INFSCI 0020 Program Design and Software Tools
Homework 5, Due April 6

In this assignment you will develop a **Deque** template class using doubly linked list, similar to the **Queue** template class shown in Fig17.13. A **Deque** object allows insertions and deletions at both the **front** as well as the **back** of a list and hence is sometimes called a **double ended queue**. The key member functions needed for a **Deque** class include:

- **pushFront**: insert an element in the front of the **Deque**
- **pushBack**: insert an element in the back of the **Deque**
- **popFront**: removes and returns an element from the front of the **Deque**
- **popBack**: removes and returns an element from the back of the **Deque**

Additionally, include the following support functions:

- **getSize**: return the size of the **Deque**
- **isDequeEmpty**: returns true if the **Deque** is empty
- **printDeque**: prints the elements of the **Deque** from front

You can define any additional member function as you deem fit. You need to define appropriate member variables – a look at the **Queue** class in Chapter 17 of the book will help you decide appropriate member variables.

To implement the **Deque** class, you will use a doubly linked list - call it class **DoublyLinkedList**. Hence you have to first implement the class **DoublyLinkedList**. You can start with the implementation of the **ListNode**, **List** and **Queue** classes explained in the book. Note that to implement **Queue**, you use **ListNode** (Fig 17.3) and **List** template classes (Fig 17.4). To implement **Deque**, you will essentially use modified versions of **ListNode**, as you now need a forward link and a backward link. Your **DoublyLinkedList** template class will essentially be a modification of the **List** template class. Note that insertion and deletion operation will be different now. Start with the zipped files provided on the web for the book’s implementation of the **Queue**.

**Client program:**

Write a client program that will allow testing for all the functions. Provide the following menu to allow choosing the operations users want to perform:

1. Push an element at the front of the **Deque**
2. Push an element at the back of the **Deque**
3. Pop an element from the front of the **Deque**
4. Pop an element from the back of the **Deque**
5. Print the size of the **Deque**
6. Print the elements of the **Deque**
7. Exit the program

Your client program should demonstrate these functionalities for **int** and **double**.

**Extra points:** Implementing the above menu and testing component as a template function will give you 10 extra points. Note that use of a template function means you will reuse your code for demonstrating above functionalities for a **Deque** of integer elements and double elements.