Problem A. Totalphone

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

The well-known mobile network operator Totalphone has set up a number of new base transceiver stations in order to cover a newly-built highway with its network. As always the programmers of Totalphone have been sloppy; as a result, the transmission power cannot be set up individually for the stations, but one can only set the transmission power to a fixed common value for all the stations. In order to minimize power consumption, the company wants to know the maximal distance of a point on the highway to the nearest base transceiver station.

Input

The first line of text file mobile.in consists of two integers N ($1 \leq N \leq 10^6$) and L ($1 \leq L \leq 10^9$) representing the number of base transceiver stations and the length of the highway, respectively. N lines follow, each containing a pair of integers x_i, y_i ($-10^9 \leq x_i, y_i \leq 10^9$) which describes the coordinates of a base transceiver station. All points are distinct. Coordinates are sorted in the non-decreasing order with respect to x_i coordinates. If two values of x_i are the same, then coordinates are sorted with respect to y_i coordinates in increasing order.

The highway is a straight line ranging from $(0, 0)$ to $(L, 0)$.

Output

The first and only line of the output file should contain a single number — the maximal distance of a point on the highway to the nearest base transceiver station. Your output will be regarded as correct if it differs by at most 10^{-3} from the precise result.

Example

standard input	standard output
2 10 0 0 11 1	5.545455

Problem B. Rural Planning

Input file: `standard input`
Output file: `standard output`
Time limit: 25 seconds
Memory limit: 256 megabytes

You have recently purchased a nice big farmyard, and you would like to build a fence around it. There are already N fence posts in your farmyard.

You will add lengths of fence in straight lines connecting the fence posts. Unfortunately, for reasons you don't fully understand, your lawyers insist you actually have to use all the fence posts, or things will go bad.

In this problem, the posts will be represented as points in a 2-dimensional plane. You want to build the fence by ordering the posts in some order, and then connecting the first with the second, second with third, and finally the last one with the first. The fence segments you create should be a polygon without self-intersections. That is, at each fence-post there are only two fence segments, and at every other point there is at most one fence segment.

Now that's easy, but you also actually want to preserve the fact your farmyard is big! It's not really fun to wall off most of your farmyard with the fences. So you would like to create the fence in such a way that the enclosed area is more than half of the maximum area you could enclose if you were allowed not to use all the posts.

Input

The first line of the input contains the number N ($3 \leq N \leq 1000$) of posts. The posts are numbered from 0 to $N-1$. Each of the next N lines contains two integers X_i and Y_i ($50000 \leq X_i, Y_i \leq 50000$) separated by a single space: the coordinates of the i-th post. All N points do not lie on the same line.

Output

Output N distinct integers from 0 to $N - 1$, separated by spaces. They are the numbers of the posts, in either clockwise or counter-clockwise direction, that you will use to build the fence. Note that the first and last posts are connected.

Examples

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

Problem C. Two circles

Input file: `standard input`
Output file: `standard output`
Time limit: 4 seconds
Memory limit: 256 megabytes

We will consider a convex polygon with N vertices. We wish to find the maximum radius R such that two circles of radius R can be placed entirely inside the polygon without overlapping.

Input

The first line of input contains the number N ($3 \leq n \leq 50000$). Each of the next N lines contains a pair of integers x_i and y_i ($-10^7 \leq x_i, y_i \leq 10^7$) — representing the coordinates of the i-th point, separated by space. Points are given in anti-clockwise order.

Output

You should output a single number R — the desired radius. Output R with absolute or relative error of 10^{-3}.

Example

standard input	standard output
4 0 0 1 0 1 1 0 1	0.293
4 0 0 3 0 3 1 0 1	0.500
6 0 0 8 0 8 6 4 8 2 8 0 4	2.189

Problem D. Aerobics

Input file:	`standard input`
Output file:	`standard output`
Time limit:	12 seconds
Memory limit:	256 megabytes

The aerobics class begins. The trainer says, "Please position yourselves on the training mat so that each one of you has enough space to move your arms around freely, and not hit anybody else."People start milling around on the mat, trying to position themselves properly. Minutes pass, and finally the trainer is so annoyed that he asks you to write a program that will position all the people correctly, hoping it will be quicker than letting them figure it out for themselves!

You are given the dimensions (width and length) of the mat on which the class takes place. For every student, there is a circular area she has to have for herself, with radius equal to the reach of her arms. These circles can not intersect, though they can touch; and the center of each circle (where the student stands) has to be on the mat. Note that the arms can reach outside the mat. You know that there's plenty of space on the mat — the area of the mat is at least five times larger than the total area of the circles required by all the people in the class. It will always be possible for all the people to position themselves as required.

Input

Test case consists of two lines. The first line contains three integers: N, W and L ($1 \le N \le 10^3$, $1 \le W, L \le 10^9$), denoting the number of students, the width of the mat, and the length of the mat, respectively. The second line contains N integers r_i ($1 \le r_i \le 10^5$), denoting the reach of the arms of the i-th student. Sum of N through all test cases does not exceed 6000. The area of the mat is at least 5 times larger than the total area of the circles.

