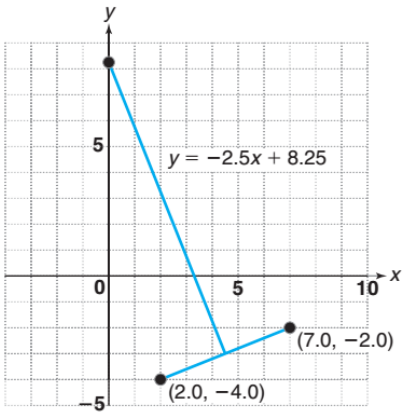
CSE 1141 - COMPUTER PROGRAMMING I
Programming Assignment # 1
DUE DATE: 15/10/2018 - 23:00 (No extension)

1. Write a program that outputs the equation of the perpendicular bisector of the line segment between two points. Your program should
   - prompt the appropriate messages and input the coordinates of the two points as (X_1, Y_1) and (X_2, Y_2) [for example, (2.0, -4.0) and (7.0, -2.0)];
   - compute the slope of the line between those two points [$slope = \frac{Y_2 - Y_1}{X_2 - X_1}$];
   - compute the coordinates of the midpoint of the line segment between the two points by averaging the two *x* coordinates and the two *y* coordinates; [$midpoint = \left(\frac{X_1 + X_2}{2}, \frac{Y_1 + Y_2}{2} \right)$]
   - compute the slope of the perpendicular bisector by taking the negative reciprocal of the slope of the line segment; [$m = -slope^{-1}$];
   - compute the y intercept of the perpendicular bisector (you now have the slope m of the bisector and the midpoint (x_{mid}, y_{mid}) on the bisector, so the y intercept is $y_{mid} - m * x_{mid}$);
   - output with labels the original two points, and output in $\mathbf{y = mx + b}$ format the equation of the perpendicular bisector. [*b* is the y intercept of the perpendicular bisector found in previous step.]
     - For example: y = -2.5x + 8.25
   - The figure below illustrates the sample line segment mentioned above and its perpendicular bisector.



Test your program to be sure it works on different pairs of points.

Example run:

```
Enter the value of x1:   2
Enter the value of y1:  -4
Enter the value of x2:   7
Enter the value of y2:  -2
The equation of the perpendicular bisector of the line segment between
(2.0,-4.0) and (7.0,-2.0) is y = -2.5x + 8.25
```

It should be noted that the figure above just shows the pictorial representation of the example given. Your program does not need to print the graph.

2. The Private Pension System, PPS for short, is a special pension system in order to enable you to spend your retirement happily by means of regular savings. Assume that you want to participate the PPS system by dedicating a certain amount of your salary (for example, 10%) each month with the monthly interest rate *x*%. Additionally, the PPS offers great advantage with the *y*% government support.

   As an example, suppose that you save 1000 TLs each month into your saving account with 25% government support and the monthly interest rate is 2%.

   After the first month, the value in the saving account becomes

   $$1000 * (1 + 0.25) * (1 + 0.02) = 1275$$

   After the second month, the value in the account becomes

   $$(1275 + 1000 * (1 + 0.25)) * (1 + 0.02) = 2575.5$$

   After the third month, the value in the account becomes

   $$(2575.5 + 1000 * (1 + 0.25)) * (1 + 0.02) = 3902.01$$

   and so on.

   Write a program that prompts the user to enter
   - the monthly salary,
   - the portion of salary to be saved,
   - the monthly interest rate and
   - the government support rate.

   Then, your program should display the value of your saving account for **six months**, the total value saved by your salary and your total profit at the end of six months.

   Example Run:

```
Enter your monthly salary, for example 3500: 10000
Enter the percent of your salary to save, for example 10%: 10
Enter the monthly interest rate, for example 8.25%: 2
Enter the monthly government support rate, for example 25%: 25

After the first  month, the account value is 1275.0 TLs
After the second month, the account value is 2575.5 TLs
After the third  month, the account value is 3902.01 TLs
After the fourth month, the account value is 5255.05 TLs
After the fifth  month, the account value is 6635.15 TLs
After the sixth  month, the account value is 8042.85 TLs

In total, you saved 6000.0 TLs and your total profit is 2042.85 TLs
```

   Note: *You should print the values with two significant digits (at most). You should declare the input variables as floating point numbers. Your program should execute correctly for different test cases.*

Submission Instructions

Please zip and submit all your files using filename YourNumberHW1.zip
(ex: 150713852HW1.zip) to Canvas system (under Assignments tab).
Your zip file should contain the followings:

1. Java source code for Problem 1 (Pro1_150713852.java)
2. Java class file for Problem 1 (Pro1_150713852.class)
3. Java source code for Problem 2 (Pro2_150713852.java)
4. Java class file for Problem 2 (Pro2_150713852.class)

Notes:

1. Write a comment at the beginning of each program to explain the purpose of the program. Write your name and student ID as a comment. Include necessary comments to explain your actions.
2. Select meaningful names for your variables.
3. You are allowed to use the materials that you have learned in lectures & labs.
4. Do not use the things that you did not learn in the course.
5. Each student should submit his/her own homework. You can discuss with your peers about the homework but you are not allowed to exchange code or pseudocode. This also applies to material found on the web. Should some submitted homework assignments be identical or suspected to be identical, all involved parties will get a grade of **ZERO** from all homeworks. In case of any forms of cheating or copying, both giver and receiver are equally culpable and suffer equal penalties.
6. No late submission will be accepted.

Grading:

Problem 1 (40 points)

- Reading four input values from the user (10 points)
- Finding the midpoint (5 points)
- Finding the slope of the line (5 points)
- Finding the slope of the perpendicular bisector (5 points)
- Finding the y intercept of the perpendicular bisector (5 points)
- Printing the output in $\mathbf{y = mx + b}$ format (10 points)

Problem 2 (50 points)

- Reading four input values from the user (10 points)
- Finding the value of saving account for six months (12 points)
- Finding the total value saved by the salary (5 points)
- Finding the total profit (5 points)
- Output format correctness (5 points)
- Correct execution for test inputs (13 points)

Submission format (10 points)

- 2 java files + 2 class files
- Make sure that your class files can be executed on another computer.
- Make sure that the input/output of your program must be the same with the examples above (all informative strings & spaces).
- Comments are necessary!