A Grachten

Damn! I did not only oversleep (and today is *the* contest day!) but I also got stuck somewhere in Delft on my way from the hotel to the contest site. Everywhere around me are grachten, these city-canals that are part of many cities in the Netherlands. I am in a bit of hurry, because the NWERC contest starts in a few minutes.

To make matters even worse, some bridges in Delft are closed due to a cycling race through the city. Thus, I decided to jump over some of the grachten instead of searching for open bridges.

Everyone knows that computer scientists like me are good at algorithms but not very good athletes. Besides, I am a bit faint-hearted and don't want to get wet. So I need your help to calculate the distance I have to jump over a gracht.

Luckily, I did attend the excursion in Delft city center yesterday, where I learned that all paving stones in Delft are squares and have the same size. This way, I can do some measurements on my side of the gracht (my units are paving stones):

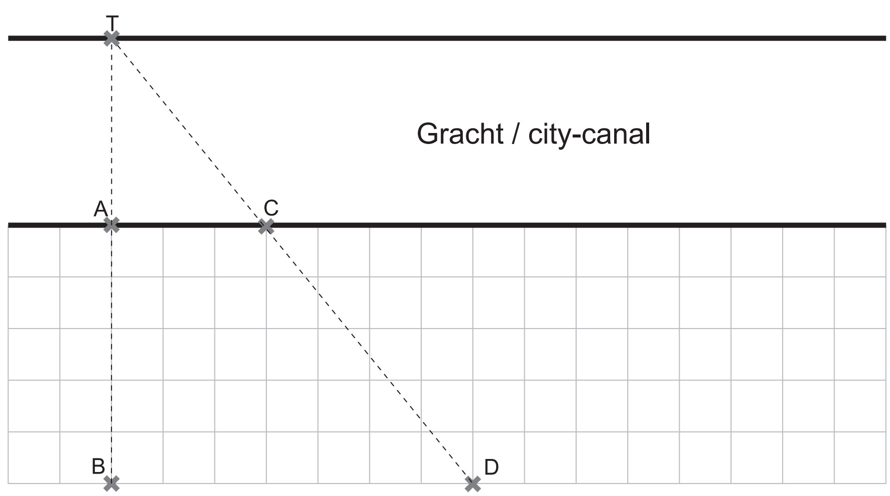


Figure 1 – Illustration of first sample input.

I walked from point C to point D via points A and B while counting the paving stones.

Points A and C are always on the edge of the gracht. Points B and D have the same distance to the gracht. The target point T is always on the edge of the other side of the canal; it is the intersection point of the line through B and A, and the line through D and C. The angle between AT and AC is 90 degrees, and the two edges of the canal are parallel lines.

Please calculate the distance between A and T (necessary jump distance) for me.

Input

For each test case, the input consists of one line containing three positive integers that specify the distances between A and B, A and C, and B and D.

You may safely assume that no distance is larger than 1000 and the distance between B and D is larger than the distance between A and C.

Output

For each test case, print one line of output: the distance between A and T as a **reduced** fraction (i.e. remove all common factors of numerator and denominator).

Example

input	output
5 3 7 5 3 8 1 2 3 23 42 47 500 500 1000 1 1 1000	15/4 3/1 2/1 966/5 500/1 1/999

