A. 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.

Sample test(s)

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.

B. 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 ($2 \leq n \leq 100\,000, 1 \leq m \leq 100\,000, n-1 \leq m \leq \frac{n(n-1)}{2}$) — 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 ($1 \leq a_j \leq 10^9, b_j = \{0, 1\}$). 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) ($1 \leq u_j, v_j \leq n, u_j \neq v_j$), 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.

Sample test(s)

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
```

C. 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 $\frac{|a-b|}{\max(1, b)} \le 10^{-6}$.

Sample test(s)

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.

D. Board Game

time limit per test: 2.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are playing a board card game. In this game the player has two characteristics, x and y — the white magic skill and the black magic skill, respectively. There are n spell cards lying on the table, each of them has four characteristics, a_i, b_i, c_i and d_i. In one move a player can pick one of the cards and cast the spell written on it, but only if first two of it's characteristics meet the requirement $a_i \leq x$ and $b_i \leq y$, i.e. if the player has enough magic skill to cast this spell. However, after casting the spell the characteristics of a player change and become equal to $x = c_i$ and $y = d_i$.

At the beginning of the game both characteristics of a player are equal to zero. The goal of the game is to cast the n-th spell. Your task is to make it in as few moves as possible. You are allowed to use spell in any order and any number of times (for example, you may not use some spells at all).

Input

The first line of the input contains a single integer n $(1 \leq n \leq 100\,000)$ — the number of cards on the table.

Each of the next n lines contains four integers a_i, b_i, c_i, d_i $(0 \leq a_i, b_i, c_i, d_i \leq 10^9)$ — the characteristics of the corresponding card.

Output

In the first line print a single integer k — the minimum number of moves needed to cast the n-th spell and in the second line print k numbers — the indices of the cards in the order in which you should cast them. In case there are multiple possible solutions, print any of them.

If it is impossible to cast the n-th spell, print -1.

Sample test(s)

input

```
4
0 0 3 4
2 2 5 3
4 1 1 7
5 3 8 8
```

output

```
3
1 2 4
```

input

```
2
0 0 4 6
5 1 1000000000 1000000000
```

output

```
-1
```

E. Intergalaxy Trips

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

The scientists have recently discovered wormholes — objects in space that allow to travel very long distances between galaxies and star systems.

The scientists know that there are n galaxies within reach. You are in the galaxy number 1 and you need to get to the galaxy number n. To get from galaxy i to galaxy j, you need to fly onto a wormhole (i, j) and in exactly one galaxy day you will find yourself in galaxy j.

Unfortunately, the required wormhole is not always available. Every galaxy day they disappear and appear at random. However, the state of wormholes does not change within one galaxy day. A wormhole from galaxy i to galaxy j exists during each galaxy day taken separately with probability p_{ij}. You can always find out what wormholes exist at the given moment. At each moment you can either travel to another galaxy through one of wormholes that exist at this moment or you can simply wait for one galaxy day to see which wormholes will lead from your current position at the next day.

Your task is to find the expected value of time needed to travel from galaxy 1 to galaxy n, if you act in the optimal way. It is guaranteed that this expected value exists.

Input

The first line of the input contains a single integer n $(1 \le n \le 1000)$ — the number of galaxies within reach.

Then follows a matrix of n rows and n columns. Each element p_{ij} represents the probability that there is a wormhole from galaxy i to galaxy j. All the probabilities are given in percents and are integers. It is guaranteed that all the elements on the main diagonal are equal to 100.

Output

Print a single real value — the expected value of the time needed to travel from galaxy 1 to galaxy n if one acts in an optimal way. 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 $\frac{|a-b|}{\max(1, b)} \le 10^{-6}$.

Sample test(s)

input

```
3
100 50 50
0 100 80
0 0 100
```

output

```
1.750000000000000
```

input

```
2
100 30
40 100
```

output

```
3.333333333333333
```

Note

In the second sample the wormhole from galaxy 1 to galaxy 2 appears every day with probability equal to 0.3. The expected value of days one needs to wait before this event occurs is $\frac{1}{0.3} = 3\frac{1}{3}$.