CSCI 2270 – Data Structures and Algorithms
Instructor: Hoenigman
Assignment 2
Part I Due Wednesday, July 16, by 10pm
Everything else due Saturday, July19, by 5pm

Stacks and queues

In this assignment, you will be implementing a stack and a queue using an array, a single linked list, and a double linked list. These data structures will be populated through user input to add or remove items one at a time.

Read the entire assignment before beginning any coding.

In main, present the user with a menu that includes the following:
1. Create stack
2. Create Queue
3. Exit program

If 1 or 2 is selected, ask the user if they want an array, single-linked list, or double-linked list implementation.

a) arrays
b) single linked lists
c) double linked lists

Each combination of data structure and implementation should be handled in a separate function. For example, you could have a function called *stackArray* if the user wants to create a stack using an array.

If the user selects “Create stack” in the menu, present the following 4 choices repeatedly:

(1) PUSH (Enter integer for insertion into stack)
(2) POP (Display integer and delete it from stack)
(3) PRINT STACK (Display stack contents without deleting anything, last element first)
(4) Exit program

If the user selects “Create queue”, then the user is given the following 4 choices repeatedly:
(1) ENQUEUE (Enter integer for insertion into queue)
(2) DEQUEUE (Display and delete integer from queue)
(3) PRINT QUEUE (Display queue contents without deleting anything, first element first)
(4) Exit program

If option 3 was selected, exit the program, not before performing garbage collection.

All functionality for that choice can be handled in that function, including the looping to get user input.

Use a class for your single and double linked list implementations. For the single linked list, you can implement the class as a double linked list and just set the pointers in the unused direction to NULL.

For the array implementations, choose a reasonable starting size for your array, such as 40. If you would like to implement an array doubling algorithm you are welcome to do so, but it is not a requirement. The array doubling would happen when the stack or queue is full and more memory needs to be allocated. Generate a new array that is double the size of the current array, and copy everything in the stack or queue to the new array.

Part 1 – Due Wednesday by 10pm
Write at least one of the functions for the data structure implementation for this assignment. The stackArray function is probably the easiest, and is a good place to start if you're unclear on how to start. For the functions that use linked lists, your class for the linked list will need a push and pop method. Once you have this function working, and you can get user input and push and pop items from the stack, move on to the other implementations.

Part II – Due Thursday by 10pm
Complete the array implementation of the queue. You will need to have the dequeue and enqueue functions. A function, such as queueArray should call these functions to add and remove from the queue.

Final Assignment 2 Submission
To submit your work, zip all three files together and submit the zip file. All files should include a comment block at the top of the file with your name, instructor's name, and lab number.