B. Rectangle Puzzle

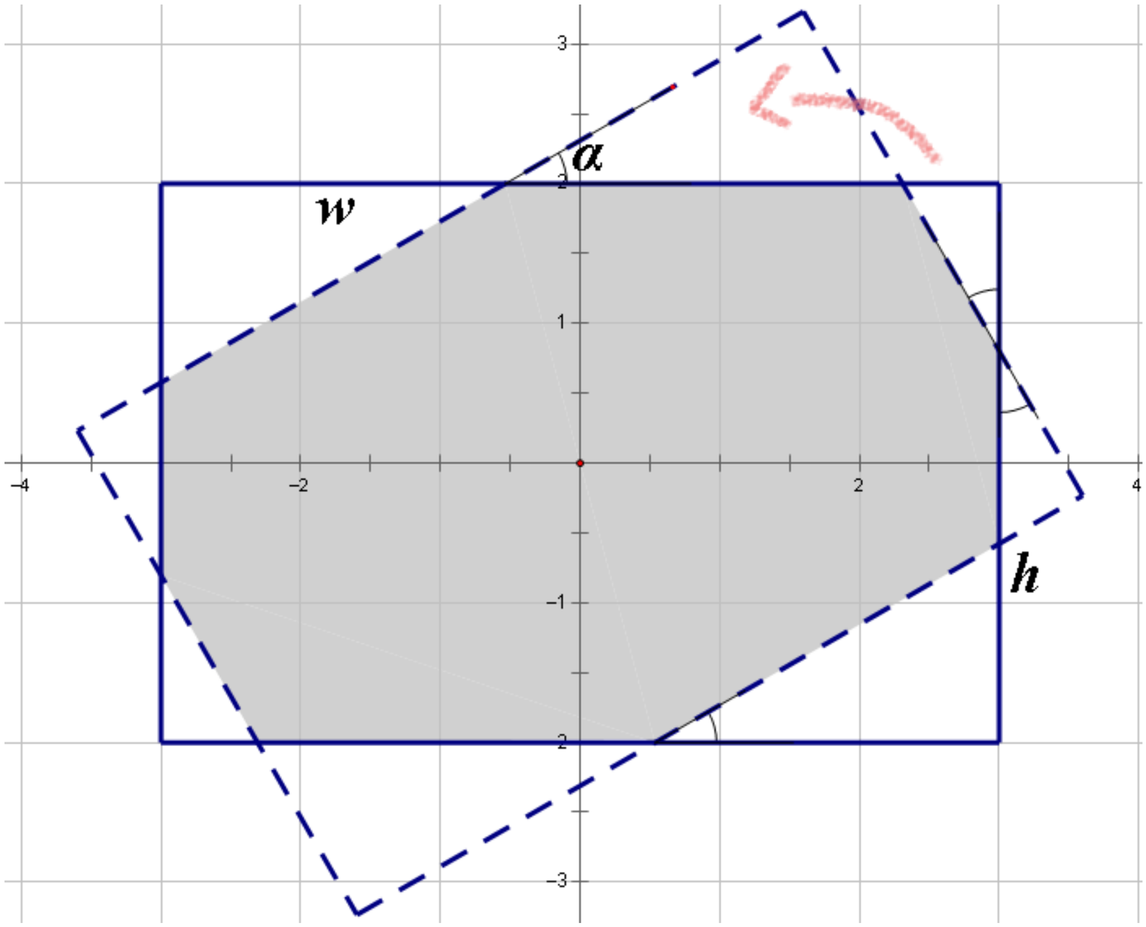
time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

You are given two rectangles on a plane. The centers of both rectangles are located in the origin of coordinates (meaning the center of the rectangle's symmetry). The first rectangle's sides are parallel to the coordinate axes: the length of the side that is parallel to the Ox axis, equals w, the length of the side that is parallel to the Oy axis, equals h. The second rectangle can be obtained by rotating the first rectangle relative to the origin of coordinates by angle α.



Your task is to find the area of the region which belongs to both given rectangles. This region is shaded in the picture.

Input

The first line contains three integers w, h, α $(1 \le w, h \le 10^6; 0 \le \alpha \le 180)$. Angle α is given in degrees.

Output

In a single line print a real number — the area of the region which belongs to both given rectangles.

The answer will be considered correct if its relative or absolute error doesn't exceed 10^{-6}.

Examples

input

```
1 1 45
```

output

```
0.828427125
```

input

```
6 4 30
```

output

```
19.668384925
```

Problem C
Finding Lines
Time limit: 4 seconds

Annabel and Richard like to invent new games and play against each other. One day Annabel has a new game for Richard. In this game there is a game master and a player. The game master draws n points on a piece of paper. The task for the player is to find a straight line, such that at least p percent of the points lie exactly on that line. Richard and Annabel have very good tools for measurement and drawing. Therefore they can check whether a point lies exactly on a line or not. If the player can find such a line then the player wins. Otherwise the game master wins the game.

There is just one problem. The game master can draw the points in a way such that it is not possible at all to draw a suitable line. They need an independent mechanism to check whether there even exists a line containing at least p percent of the points, i.e., $\lceil n \cdot p/100 \rceil$ points. Now it is up to you to help them and write a program to solve this task.

Input

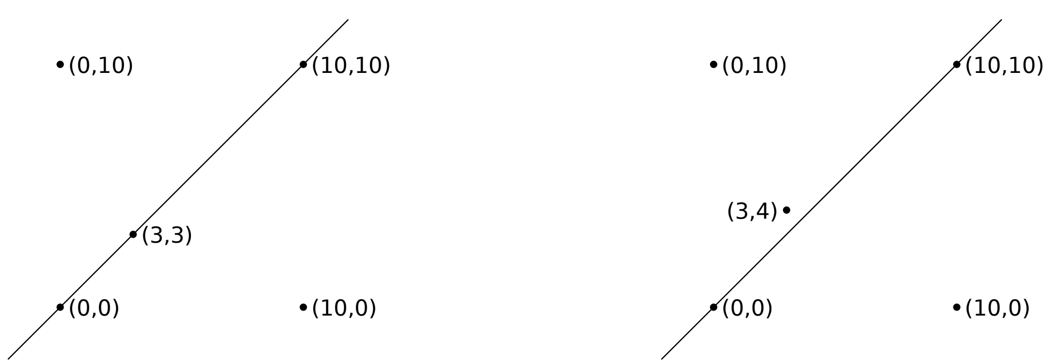
The input consists of:

- one line with one integer n ($1 \leq n \leq 10^5$), the number of points the game master has drawn;
- one line with one integer p ($20 \leq p \leq 100$), the percentage of points which need to lie on the line;
- n lines each with two integers x and y ($0 \leq x, y \leq 10^9$), the coordinates of a point.

