A. Magic Spheres

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

Carl is a beginner magician. He has a blue, b violet and c orange magic spheres. In one move he can transform two spheres **of the same color** into one sphere of any other color. To make a spell that has never been seen before, he needs at least x blue, y violet and z orange spheres. Can he get them (possible, in multiple actions)?

Input

The first line of the input contains three integers a, b and c ($0 \le a, b, c \le 1\,000\,000$) — the number of blue, violet and orange spheres that are in the magician's disposal.

The second line of the input contains three integers, x, y and z ($0 \le x, y, z \le 1\,000\,000$) — the number of blue, violet and orange spheres that he needs to get.

Output

If the wizard is able to obtain the required numbers of spheres, print "Yes". Otherwise, print "No".

Examples

input

```
4 4 0
2 1 2
```

output

```
Yes
```

input

```
5 6 1
2 7 2
```

output

```
No
```

input

```
3 3 3
2 2 2
```

output

```
Yes
```

Note

In the first sample the wizard has 4 blue and 4 violet spheres. In his first action he can turn two blue spheres into one violet one. After that he will have 2 blue and 5 violet spheres. Then he turns 4 violet spheres into 2 orange spheres and he ends up with 2 blue, 1 violet and 2 orange spheres, which is exactly what he needs.

B. Testing Robots

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

The Cybernetics Failures (CF) organisation made a prototype of a bomb technician robot. To find the possible problems it was decided to carry out a series of tests. At the beginning of each test the robot prototype will be placed in cell (x_0, y_0) of a rectangular squared field of size $x \times y$, after that a mine will be installed into one of the squares of the field. It is supposed to conduct exactly $x \cdot y$ tests, each time a mine is installed into a square that has never been used before. The starting cell of the robot always remains the same.

After placing the objects on the field the robot will have to run a sequence of commands given by string s, consisting only of characters 'L', 'R', 'U', 'D'. These commands tell the robot to move one square to the left, to the right, up or down, or stay idle if moving in the given direction is impossible. As soon as the robot fulfills all the sequence of commands, it will blow up due to a bug in the code. But if at some moment of time the robot is at the same square with the mine, it will also blow up, but not due to a bug in the code.

Moving to the left decreases coordinate y, and moving to the right increases it. Similarly, moving up decreases the x coordinate, and moving down increases it.

The tests can go on for very long, so your task is to predict their results. For each k from 0 to $length(s)$ your task is to find in how many tests the robot will run exactly k commands before it blows up.

Input

The first line of the input contains four integers x, y, x_0, y_0 ($1 \le x, y \le 500, 1 \le x_0 \le x, 1 \le y_0 \le y$) — the sizes of the field and the starting coordinates of the robot. The coordinate axis X is directed downwards and axis Y is directed to the right.

The second line contains a sequence of commands s, which should be fulfilled by the robot. It has length from 1 to $100\,000$ characters and only consists of characters 'L', 'R', 'U', 'D'.

Output

Print the sequence consisting of $(length(s) + 1)$ numbers. On the k-th position, starting with zero, print the number of tests where the robot will run exactly k commands before it blows up.

Examples

input

```
3 4 2 2
UURDRDRL
```

output

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

input

```
2 2 2 2
ULD
```

output

```
1 1 1 1
```

Note

In the first sample, if we exclude the probable impact of the mines, the robot's route will look like that: $(2,2) \rightarrow (1,2) \rightarrow (1,2) \rightarrow (1,3) \rightarrow (2,3) \rightarrow (2,4) \rightarrow (3,4) \rightarrow (3,4) \rightarrow (3,3)$.

C. Sorting Railway Cars

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

An infinitely long railway has a train consisting of n cars, numbered from 1 to n (the numbers of all the cars are distinct) and positioned in arbitrary order. David Blaine wants to sort the railway cars in the order of increasing numbers. In one move he can make one of the cars disappear from its place and teleport it either to the beginning of the train, or to the end of the train, at his desire. What is the minimum number of actions David Blaine needs to perform in order to sort the train?

Input

The first line of the input contains integer n ($1 \le n \le 100\,000$) — the number of cars in the train.

The second line contains n integers p_i ($1 \le p_i \le n$, $p_i \ne p_j$ if $i \ne j$) — the sequence of the numbers of the cars in the train.

Output

Print a single integer — the minimum number of actions needed to sort the railway cars.

Examples

input

```
5
4 1 2 5 3
```

output

```
2
```

input

```
4
4 1 3 2
```

output

```
2
```

Note

In the first sample you need first to teleport the 4-th car, and then the 5-th car to the end of the train.

