**ESE 224: Project 6:** **Linked lists, stacks, queues**

**ECE, SBU, M. Subbarao, Fall 2018, Max marks: 16 points**

Weight: 16%   Due 11/28/2018

1. (7 points) Modify the Doubly linked list Class in the text book to complete this project, but it should be a **doubly linked Circular list** in this project. Define a **template class** for a doubly linked circular list to store a list of unordered items such as integers, floats, chars, etc. Multiple occurrences of an item are permitted; all items are stored in order determined by the commands that leads to its creation. Include methods for the following  operations. If the specifications of any of the following methods are not clear (e.g. what are the parameters and what should be returned), then use the implementation of the corresponding operations in the text book using singly linked lists in Chapters 16 and 17 in the text book.

1. Initialize the list (to an empty state).
2. Determine if the list is empty.
3. Destroy the list.
4. Print the list.
5. Find the length of the list.
6. Search the list for a given item (return the position of its first occurrence)
7. Insert an item with a given item/integer value in the list at a specified position. Position value goes from 0 to Length where Length is the number of elements in the list.
8. Delete an item at a specified position from the list.
9. Delete the first occurrence of an item with a specified value from the list if it exists.
10. Copy the list.
11. Write the overloaded assignment **operator=** .

2. (4 points) Write a menu driven program to test the class defined above by creating two unordered list of integers L1 and L2. First L1 and L2 are created as empty lists without any command. Then L1 is built to contain many integers from input, and then command 10 or 11 is used to copy L1 to L2, or assign L1 to L2 with L2=L1, and the resulting L2 is printed as part of command 10 or 11. The main() function should ask for entering a command number 1 to 11 as above (e.g. 8 for deleting an item) and associated input data, and carryout the operation on L1 object of the class.

3. (3 points) **Stack:** add methods (with command numbers 12, 13, 14,below, and 4 for Print stack):

12. push(),

13. pop(), and

14. top()

to the class above, and extend the mani() function above to test these three methods. You have to add appropriate parameters to these methods. See the singly-linked list implementation of stack data structure in Ch. 17 for details.

4. (3 points) **Queue:** add methods (and their command numbers 15, 16, 17):

15. addQ(),

16. deleteQ() and

17. front()

to the class above, and extend the mani() function above to test these three methods. Use command number 4 for printing the queue. You have to add appropriate parameters to these methods. See the singly-linked list implementation of the Queue data structure in Ch. 17 for details.

In all parts above, you may add additional commands and command numbers so that the main() function can call the operations above and print the contents of the lists.

Run the program and test it with your own input data. If you make up a good input data set, post it on piazza.com for others to use it. Print the lists periodically after two or three commands to show its content. Submit the source code, input test data, and output for the test data. You may use **cin/cout** and input/output redirection on command line arguments to test your program.

When all parts are completed, there will be only one main() function, and only one class definition of circularly linked doubly linked list. Use a suitable node structure.