/\* 4. A class called MyPoint, which models a 2D point with x and y coordinates, is designed as follows:  
● Two instance variables x (int) and y (int).  
● A default (or "no-arg") constructor that construct a point at the default location of (0, 0).  
● A overloaded constructor that constructs a point with the given x and y coordinates.  
● A method setXY() to set both x and y.  
● A method getXY() which returns the x and y in a 2-element int array.  
● A toString() method that returns a string description of the instance in the format "(x, y)".  
● A method called distance(int x, int y) that returns the distance from this point to another point at the  
given (x, y) coordinates  
● An overloaded distance(MyPoint another) that returns the distance from this point to the given  
MyPoint instance (called another)  
● Another overloaded distance() method that returns the distance from this point to the origin (0,0)  
Develop the code for the class MyPoint.   
Also develop a JAVA program (called TestMyPoint) to test all the methods defined in the class.  
 \*/  
class MyPoint

{ int x, y;  
  
 MyPoint()

{ x = 0; y = 0;  
 }  
  
 MyPoint(int p, int q)

{ x = p; y = q;  
 }  
  
 void setxy(int p, int q)

{ x = p; y = q;  
 }  
  
 int[] getxy()

{ int[] point = new int[2];  
 point[0] = x; point[1] = y;  
 return point;  
 }  
  
 double distance(int p, int q)

{ double dist;  
 dist = Math.sqrt((double) ((x - p) \* (x - p) + (y - q) \* (y - q)));  
 return dist;  
 }  
  
 double distance(MyPoint point)   
 { double dist;  
 dist=Math.sqrt((double)((x-point.x)\*(x-point.x)+(y-point.y)\*(y-point.y)));  
 return dist;  
 }  
  
 double distance()

{  
 double dist;  
 dist = Math.sqrt((double) ((x - 0) \* (x - 0) + (y - 0) \* (y - 0)));  
 return dist;  
 }  
  
 public String toString() {  
 // overriding the toString() method  
 return ("\n\nThe coordinate value of x = " + x + " and y=" + y);  
 }  
}  
  
public class TestMyPoint   
{  
 public static void main(String args[])   
 {   
 MyPoint mypoint1 = new MyPoint(4, 5);   
 System.out.println(mypoint1);  
   
 int[] point1 = mypoint1.getxy();  
 System.out.print("Dist between Point 7,6 & "+point1[0]+ ", "+point1[1]+" is ");  
 double distance = mypoint1.distance(7, 6);  
 System.out.println(distance);  
  
 System.out.print("Dist between Origin(0,0) & "+point1[0]+ ", "+point1[1]+" is ");  
 distance = mypoint1.distance();  
 System.out.println(distance);  
   
 MyPoint mypoint2 = new MyPoint(8, 9);  
 int[] point2 = mypoint2.getxy();  
 System.out.println(mypoint2);  
 System.out.print("Distance between point " + point1[0] + ", " + point1[1]);  
 System.out.print(" and " + point2[0] + ", " + point2[1]+ " is ");  
 distance = mypoint2.distance(mypoint1);  
 System.out.println(distance);   
  
 System.out.print("Dist between Origin(0,0) and "+point2[0] + ", " + point2[1]);  
 System.out.print(" is ");  
 distance = mypoint2.distance();  
 System.out.println(distance);  
   
 MyPoint mypoint3 = new MyPoint();  
 mypoint3.setxy(10, 20);  
   
 System.out.println(mypoint3);  
 int[] point3 = mypoint3.getxy();  
 System.out.print("Dis between point " + point1[0] + ", " + point1[1]);  
 System.out.print(" and " + point3[0] + ", " + point3[1]+ " is ");  
 distance = mypoint3.distance(mypoint2);  
 System.out.println(distance);   
  
 System.out.print("Dist between Origin(0,0)& " + point3[0] + ", " + point3[1]);  
 System.out.print(" is ");  
 distance = mypoint3.distance();  
 System.out.println(distance);  
 }  
}