Assignment #1 – Recursion

Develop Java program for each of the following problems. Please **name the programs as indicated** and add proper **program headers** and **output labels** as shown below.

Use <u>only</u> concepts and programming constructs/syntax we discussed in class.

Program name (= Class name) must be same as the java filename.

At first, **design** your strategy of this problem with **recursive thinking**, then write code.

Programming #1 (50 points)

Write your Java program named ***PrintReverseYourName*** as follows. The main method prompts the user to enter a string. The main method then passes the input string to method `void printCharsReverse(String str)` to **recursively** print given string in reverse. Again, method `void printCharsReverse(String str)` is a recursive method which print 'one character' for each invoke.

Format the outputs as follows. Test data shows values for illustration, user may enter values one per line. Test strings does not show input prompts.

Test data:

```
    Entered string:         Hello world    //Red characters are user input
    Reversed string:        dlrow olleH
    Try again(Y/N):         Y

    Entered string:         Data Structures and Algorithms
    Reversed string:        smhtiroglA dna serutcurtS ataD
    Try again(Y/N):         N
```

Document your code, use proper prompts for input, format outputs as shown above, use sound coding practices we learned thus far, do not hard code inputs, allow program re-runs, and test your code thoroughly.

Programming #2 (50 points)

Write a java program (name it ***AverageGradeYourName***) as follows: The main method prompts the user to enter number of students in a class (class size is integer value), then prompts the user to enter the grades (between 0 and 100) into an array of type integer. The entered class size determines the array size. Next, the main method passes the filled array to method `computeAverage(…)` to **recursively** determine and return the class average as a double value. Again, method `computeAverage(…)` is a **recursive** method.

Format the outputs as follows. Shown input values are just for illustration, user may enter values one per line. Test data does not show input prompts.

Test data:

```
Class size:            3
Entered grades:        100 100 100
Class average:         100.00

Class size:            7
Entered grades:        50 75 80 80 40 35 85
Class average:         63.57

Class size:            8
Entered grades:        0 100 25 90 55 30 90 35
Class average:         53.13
```

Document your code, use proper prompts for input, format outputs as shown above, use sound coding practices we learned thus far, do not hard code inputs, allow program re-runs, and test your code thoroughly.

(Spoiler Alert) On the last page, there is a strategy to get average recursively. Please think yourself first, and check the last page if you have no idea!

Submission: Before submitting your programs, make sure you review the assignment submission requirements and grading guidelines on the course webpage. The grading guidelines explain some of the common errors found in programming assignments.
1. The assignment is due no later than **11:59 PM** on the due day posted in D2L.
2. Please compile and run your java files (only the .java files) right before you upload to the assignment submission folder in D2L.

Do not forget to include author header in each submitted file as shown, **no header, no points!**
// Name: <your name>

// Class: CS 3305/Section#

// Term:

// Instructor: Umama Tasnim

1	2	3	4	5

1	2	3	4		5

$$\text{Mean}(\{1,2,3,4,5\}) = \frac{\text{Mean}(\{1,2,3,4\}) * 4 + 5}{5}$$

1	2	3		4

$$\text{Mean}(\{1,2,3,4\}) = \frac{\text{Mean}(\{1,2,3\}) * 3 + 4}{4}$$

1	2		3

$$\text{Mean}(\{1,2,3\}) = \frac{\text{Mean}(\{1,2\}) * 2 + 3}{3}$$

1		2

$$\text{Mean}(\{1,2\}) = \frac{\text{Mean}(\{1\}) * 1 + 2}{2}$$