A. Round House

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Vasya lives in a round building, whose entrances are numbered sequentially by integers from 1 to n. Entrance n and entrance 1 are adjacent.

Today Vasya got bored and decided to take a walk in the yard. Vasya lives in entrance a and he decided that during his walk he will move around the house b entrances in the direction of increasing numbers (in this order entrance n should be followed by entrance 1). The negative value of b corresponds to moving $|b|$ entrances in the order of decreasing numbers (in this order entrance 1 is followed by entrance n). If $b = 0$, then Vasya prefers to walk beside his entrance.

Illustration for $n = 6$, $a = 2$, $b = -5$.

Help Vasya to determine the number of the entrance, near which he will be at the end of his walk.

Input

The single line of the input contains three space-separated integers n, a and b ($1 \le n \le 100$, $1 \le a \le n$, $-100 \le b \le 100$) — the number of entrances at Vasya's place, the number of his entrance and the length of his walk, respectively.

Output

Print a single integer k ($1 \le k \le n$) — the number of the entrance where Vasya will be at the end of his walk.

Examples

input
```
6 2 -5
```
output
```
3
```

input
```
5 1 3
```
output
```
4
```

input
```
3 2 7
```
output
```
3
```

Note

The first example is illustrated by the picture in the statements.

B. Qualifying Contest

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Very soon Berland will hold a School Team Programming Olympiad. From each of the m Berland regions a team of two people is invited to participate in the olympiad. The qualifying contest to form teams was held and it was attended by n Berland students. There were at least two schoolboys participating from each of the m regions of Berland. The result of each of the participants of the qualifying competition is an integer score from 0 to 800 inclusive.

The team of each region is formed from two such members of the qualifying competition of the region, that none of them can be replaced by a schoolboy of the same region, not included in the team and who received a **greater** number of points. There may be a situation where a team of some region can not be formed uniquely, that is, there is more than one school team that meets the properties described above. In this case, the region needs to undertake an additional contest. The two teams in the region are considered to be different if there is at least one schoolboy who is included in one team and is not included in the other team. It is guaranteed that for each region at least two its representatives participated in the qualifying contest.

Your task is, given the results of the qualifying competition, to identify the team from each region, or to announce that in this region its formation requires additional contests.

Input

The first line of the input contains two integers n and m ($2 \leq n \leq 100\,000$, $1 \leq m \leq 10\,000$, $n \geq 2m$) — the number of participants of the qualifying contest and the number of regions in Berland.

Next n lines contain the description of the participants of the qualifying contest in the following format: Surname (a string of length from 1 to 10 characters and consisting of large and small English letters), region number (integer from 1 to m) and the number of points scored by the participant (integer from 0 to 800, inclusive).

It is guaranteed that all surnames of all the participants are distinct and at least two people participated from each of the m regions. The surnames that only differ in letter cases, should be considered distinct.

Output

Print m lines. On the i-th line print the team of the i-th region — the surnames of the two team members in an arbitrary order, or a single character "?" (without the quotes) if you need to spend further qualifying contests in the region.

Examples

input

```
5 2
Ivanov 1 763
Andreev 2 800
Petrov 1 595
Sidorov 1 790
Semenov 2 503
```

output

```
Sidorov Ivanov
Andreev Semenov
```

input

```
5 2
Ivanov 1 800
Andreev 2 763
Petrov 1 800
Sidorov 1 800
Semenov 2 503
```

output

```
?
Andreev Semenov
```

Note

In the first sample region teams are uniquely determined.

In the second sample the team from region 2 is uniquely determined and the team from region 1 can have three teams: "Petrov"-"Sidorov", "Ivanov"-"Sidorov", "Ivanov" -"Petrov", so it is impossible to determine a team uniquely.

C. Tanya and Toys

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

In Berland recently a new collection of toys went on sale. This collection consists of 10^9 types of toys, numbered with integers from 1 to 10^9. A toy from the new collection of the i-th type costs i bourles.

Tania has managed to collect n different types of toys $a_1, a_2, ..., a_n$ from the new collection. Today is Tanya's birthday, and her mother decided to spend no more than m bourles on the gift to the daughter. Tanya will choose several different types of toys from the new collection as a gift. Of course, she does not want to get a type of toy which she already has.

Tanya wants to have as many distinct types of toys in her collection as possible as the result. The new collection is too diverse, and Tanya is too little, so she asks you to help her in this.

Input

The first line contains two integers n ($1 \le n \le 100\,000$) and m ($1 \le m \le 10^9$) — the number of types of toys that Tanya already has and the number of bourles that her mom is willing to spend on buying new toys.

The next line contains n distinct integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 10^9$) — the types of toys that Tanya already has.

Output

In the first line print a single integer k — the number of different types of toys that Tanya should choose so that the number of different types of toys in her collection is maximum possible. Of course, the total cost of the selected toys should not exceed m.

In the second line print k distinct space-separated integers $t_1, t_2, ..., t_k$ ($1 \le t_i \le 10^9$) — the types of toys that Tanya should choose.

If there are multiple answers, you may print any of them. Values of t_i can be printed in any order.

Examples

input

```
3 7
1 3 4
```

output

```
2
2 5
```

input

```
4 14
4 6 12 8
```

output

```
4
7 2 3 1
```

Note

In the first sample mom should buy two toys: one toy of the 2-nd type and one toy of the 5-th type. At any other purchase for 7 bourles (assuming that the toys of types 1, 3 and 4 have already been bought), it is impossible to buy two and more toys.

D. Bicycle Race

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Maria participates in a bicycle race.

The speedway takes place on the shores of Lake Lucerne, just repeating its contour. As you know, the lake shore consists only of straight sections, directed to the north, south, east or west.

