Problem F: Fair Fruitcake Fragmenting

Time limit: 3 seconds

Frida's birthday is just coming up, and as her best friend, you obviously baked a cake for her. Since you know that Frida loves rotational symmetry, you thought to bake a cake that looks the same from above when rotated by $180°$. Of course, you could have simply baked a boring round cake, but without a perfectly round cake tin, this sounds easier than done. Therefore, you decided to bake a cake whose shape can be described by straight line segments.

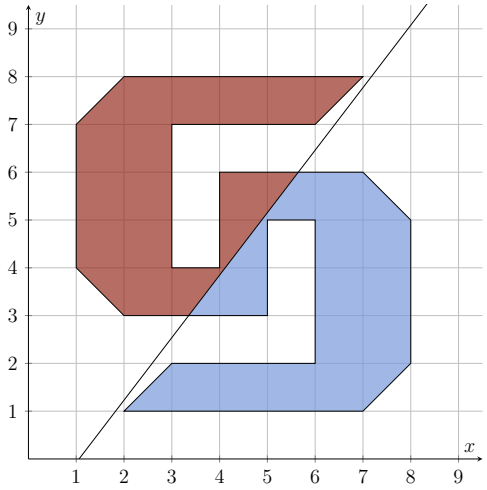


Figure F.1: Visualization of Sample Input 2. The swirly cake looking like an *S* can be cut into the red and blue part with a single cut.

However, after you are done with your cake, you notice that you also want to cut the cake into two equal pieces, one for Frida and one for yourself. More precisely, you wonder if it is possible to cut the cake along an infinite line such that it splits into exactly two parts of equal weight. You can assume that the cake has uniform density and height.

Input

The input consists of:

- One line containing an *even* integer n ($4 \leq n \leq 10^5$), the number of points needed to describe the cake's shape.
- n lines, each containing two integers x, y ($0 \leq x, y \leq 10^6$), the x and y coordinates of a point on the border of the cake's shape.

The following additional guarantees are given for the shape of the cake:

- The cake has a $180°$ rotational symmetry.
- The points are given in counterclockwise order.
- No three consecutive points are collinear.
- The shape is simple (no segments intersect and only consecutive segments touch at their ends).

Output

Output two different points on the desired line as $x_1/c_1 \;\; y_1/d_1 \;\; x_2/c_2 \;\; y_2/d_2$, where $|x_i|$, $|y_i|$, $|c_i|$ and $|d_i|$ are integers and at most 10^9, and x_i/c_i is the first coordinate of point i and y_i/d_i is the second ($1 \leq i \leq 2$). If the denominator of a fraction is 1 you may output only the numerator. Fractions do not have to be reduced. If there is no such line, output "`impossible`" instead.

It can be shown that if there is a line as desired, it is possible to represent it in the given format.

Sample Input 1

```
4
0 0
2 0
2 2
0 2
```

Sample Output 1

```
1 1 1337/42 3141/1000
```

Sample Input 2

```
20
7 1
8 2
8 5
7 6
4 6
4 4
3 4
3 7
6 7
7 8
2 8
1 7
1 4
2 3
5 3
5 5
6 5
6 2
3 2
2 1
```

Sample Output 2

```
11 13 -2 -4
```

Sample Input 3

```
10
11 5
10 2
12 6
2 2
7 3
1 1
2 4
0 0
10 4
5 3
```

Sample Output 3

```
impossible
```