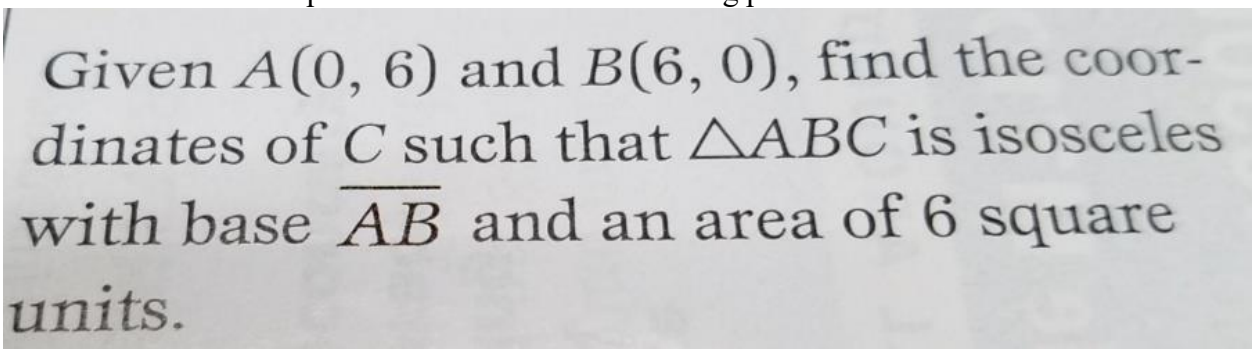
Fun With Triangles

This problem has 13 points: it has 12 different testers and a bonus Point for successfully passing all 12 tests ☺

The motivation for this problem came from the following puzzle.



According to Wikipedia, an isosceles triangle is a triangle that has two sides of equal length. The equal sides are called legs and the third side is called the base.

In this problem, you are given two vertices, A and B, of a triangle and are to find the coordinates of the third vertex C, such that the triangle ABC is isosceles with base $\overline{AB}$ and area `myArea` square units. Since two solutions exist, you will implement two methods. The first method returns the vertex C with the largest x value (if both vertices have the same x value, return the vertex with the larger y value) and the second method returns the vertex with the smallest x value.

This problem uses the completed `TriangleCoordinate` helper class representing the coordinates of a vertex of a Triangle. The `TriangleCoordinate` has accessor methods `getX()` (returns the x-value) and `getY()` (returns the y-value) and a correctly implemented `equals` method. `getX()` and `getY()` both return `double` values.

In this problem you will complete two methods in the `FunWithTriangles` class. Both `vertexA` and `vertexB` are `TriangleCoordinate` objects and are instance variables in the `FunWithTriangles` class. The instance variables are initialized by the constructor.

Note: all sample data shown in this problem uses whole numbers, the actual test data will include non integer values.

Fun With Triangles

The first method to complete is:
`TriangleCoordinate generateAreaWithLargestXcoordinate(double myArea)`.
`generateAreaWithLargestXcoordinate` returns the `TriangleCoordinate` such that, along with `vertexA` and `vertexB` forms an isosceles triangle with base formed by `vertexA` and `vertexB` and area equal to `myArea`. Since two different triangles (and therefore two different vertices) exist, this method returns the `TriangleCoordinate` representing the vertex with the largest x coordinate. If the vertex of both triangles has the same x value, return the vertex with the largest y value.

The following shows the results of `generateAreaWithLargestXcoordinate` method. In this example, the base formed by the given `TriangleCoordinate` has slope = -1.

The following code	Returns

`TriangleCoordinate x = new TriangleCoordinate(6, 0);`
`TriangleCoordinate y = new TriangleCoordinate(0, 6);`
 	
`FunWithTriangles fwt = new FunWithTriangles(x, y);`
 	
`TriangleCoordinate ans =`
`fwt.generateAreaWithLargestXcoordinate(6);`	

`ans.getX();`	4.0
`ans.getY();`	4.0

The following shows the results of `generateAreaWithLargestXcoordinate` method. In this example, the base formed by the given `TriangleCoordinate` is horizontal (has slope = 0).