No two points will coincide.

Output

Output one line containing either "`possible`" if it is possible to find a suitable line or "`impossible`" otherwise.



(a) Sample input 1: A line with (at least) 3 of the points exists.

(b) Sample input 2: No line with at least 3 points exists.

Figure F.1: Illustration of the sample inputs

Sample Input 1

```
5
55
0 0
10 10
10 0
0 10
3 3
```

Sample Output 1

```
possible
```

Sample Input 2

```
5
45
0 0
10 10
10 0
0 10
3 4
```

Sample Output 2

```
impossible
```

D Bribing Eve

Eve works at a magazine that does product reviews and publishes recommendations to consumers. They are working on a new mobile phones review and have decided on two reproducible tests that score each device's battery lifetime and performance using an integer between 1 and 1000.

These two scores, x_1 and x_2, are then combined with a weights vector $w = [w_1, w_2]$ to produce an overall score:

$$s = w_1 x_1 + w_2 x_2 .$$

The final review ranking is then obtained by sorting the products by decreasing order of s. Additionally, when multiple products get exactly the same score, Eve decides how to order them.

Maria (a fake name to mask her identity) tried to bribe Eve to tweak the results to get her product higher on the list. Eve argued that she was not able to tamper the evaluation of each test, but Maria suggested to tweak the weights w used when computing the overall score. The weights w must be non-negative and at least one of them must be positive, but the values are decided by Eve.

Eve is thinking whether to modify the weights in Maria's benefit or not, and asked you to determine what are the best and worst possible ranking positions for Maria's product.

Task

Given a list of all products scores in battery and performance $[x_1, x_2]$ tests, find out what are the best and worst positions in the ranking that can be given to Maria's product when the weights $[w_1, w_2]$ and the order for tied products are left for Eve to decide.

Input

The first line has the number N of products being compared. N lines follow, each containing two integers x_1 and x_2 indicating a product's score in the battery and performance tests. Maria's product is the first on the list.

Constraints

$1 \leq N \leq 100\,000$	Number of products
$1 \leq x_1, x_2 \leq 1\,000$	Performance of a product in the tests

Output

The output consists of two numbers A and B, indicating the best and worst possible positions that Maria's product can get on the ranking given Eve's ability to modify the weights and ordering in case of a tie.

Sample Input

```
5
7 7
11 10
8 5
1 1
12 12
```

Sample Output

```
3 4
```

E Kissing Circles

If you think the past winter was crazy, just ask the farmers of Punxsutawney about last summer. Usually a summer burns into one's memory due to plentiful harvest, severe drought or hail. However the last summer was so extraordinary because of strange shapes, which appeared at several wheat fields. As a expert in all unusual problems, Bill Murray decided to explain this phenomenon on the scientifically. In order to do this, he thoroughly inspected each wheat field and noticed that each shape was made by crumbling all wheat belonging to a circular region. Every two circles touch in at most one point (in particular no circle can be contained in a different circle).

Bill suspects that the circles describe messages sent by aliens. Unfortunately, understanding their language is very hard. At this point Bill collected all the information about the shapes he found and he is going to use some tools of statistical analysis. The more interesting data he collects, the better. Bill asked you to write a program, which given the description of all the circles computes the number of pairs of circles having a common point.

Input

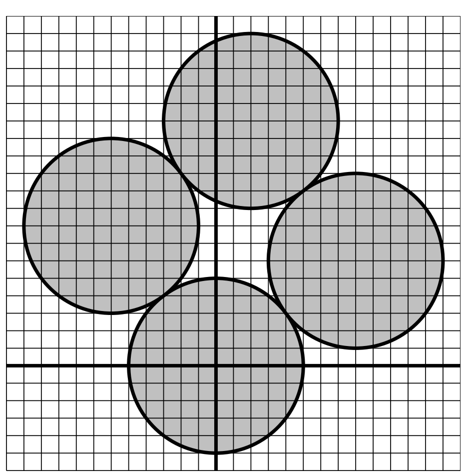
The first line of the input contains a single integer n ($1 \leq n \leq 200,000$), the number of circles. Each of the following n lines describes a circle. In the i-th of those lines there are three integers x_i, y_i, r_i ($-10^9 \leq x_i, y_i \leq 10^9$, $1 \leq r_i \leq 10^9$), meaning that the center of the i-th circle has coordinates (x_i, y_i) while its radius equals r_i.

