Problem A. Vectors

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

You are given two non-zero vectors in the plane. Compute:

- Lengths of both vectors (two real numbers).
- The sum of the two vectors.
- Dot-product and cross-product of vectors.
- Area of the triangle with two sides equal to the given vectors.

Input

The two lines of input contain four integers — coordinates of the start point and the end point of the respective vector. All coordinates do not exceed 10^4 by absolute value.

Output

On each line print the answer to the respective question. All numbers have to be precise within 10^{-6} relative or absolute tolerance.

Examples

stdin	stdout
5 1 2 6 1 1 7 8	5.830951895 9.219544457 3.000000000 12.000000000 17.000000000 -51.000000000 25.500000000

Problem B. 16

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

You are given four points A, B, C, D.

Compute

- The distance from the point A to the point C.
- The distance from the point A to the segment CD.
- The distance from the point A to the half-infinite ray CD.
- The distance from the point A to the line CD.
- The distance from the segment AB to the point C.
- The distance from the segment AB to the segment CD.
- The distance from the segment AB to the half-infinite ray CD.
- The distance from the segment AB to the line CD.
- The distance from the half-infinite ray AB to the point C.
- The distance from the half-infinite ray AB to the segment CD.
- The distance from the half-infinite ray AB to the half-infinite ray CD.
- The distance from the half-infinite ray AB to the line CD.
- The distance from the line AB to the point C.
- The distance from the line AB to the segment CD.
- The distance from the line AB to the half-infinite ray CD.
- The distance from the line AB to the line CD.

Input

Each line contains two integers — coordinates of points A, B, C, D respectively. All coordinates do not exceed 10^4 by absolute value.

Output

Print 16 numbers, one per line. The numbers have to be precise to 6 decimal digits.

Examples

stdin	stdout
1 2 7 1 5 6 8 2	5.6568542495 5.6000000000 5.6000000000 5.6000000000 4.6031716446 1.4142135624 1.4000000000 1.4000000000 4.6031716446 1.1507929111 0.0000000000 0.0000000000 4.6031716446 1.1507929111 0.0000000000 0.0000000000

Problem C. Segments intersection

Input file:	stdin
Output file:	stdout
Time limit:	1 second
Memory limit:	256 megabytes

You are given two segments AB and CD. Find the intersection of these segments.

Input

Each of the four lines contains integer coordinates of A, B, C, D respectively. All coordinates do not exceed 10^4 by absolute value. Note that the point may coincide (including the endpoints of the same segment).

Output

If the intersection is empty, print a single line "Empty". If the intersection consists of a single point, print two numbers — coordinates of the intersection point. If the intersection is a segment, print coordinates of the endpoints of the intersection points (the first point must have a smaller x-coordinate, in case of equal x-coordinates, the point with smaller y-coordinate should come first). The answer has to be precise up to 6 digits after decimal point.

Examples

stdin	stdout
0 0 9 9 9 5 0 5	5.0000000000 5.0000000000
0 0 9 9 15 15 7 7	7.0000000000 7.0000000000 9.0000000000 9.0000000000
0 0 9 9 10 10 10 10	Empty

Problem D. Incircle

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 second
Memory limit:	256 megabytes

You are given coordinates of three vertices of a triangle. Find center and radius of an incircle of the triangle.

Input

The only line contains six integer coordinates of the triangle vertices. All coordinates do not exceed 1 000 by absolute value.

Output

Print the coordinates of the incircle radius, followed by its radius. The answer has to be within 10^{-4} absolute or relative tolerance.

Examples

stdin	stdout
0 0 0 15 20 0	5 5 5

Problem E. Two circles

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

You are given two circles in the plane. Find all point of their intersection.

Input

The first line contains the number of test cases K ($1 \le K \le 10000$). Each test case desription consists of two lines, each of the lines describes a circle. Each circle is described by three integers x, y, r ($-1000 \le x, y \le 1000$, $0 < r \le 1000$) — coordinates of the center and the radius.

Output

For each test case output one of the following:

- "`There are no points!!!`" if the intersection is empty.
- "`There are only` i `of them....`" — if the intersection contains exactly i points. The next i lines should describe the intersection points x'_j and y'_j. The points should be ordered lexicographically (the points with smaller x-coordinates come first, in case of equal x-coordinates points with smalelr y-coordinates should come first). Output the number with at least four decimal places.
- "`I can't count them - too many points :(`" if there are infinitely many intersection points.

Don't output the quotes. Separate the answers for separate cases with an empty line.

Examples

stdin	stdout
2 0 0 2 4 0 2 0 0 1 1000 1000 1	There are only 1 of them.... 2.0 0.0 There are no points!!!

Problem F. Avoid the circle!

Input file:	stdin
Output file:	stdout
Time limit:	2 seconds
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

stdin	stdout
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 G. Convex hull

Input file:	stdin
Output file:	stdout
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.

Examples

stdin	stdout
5 0 0 1 1 2 2 1 0 0 1	4 3 5 1 4 6.472135954999580000 00 2.0

Problem H. 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).

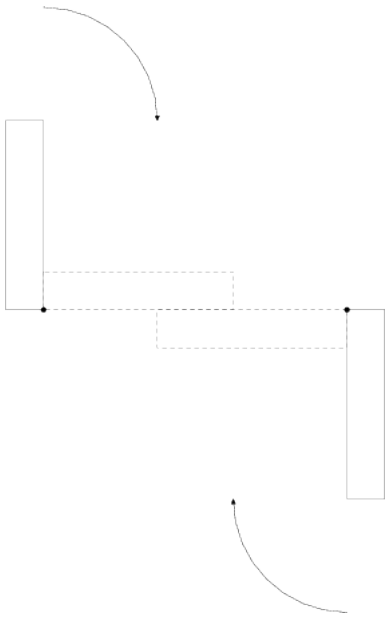
Examples

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:



Disclaimer: this problem is borrowed from a past Codeforces round. Please refrain from looking up the solution, or copying the solution code. Value your practice opportunities!