6839 Cornering at Poles

You are invited to a robot contest. In the contest, you are given a disc-shaped robot that is placed on a flat field. A set of poles are standing on the ground. The robot can move in all directions, but must avoid the poles. However, the robot can make turns around the poles touching them.

Your mission is to find the shortest path of the robot to reach the given goal position. The length of the path is defined by the moving distance of the center of the robot. Figure H.1 shows the shortest path for a sample layout. In this figure, a red line connecting a pole pair means that the distance between the poles is shorter than the diameter of the robot and the robot cannot go through between them.

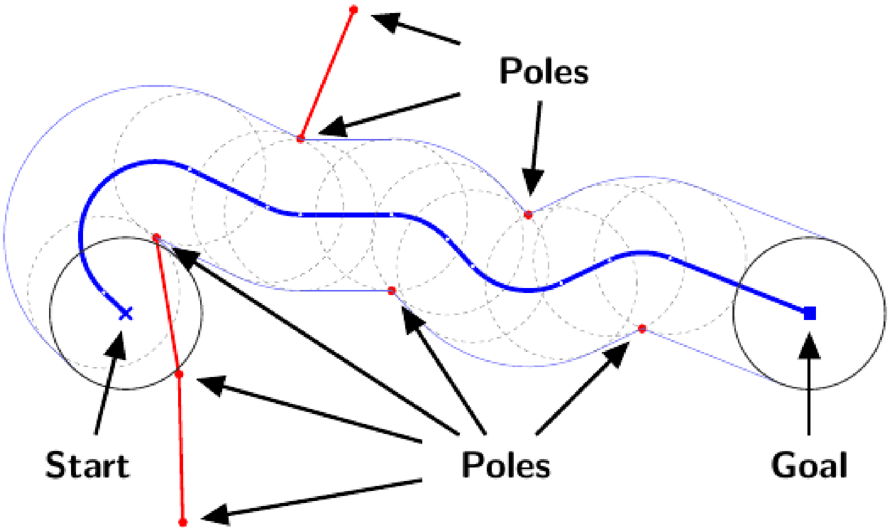

Figure H.1. The shortest path for a sample layout

Input

The input will contain several test cases, each of them formatted as follows.

$N\ G_x\ G_y$
$x_1\ y_1$
$\vdots$
$x_N\ y_N$

The first line contains three integers. N represents the number of the poles ($1 \leq N \leq 8$). (G_x, G_y) represents the goal position. The robot starts with its center at (0, 0), and the robot accomplishes its task when the center of the robot reaches the position (G_x, G_y). You can assume that the starting and goal positions are not the same.

Each of the following N lines contains two integers. (x_i, y_i) represents the standing position of the i-th pole. Each input coordinate of (G_x, G_y) and (x_i, y_i) is between -1000 and 1000, inclusive. The radius of the robot is 100, and you can ignore the thickness of the poles. No pole is standing within a 100.01 radius from the starting or goal positions. For the distance $d_{i,j}$ between the i-th and j-th poles $(i \neq j)$, you can assume $1 \leq d_{i,j} < 199.99$ or $200.01 < d_{i,j}$.

Figure H.1 shows the shortest path for Sample Input 1 below, and Figure H.2 shows the shortest paths for the remaining Sample Inputs.

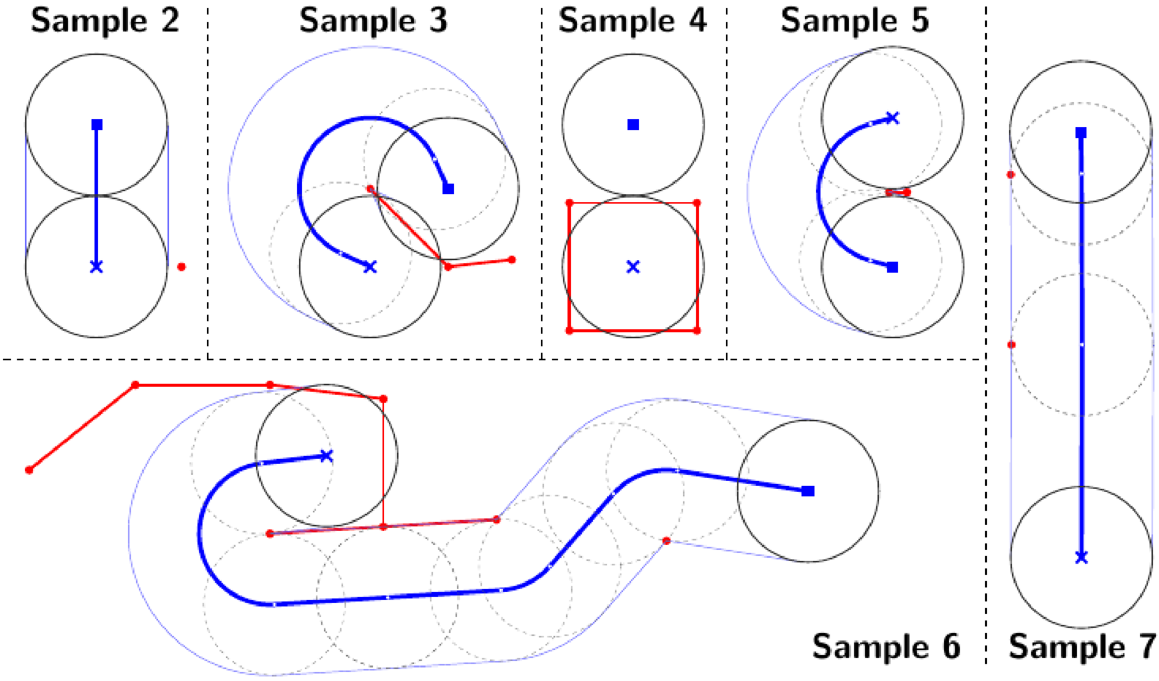

Figure H.2. The shortest paths for the sample layouts

Output

For each test case, output the length of the shortest path to reach the goal. If the robot cannot reach the goal, output '`0.0`'. The output should not contain an error greater than 0.0001.

Sample Input

```
8 900 0
40 100
70 -80
350 30
680 -20
230 230
300 400
530 130
75 -275
1 0 200
120 0
3 110 110
0 110
110 0
200 10
4 0 200
90 90
-90 90
-90 -90
90 -90
2 0 -210
20 -105
-5 -105
```

```
8 680 -50
80 80
80 -100
480 -120
-80 -110
240 -90
-80 100
-270 100
-420 -20
2 -1 600
-99 300
-100 540
```

Sample Output

```
1210.99416
200.0
476.95048
0.0
325.81116
1223.53071
600.01216
```