A. Two Problems

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

A boy Valera registered on site `Codeforces` as `Valera`, and wrote his first `Codeforces Round #300`. He boasted to a friend Arkady about winning as much as x points for his first contest. But Arkady did not believe his friend's words and decided to check whether Valera could have shown such a result.

He knows that the contest number 300 was unusual because there were only two problems. The contest lasted for t minutes, the minutes are numbered starting from zero. The first problem had the initial cost of a points, and every minute its cost reduced by d_a points. The second problem had the initial cost of b points, and every minute this cost reduced by d_b points. Thus, as soon as the zero minute of the contest is over, the first problem will cost $a - d_a$ points, and the second problem will cost $b - d_b$ points. It is guaranteed that at any moment of the contest each problem has a non-negative cost.

Arkady asks you to find out whether Valera could have got exactly x points for this contest. You should assume that Valera could have solved any number of the offered problems. You should also assume that for each problem Valera made no more than one attempt, besides, he could have submitted both problems at the same minute of the contest, starting with minute 0 and ending with minute number $t - 1$. Please note that Valera can't submit a solution exactly t minutes after the start of the contest or later.

Input

The single line of the input contains six integers x, t, a, b, d_a, d_b ($0 \le x \le 600$; $1 \le t, a, b, d_a, d_b \le 300$) — Valera's result, the contest's duration, the initial cost of the first problem, the initial cost of the second problem, the number of points that the first and the second problem lose per minute, correspondingly.

It is guaranteed that at each minute of the contest each problem has a non-negative cost, that is, $a - i \cdot d_a \ge 0$ and $b - i \cdot d_b \ge 0$ for all $0 \le i \le t - 1$.

Output

If Valera could have earned exactly x points at a contest, print "YES", otherwise print "NO" (without the quotes).

Examples

input

```
30 5 20 20 3 5
```

output

```
YES
```

input

```
10 4 100 5 5 1
```

output

```
NO
```

Note

In the first sample Valera could have acted like this: he could have submitted the first problem at minute 0 and the second problem — at minute 2. Then the first problem brings him 20 points and the second problem brings him 10 points, that in total gives the required 30 points.

B. Game on Paper

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

One not particularly beautiful evening Valera got very bored. To amuse himself a little bit, he found the following game.

He took a checkered white square piece of paper, consisting of $n \times n$ cells. After that, he started to paint the white cells black one after the other. In total he painted m different cells on the piece of paper. Since Valera was keen on everything square, he wondered, how many moves (i.e. times the boy paints a square black) he should make till a black square with side 3 can be found on the piece of paper. But Valera does not know the answer to this question, so he asks you to help him.

Your task is to find the minimum number of moves, till the checkered piece of paper has at least one black square with side of 3. Otherwise determine that such move does not exist.

Input

The first line contains two integers n and m $(1 \le n \le 1000, 1 \le m \le min(n \cdot n, 10^5))$ — the size of the squared piece of paper and the number of moves, correspondingly.

Then, m lines contain the description of the moves. The i-th line contains two integers x_i, y_i $(1 \le x_i, y_i \le n)$ — the number of row and column of the square that gets painted on the i-th move.

All numbers on the lines are separated by single spaces. It is guaranteed that all moves are different. The moves are numbered starting from 1 in the order, in which they are given in the input. The columns of the squared piece of paper are numbered starting from 1, from the left to the right. The rows of the squared piece of paper are numbered starting from 1, from top to bottom.

Output

On a single line print the answer to the problem — the minimum number of the move after which the piece of paper has a black square with side 3. If no such move exists, print -1.

Examples

input

```
4 11
1 1
1 2
1 3
2 2
2 3
1 4
2 4
3 4
3 2
3 3
4 1
```

output

```
10
```

input

```
4 12
1 1
1 2
1 3
2 2
2 3
1 4
2 4
3 4
3 2
4 2
4 1
3 1
```

output

```
-1
```

C. Photographer

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Valera's lifelong ambition was to be a photographer, so he bought a new camera. Every day he got more and more clients asking for photos, and one day Valera needed a program that would determine the maximum number of people he can serve.

The camera's memory is d megabytes. Valera's camera can take photos of high and low quality. One low quality photo takes a megabytes of memory, one high quality photo take b megabytes of memory. For unknown reasons, each client asks him to make several low quality photos and several high quality photos. More formally, the i-th client asks to make x_i low quality photos and y_i high quality photos.

Valera wants to serve as many clients per day as possible, provided that they will be pleased with his work. To please the i-th client, Valera needs to give him everything he wants, that is, to make x_i low quality photos and y_i high quality photos. To make one low quality photo, the camera must have at least a megabytes of free memory space. Similarly, to make one high quality photo, the camera must have at least b megabytes of free memory space. Initially the camera's memory is empty. Valera also does not delete photos from the camera so that the camera's memory gradually fills up.

Calculate the maximum number of clients Valera can successfully serve and print the numbers of these clients.

Input

The first line contains two integers n and d $(1 \le n \le 10^5, 1 \le d \le 10^9)$ — the number of clients and the camera memory size, correspondingly. The second line contains two integers a and b ($1 \le a \le b \le 10^4$) — the size of one low quality photo and of one high quality photo, correspondingly.

Next n lines describe the clients. The i-th line contains two integers x_i and y_i ($0 \le x_i, y_i \le 10^5$) — the number of low quality photos and high quality photos the i-th client wants, correspondingly.

All numbers on all lines are separated by single spaces.

Output

On the first line print the answer to the problem — the maximum number of clients that Valera can successfully serve. Print on the second line the numbers of the client in any order. All numbers must be distinct. If there are multiple answers, print any of them. The clients are numbered starting with 1 in the order in which they are defined in the input data.

