CS 260 Lab #2 Double Linked Lists

General

For this lab you will gain an introduction to Abstract Data Types and coding a pointer-based doubly-linked list and queue.

General requirements for all labs:
At the beginning of any project files that you create or modify, include a header comment including your name, creation date, Lab #, and purpose of this project.

Each public class and method that you write should have a properly formatted comment specifying at least what the class/method's purpose is. These comments can be fairly short at (one sentence). Use @return and @param where appropriate. You should additionally have in-line comments at any point in the code where the complexity is not easily obvious from the code itself.

Concepts

This lab will introduce you to programming techniques for a doubly-linked list ADT and give you practice using the Iterator along with experience implementing the List and Iterable structures.

Background

Review the material in the online text. Be sure to complete the sections on Nodes, Overview of List ADTs and LinkList sections prior to doing this lab. Each node in a doubly linked list contains at least three fields: the data, and two pointers. One pointer points to the previous node in the list, and the other pointer points to the next node in the list. The previous pointer of the first node, and the next pointer of the last node are both null.

Here's the C++ class definition for a doubly linked list node:

```
template<typename T>
class Node

{
public:
      T element; // Element contained in the node
      Node* next; // Pointer to the next node
      Node* prev; // Pointer to the previous node

      Node() // No-arg constructor
      {
            next = NULL;
```

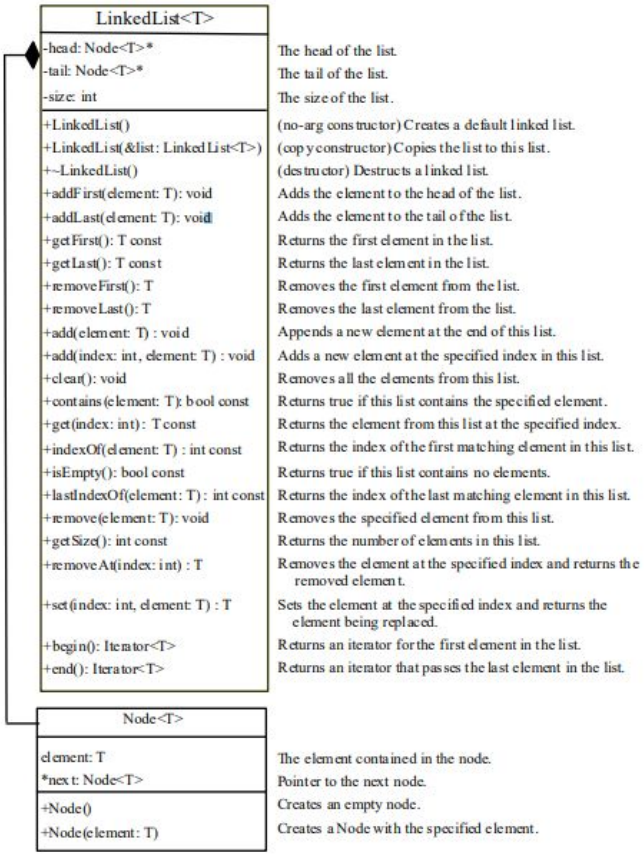
```
        prev = NULL;
}

Node(T element) // Constructor
{
        this->element = element;
        next = NULL;
        prev = NULL;
        }
};
```

Assignment

#2a) Write a new class called DLList that implements the double linked list container structure based on the picture below. You don't have to implement all the methods, and I encourage you to think of different logic for the add and remove methods than are suggested at here.

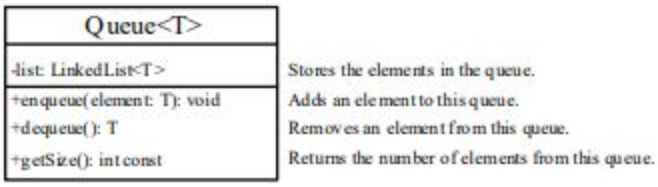
You only need to implement the following:
add(T), add(pos, T), clear(), contains(T), get(pos), isEmpty(), iterator(), remove(T), remove(post), and getSize()



Discussion/notes on the entire lab:
Most of the Queue methods are implemented simply by calling methods that you have already written in your double linked list container class. The code for all of the queue methods should be very short. Include the methods as shown below as well as:

front(), back()

Which return the first and last element in the list.



Testing

I will be providing you with a simple unit test that will test all the required methods, which is how we will be grading the lab. I also highly suggest doing simple output tests as you implement your methods. If you have trouble with the unit test, please demonstrate your working code via output testing by printing your list and demonstrating the add/remove/contains/size methods.

Use this function to print the list (it uses functions that you must write for the linked list):

```
void printList(const LinkedList<string>& list)
{
      for (int i = 0; i < list.size(); i++)
      {
            cout << list.get(i) << " ";
      }
      cout << endl;
 }
```

Submission

The entire QT project folder, including the report pdf. This zipped file should be turned in using the same method and done for previous labs.

Requirements for 2a): For full credit, you must have the DLList class header properly defined as described above. You must show work on all of the required List methods. Your code does not need to compile at this time.

Requirements for 2b): You must finish the remaining requirements, and the main method must output the some testing of your code.

Report: Write a lab report explaining your process, including an resources used (book, internet, peers, tutors, etc). The lab report should include screenshots of your code and individual screenshots of your own testing if completed. The lab report should be written using grammatical conventions.