CS 260
Programming Lab 2

This exercise builds upon the simple queue created in Part A to create a double ended queue.

This lab is worth a total of 100 points, 10 points are the self-evaluation, 50 points for the basic lab as specified below, 30 points for the advanced version of the lab, and 10 points for a solution to the thinking problem.

Base Lab Specification

For the base lab start with your Lab Part A and add the following functionality.

1) Write a method resize(). Revise your addTail method to create a new array twice as large and copy the elements to it rather than throwing an exception. This should properly deal with situations where there has been wrapping and the tail is at an index less than the head. Note, this is not a simple exercise where you can copy index 0 old array to index 0 new array and so forth. You will need to end up after copying with the head at index 0 and tail appropriately located. There are several ways to accomplish this. You should write a private method for this that is called, since you will end up needing the exact same code in two locations.

2) Write a listQueue method that provides a listing of the elements from head to tail. It should return a string to the calling program. This should properly deal with situations where there has been wrapping and the tail is at an index less than the head.

Advanced Lab Specification

For the advanced lab, start with your base lab class and add the following functionality.

1) Add a method addHead(int value) that adds a new value at the head. This should not overwrite data that was previously in the queue. If the array is full, call your resize method to double the array and copy the data as needed.

2) Add a method removeTail() – save the value at the tail of the queue, update the queue to remove the item at the tail, and return the saved value. Wrapping if necessary. If the queue is empty, throw an exception. (C++ use ***out_of_range*** , C# use ***IndexOutOfRange***, Python use ***IndexError***).

Thinking problem

How could you implement a stack using your double ended queue? Use this concept to list a provided set of numbers in reverse order.

Development and Testing

There is a driver provided in Moodle for this lab. It has multiple sections; your class needs to pass all of them to get full credit.

Complete each section of testing before you begin working on implementing the next features in your class.

There is a place in the driver for you to add code for the thinking part of the lab.

For this lab, there are explanations of how such a double ended queue would work, but there are no details, you need to work it out on your own. Suggestion, draw pictures of the queue and work through the problem with pseudocode before you start actually coding it.