Examples

input

```
3 10
2 3
1 4
2 1
1 0
```

output

```
2
3 2
```

input

```
3 6
6 6
1 1
1 0
1 0
```

output

```
1
2
```

D. Hit Ball

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

When Valera was playing football on a stadium, it suddenly began to rain. Valera hid in the corridor under the grandstand not to get wet. However, the desire to play was so great that he decided to train his hitting the ball right in this corridor. Valera went back far enough, put the ball and hit it. The ball bounced off the walls, the ceiling and the floor corridor and finally hit the exit door. As the ball was wet, it left a spot on the door. Now Valera wants to know the coordinates for this spot.

Let's describe the event more formally. The ball will be considered a point in space. The door of the corridor will be considered a rectangle located on plane xOz, such that the lower left corner of the door is located at point $(0, 0, 0)$, and the upper right corner is located at point $(a, 0, b)$. The corridor will be considered as a rectangular parallelepiped, infinite in the direction of increasing coordinates of y. In this corridor the floor will be considered as plane xOy, and the ceiling as plane, parallel to xOy and passing through point $(a, 0, b)$. We will also assume that one of the walls is plane yOz, and the other wall is plane, parallel to yOz and passing through point $(a, 0, b)$.

We'll say that the ball hit the door when its coordinate y was equal to 0. Thus the coordinates of the spot are point $(x_0, 0, z_0)$, where $0 \le x_0 \le a, 0 \le z_0 \le b$. To hit the ball, Valera steps away from the door at distance m and puts the ball in the center of the corridor at point $(\frac{a}{2}, m, 0)$. After the hit the ball flies at speed (v_x, v_y, v_z). This means that if the ball has coordinates (x, y, z), then after one second it will have coordinates $(x + v_x, y + v_y, z + v_z)$.

See image in notes for clarification.

When the ball collides with the ceiling, the floor or a wall of the corridor, it bounces off in accordance with the laws of reflection (the angle of incidence equals the angle of reflection). In the problem we consider the ideal physical model, so we can assume that there is no air resistance, friction force, or any loss of energy.

Input

The first line contains three space-separated integers a, b, m $(1 \le a, b, m \le 100)$. The first two integers specify point $(a, 0, b)$, through which the ceiling and one of the corridor walls pass. The third integer is the distance at which Valera went away from the door.

The second line has three space-separated integers v_x, v_y, v_z $(|v_x|, |v_y|, |v_z| \le 100, v_y < 0, v_z \ge 0)$ — the speed of the ball after the hit.

It is guaranteed that the ball hits the door.

Output

Print two real numbers x_0, z_0 — the x and z coordinates of point $(x_0, 0, z_0)$, at which the ball hits the exit door. The answer will be considered correct, if its absolute or relative error does not exceed 10^{-6}.

Examples

input

```
7 2 11
3 -11 2
```

output

```
6.5000000000 2.0000000000
```

input

```
7 2 11
4 -3 3
```

output

```
4.1666666667 1.0000000000
```

Note

E. Transportation

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Valera came to Japan and bought many robots for his research. He's already at the airport, the plane will fly very soon and Valera urgently needs to bring all robots to the luggage compartment.

The robots are self-propelled (they can potentially move on their own), some of them even have compartments to carry other robots. More precisely, for the i-th robot we know value c_i — the number of robots it can carry. In this case, each of c_i transported robots can additionally carry other robots.

However, the robots need to be filled with fuel to go, so Valera spent all his last money and bought S liters of fuel. He learned that each robot has a restriction on travel distances. Thus, in addition to features c_i, the i-th robot has two features f_i and l_i — the amount of fuel (in liters) needed to move the i-th robot, and the maximum distance that the robot can go.

Due to the limited amount of time and fuel, Valera wants to move the maximum number of robots to the luggage compartment. He operates as follows.

- First Valera selects some robots that will travel to the luggage compartment on their own. In this case the total amount of fuel required to move all these robots must not exceed S.
- Then Valera seats the robots into the compartments, so as to transport as many robots as possible. Note that if a robot doesn't move by itself, you can put it in another not moving robot that is moved directly or indirectly by a moving robot.
- After that all selected and seated robots along with Valera go to the luggage compartment and the rest robots will be lost.

There are d meters to the luggage compartment. Therefore, the robots that will carry the rest, must have feature l_i of not less than d. During the moving Valera cannot stop or change the location of the robots in any way.

Help Valera calculate the maximum number of robots that he will be able to take home, and the minimum amount of fuel he will have to spend, because the remaining fuel will come in handy in Valera's research.

Input

The first line contains three space-separated integers n, d, S ($1 \le n \le 10^5$, $1 \le d, S \le 10^9$). The first number represents the number of robots, the second one — the distance to the luggage compartment and the third one — the amount of available fuel.

Next n lines specify the robots. The i-th line contains three space-separated integers c_i, f_i, l_i ($0 \le c_i, f_i, l_i \le 10^9$) — the i-th robot's features. The first number is the number of robots the i-th robot can carry, the second number is the amount of fuel needed for the i-th robot to move and the third one shows the maximum distance the i-th robot can go.

Output

Print two space-separated integers — the maximum number of robots Valera can transport to the luggage compartment and the minimum amount of fuel he will need for that. If Valera won't manage to get any robots to the luggage compartment, print two zeroes.

Examples

input

```
3 10 10
0 12 10
1 6 10
0 1 1
```

output

```
2 6
```

input

```
2 7 10
3 12 10
5 16 8
```

output

```
0 0
```

input

```
4 8 10
0 12 3
1 1 0
0 3 11
1 6 9
```

output

```
4 9
```