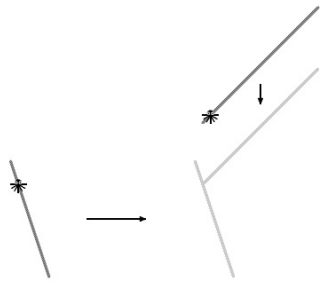
Problem A. Branches

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Two branches are floating on the surface of a lake. Assuming that branches are line segments that move with constant speeds, determine the time until the branches intersect.



Input

The input contains 12 integers $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4, v_{1x}, v_{1y}, v_{2x}, v_{2y}$. The endpoints of the first segment are (x_1, y_1) and (x_2, y_2), the endpoints of the second segment are (x_3, y_3) and (x_4, y_4), the velocity vector of the first segment is (v_{1x}, v_{1y}), the velocity vector of the second segment is (v_{2x}, v_{2y}). All numbers do not exceed 10^4 by absolute value. It is guaranteed that the segments do not have common point at the initial moment, and that the segments have non-zero length.

Output

Print the time until the first moment when the branches have a common point with relative or absolute precision 10^{-4}. If the branches never touch each other, print -1.

Examples

standard input	standard output
0 0 -1 3 4 4 7 7 3 0 0 -1	1.6000000000
0 0 -1 3 4 4 7 7 1 0 0 -3	-1

Problem B. Avoid the circle!

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You have travel from a point A to a point B in the plane. There is a large circular hole in the plane with radius R centered at the point C. Find the length of the shortest path from A to B avoiding the hole.

Input

The input file contains coordinates of A and B, followed by coordinates of C, followed by the radius R. All coordinates are integers not exceeding 32 000 by absolute value. The radius R is a positive integer not exceeding 32 000.

The points A and B are not inside the circle, but may be located on its border.

Output

Print a single real number — the answer to the problem.

Examples

standard input	standard output
0 0 0 1 10 10 1	1.000000
5 0 0 5 0 0 5	7.853982
-5 0 5 0 0 0 3	11.861007

Problem C. Convex Hull

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Find the convex hull of a set of points in the plane.

Input

The first line contains the number of points n ($3 \le n \le 200\,000$). The next n lines describes the given points, two integer coordinates of a point per line. All coordinates do not exceed 10^9 by absolute value. It is guaranteed that the points do not belong to a common straight line. Some of the points may coincide.

Output

On the first line print the number of vertices of the convex hull. On the second line print space-separated indices of vertices of the convex hull in the counter-clockwise order.

On the third line print the perimeter length, and on the fourth line print the area of the convex hull.

The perimeter has to be within 10^{-9} absolute or relative tolerance. The area has to be absolutely precise.

Example

standard input	standard output
5 0 0 1 1 2 2 1 0 0 1	4 3 5 1 4 6.472135954999580000 0 2.0

Problem D. Sea Battle

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Flatlandians are battling against an enemy ship. The ship is a convex polygon with n vertices. Flatlandians have fired m rockets at the ship and determined the points where the rockets landed. A rocket hits the ship if it lands inside the ship or on its border. At least k rockets must have hit the ship in order to destroy it. Find out if the ship was destroyed or not.

Input

The first line contains integers n, m, k ($3 \leqslant n \leqslant 10^5$, $0 \leqslant k \leqslant m \leqslant 10^5$). The next n lines contain coordinates of vertices of the ship in counter-clockwise order. The next m lines contain coordinates of rocket hits. All coordinates are integers that do not exceed 10^9 by absolute value.

Output

Print "`YES`" if the ship was destroyed, and "`NO`" otherwise.

Example

standard input	standard output
5 4 2 1 -1 1 2 0 4 -1 2 -1 -1 -2 -1 1 -1 0 1 2 3	YES

Problem E. Gears

Input file:	`stdin`
Output file:	`stdout`
Time limit:	2 seconds
Memory limit:	256 megabytes

There are two polygons on the plane, A and B. Polygon A rotates around point P, and polygon B rotates around point Q. Each polygon rotates with the constant rotational speed in the clockwise direction around its point, the rotational speed values of the polygons' rotation are equal.

Your task is to determine if there will be a collision between polygons. A collision is a situation when the polygons have at least one common point.

It is guaranteed that at the moment 0 the polygons A and B do not intersect and no polygon is fully contained inside another one.

Note that:

- the polygons are not necessarily convex;
- points P and Q can be located on the border of or outside their polygons.

Input

The first line contains space-separated coordinates of point P.

The second line contains a single integer n ($3 \le n \le 1000$) — the number of vertices of polygon A.

Each of the next n lines contains two space-separated integers — the coordinates of the corresponding vertex of polygon A.

The next line is empty.

Then follow space-separated coordinates of point Q.

The next line contains a single integer m ($3 \le m \le 1000$) — the number of vertices of polygon B. Next m lines contain the coordinates of the vertices of the polygon B.

