Problem B. LaLa and Magic Circle (LaLa Version)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

LaLa has a pile of magic circles in her laboratory.

A magic circle can be represented as a simple polygon drawn with special ink, and is **usable** if and only if it is convex. i.e. all of its internal angles are equal or less than π.

LaLa plans to turn every magic circle into a usable one. However, it may lose all its magical power if done incorrectly. Thankfully, LaLa has the perfect magical tool for that.

The tool works as follows. When you toss in a magic circle, if it's usable, it reports that it is. Otherwise, it takes two distinct points u and v such that

- u and v lie on the boundary of the convex hull of the magic circle, and
- none of the points on the path from u to v through the boundary of the magic circle in counterclockwise order lie on the boundary of the convex hull of the magic circle, except for u and v.

And then it rotates the u-v path by π around the midpoint of u and v. In other words, for each point w on the u-v path, w becomes $u + v - w$ where the addition is done coordinate-wise over the two dimensional coordinate system over the paper the magic circle is drawn on. Note that the result of this modification is also a simple polygon.

LaLa got annoyed by how long it takes to convert them. In order to finish and take a nap ASAP, LaLa made the following observations.

1. A magic circle always turns into a usable one within a finite number of applications of the tool.
2. The set of points on the final magic circle is independent of the intermediate modifications. In other words, the shape and location of the final magic circle is a function of the initial magic circle.

Therefore, LaLa doesn't have to manually turn magic circles into usable ones with the tool. Instead, LaLa will compute the usable magic circle that can be made from the initial magic circle by a sequence of modifications by the tool and modify it in one go.

Write a program to help LaLa compute the final magic circle so that she can go take a nap.

Input

The input is given in the following format:

$$\begin{array}{ll} N & \\ x_0 & y_0 \\ x_1 & y_1 \\ & \vdots \\ x_{N-1} & y_{N-1} \end{array}$$

where the initial magic circle is the union of N line segments connecting points (x_i, y_i) and $(x_{(i+1 \bmod N)}, y_{(i+1 \bmod N)})$ for all integers $0 \le i < N$.

The input satisfies the following constraints:

- All numbers in the input are integers.

- $3 \le N \le 100\,000$
- $0 \le x_i \le 300\,000$ and $0 \le y_i \le 300\,000$ for all integers $0 \le i < N$.
- $x_i \ne x_j$ or $y_i \ne y_j$ for all integers $0 \le i < j < N$.
- The input defines a counterclockwise traversal of the boundary of a simple polygon. In particular, it does not intersect with itself.

Output

The output should be in the following format:

$$\begin{array}{ll} M & \\ z_0 & w_0 \\ z_1 & w_1 \\ & \vdots \\ z_{M-1} & w_{M-1} \end{array}$$

where the final usable magic circle is the union of M line segments connecting points (z_i, w_i) and $(z_{(i+1 \bmod M)}, w_{(i+1 \bmod M)})$ for all integers $0 \le i < M$,

The output should satisfy the following constraints:

- All the numbers in the output are integers.
- The point (z_0, w_0) is lexicographically smaller than the point (z_i, w_i) for all integers $1 \le i < M$. i.e. $(z_0 < z_i)$ or $(z_0 = z_i$ and $w_0 < w_i)$.
- Points (z_i, w_i), $(z_{(i+1 \bmod M)}, w_{(i+1 \bmod M)})$, and $(z_{(i+2 \bmod M)}, w_{(i+2 \bmod M)})$ are not collinear for all integers $0 \le i < M$.
- The output defines a counterclockwise traversal of the boundary of a convex polygon.

It can be proved that the output satisfying the above constraints is unique.

