Worksheet 6 **Name:** ______________________________

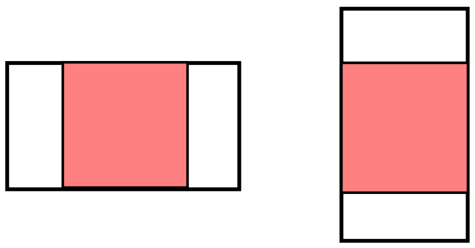
1. At some point in your math studies, you learned that the standard quadratic equation

$$ax^2 + bx + c = 0 \quad \text{can be solved by the } \textit{quadratic formula}: \quad x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$$

The quantity under the radical sign ($\sqrt{\ }$) is called a radicand. In this case, it is $b^2 - 4ac$, which can be used to determine **how many real roots** the equation has.

Given the coefficient values	Value of $b^2 - 4ac$	Number of of real roots	Roots, if any:
$a = 1$, $b = 1$, and $c = -6$			
$a = 1$, $b = 4$, and $c = 4$			
$a = 6$, $b = 5$, and $c = 2$			

2. Assume we desire a background square that fills the graphics window in the smaller dimension – in other words, *select the smaller of the window's width and height* for the size of the background, then **center the square in the larger dimension**, as shown below. On the left, the height is the smaller dimension, on the right the width is smaller.



(a) Complete the code fragment below, to determine the square size based on the window's width and height.

```
double wWidth = getWidth();
double wHeight = getHeight();
double sqSize;
```

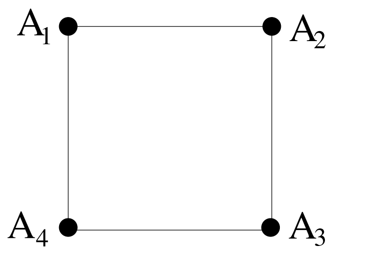
(b) Give the code to determine the x-value of the location of the square, regardless of which dimension it is going to fill:

```
double x =
```

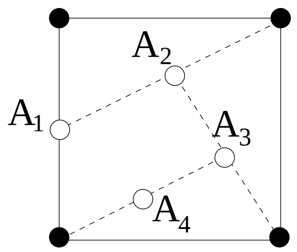
(c) Give the code to determine the y-value of the location of the square, regardless of which dimension it is going to fill:

```
double y =
```

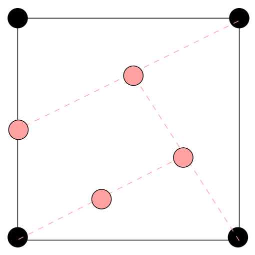
3. Four ants are forming a Dance Club, and will perform in their own unique style for us. They begin at the corners of a square, then each advances in turn (clockwise ordering) – the first ant A_1 in the upper left corner moving toward the fourth ant A_4 below it. Then A_2 moves toward A_1, A_3 steps toward A_2, and finally, A_4 dances toward A_3. Then the process repeats itself.



The Ants begin their dance.



Round #1


Round #2

(a) Assuming the ants move a fraction of the distance between them, in this case 50%, find the approximate position of each ant after the second round of steps, and label who's who, in the third picture above.

(b) Let P be the point (x_P, y_P), Q be the point (x_Q, y_Q) and suppose the distance between these points is D. If P moves toward Q a distance rD, it stops at the point $(x_P + r(x_Q - x_P), y_P + r(y_Q - y_P))$. Assume that before anyone moves, ant A_1 is at the origin, (0.0, 0.0), A_2 is at (100.0, 0.0), A_3 is at (100.0, 100.0), and A_4 is at (0.0, 100.0). Calculate their new positions after they have each taken their first step in the dance, as shown in Round # 1 above, where r is 0.5. Note that each ant moves towards the next ant's newest position.

Order of Movement	Original Coordinates	Move	Toward by 50%	New Position
1	(0.0, 0.0)	A_1	A_4	A_1 :
2	(100.0, 0.0)	A_2	A_1	A_2 :
3	(100.0, 100.0)	A_3	A_2	A_3 :
4	(0.0, 100.0)	A_4	A_3	A_4 :

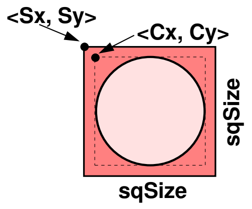
4. A checkerboard has 8 rows and 8 columns. To set up for a game of checkers, playing pieces are placed in the first three and last three rows of the checkerboard, on same color squares.

(a) Assume the following loop is in effect:

```
for (int row = 0; row < MAXSIZE; row++)
```

What boolean expression can be used to ascertain whether the current value of `row` falls into the "first three" or "last three" rows of the board?

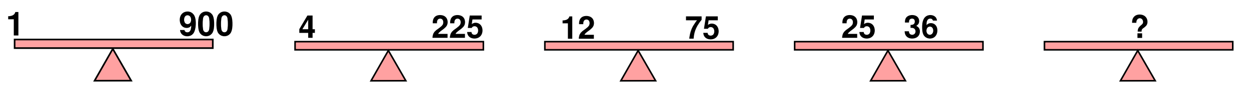
(b) Assume the coordinate of a square, `<Sx, Sy>`, and its size, `sqSize`, are known, and we wish to center a circle that is 80% of the size of the square in the square. Give the Java statements which will accomplish this.



5. You may recall a lab exercise that required you to print the digits of a positive integer in reverse. The code to do that is given on the left below. Give the modification of that code which will *sum* the digits, then in the third column, the code which will *count* the number of digits in an integer. Declare and initialize any necessary objects.

Reverse Digits	Sum Digits	Count Digits
`while (n > 0) {`		
`  print(n % 10);`		
`  n /= 10`		
`}`		
`println();`		

6. If we divide any number by successively larger divisors (but still less than the number), the quotients becomes successively smaller, as shown on the balances below for the number 900. At some point, these divisors and quotients meet. What is the value of the divisor and quotient at the middle of the balance?



```
i.e., when are the divisor and quotient the same (?) :
```

7. We have already seen how to determine whether an integer is even or odd in the checkerboard program, which used the expression `((row + col) % 2) == 0)` to alternate between black and white squares.
   (a) If we want to determine if an integer `d` divides another integer `n` evenly, what test could be used?

   (b) We want to find *all* the integers which divide `n`, a positive integer, evenly. Give the loop which will find and display all the divisors of `n` between 2 and `n-1` inclusive: