Programming Assignment #5

Linked List of string

CS 2308.255 + CS5301 Spring 2020
Instructor: Jill Seaman

Due: Wednesday, 4/15/2020: upload electronic copy by 11:59pm!

Problem: Implement an interface that manipulates a list of strings. You will be provided with the following files on the class website:

- **StringList.h** containing a class declaration, set up for a linked list representation.
- **Driver.cpp** containing a main function you can use to test your implementation.

You will be responsible for providing the StringList.cpp file, including the implementation of the StringList member functions (described below):

StringList and **~StringList**: creates an empty list, and deallocates all the nodes in the list, respectively.

count: returns the total number of strings (nodes) in the list. Any duplicate strings should be counted.

pushFront(string) Adds a new node containing the string to the front of the list.

popFront() removes the first node, if the list is not empty (else does nothing).

display(): displays the strings in the list to the screen, one string per line.

concatenate(char x): returns a string that is constructed by appending all of the strings from the list together, with character **x** in between each original string.

moveToFront(String s) removes the node that contains string s from the linked list and inserts it at the front of the list. Does nothing if s is not in the list, or if the list is empty.

maximumStartingAt(int i): returns the string that would come last in alphabetical (ascii) ordering, starting with the node in position i (it ignores strings in position 0 through i-1). Does not change the list! Returns the empty string if the list is empty, or if i is greater than the length of the list.

selectionSort(): Here is the algorithm you **must** use for implementing the selectionSort function:

For each position in the list, starting with the first string (in position 0).

a. Find the maximum string in the **remaining** portion of the list.
b. Move the node containing the string found in part a to the beginning of the list.

Hint: Call functions you have already defined to do these two steps

Input/Output:

Use the provided Driver.cpp file to test your code. I recommend trying to implement one or two functions at a time, and testing them, rather than implementing all the functions and then trying to debug them all at once.

NOTES:

- This program must be done in a **Linux or Unix** environment, using a command line compiler like g++. Do not use codeblocks, eclipse, or Xcode to compile.
- Put your code in a file named **StringList.cpp.**
- Your StringList.cpp file **must compile** with the (unchanged) provided files, otherwise you may receive a score of 0.
- You may re-use code from the **NumberList** class (source: book/slides/website).

Logistics:

For this assignment you need to submit only the **StringList.cpp** file. You do not need a zip file, you do not need a makefile, you do not need to provide your driver. You do not need to rename the StringList.cpp file.

Submit an **electronic copy** using the Assignments tool on the Canvas website for this class (canvas.txstate.edu).

See the assignment policy on the course website (cs.txstate.edu/~js236/cs2308) for more details, including late deadlines and penalties.