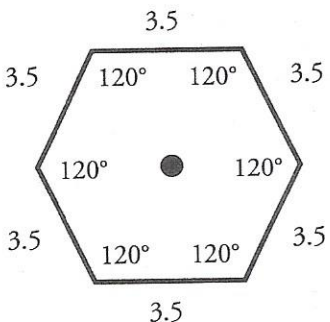


1. A regular hexagon is a six-sided geometric figure in which all six sides are the same length and all the interior angles measure 120 degrees.



A regular hexagon is represented by the `RegularHexagon` class. The `RegularHexagon` class uses the `CoordinatePoint` class to specify its vertices (corners). The `CoordinatePoint` class represents a point on the coordinate plane.

```
public class RegularHexagon
{
    /** ArrayList points holds CoordinatePoint objects that represent
     * the vertices of the RegularHexagon. The CoordinatePoint objects are in
     * consecutive order. That is, if you connect the element at index 0 with
     * the element at index 1 and so on to the element at index 5 and then
     * back to the element at index 0, the hexagon will be formed correctly.
     */
    private ArrayList<CoordinatePoint> points;

    public RegularHexagon(ArrayList<CoordinatePoint> pts)
    {
        points = pts;
    }

    /** Returns the side length for a RegularHexagon
     *
     * @return the length of each side of the RegularHexagon
     * Precondition: The CoordinatePoint objects in points are in
     * consecutive order.
     */
    public double getSideLength()
    {
        /* to be implemented in part (a) */
    }

    /** Returns the area of a RegularHexagon
     *
     * @return the area of the RegularHexagon
     * Precondition: The CoordinatePoint objects in points are in
     * consecutive order.
     */
    public double getArea()
    {
        /* to be implemented in part (b) */
    }
}
```

```

/** Returns the center of the RegularHexagon.
 *
 * @return the CoordinatePoint that is the center of the RegularHexagon
 * Precondition: The CoordinatePoint objects in points are in
 *               consecutive order.
 */
public CoordinatePoint getCenter()
{
    /* to be implemented in part (c) */
}

public class CoordinatePoint
{
    private double x, y;

    public CoordinatePoint(double myX, double myY)
    {
        x = myX;
        y = myY;
    }

    public double getX()
    { return x; }

    public double getY()
    { return y; }
}

```

- (a) Write the `getSideLength` method for the `RegularHexagon` class. The side length is found by calculating the distance between two consecutive coordinate points. The formula for computing distance is:

Example

Suppose two consecutive coordinate points of the regular hexagon are (1, 3) and (2, 4). The side length of the regular hexagon is:

$$distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Complete the method `getSideLength`.

$$distance = \sqrt{(2 - 1)^2 + (4 - 3)^2} = \sqrt{2} = 1.414\dots$$

```

/** Returns the side length for a RegularHexagon
 * @return the length of each side of the RegularHexagon
 */
public double getSideLength()

```

- (b) Write the `getArea` method for the `RegularHexagon` class. The formula for computing the area of a regular hexagon is:

$$Area = \frac{(side\ length)^2 * 3\sqrt{3}}{2}$$

Example:

Suppose a regular hexagon has a side length of 5. Its area is:

$$Area = \frac{5^2 * 3\sqrt{3}}{2} = 64.9519\dots$$

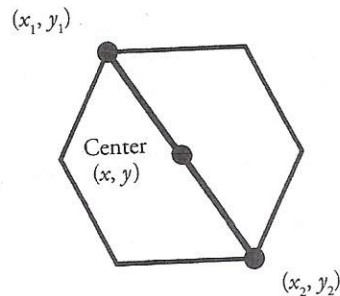
You may assume that the `getSideLength` method works as intended, regardless of what you wrote in part (a).

Complete the method `getArea`.

```
/** Returns the area of a RegularHexagon
 * @return the area of the RegularHexagon
 */
public double getArea()
```


- (c) Write the `getCenter` method for the `RegularHexagon` class. The center of a regular hexagon is located at the midpoint of any diagonal of a regular hexagon.

A diagonal of a regular hexagon is a line segment that connects two points that lie opposite each other on the hexagon.



Point 0 and point 3 are endpoints of a diagonal, as are points 1 and 4, and points 2 and 5.

The midpoint formula is:

$$\text{midpoint} = \left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right)$$

Example

Given a `RegularHexagon` object with its first vertex at (2, 5) and its fourth vertex at (4, 3), the center of the hexagon is:

$$\text{midpoint} = \left(\frac{2 + 4}{2}, \frac{5 + 3}{2} \right) = (3, 4)$$

```
/** Returns the center of the RegularHexagon.
 * @return the CoordinatePoint that is the center of the RegularHexagon
 */
public CoordinatePoint getCenter()
```