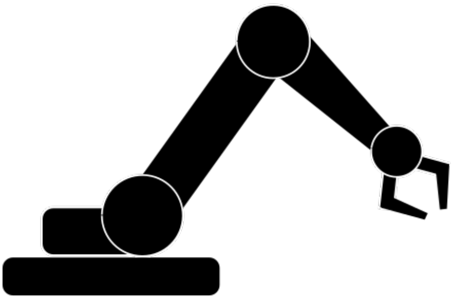


João wants to join the robotic football team of his university. However, since he knows little about robotics and mathematics, he decided to build a 2-dimensional robotic arm to bootstrap his knowledge.

The robotic arm is composed of N segments of various lengths. The segments can form any angle between them, including configurations that make it appear to self-intersect when viewed from above. The robotic arm works great, but it is not trivial to position the arm's tip as close as possible to given x, y target coordinates with so many joints to control. Can you help João?

Given the robotic arm description and target coordinates relative to the arm's origin, calculate a configuration that places the arm's tip as close as possible to the target.

Input

The input file contains several test cases, each of them as described below.

The first line contains N, the number of segments composing the robotic arm. N lines follow, each with an integer L_i describing the length of the i-th segment from the fixed point until the arm's tip. There is one more line with 2 integers: the x, y coordinates of the target point to reach.

Constraints

$1 \leq N \leq 20$	Number of segments in the robotic arm
$1 \leq L_i \leq 1\,000$	Length of the i-th segment
$-20\,000 \leq x, y \leq 20\,000$	Target coordinates to attempt to reach

Output

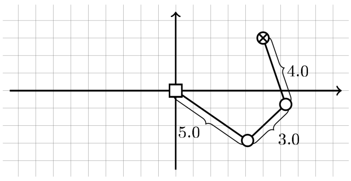
For each test case, the output must follow the description below.

The output should contain N lines, each containing two real numbers x_i, y_i indicating the coordinates of the tip of the i-th segment.

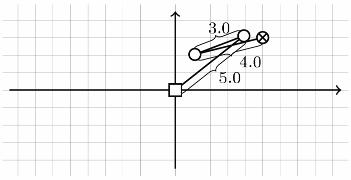
The length of the i-th segment computed from the solution and input L_i may not differ by more than 0.01. Similarly, the absolute error between the solution's distance to the target and the minimum possible distance to the target cannot exceed 0.01.

Note that, in general, there are many solutions. Your program may output any of them.

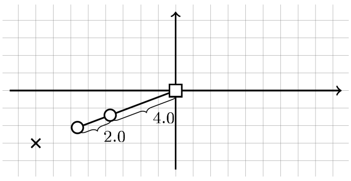
Sample Explanation: The pictures below show 2 different solutions for the first sample input and 1 solution for second.



Sample Output 1a



Sample Output 1b



Sample Output 2

Sample Input

```
3
5
3
4
5 3
2
4
2
-8 -3
```

Sample Output

```
4.114 -2.842
6.297 -0.784
5.000 3.000
-3.745 -1.404
-5.618 -2.107
```