D. Lazy Student

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

Student Vladislav came to his programming exam completely unprepared as usual. He got a question about some strange algorithm on a graph — something that will definitely never be useful in real life. He asked a girl sitting next to him to lend him some cheat papers for this questions and found there the following definition:

The minimum spanning tree T of graph G is such a tree that it contains all the vertices of the original graph G, and the sum of the weights of its edges is the minimum possible among all such trees.

Vladislav drew a graph with n vertices and m edges containing no loops and multiple edges. He found one of its minimum spanning trees and then wrote for each edge its weight and whether it is included in the found tree or not. Unfortunately, the piece of paper where the graph was painted is gone and the teacher is getting very angry and demands to see the original graph. Help Vladislav come up with a graph so that the information about the minimum spanning tree remains correct.

Input

The first line of the input contains two integers n and m ($2 \le n \le 100\,000, 1 \le m \le 100\,000, n-1 \le m \le \frac{n(n-1)}{2}$) — the number of vertices and the number of edges in the graph.

Each of the next m lines describes an edge of the graph and consists of two integers a_j and b_j ($1 \le a_j \le 10^9, b_j = \{0, 1\}$). The first of these numbers is the weight of the edge and the second number is equal to 1 if this edge was included in the minimum spanning tree found by Vladislav, or 0 if it was not.

It is guaranteed that exactly $n - 1$ number $\{b_j\}$ are equal to one and exactly $m - n + 1$ of them are equal to zero.

Output

If Vladislav has made a mistake and such graph doesn't exist, print -1.

Otherwise print m lines. On the j-th line print a pair of vertices (u_j, v_j) ($1 \le u_j, v_j \le n, u_j \ne v_j$), that should be connected by the j-th edge. The edges are numbered in the same order as in the input. The graph, determined by these edges, must be connected, contain no loops or multiple edges and its edges with $b_j = 1$ must define the minimum spanning tree. In case there are multiple possible solutions, print any of them.

Examples

input

```
4 5
2 1
3 1
4 0
1 1
5 0
```

output

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

input

```
3 3
1 0
2 1
3 1
```

output

```
-1
```

E. Freelancer's Dreams

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

Mikhail the Freelancer dreams of two things: to become a cool programmer and to buy a flat in Moscow. To become a cool programmer, he needs at least p experience points, and a desired flat in Moscow costs q dollars. Mikhail is determined to follow his dreams and registered at a freelance site.

He has suggestions to work on n distinct projects. Mikhail has already evaluated that the participation in the i-th project will increase his experience by a_i per day and bring b_i dollars per day. As freelance work implies flexible working hours, Mikhail is free to stop working on one project at any time and start working on another project. Doing so, he receives the respective share of experience and money. Mikhail is only trying to become a cool programmer, so he is able to work only on one project at any moment of time.

Find the real value, equal to the minimum number of days Mikhail needs to make his dream come true.

For example, suppose Mikhail is suggested to work on three projects and $a_1 = 6$, $b_1 = 2$, $a_2 = 1$, $b_2 = 3$, $a_3 = 2$, $b_3 = 6$. Also, $p = 20$ and $q = 20$. In order to achieve his aims Mikhail has to work for 2.5 days on both first and third projects. Indeed, $a_1 \cdot 2.5 + a_2 \cdot 0 + a_3 \cdot 2.5 = 6 \cdot 2.5 + 1 \cdot 0 + 2 \cdot 2.5 = 20$ and $b_1 \cdot 2.5 + b_2 \cdot 0 + b_3 \cdot 2.5 = 2 \cdot 2.5 + 3 \cdot 0 + 6 \cdot 2.5 = 20$.

Input

The first line of the input contains three integers n, p and q ($1 \le n \le 100\,000$, $1 \le p, q \le 1\,000\,000$) — the number of projects and the required number of experience and money.

Each of the next n lines contains two integers a_i and b_i ($1 \le a_i, b_i \le 1\,000\,000$) — the daily increase in experience and daily income for working on the i-th project.

Output

Print a real value — the minimum number of days Mikhail needs to get the required amount of experience and money. Your answer will be considered correct if its absolute or relative error does not exceed 10^{-6}.

Namely: let's assume that your answer is a, and the answer of the jury is b. The checker program will consider your answer correct, if $\frac{|a-b|}{\max(1,b)} \le 10^{-6}$.

Examples

input

```
3 20 20
6 2
1 3
2 6
```

output

```
5.000000000000000
```

input

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

output

```
0.400000000000000
```

Note

First sample corresponds to the example in the problem statement.