Labwork 2

Add the following member functions to the DoublyLinkedList class provided to you. You will need to change both header and cpp files. Submit both of these files. **Your work should be able to compile and run with the example main file provided**. Your work will be evaluated using Visual Studio 2015.

You are **NOT** allowed to use any pre-written functions from DoublyLinkedList class.

Q1) Add the member function `void append(const DoublyLinkedList<T> &)` to DoublyLinkedList class. This function should append all elements in the parameter list to the caller list. After this function, parameter list should stay intact. Lists should also remain independent after the operation (adding/removing an element from one list should not affect the other). Remember to take care of memory leaks and corner cases.

Before	**After**
A: 5 8 2 4	A: 5 8 2 4 6 7 11 0
B: 6 7 11 0	B: 6 7 11 0

Q2) Add the member function `DoublyLinkedList<T> * split()` to DoublyLinkedList class. This function should split the linked list into two: first half of elements should remain in the caller list whereas second half will be transferred to a new list whose address will be returned by the function. If there are an odd number of elements in the list, caller list should have one more element than the new list (as shown in the example). Lists should remain independent after the operation (adding/removing an element from one list should not affect the other). Remember to take care of memory leaks and corner cases.

Before	**After**
A: 5 8 2 4 6 7 11 0 15	A: 5 8 2 4 6
	C: 7 11 0 15