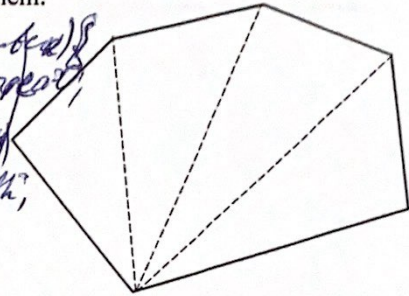


**Problem 3:** Write a Java function `public static double area(double[][] vertex)` that takes as its argument a 2-by-n array of a convex polygon's vertex coordinates `double[][] vertex` - the x coordinates in the first row and y coordinates in the second row. It returns polygon's area as follows:

1. Divides the polygon into triangles by connecting the **first** vertex with the  $n^{th}$  and  $(n+1)^{st}$  vertices;
2. Adds the areas of the constructed triangles using the formula  $area = \sqrt{p(p-a)(p-b)(p-c)}$ , where  $a, b$  and  $c$  are the sides and  $p = (a + b + c) / 2$ .

You may assume and use a method `double dist(double x1, double y1, double x2, double y2)` that takes as its arguments coordinates of two points and returns the distance between them.

```
(public static double area(double[][] vertex) {
    double first = vertex[0][0]; double second = vertex[1][0];
    for (int row = 0; row < vertex.length; row++)
    {
        for (int col = 0; col < vertex[0].length; col++)
        {
            if (col != 2)
            {
                double a =
```



```
public static double area(double[][] vertex) {
    double first = vertex[0][0]; double second = vertex[1][0];
    double area = 0;
    double a, b, c, p;
    for (int i = 0; i < 2; i++) {
    for (int j = 0; j < vertex[0].length; j++) {
        a = dist(first, second, vertex[0][j+1], vertex[1][j+1]);
        b = dist(first, second, vertex[0][j+2], vertex[1][j+2]);
        c = dist(vertex[0][j+1], vertex[1][j+1], vertex[0][j+2], vertex[1][j+2]);
        p = (a + b + c) / 2;
        area += Math.sqrt(p * (p - a) * (p - b) * (p - c));
    }
    return area;
}
```