Lab 06 - Deque

Direction: Submit typed work in the Labs directory of your github repository and/or as an attachment on Google classroom under the accurate Lab06 assessment. All submissions should have their appropriate extension (h).

Group Lab

A deque is a double ended queue, which means it allows insertions to, removals from and views of either end of its collection. To allow all of its standard methods to have a constant big-O runtime, it is implemented using a doubly linked list with a reference to its head node named *front* and another reference to its last node named *back*. Your objective is to define each of the following methods in separate header files that must include the Deque.h file provided. Each member of the group is responsible for defining exactly 1 method. If more than one person submit work for the same problem, no points will be awarded for the problem and there will be a point penalty.

The methods are

a. a void method named `InsertFirst()` whose header is

```
template<class T>
void InsertFirst(const T& itm)
```

It inserts *itm* to the beginning of the deque.

b. a void method named `InsertLast()` whose header is

```
template<class T>
void InsertLast(const T& itm)
```

It inserts *itm* to the end of the deque.

c. a void method named `RemoveFirst()` whose header is

```
template<class T>
void RemoveFirst()
```

If the deque is not empty, it removes the head node of the deque.

d. a void method named `RemoveLast()` whose header is

```
template<class T>
void RemoveLast()
```

If the deque is not empty, it removes the last node of the deque.

Remember when defining the methods to consider all scenarios, make sure there are no memory leaks and that the linked list is in a stable state when the methods terminate.