be empty
test lines
that means you just need to sort this file alphabetically
```

    Linux Command:

```
./files “input=input21.txt;command=command21.txt;output=output21.txt”
```

    ii. Ex2:

    input22.txt

```
Length

this file will be testing each of the different commands with your sentences
test sentence number one
```

```
another sentence
longer sentence than another sentence
most of the sentences have the word sentence in them
last sentence of the input
```

command22.txt

```
Add (15) [this sentence shouldn't be added because it's out of bounds]
Sort (alphabetically)
Add (6) [this sentence should be added to the end of the list]
Remove [test]
Add (0) [another sentence]
```

ans22.txt

```
another sentence
last sentence of the input
longer sentence than another sentence
most of the sentences have the word sentence in them
this sentence should be added to the end of the list
```

Linux Command:

```
./files “input=input22.txt;command=command22.txt;output=output22.txt”
```

iii. Ex3:

input23.txt

```
Beginning
here i go typing out a bunch of
random sentences so that this assignment can have
enough inputs
I hope this assignment isn't too challenging
but if you're having trouble with it don't give up
just take a deep breath and you'll get it
however this giant message
is just
going
to
be
a
random
jumble of
words
in
the
output
file
```

command23.txt

```
Remove [us]
Sort (length)
Sort (alphabetically)
Add (0) [this should go to the beginning until it gets sorted]
Add (70) [this should not be added]
```

```
Sort (length)
Remove [a]
Remove [e]
Remove [i]
Remove [o]
Remove [u]
Remove [y]
Add (0) [first]
Add (1) [second]
Add (2) [third]
Add (4) [Not added]
```

ans23.txt

```
first
second
third
```

Linux Command:

```
./files "input=input23.txt;command=command23.txt;output=output23.txt"
```

3. Requirements

Homework is individual. Your homework will be automatically screened for code plagiarism against code from the other students and code from external sources. Code that is copied from another student (for instance, renaming variables, changing for and while loops, changing indentation, etc, will be treated as copy) will be detected and result in ”0” in this homework. The limit is 50% similarity. Here are some previous homework which have been found to copy each other (the main function has been deleted).

4. Turn in your homework

Homework 2 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 hw2 (name need to be lower case), only copy your code to this folder, no testcase or other files needed.

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