The following code	Returns

`TriangleCoordinate x = new TriangleCoordinate(-5, 0);`
`TriangleCoordinate y = new TriangleCoordinate(5, 0);`
 	
`FunWithTriangles fwt = new FunWithTriangles(x, y);`
 	
`TriangleCoordinate ans =`
`fwt.generateAreaWithLargestXcoordinate(35);`	

`ans.getX();`	0.0
`ans.getY();`	7.0

The following shows the results of `generateAreaWithLargestXcoordinate` method. In this example, the base formed by the given `TriangleCoordinate` is vertical (has undefined slope).

The following code	Returns

`TriangleCoordinate x = new TriangleCoordinate(5, 3);`
`TriangleCoordinate y = new TriangleCoordinate(5, 9);`
 	
`FunWithTriangles fwt = new FunWithTriangles(x, y);`
 	
`TriangleCoordinate ans =`
`fwt.generateAreaWithLargestXcoordinate(30);`	

`ans.getX();`	15
`ans.getY();`	6

The following shows the results of `generateAreaWithLargestXcoordinate` method. In this example, the base formed by the given `TriangleCoordinate` has a slope = 3/4.

The following code	Returns

`//  base with random slope`
`TriangleCoordinate  x = new TriangleCoordinate(-10, 3);`
`TriangleCoordinate y = new TriangleCoordinate(-2, 9);`
 	
`FunWithTriangles fwt = new FunWithTriangles(x, y);`
 	
`TriangleCoordinate ans =`
`fwt.generateAreaWithLargestXcoordinate(50);`	

`ans.getX();`	0.0
`ans.getY();`	-2.0

The second method to complete is:
`TriangleCoordinate generateAreaWithSmallestXcoordinate(double myArea)`.
`generateAreaWithSmallestXcoordinate` returns the `TriangleCoordinate` such that, along with `vertexA` and `vertexB` forms an isosceles triangle with base formed by `vertexA` and `vertexB` and area equal to `myArea`. Since two different triangles (and therefore two different vertices) exist, this method returns the `TriangleCoordinate` representing the vertex with the smallest x coordinate. If the vertex of both triangles has the same x value, return the vertex with the smallest y value.

The following shows the results of `generateAreaWithSmallestXcoordinate` method. In this example, the base formed by the given `TriangleCoordinate` has slope = -1.

The following code	Returns

`TriangleCoordinate x = new TriangleCoordinate(6,0);`
`TriangleCoordinate y = new TriangleCoordinate(0, 6);`
 	
`FunWithTriangles fwt = new FunWithTriangles(x, y);`
 	
`TriangleCoordinate ans`
`fwt.generateAreaWithSmallestXcoordinate(6);`	

`ans.getX();`	2.0
`ans.getY();`	2.0

The following shows the results of `generateAreaWithSmallestXcoordinate` method. In this example, the base formed by the given `TriangleCoordinate` is horizontal (has slope = 0).

The following code	Returns

`TriangleCoordinate x = new TriangleCoordinate(-5,0);`
`TriangleCoordinate y = new TriangleCoordinate(5, 0);`
 	
`FunWithTriangles fwt = new FunWithTriangles(x, y);`
 	
`TriangleCoordinate ans =`
`fwt.generateAreaWithLargestXcoordinate(6);`	

`ans.getX();`	0.0
`ans.getY();`	-7.0

Fun With Triangles

The following shows the results of `generateAreaWithSmallestXcoordinate` method. In this example, the base formed by the given `TriangleCoordinate` is vertical (has undefined slope).

The following code	Returns

`TriangleCoordinate x = new TriangleCoordinate(5, 3);`
`TriangleCoordinate y = new TriangleCoordinate(5, 9);`
 	
`FunWithTriangles fwt = new FunWithTriangles(x, y);`
 	
`TriangleCoordinate ans =`
`fwt.generateAreaWithSmallestXcoordinate(30);`	

`ans.getX();`	-5
`ans.getY();`	6

The following shows the results of `generateAreaWithSmallestXcoordinate` method. In this example, the base formed by the given `TriangleCoordinate` has a slope = 3/4.

The following code	Returns

`TriangleCoordinate x = new TriangleCoordinate(-10,3);`
`TriangleCoordinate y = new TriangleCoordinate(-2, 9);`
 	
`FunWithTriangles fwt = new FunWithTriangles(x, y);`
 	
`TriangleCoordinate ans =`
`fwt.generateAreaWithSmallestXcoordinate(50);`	

`ans.getX();`	-12.0
`ans.getY();`	14.0