Programming Assignment #2
CS 163 Data Structures

Submit your assignment to the D2L Dropbox (sign on via d2l.pdx.edu)

Goal: The purpose of the second programming assignment is to experience stacks, queues, and new data structures. The data structures for this assignment are a linear linked list of arrays and a circular linked list.

Problem Statement:
We were talking about this in class. We get so many messages. Maybe they are text messages. Or, they could be email messages. Of course, you might use facebook messenger. Regardless, in most situations the default mode is to have the most recently sent message appear on your screen as if it was what you needed to look at first. This represents how a stack behaves. The last item pushed is the first item examined.

For program #2, we will use a both a Stack ADT and a Queue ADT to assist with managing messages. First, messages arrive. As a message arrives from a client program, they are pushed onto the stack. As the user decides they want to process the messages, they are popped from the stack. Junk messages (spam, advertisements, etc.) may be just discarded. But, if the message is one that the user wants to respond to, then will enqueue the message onto a queue to keep track of the messages handled for this session.

Programming: There are two ADTs in this program (Queue and Stack ADT).

1. The information kept includes:
   a. The sender (email address for an email, phone number for a text)
   b. The subject
   c. The date
   d. The message itself

- Please keep in mind that because we are creating ADTs, the **user** of the program must not be aware that stacks and queues are being used. However, the client program knows that they are using these ADTs.
- You should support complete implementations of the Stack and Queue ADT. Make sure to thoroughly test each of the stack and queue functions from the client program!

Programming – Data Structures:

- The stack should be implemented using a linear linked list of arrays (lab 3), where each element in the array is a message. The array must be dynamically allocated and each array must be of the same size. I recommend that the arrays be relatively small (e.g., 5) so that you can properly test the code. Stack ADT functions should include **push, pop, peek, and display.**
- The queue should be implemented using a circular linked list (lab 3 and 4), where the rear pointer points to the last segment we will take, and rear->next points to the next segment (or first) we are going to take. You must implement **enqueue, dequeue, peek, and display.**
- Remember the idea of your client program is to test out your ADT functionality so that a real application can be developed off site by another team of Software Engineers based on the ADT public functions. You want to make sure that your stack and queue ADTs perform well!

<u>Things you should know...as part of your program:</u>

1) **Do not use statically allocated arrays in your classes or structures used by the ADT.**
2) All data members in a class must be private
3) Never perform input operations from your class in CS163
4) Global variables are not allowed in CS163
5) **Do not use the String class! (use arrays of characters instead!) You may use the cstring library.**
6) Use modular design, separating the .h files from the .cpp files. Remember, .h files should contain the class header and any necessary prototypes. The .cpp files should contain function definitions. **Never "#include" .cpp files!**