

Union Laser

Time limit: 1.0s **Memory limit:** 256M

Doctor Y, a leading expert in the field of boxology, is investigating the properties of boxen with his expensive super-precision laser. In particular, he plans to position the laser inside the union of a bunch of axis-aligned boxen, and see where the beam collides with itself.

Since Doctor Y blew all his cash on the laser, he can't actually afford to purchase boxen to conduct his experiment - instead, he will simulate it on his old computer, which uses an integer grid system. He will give his computer an integer N , the number of boxen ($1 \leq N \leq 10\,000$), as well as their descriptions. Each box is described by 4 integers - x_1, y_1, x_2 and y_2 , such that $(-3\,000 \leq x_1, x_2 \leq 3\,000)$ and $(-3\,000 \leq y_1, y_2 \leq 3\,000)$, where (x_1, y_1) and (x_2, y_2) are any two opposite corners of the box. Note that some boxen may have an area of 0 - these do not contribute to the union. He will also input the location of the laser (A, B) such that $(-3\,000 \leq A, B \leq 3\,000)$ and such that it will be positioned strictly within the box union. The laser points down and right at a 45° angle.

As it turns out, boxen are made of a perfectly reflective material (a fact which Doctor Y will discover upon the completion of his experiment) - as such, whenever the laser beam hits the edge of the box union, it bounces off. For example, if it hits a vertical edge from the left, it bounces to the right, with the up-down direction preserved. If it hits a corner head-on, it will reverse direction, hence colliding with itself right at the corner. Normally, light travels quite fast, but due to the mysterious properties of boxen, the speed of light when inside a box union is only $\sqrt{2}$ units/second.

Doctor Y plans to use his computer to simulate the paths of the lasers and see when and where they first collide, but there's one problem - the only program on his computer is C++! Offering non-cafeteria food, he lures a desperate Waterloo student into his lab, where he forces him (or her?) to write the laser simulation program. That's you!

Input Specification

Line 1: A single integer - N

Line 2: Two integers - A and B

The next N lines: Four integers - x_1, y_1, x_2 and y_2

Output Specification

If the laser beam never collides with itself of the laser, simply output `Time limit exceeded`.

If the laser beam collides with the actual laser before it collides with itself, the first line of output should consist of a single number - the amount of time that passes before the laser beam collides with the laser, in seconds. The second line should read `Broken equipment`.

Otherwise, the first line of output should consist of a single number - the amount of time that passes before the laser beam first collides with itself, in seconds. The second line should contain a pair of numbers - the X and Y coordinates of where this takes place, respectively.

Each number should be rounded off to 1 decimal place.

Sample Input

```

7
0 4
-1 -1 1 5
0 -2 4 2
0 2 1 -4
5 -4 2 -1
5 2 7 4
3 -5 4 -4
0 -6 3 -4

```

Sample Output

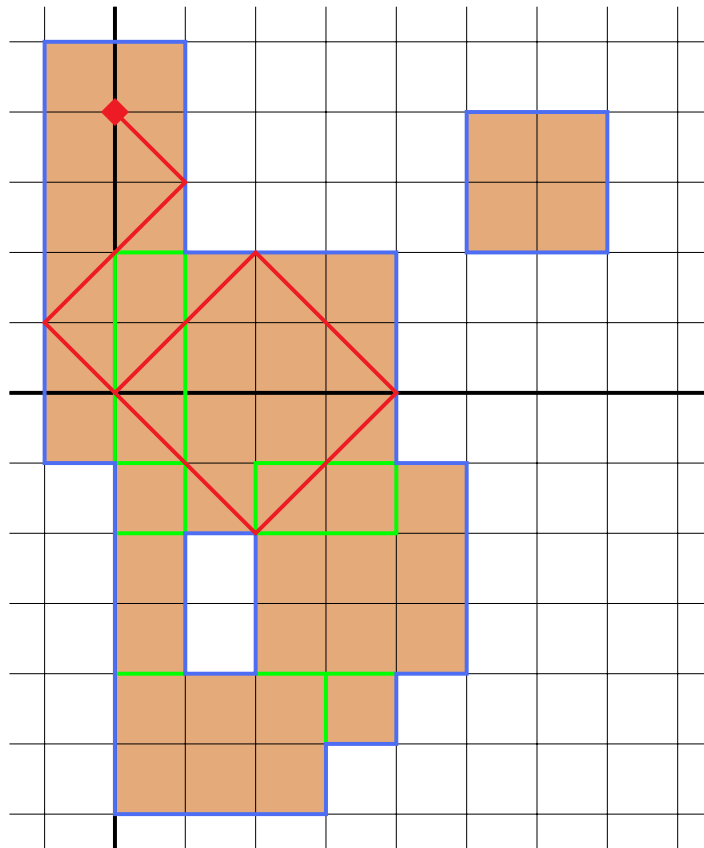
```

12.0
0.0 0.0

```

Explanation

The grid looks like this:



The filled-in squares represent squares that are part of at least one box - together, they make up the box union. Note that the union can have holes in it, and that it can be disjoint.

The blue lines denote the boundaries of the union - these are the lines that the laser can bounce off of.

The green lines are edges of boxes that do not contribute to the union - the laser can go through these freely.

The red diamond is the position of the laser, and the red line is the path that the laser beam follows.

Note that when it bounces off the corner at $(2, -2)$, since it didn't hit it head-on, it is the same as bouncing off of a horizontal edge.

Following the path of the laser beam, it can be seen that it collides with itself at $(0, 0)$. Before this, it has travelled $12\sqrt{2}$ units, which takes exactly 12 seconds.

More Examples

If the laser were positioned at $(0, 3)$, the output should be:

```
12.0  
0.0 -1.0
```

If the laser were positioned at $(0, 1)$, the output should be:

```
5.0  
5.0 -4.0
```

If the laser were positioned at $(0, 0)$, the output should be:

```
8.0  
Broken equipment
```