Week 12: Welcome to second Review Session!

Due: at the end of the lab session when this assignment was released to you.

How: via link **bit.ly/19w_cs2-mw**

What do you have to do?

1/ Take the Socrative quiz called "Review Session 2"
Room: **CS2401LABMW**

2/ Download files from piazza:
For those taking labs on MW: file **W12-MW.zip**

3/ Work with your favorite IDE / editor

4/ Turn in your work as a pair work or individual work depending on what you decided on: **bit.ly/19w_cs2-mw**

Below is a set of 4 problems. Complete as many as you can. A problem is considered solved when it has passed all test cases for this problem. The java file that is given to you contains all dummy stubs for these methods. You just have to change the body of these methods. You are also given test files that you can run from your favorite IDE to check if your code passes all of our test cases.

maxMod5

Given two int values, return whichever value is larger. However if the two values have the same remainder when divided by 5, then the return the smaller value. However, in all cases, if the two values are the same, return 0. Note: the % "mod" operator computes the remainder, e.g. 7 % 5 is 2.

maxMod5(2, 3) → 3
maxMod5(6, 2) → 6
maxMod5(3, 2) → 3

repeatFront

Given a string and an int n, return a string made of the first n characters of the string, followed by the first n-1 characters of the string, and so on. You may assume that n is between 0 and the length of the string, inclusive (i.e. n >= 0 and n <= str.length()).

repeatFront("Chocolate", 4) → "ChocChoChC"
repeatFront("Chocolate", 3) → "ChoChC"
repeatFront("Ice Cream", 2) → "IcI"

canBalance

Given a non-empty array, return true if there is a place to split the array so that the sum of the numbers on one side is equal to the sum of the numbers on the other side.

canBalance([1, 1, 1, 2, 1]) → true
canBalance([2, 1, 1, 2, 1]) → false
canBalance([10, 10]) → true

array220

Given an array of ints, compute recursively if the array contains somewhere a value followed in the array by that value times 10. We'll use the convention of considering only the part of the array that begins at the given index. In this way, a recursive call can pass index+1 to move down the array. The initial call will pass in index as 0.

array220([1, 2, 20], 0) → true
array220([3, 30], 0) → true
array220([3], 0) → false