600.107 Introductory Programming in Java, Summer 2015
Homework 2
Due: Thursday, July 2nd at 11:59pm

Please review the **Homework Guidelines** handout for this course before you begin.

> **Restriction:** For all tasks in this assignment, you are not permitted use decision statements (if, if-else, or switch), repetition statements (while, for, do-while), arrays or ArrayLists, or methods from the Java API's `Math` class other than `pow` and `sqrt`.

Task 1. Write a Java class called `AddFraction` that asks the users for two integer values, representing the numerator and the denominator, respectively, of a fraction. Then, ask the user for two more integer values, representing the numerator and denominator, respectively, of a second fraction. *You must use only int type variables in this program; don't use any variables of type double or float.* Your program should first echo back the two fractions as output, and then compute and output the sum of the two fractions, with an appropriate label. You need not reduce the sum to lowest terms. For example, if the user enters the integers 10, 20, 4 and 7, the fractions represented are 10/20 and 4/7. Their sum would be printed as 150/140.

Here is a sample run where user enters 1 2 15 16 as shown below in bold:

```
Please enter the numerator of the first fraction:  1
Please enter the denominator of the first fraction:  2
Please enter the numerator of the second fraction:  15
Please enter the denominator of the second fraction:  16

Now adding fractions 1/2 and 15/16...
Their sum is 46/32.
```

Please note that no actual division is taking place during the execution of this program. The / sign is just used to denote a fraction with integer numerator and denominator.

You should assume that no denominator entered by the user is a zero value. You don't have to check for this. Later in the course, we'll learn how to detect and deal with improper input values.

Task 2. Write a Java class called `QuadEqn` that takes three *double* numbers from the Java console representing, respectively, the three coefficients a, b, and c, of a quadratic equation $ax^2 + bx + c = 0$. Solve the equation using the quadratic formula

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

and print out the two roots you obtain. You'll find the `Math.sqrt()` method helpful here. First, sketch out three sample runs with the following input values (you'll have to determine the proper output values yourself), so you know what answers you should be getting when you execute your program.

Inputs for sample run 1: $a = 1$, $b = 3$, $c = 2$
Inputs for sample run 2: $a = 0.5$, $b = 0.5$, $c = 0.125$
Inputs for sample run 3: $a = 2$, $b = 19$, $c = -91$

For the purposes of this problem, you may assume that the a coefficient entered is always nonzero, and that the three coefficients entered always give you a nonnegative *discriminant*; that is, the $b^2 - 4ac$ part is never negative. If the entered coefficients give you a discriminant value of 0, your program will simply output the single root twice. (Remember, you're not allowed to use `if` statements!)

Task 3. Write a Java class called `Mortgage` that reads the amount still owed on a mortgage (the outstanding balance), the monthly mortgage payment, and the annual interest rate on the loan, and then displays the amount of the payment that goes to interest and the amount that goes to principal (i.e., the amount that goes to reducing debt) for both the first and the second month. Note that payments are made monthly, so the monthly interest is only one twelfth of the annual interest percentage.

You should assume that the user will not enter any commas in the numeric values she or he provides. You must use `printf` formatting (see pages 184-186 of Section 5.7 of the text) to format the dollar amounts you output so that exactly two decimal places are shown.

Below is a sample execution, with user input shown in bold:

```
Please enter the outstanding balance (no commas, please):  250000.00
Now, enter the monthly mortgage payment (again, no commas):  1229.85
Next, we need the annual interest rate.  For example, enter 4.98% as 0.0498.
Please enter the rate:  0.0425

First Month
Amount going to interest:  $885.42
Amount going to principal: $344.43

Second Month
Amount going to interest:  $884.20
Amount going to principal: $345.65
```

Turn-in. Before the due date listed at the top of this file, please submit via the course Blackboard site one complete `.zip` file named `HW2-jhed.zip` (where you replace 'jhed' with your own personal JHED) containing all `.java` files required for the above tasks. Be sure that all submitted code compiles! If you were unable to complete a task, please include as part of your `.zip` submission a text file named `README` explaining anything you'd like the graders to know.

Need to fix something that you already submitted? At any point up until the due date for this assignment, you may re-submit a complete zip file. The submission most recently received is the one that will be graded. Any earlier submissions will be ignored, so please include *all* files in each submission.