COSC3331 - Data Structures & Algorithms I
Programming Assignment 3 **Due Date: Friday, Oct. 15 @ 11:59 pm**

Target Topic: Linked List Operations

Here is the description of this two part assignment:

PART I: Word Game

Prompt the user for a string.
Dissect it into characters and build a **SINGLE-ENDED SINGLY LINKED LIST** out of them.
When we have a linked list of characters, we would like to implement the following rules:

1. Upon encountering a sequence of `"rm"` remove the character that follows this sequence as well as `"rm"`. In case it appears as the last segment of the word with no character following it, just remove `"rm"`.

2. No vowels should be in between consonants. In case you find such occurrence, delete the vowel.

Here is an example: The first string is the original user input, the second is the result of one round of implementation of the above rules and the third is the final version of the string after complete elimination of illegal occurrences:

```
coppermakerimprint
cppkrmprnt
cppkrnt
```

As you notice you may need to reapply the rules to make sure that the final version of the string complies with our rules. So you keep going over the linked list as many times as needed until the string does not violate any of the above rules. For our example here, it took 2 rounds of "scanning" to get to the final product.

You are required to display the current state of your linked list after completion of each round of implementation of the rules, until the last one which is the final product of this game.

PART II: Numbers Game

Prompt the user for two sets of sorted numbers. You will first ask the number of items for the first list and then read those numbers in a **SINGLE-ENDED SINGLY LINKED LIST**. You repeat this process for the second linked list.

When you have your two sorted linked lists of integers. Write a segment of code to merge them together into a single sorted linked list.
Next you must display the possible segments of duplicate values found in the merged list on separate lines. In case there were not such occurrences, display a proper message.

Here is an example:

```
List 1:  3 4 6 11 11 19 24 24
List 2:  1 6 6 6 19 22 28 28 28 31

Merged List:  1 3 4 6 6 6 6 11 11 19 19 22 24 24 28 28 28 31
```

```
Sublists of duplicates:
6 6 6 6
11 11
19 19
24 24
28 28 28
```

No report is required for this assignment. You're Welcome!

You must use only the textbook code and not any Java classes. The violation of this rule, makes your submission void.

- *You must upload all the required files on blackboard by the due date.*
- *Programs that do not compile will be considered void.*
- *This is an individual assignment. If students submit fully or partially identical programs, ALL individuals involved will earn a grade of zero.*