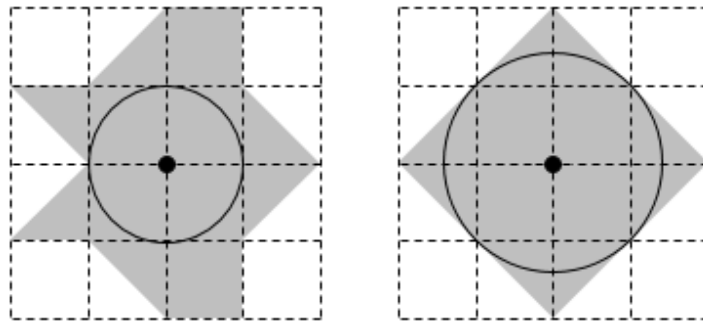


A: ACIS, A Contagious vIruS

Source file name: `acis.c`, `acis.cpp`, `acis.java`, or `acis.py`

Author: A. Sotelo

Scientists from REDIS (*REsearch of DISeases*), a famous investigation center in Raccoon City, accidentally caused the mutation of a very contagious virus known as ACIS (*A Contagious vIruS*), just when they were manipulating ACIS' DNA. Raphael, the main researcher at REDIS, was infected with ACIS while he was treating inoculated rats. After that, all persons at REDIS were infected in less than an hour. Immediately he discovered the issue, Raphael contacted the Major, who decided to quarantine the largest possible circular region centered at REDIS that is totally inside Raccoon City, whose boundaries are described with a polygon.



The Major wants to know the maximum radius of such circular region. Can you help him?

Input

The input consists of several test cases. The first line of a test case contains a single integer N indicating the number of vertices of the polygon describing the boundaries of Raccoon City ($3 \leq N \leq 16$). The second line of a test case contains two blank-separated integers x_R and y_R ($0 \leq x_R \leq 50$, $0 \leq y_R \leq 50$) indicating the position (x_R, y_R) where REDIS is located. Then follow N lines: line i contains exactly two blank-separated integers x_i and y_i , where (x_i, y_i) is the position of the i -th vertex of the polygon describing the boundaries of Raccoon City ($0 \leq x_i \leq 50$, $0 \leq y_i \leq 50$). You may assume that there are not two vertices located at the same position, and that REDIS is located inside the polygon excluding its boundaries. The input ends with a line containing a single asterisk (*).

The input must be read from standard input.

Output

For each test case, print a single line with a number indicating the radius of the largest possible circular region centered at REDIS that is totally inside Raccoon City. The answer should be formatted and approximated to three decimal places. The floating point delimiter must be '.' (i.e., the dot). The rounding applies towards the *nearest neighbor* unless both neighbors are equidistant, in which case the result is rounded up (e.g., 78.3712 is rounded to 78.371; 78.5766 is rounded to 78.577; 78.3745 is rounded to 78.375, etc.).

The output must be written to standard output.

Sample Input	Sample Output
12	1.000
2 2	1.414
0 1	
1 1	
2 0	
3 0	
3 1	
4 2	
3 3	
3 4	
2 4	
1 3	
0 3	
1 2	
4	
2 2	
0 2	
2 0	
4 2	
2 4	
*	