Output

Output one line containing $2N$ real numbers $x_1, y_1, \ldots, x_N, y_N$, where the pair (x_i, y_i) is the position where i-th student should stand with $1 \le x_i \le W$ and $1 \le y_i \le L$.

Examples

standard input	standard output
2 6 6 1 1	0.0 0.0 6.0 6.0
3 320 2 4 3 2	0.0 0.0 7.0 0.0 12.0 0.0

Problem E. Twirling Towards Freedom

Input file:	`standard input`
Output file:	`standard output`
Time limit:	45 seconds
Memory limit:	256 megabytes

After hearing this inspirational quote from America's first presidential nominee from the planet Rigel VII, you have decided that you too would like to twirl (rotate) towards freedom. For the purposes of this problem, you can think of "freedom"as being as far away from your starting location as possible.

The galaxy is a two-dimensional plane. Your space ship starts at the origin, position $(0, 0)$. There are N stars in the galaxy. Every minute, you can choose a star and rotate your space ship 90 degrees clockwise around the star. You may also choose to stay where you are.

How far away can you move from the origin after M minutes?

Input

The first line of the input contains two integers N and M ($1 \le N \le 5\,000$, $1 \le M \le 10^8$). The next N lines each contain two integers, x_i and y_i ($-1000 \le x_i, y_i \le 1000$), representing locations of starts.

Output

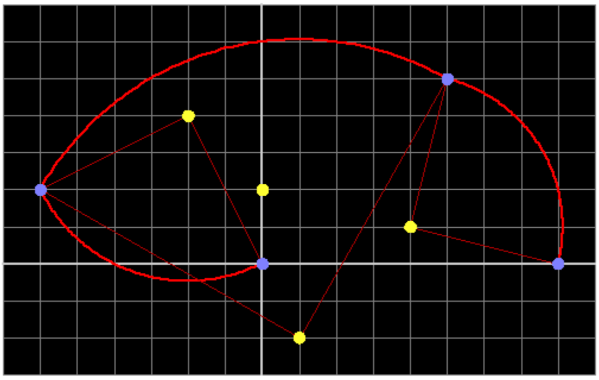
The sole line of the output should contain one real number — the distance from the origin to the optimal final position. Relative or absolute error should not exceed 10^{-6}.

Examples

standard input	standard output
4 1 -2 4 1 -2 4 1 0 2	6.324555
1 4 -5 0	10.000000
2 5 -1 1 -2 2	6.324555

Note

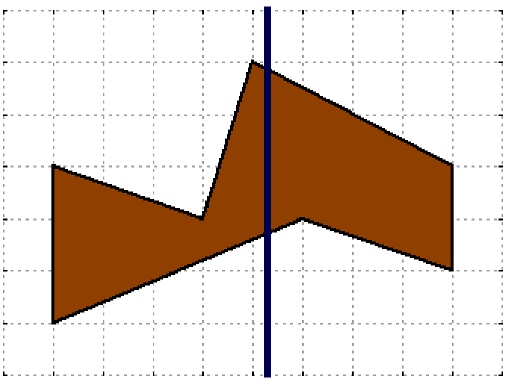
The image illustrates the first 3 rotations for a possible path in sample case 1. Note that this path is not necessarily a part of any optimal solution.



Problem F. Irregular cake

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

Mary the Mathematician has a bakery that she founded some years ago, but after all this time she has become bored with always baking the same rectangular and circular cakes. For her next birthday, she wants to bake an irregular cake, which is defined as the area between two "polylines"between $x = 0$ and $x = W$. These polylines will be called the lower boundary and the upper boundary.



Formally, a polyline is defined by a sequence of points $(P_0, P_1, ..., P_n)$ going from left to right. Consecutive points are connected to form a sequence of line segments, which together make up the polyline.

Today is Mary's birthday and she has baked an irregular cake bounded by two polylines with L points and U points respectively. After singing "Happy Birthday,"she wants to make G-1 vertical cuts to split the cake into G slices with equal area. She can then share these cake slices with all her guests. However, the irregular cake shape makes this task pretty tricky. Can you help her decide where to make the cuts?

Input

The first line of the input contains four integers: W (the cake's width, $1 \le W \le 1000$), L (the number of points on the lower boundary, $1 \le L \le 100$), U (the number of points on the upper boundary, $1 \le U \le 100$) and G (the number of guests at the party, $1 \le G \le 101$).

This is followed by L lines specifying the lower boundary. The i-th line contains two integers x_i and y_i, representing the coordinates of the i-th point on the lower boundary ($-1000 \le x_i, y_i \le 1000$). The x-coordinate of the leftmost point of both boundaries will be 0. The x-coordinate of the rightmost point of both boundaries will be W. Points in the same boundary will be sorted increasingly by x-coordinate. Points in the same boundary will have different x-coordinates. The lower boundary will always be strictly below the upper boundary for all x between 0 and W, inclusive. (In other words, the lower boundary will have a smaller y-coordinate than the upper boundary at every x position.)