The vertices of both polygons are listed in the counterclockwise order. Coordinates of all points are integers, their absolute values don't exceed 10^4.

Output

Print "`YES`", if the collision takes place and "`NO`" otherwise (don't print the quotes).

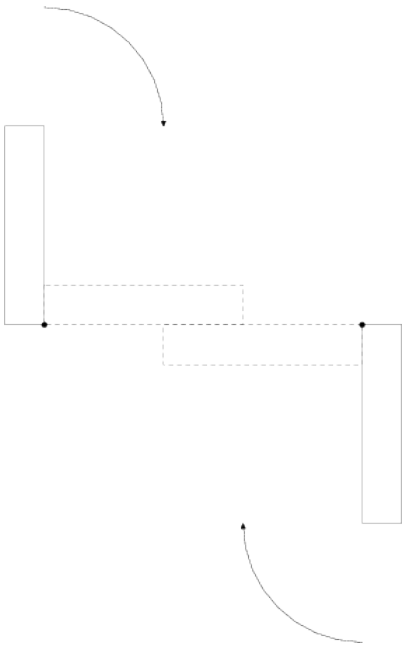
Examples

stdin	stdout
1 0 4 0 0 1 0 1 5 0 5 9 0 4 9 0 9 -5 10 -5 10 0	YES
0 0 3 1 0 2 -1 2 1 0 0 3 -1 0 -2 1 -2 -1	NO

Note

A polygon is a closed polyline that doesn't intersect itself and doesn't touch itself.

Picture to the first sample:



Problem F. Closest Points

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

You are given n points in the plane. Find a pair of points at the smallest distance.

Input

The first line contains an integer n ($2 \le n \le 10^5$).

The next n lines contain coordinates of the points. It is guaranteed that all coordinates are integers that do not exceed 10^9 by absolute value, and that all points are distinct.

Output

Print coordinates of two closest points. If there are several possible pairs, print any of them.

Example

standard input	standard output
4 0 0 0 1 1 1 1 0	0 0 0 1

Problem G. Largest Quad

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	256 megabytes

There are n points in the plane. Find the largest area of a quadrilateral with vertices in four of the given points. The quadrilateral should not have self-intersections.

Input

The first line contains an integer n ($4 \le n \le 5\,000$).

The next n lines contain coordinates of the given points. All points are distinct, and all coordinates do not exceed 10^8 by absolute value.

Output

Print one real number — the answer to the problem with exactly one decimal place. Your answer should be absolutely precise. It is guaranteed that the answer will be positive.

Example

standard input	standard output
5 0 0 2 0 0 2 2 2 1 1	4.0

Problem H. Highways

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

There are n highways which can be modeled as infinite straight lines passing near a city. In the city there are m subway stations. We say that a highway passes through the city if at least one subway station lies on the highway, or there are two stations on opposite sides of the highway.

Determine which highways pass through the city.

Input

The first line contains two integers n and m ($1 \le n, m \le 10^5$).

The next n lines describe the highways. Each of these lines contain three integers a, b and c and describe a highway given by the line equation $ax + by + c = 0$ ($|a|, |b|, |c| \le 10^9$).

The next m lines describe the subway stations. Each station is described by its integer coordinates x and y ($|x|, |y| \le 10^9$).

Output

The first line should contain the number of highways that pass through the city. The next line should contain indices of these highways in ascending order. The highways are numbered 1 through n as listed in the input.

Example

standard input	standard output
4 2 0 1 0 1 0 1 1 1 0 1 1 -1 0 0 2 0	3 1 3 4

Problem I. Most Far

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

There are n points in the plane. Your task is to process the following queries:

- `get a b` — find the maximum value $a \cdot x + b \cdot y$ over all points in the set.
- `add x y` — add a new point to the set.

Input

The first line contains an integer n ($1 \le n \le 10^5$). The next n lines contain coordinates of the given points.

The next line contains an integer m ($1 \le m \le 10^5$) — the number of queries. Next m lines describe the queries in the format described above. All coordinates and parameters of queries are integers not exceeding 10^9 by absolute value.

Output

For each `get` query print one integer — the answer to the query.

Example

standard input	standard output
3 0 0 1 0 0 1 10 get 1 1 get -1 -1 get 1 -1 get -1 1 add 2 2 add -2 -2 get 1 1 get -1 -1 get 1 -1 get -1 1	1 0 1 1 4 4 1 1

Problem J. Circle Cover

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Find the smallest circle covering a given collection of points in the plane.

Input

The first line contains the number of points n ($1 \le n \le 10^5$).

The next n lines contain coordinates of the points. All coordinates are integer that do not exceed 10^6 by absolute value.

Output

On the first line print the coordinates of the center of the circle. On the second line print the radius of the circle. Your answer should be correct within 10^{-6} absolute or relative precision.

Example

standard input	standard output
3 0 2 0 0 2 0	1.00000000000000000000 1.00000000000000000000 1.41421356237309510000