Drawing Borders

Somewhere in the great North American plains live the tribes of chiefs *Blue Eagle*, *Red Beaver*, and *Green Serpent*. Their population is scattered over numerous villages all over the land and conflict arises whenever members of different tribes meet while traveling across the plains.

To put an end to the constant animosities the chiefs have decided that the land should be divided between the tribes so that they can avoid each other when moving between villages belonging to the same tribe. More precisely, they want to construct two border fences – thus dividing the land into three regions – such that two villages lie in the same region precisely when they belong to the same tribe.

The villages are represented by points in the Euclidean plane that are colored blue, red or green, depending on the tribe, and the fences should be drawn in the form of two polygons. The polygons may not touch or intersect themselves or each other and none of the points may lie on their boundary. (Make sure to read the constraints in the Output section!)

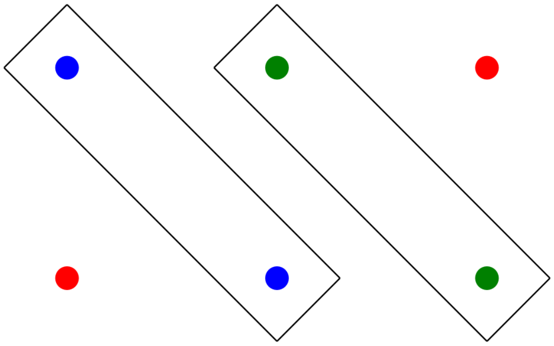

Figure 1: Illustration of the sample.

Input

The input consists of:

- one line with an integer n ($3 \leq n \leq 100$), the number of villages.
- n lines, each with three integers x, y, c ($-1\,000 \leq x, y \leq 1\,000, 1 \leq c \leq 3$), representing a village at coordinates (x, y) of color c ($1 =$ blue, $2 =$ red, $3 =$ green). All positions are unique. There is at least one village of each color.

Output

If there is no solution, print `impossible`. Otherwise, print the two polygons in the following format:

- one line with an integer m ($3 \leq m \leq 1\,000$), the number of vertices of the polygon.
- m lines, each with two real numbers x, y ($-3\,000 \leq x, y \leq 3\,000$), the vertices of the polygon in either clockwise or counter-clockwise order. The numbers may be given with up to five decimal places (additional places will be rounded off).

Sample Input 1

```
6
0 0 2
0 1 1
1 0 1
1 1 3
2 0 3
2 1 2
```

Sample Output 1

```
4
-0.3 1.0
1.0 -0.3
1.3 0.0
0.0 1.3
4
0.7 1.0
2.0 -0.3
2.3 0.0
1.0 1.3
```