Output

$G-1$ lines of the output should contain the x-coordinates at which cuts must be made, ordered from the leftmost cut to the rightmost cut.

Answers with a relative or absolute error of at most 10^{-6} will be considered correct.

Examples

standard input	standard output

`15 3 3 3`
`0 6`
`10 8`
`15 9`
`0 10`
`5 11`
`15 13`
	`5.000000`
	`10.000000`

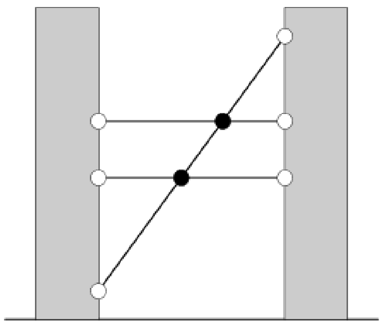
`8 3 4 2`
`0 2`
`5 4`
`8 3`
`0 5`
`3 4`
`4 7`
`8 5`
	`4.290588`

Problem G. Rope Intranet

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

A company is located in two very tall buildings. The company intranet connecting the buildings consists of many wires, each connecting a window on the first building to a window on the second building.

You are looking at those buildings from the side, so that one of the buildings is to the left and one is to the right. The windows on the left building are seen as points on its right wall, and the windows on the right building are seen as points on its left wall. Wires are straight segments connecting a window on the left building to a window on the right building.



You've noticed that no two wires share an endpoint (in other words, there's at most one wire going out of each window). However, from your viewpoint, some of the wires intersect midway. You've also noticed that exactly two wires meet at each intersection point.

On the above picture, the intersection points are the black circles, while the windows are the white circles.

How many intersection points do you see?

Input

The first line of the input contains an integer N ($1 \le N \le 1000$), denoting the number of wires you see.

The next N lines each describe one wire with two integers A_i and B_i ($1 \le A_i, B_i \le 10^4$). These describe the windows that this wire connects: A_i is the height of the window on the left building, and B_i is the height of the window on the right building. All A_i are different. All B_i are different. No three wires intersect at the same point.

Output

The sole line of the output should contain one number — the number of intersection points you see.

Examples

standard input	standard output
3 1 10 5 5 7 7	2
2 1 1 2 2	0

Problem H. Watering plants

Input file: standard input
Output file: standard output
Time limit: 12 seconds
Memory limit: 256 megabytes

In your greenhouse you have a number of plants which you need to water.

Each of the plants takes up an area which is a circle. No two of the plants overlap or touch each other.

You are going to buy two sprinklers. Each of the sprinklers will spray everything within a circle of radius R with water.

One of the sprinklers will run in the morning, and one will run at night. For you to be satisfied that a plant will get enough water, either the whole area of the plant must be watered in the morning, or the the whole area of the plant must be watered at night. So each of the circles representing a plant must be completely in one or both of the two circles representing the area the sprinklers can water.

Given the location and radius of each of the plants, find the minimum radius R so that it is possible to place the two sprinklers to water all the plants. The sprinklers will be installed on the ceiling, so a sprinkler's position can be inside the area of a plant.

Input

The first line of the input contains the number of plants N ($1 \le N \le 40$). Then N lines contain three integers x, y, r each ($1 \le x, y \le 1000$, $1 \le r \le 100$) — the coordinates of the center of the plant and the radius of the plant.

Output

The sole line of the output should contain one integer — the minimal radius of sprinkles. An absolute or relative error should not exceed 10^{-4}.

Examples

standard input	standard output
3 20 10 2 20 20 2 40 10 3	6.999999999
3 20 10 3 30 10 3 40 10 3	7.999999999
5 100 100 1 140 100 1 100 130 1 100 500 1 150 500 1	25.9999999998
8 100 100 1 110 100 1 100 110 1 110 110 1 200 200 1 210 200 1 200 210 1 210 210 1	8.071067806865473
4 100 100 1 200 100 1 200 103 1 300 103 1	50.999999999900005

Problem I. Pinball

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

Maria is quite addicted to pinball. She can shoot the ball to any position at the top of the board, but she cannot predict where the ball will end when it falls down, because it hits many bumpers on its way down.

She decided to model the pinball table as line segments and assume that the ball is a point that falls from infinite height. The ball falls straight vertically unless there is a segment immediately below it, in which case it follows the direction of the segment downwards until its end.

As you would expect the segments are closed, that is an endpoint is part of its segment. Pairs of segments do not intersect, not even at endpoints, and none is vertical or horizontal. Segments are not given in any specific order.

Input

The first line contains an integer N ($0 \le N \le 100\,000$), the number of segments. Then N lines follow, each with four integers x_1 y_1 x_2 y_2 the coordinates of a segment ($-10^6 \le x_i, y_i \le 10^6$).

The last line contains an integer x_0 ($-10^6 \le x_0 \le 10^6$), the initial x-coordinate of the ball.

Output

Output a single integer x_T, the final x-coordinate of the ball.

Example

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