**Linked List Quiz**

You will be adding to the simplified MyLinkedList class (portion shown below, **code in folder**):

public class MyLinkedList

{

private ListNode front; //first element in the list

public class ListNode {

int data; //data stored in this node

ListNode next; //a link to the next node in the list

public ListNode(int data) { ... }

public ListNode(int data, ListNode next) { ... }

}

//\*\*\*\*\* student methods added here \*\*\*\*\*

}

***You may not use any API data structure (e.g. ArrayList) or add any other methods to the MyLinkedList class (e.g.*** *add* ***or*** *remove****).***

***You only have access to the first element (the*** *front* ***node). For all methods, you can assume the list will contain at least one element (you don't have to worry about the*** *front* ***node being*** *null****).***

1. Initialize the public static final variables with your name and student ID.
2. Your goal is to complete as many of the following problems as you can. You do NOT need to complete them in order (if you get stuck on one, skip it). Note that your code MUST compile.
3. Complete the method void replaceLast(int n) that replaces the value of the last element in an arbitrarily long list with the value of parameter n.

For example, if a variable list stores the sequence of values below, the call of list.replaceLast(42) should modify the list as follows:

[1, 2, 3, 4, 5, 6] >>> [1, 2, 3, 4, 5, 42]

1. Complete the method int countDuplicates() that returns the number of duplicates in a *sorted* list. The list will be in sorted order, so all of the duplicates will be grouped together.

For example, if a variable list stores the sequence of values below, the call of list.countDuplicates() should return 7.

[1, 1, 1, 3, 3, 6, 9, 15, 15, 23, 23, 23, 40, 40]

1. Complete the method int lastIndexOf(int n) that returns the index in the list of the last occurrence of that value, or -1 if the value is not found in the list.

For example, if a variable list stores the following sequence of values, then the call of list.lastIndexOf(18) should return 6 because that is the index of the *last* occurrence of 18:

[1, 18, 2, 7, 18, 39, 18, 40]

If the call had instead been list.lastIndexOf(3), the method would return -1 because 3 does not appear in the list.

1. Complete the method void stutter() that doubles the size of a list by replacing every integer in the list with two of that integer.

For example, suppose a variable list stores the following sequence of integers:

[1, 8, 19, 4, 17]

After a call of list.stutter(), it should store the following sequence of integers:

[1, 1, 8, 8, 19, 19, 4, 4, 17, 17]

1. Complete the method void removeAll(int n) that removes all occurrences of a particular value.

For example, if a variable list contains the following values:

[3, 9, 4, 2, 3, 8, 17, 4, 3, 18]

The call of list.removeAll(3) would remove all occurrences of the value 3 from the list, yielding the following values:

[9, 4, 2, 8, 17, 4, 18]

You should preserve the original relative order of the elements of the list.

1. Complete the method int deleteLast() that will remove and return the last element in the list.

For example, if the variable list initially stores this sequence of integers:

[1, 8, 19, 4, 17]

It should store the following sequence of integers after deleteLast is called (and return 17):

[1, 8, 19, 4]

**(more on next page)**

When you're done, FOLLOW THE INSTRUCTIONS BELOW TO SUBMIT (failure to follow submission instructions will cause you to lose points):

1. Click the link for your section; you will upload files via Dropbox:

**1b:** [Click here](https://www.dropbox.com/request/oREXUh64qWLASDk36xJg)

**3b**: [Click here](https://www.dropbox.com/request/4q2pyZ1bGu2ix4JQezdm)

1. Upload ONLY the following files:
   1. Your MyLinkedList.java source file.