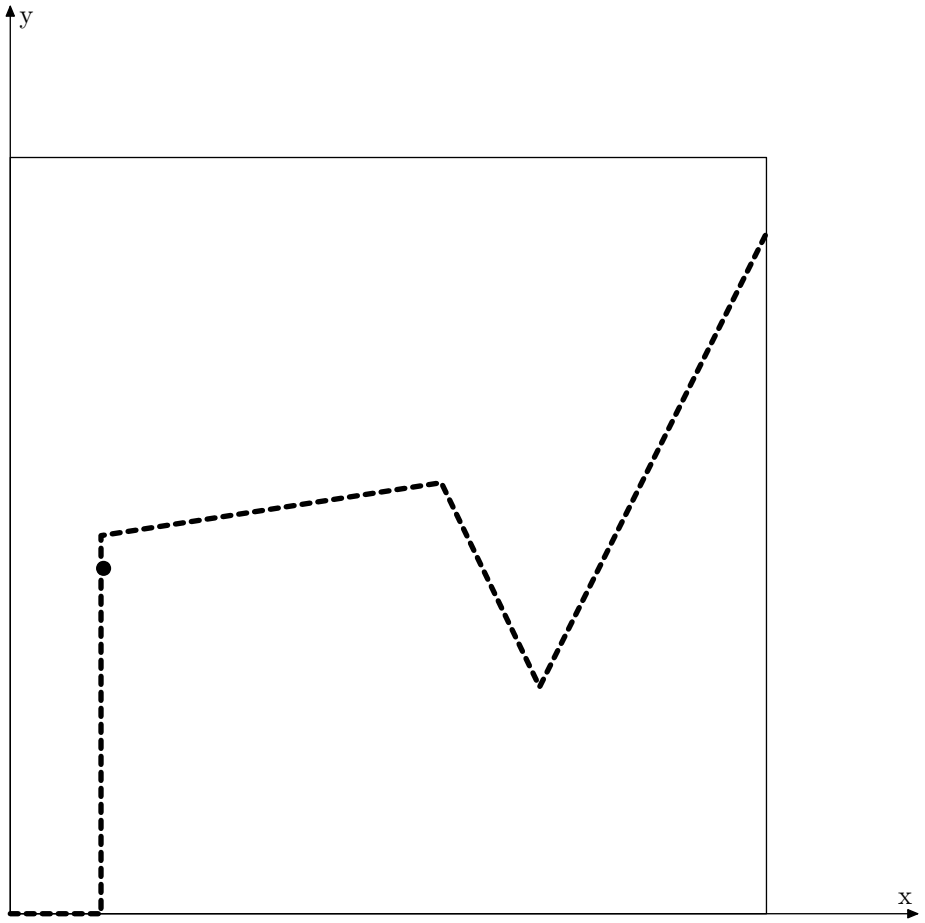
Problem B. Mond

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

You're part of a Moon expedition, and you are now orbiting the Moon preparing for landing. The optimal landing point has been precisely calculated, and a beacon was previously dispatched to that point to guide your landing. However, it turned out that the beacon is too weak and its signal doesn't reach the orbit — more precisely, the signal can only be received if one is within a sphere with radius one kilometer from the beacon, which means within a circle with radius one kilometer from the beacon if one is on the Moon's surface.

You've identified an area where the beacon can be as a 100 kilometer by 100 kilometer square on the side of a crater, but it can be anywhere within that square. Your only means of figuring out the precise location of the beacon is to send unmanned Moon landers to the surface that would try to catch its signal.

Each Moon lander does not have a link to the orbit, and has to be pre-programmed. Its program is a sequence of waypoints on the Moon's surface, and it moves between waypoints in straight lines. Because the crater is quite steep, the lander can never decrease its x-coordinate. In other words, the x-coordinates of consecutive waypoints must be non-decreasing (they can be identical), the x-coordinate of the first waypoint must be equal to 0 (the high side of the crater), and the x-coordinate of the last waypoint must be equal to 100 (the low side of the crater). The y-coordinates of waypoints can be arbitrary numbers between 0 and 100. Coordinates are expressed in kilometers.



When moving between the waypoints, the Moon lander listens for the beacon. If at any point the lander is within 1 kilometer from the beacon, it registers this fact. After its entire trip is done, the Moon lander

transmits one single bit of information to the orbit: was it within 1 kilometer from the beacon at any point along its route?

This bit still doesn't allow you to determine the beacon's location well. However, by sending several Moon landers with different programs, you can actually figure out the beacon's location very precisely. You need to do so. You're allowed to base the program for each Moon lander on the results transmitted by the previous Moon landers.

You can send at most 60 Moon landers, and need to determine the beacon's location with error not exceeding one millimeter (10^{-6} kilometers).

Interaction Protocol

Initially your program should not read anything from the standard input.

In order to send a Moon lander, your program should print its route. The route starts with the number n of waypoints, which must not exceed 250, followed by n pairs of floating-point numbers: the x- and y- coordinates of waypoints. All coordinates must be between 0 and 100, inclusive, the first x-coordinate must be exactly 0, the last x-coordinate must be exactly 100, and the x-coordinates must be non-decreasing. These conditions will be checked without any additional tolerances. It's OK for waypoints to coincide. Remember to flush standard output after printing your route.

You can then read the Moon lander's trip result from the standard input as a single integer: 0 if the beacon was never within 1 kilometer from the lander during the trip, or 1 if it was. The interactor will use at least double-precision floating-point numbers to make this comparison, and will not have any additional tolerances when comparing, so in case the lander passes within exactly 1 kilometer from the beacon, your program can receive either result.

When you've determined the location of the beacon precisely enough, you should output the number 0 followed by two floating-point numbers: your guess for the location of the beacon. Your solution will be accepted if both coordinates are between 0 and 100, inclusive, and if the distance between this location and the actual location of the beacon does not exceed 10^{-6} kilometers. This comparison will be performed using at least double-precision floating-point numbers without any additional tolerances, too. Remember to flush the standard output and terminate your program gracefully after outputting your guess.

You can send at most 60 Moon landers.

On the sample case, the beacon is in the location shown in the sample output, but your solution will pass the sample case simply if it sends at most 60 Moon landers, and then outputs any location with both coordinates between 0 and 100, inclusive.

There are 50 non-sample cases in this problem. Both coordinates of the location of the beacon for each case are fixed, and were generated independently and uniformly at random.

Example

standard input	standard output
1 1	2 0.0 45.0 100.0 46.0 6 0.0 0.0 12.0 0.0 12.0 50.0 57.0 57.0 70.0 30.0 100.0 90.0 0 12.34 45.67

Note

The picture above corresponds to the second Moon lander in the sample output.