Exercise 1 <25 points>

You will implement a queue data structure, as described in class. The structure and function declarations are the following (queue.h in the homework files):

```c
struct ListNode {
    int data;
    ListNode *next;
};

struct Queue {
    ListNode *head;
    ListNode *tail;
};

// Creates a new queue
void createQueue(Queue &q);

// Enqueues a new element
void enqueue(Queue &q, int data);

// Dequeues a new element
// Assumes queue is non-empty
int dequeue(Queue &q);

// Returns the value of the first element in the queue,
// without dequeueing
int peek(Queue &q);

// Returns true if the queue is empty
bool isEmpty(Queue &q);

// Prints out the contents of the queue
void printQueue(Queue &q);

// Destroys queue
void destroyQueue(Queue &q);
```
Don't reinvent the wheel! You should be able to implement this queue data structure reusing practically all the code from the double-ended list seen in class.

To test your list implementation, a main_queue.cpp is provided in the lab files. Running this program with a correct queue implementation should yield the following:

```
1 2 3 4 5 6 7 8 9
First element is 1
1 2 3 4 5 6 7 8 9
Dequeued element 1
2 3 4 5 6 7 8 9
Dequeued element 2
3 4 5 6 7 8 9
Dequeued element 3
4 5 6 7 8 9
Dequeued element 4
5 6 7 8 9
Dequeued element 5
6 7 8 9
Dequeued element 6
7 8 9
Dequeued element 7
8 9
Dequeued element 8
9
Dequeued element 9
Queue is empty!
1 2 3
Queue is empty!
```

**Exercise 2  <<10 points>>**

Add the following function to the linked list implementation seen in class (available on the course website, in the “Files” section):

```cpp
void append(List &l1, List &l2);
```

This function will take the contents of list2 and append them at the end of list1. This does *not* mean that you simply have to link the last element of list1 with the first element of list2. You have to **copy** the contents of list2 and place them in the end of list1. This means that, for example, if we were to modify the contents of list2 (after doing an append operation), this will not affect the contents of list1.