// A Java program to check if a given point lies inside a given polygon or not

package challenges;

public class Challenges {

static int INF = 10000;

static class Point

{

int x;

int y;

public Point(int x, int y)

{

this.x = x;

this.y = y;

}

};

// Given three colinear points p, q, r,

// the function checks if point q lies

// on line segment 'pr'

static boolean onSegment(Point p, Point q, Point r)

{

if (q.x <= Math.max(p.x, r.x) &&

q.x >= Math.min(p.x, r.x) &&

q.y <= Math.max(p.y, r.y) &&

q.y >= Math.min(p.y, r.y))

{

return true;

}

return false;

}

// To find orientation of ordered triplet (p, q, r).

static int orientation(Point p, Point q, Point r)

{

int val = (q.y - p.y) \* (r.x - q.x)

- (q.x - p.x) \* (r.y - q.y);

if (val == 0)

{

return 0; // colinear

}

return (val > 0) ? 1 : 2; // clock or counterclock wise

}

// The function that returns true if

// line segment 'p1q1' and 'p2q2' intersect.

static boolean doIntersect(Point p1, Point q1, Point p2, Point q2)

{

// Find the four orientations needed for

// general and special cases

int o1 = orientation(p1, q1, p2);

int o2 = orientation(p1, q1, q2);

int o3 = orientation(p2, q2, p1);

int o4 = orientation(p2, q2, q1);

// General case

if (o1 != o2 && o3 != o4)

{

return true;

}

// Special Cases

// p1, q1 and p2 are colinear and

// p2 lies on segment p1q1

if (o1 == 0 && onSegment(p1, p2, q1))

{

return true;

}

// p1, q1 and p2 are colinear and

// q2 lies on segment p1q1

if (o2 == 0 && onSegment(p1, q2, q1))

{

return true;

}

// p2, q2 and p1 are colinear and

// p1 lies on segment p2q2

if (o3 == 0 && onSegment(p2, p1, q2))

{

return true;

}

// p2, q2 and q1 are colinear and

// q1 lies on segment p2q2

if (o4 == 0 && onSegment(p2, q1, q2))

{

return true;

}

// Doesn't fall in any of the above cases

return false;

}

// Returns true if the point p lies

// inside the polygon[] with n vertices

static boolean isInside(Point polygon[], int n, Point p)

{

// There must be at least 3 vertices in polygon[]

if (n < 3)

{

return false;

}

// Create a point for line segment from p to infinite

Point extreme = new Point(INF, p.y);

// Count intersections of the above line with sides of polygon

int count = 0, i = 0;

do

{

int next = (i + 1) % n;

// Check if the line segment from 'p' to 'extreme' intersects with the line

// segment from 'polygon[i]' to 'polygon[next]'

if (doIntersect(polygon[i], polygon[next], p, extreme))

{

// If the point 'p' is colinear with line segment 'i-next', then check if it lies

// on segment. If it lies, return true, otherwise false

if (orientation(polygon[i], p, polygon[next]) == 0)

{

return onSegment(polygon[i], p,

polygon[next]);

}

count++;

}

i = next;

} while (i != 0);

// Return true if count is odd, false otherwise

return (count % 2 == 1); // Same as (count%2 == 1)

}

// Driver Code

public static void main(String[] args)

{

Point polygon1[] = {new Point(0, 0), new Point(10, 0),

new Point(10, 10),new Point(0, 10)};

int n = polygon1.length;

Point p = new Point(20, 20);

if (isInside(polygon1, n, p))

{

System.out.println("Yes,this point is Inside the polygon");

}

else

{

System.out.println("No,this point is Outside the polygon");

}

p = new Point(15, 15);

if (isInside(polygon1, n, p))

{

System.out.println("Yes,this point is Inside the polygon");

}

else

{

System.out.println("No,this point is Outside the polygon");

}

Point polygon2[] = {new Point(0, 0),new Point(5, 5),

new Point(5, 0)};

p = new Point(3, 3);

n = polygon2.length;

if (isInside(polygon2, n, p))

{

System.out.println("Yes,this point is Inside the polygon");

}

else

{

System.out.println("No,this point is Outside the polygon");

}

p = new Point(5, 5);

if (isInside(polygon2, n, p))

{

System.out.println("Yes,this point is Inside the polygon");

}

else

{

System.out.println("No,this point is Outside the polygon");

}

p = new Point(8, 10);

if (isInside(polygon2, n, p))

{

System.out.println("Yes,this point is Inside the polygon");

}

else

{

System.out.println("No,this point is Outside the polygon");

}

Point polygon3[] = {new Point(0, 0),new Point(10, 0),

new Point(10, 10),new Point(0, 10)};

p = new Point(1, -10);

n = polygon3.length;

if (isInside(polygon3, n, p))

{

System.out.println("Yes,this point is Inside the polygon");

}

else

{

System.out.println("No,this point is Outside the polygon");

}

Point polygon4[] = {new Point(0, 0),new Point(10, 0),

new Point(10, 10),new Point(0, 10)};

p = new Point(2, 3);

n = polygon4.length;

if (isInside(polygon4, n, p))

{

System.out.println("Yes,this point is Inside the polygon");

}

else

{

System.out.println("No,this point is Outside the polygon");

}

}

}

/\*OUTPUT-

No,this point is Outside the polygon

No,this point is Outside the polygon

Yes,this point is Inside the polygon

Yes,this point is Inside the polygon

No,this point is Outside the polygon

No,this point is Outside the polygon

Yes,this point is Inside the polygon

\*/