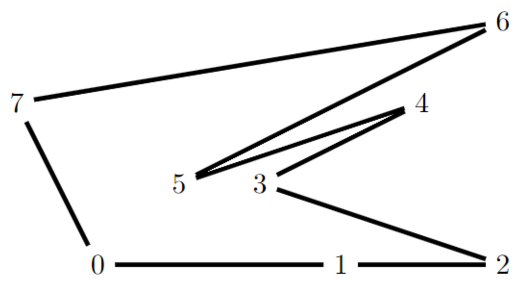
Example

standard input	standard output

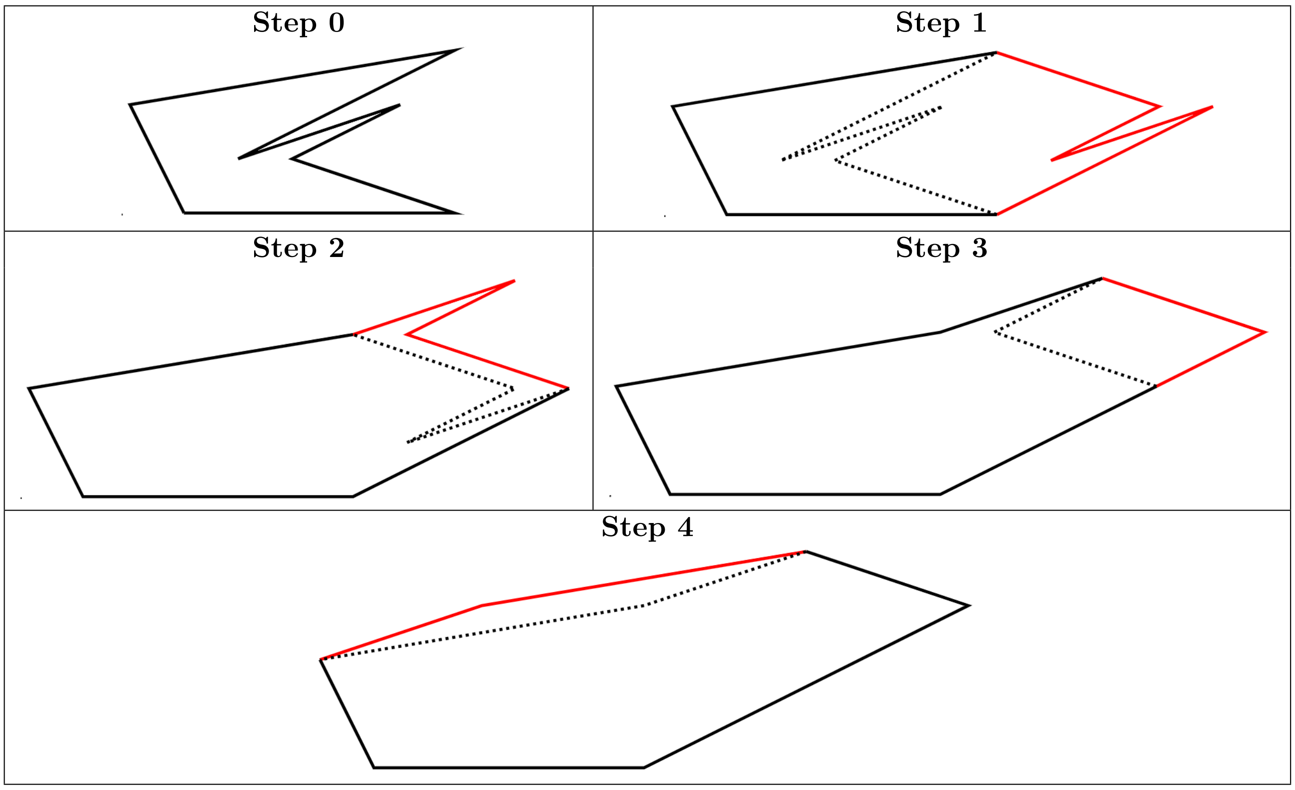
8
1 0
4 0
6 0
3 1
5 2
2 1
6 3
0 2
	6
	0 2
	1 0
	6 0
	12 3
	9 4
	3 3

Note

The following illustrates the initial magic circle for the first sample.



The following illustrates the sequence of usage of the tool to make it usable. The dotted part of the boundary is the path modified by the tool, which becomes the red part after the modification.


Step 0
Step 1
Step 2
Step 3
Step 4