Output

Your program should output the number of pairs of circles having a common point.

Example

Input	Output
4 0 0 5 8 6 5 -6 8 5 2 14 5	4



F. Very simple problem

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given a convex polygon. Count, please, the number of triangles that contain a given point in the plane and their vertices are the vertices of the polygon. It is guaranteed, that the point doesn't lie on the sides and the diagonals of the polygon.

Input

The first line contains integer n — the number of vertices of the polygon ($3 \le n \le 100000$). The polygon description is following: n lines containing coordinates of the vertices in clockwise order (integer x and y not greater than 10^9 by absolute value). It is guaranteed that the given polygon is nondegenerate and convex (no three points lie on the same line).

The next line contains integer t ($1 \le t \le 20$) — the number of points which you should count the answer for. It is followed by t lines with coordinates of the points (integer x and y not greater than 10^9 by absolute value).

Output

The output should contain t integer numbers, each on a separate line, where i-th number is the answer for the i-th point.

Please, do not use `%lld` specificator to read or write 64-bit integers in C++. It is preffered to use `cin` (also you may use `%I64d`).

Examples

input

```
4
5 0
0 0
0 5
5 5
1
1 3
```

output

```
2
```

input

```
3
0 0
0 5
5 0
2
1 1
10 10
```

output

```
1
0
```

input

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

output

```
5
3
3
4
```

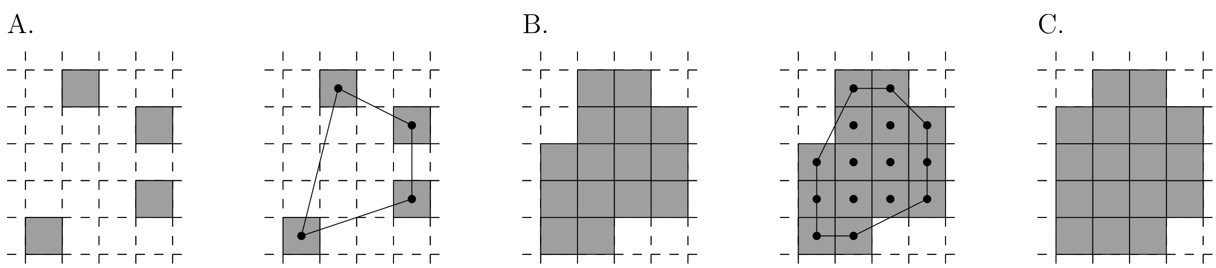
G Fields and Farmers

Being a farmer in Ardenia is a tough job. We do not mean only the dry and hostile environment where they have to pasture sheep. The government (or actually the king himself) wants his people to invade foreign lands rather than to harvest theirs, and thus tries to make the lives of poor farmers as hard as possible. All difficulties start with a seemingly simple task: purchasing a piece of land, called farming parcel.

The whole farming territory is a huge rectangular grid consisting of square fields; a farming parcel consists of some of these fields. At the beginning a farmer buys an *initial set of fields*; his parcel consists initially just of these fields. However, the actual farming parcel is determined with the help of string and poles, by repeating the following steps.

1. Stick a pole into the middle of each field from the farming parcel.
2. Surround the poles with a string, creating the smallest region enclosing all the poles.
3. The new farming parcel is the set of all fields having a non-empty intersection with this region. A field sharing just the edge or a corner with the region does not count.

Of course, the parcel may only increase by implementing the operation above, so each farmer makes sure these steps are repeated till the farming parcel does not change; we call such a parcel *final*. An example is depicted below. The initial farming parcel consists of four fields (figure A), after one iteration it grows (figure B), and after another one it becomes final (figure C).



It appears, however, that the final farming parcel would sometimes be the same even if the farmer did not buy all the initial fields but just a subset of them. A subset of this property is called *valid*. The farmer wants to know in how many ways he may choose a valid subset of initial fields.

Multiple Test Cases

The input contains several test cases. The first line of the input contains a positive integer $Z \leq 50$, denoting the number of test cases. Then Z test cases follow, each conforming to the format described in section *Single Instance Input*. For each test case, your program has to write an output conforming to the format described in section *Single Instance Output*.

Single Instance Input

The first line of the input instance contains a positive integer $n \leq 10^6$, being the number of initial fields of the parcel. Each of the following n lines contain two integers x_i, $y_i \in [-10^9, 10^9]$, being the coordinates of these fields. All initial fields are different.

Single Instance Output

Let k be the number of valid subsets of the initial fields. You should output a line containing the number $k \mod (10^9 + 7)$.

Example

Input	Output
2 4 0 0 0 1 0 2 0 3 5 0 0 -1 0 1 0 0 -1 0 1	4 2