Let's introduce a system of coordinates, directing the Ox axis from west to east, and the Oy axis from south to north. As a starting position of the race the southernmost point of the track is selected (and if there are several such points, the most western among them). The participants start the race, moving to the north. At all straight sections of the track, the participants travel in one of the four directions (north, south, east or west) and change the direction of movement only in bends between the straight sections. The participants, of course, never turn back, that is, they do not change the direction of movement from north to south or from east to west (or vice versa).

Maria is still young, so she does not feel confident at some turns. Namely, Maria feels insecure if at a failed or untimely turn, she gets into the water. In other words, Maria considers the turn dangerous if she immediately gets into the water if it is ignored.

Help Maria get ready for the competition — determine the number of dangerous turns on the track.

Input

The first line of the input contains an integer n ($4 \le n \le 1000$) — the number of straight sections of the track.

The following $(n+1)$-th line contains pairs of integers (x_i, y_i) ($-10\,000 \le x_i, y_i \le 10\,000$). The first of these points is the starting position. The i-th straight section of the track begins at the point (x_i, y_i) and ends at the point (x_{i+1}, y_{i+1}).

It is guaranteed that:

- the first straight section is directed to the north;
- the southernmost (and if there are several, then the most western of among them) point of the track is the first point;
- the last point coincides with the first one (i.e., the start position);
- any pair of straight sections of the track has no shared points (except for the neighboring ones, they share exactly one point);
- no pair of points (except for the first and last one) is the same;
- no two adjacent straight sections are directed in the same direction or in opposite directions.

Output

Print a single integer — the number of dangerous turns on the track.

Examples

input

```
6
0 0
0 1
1 1
1 2
2 2
2 0
0 0
```

output

```
1
```

input

```
16
1 1
1 5
3 5
3 7
2 7
2 9
6 9
6 7
5 7
5 3
4 3
4 4
3 4
3 2
5 2
5 1
1 1
```

```
output
```

```
6
```

Note

The first sample corresponds to the picture:

The picture shows that you can get in the water under unfortunate circumstances only at turn at the point $(1, 1)$. Thus, the answer is 1.

E. New Reform

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Berland has n cities connected by m bidirectional roads. No road connects a city to itself, and each pair of cities is connected by no more than one road. It is **not guaranteed** that you can get from any city to any other one, using only the existing roads.

The President of Berland decided to make changes to the road system and instructed the Ministry of Transport to make this reform. Now, each road should be unidirectional (only lead from one city to another).

In order not to cause great resentment among residents, the reform needs to be conducted so that there can be as few separate cities as possible. A city is considered *separate*, if no road leads into it, while it is allowed to have roads leading from this city.

Help the Ministry of Transport to find the minimum possible number of separate cities after the reform.

Input

The first line of the input contains two positive integers, n and m — the number of the cities and the number of roads in Berland ($2 \le n \le 100\,000$, $1 \le m \le 100\,000$).

Next m lines contain the descriptions of the roads: the i-th road is determined by two distinct integers x_i, y_i ($1 \le x_i, y_i \le n$, $x_i \ne y_i$), where x_i and y_i are the numbers of the cities connected by the i-th road.

It is guaranteed that there is no more than one road between each pair of cities, but it is not guaranteed that from any city you can get to any other one, using only roads.

Output

Print a single integer — the minimum number of separated cities after the reform.

Examples

input

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

output

```
1
```

input

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

output

```
0
```

input

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

output

```
1
```

Note

In the first sample the following road orientation is allowed: $1 \to 2, 1 \to 3, 3 \to 4$.

The second sample: $1 \to 2, 3 \to 1, 2 \to 3, 2 \to 5, 3 \to 4$.

The third sample: $1 \to 2, 2 \to 3, 4 \to 5, 5 \to 6, 6 \to 4$.

F. Polycarp and Hay

time limit per test: 4 seconds

memory limit per test: 512 megabytes

input: standard input

output: standard output

The farmer Polycarp has a warehouse with hay, which can be represented as an $n \times m$ rectangular table, where n is the number of rows, and m is the number of columns in the table. Each cell of the table contains a haystack. The height in meters of the hay located in the i-th row and the j-th column is equal to an integer $a_{i,j}$ and coincides with the number of cubic meters of hay in the haystack, because all cells have the size of the base 1×1. Polycarp has decided to tidy up in the warehouse by removing an arbitrary integer amount of cubic meters of hay from the top of each stack. You can take different amounts of hay from different haystacks. Besides, it is allowed not to touch a stack at all, or, on the contrary, to remove it completely. If a stack is completely removed, the corresponding cell becomes empty and no longer contains the stack.

Polycarp wants the following requirements to hold after the reorganization:

- the total amount of hay remaining in the warehouse must be **equal** to k,
- the heights of all stacks (i.e., cells containing a non-zero amount of hay) should be the same,
- the height of at least one stack must remain the same as it was,
- for the stability of the remaining structure all the stacks should form one connected region.

The two stacks are considered adjacent if they share a side in the table. The area is called connected if from any of the stack in the area you can get to any other stack in this area, moving only to adjacent stacks. In this case two adjacent stacks necessarily belong to the same area.

Help Polycarp complete this challenging task or inform that it is impossible.

Input

The first line of the input contains three integers n, m $(1 \le n, m \le 1000)$ and k $(1 \le k \le 10^{18})$ — the number of rows and columns of the rectangular table where heaps of hay are lain and the required total number cubic meters of hay after the reorganization.

Then n lines follow, each containing m positive integers $a_{i,j}$ $(1 \le a_{i,j} \le 10^9)$, where $a_{i,j}$ is equal to the number of cubic meters of hay making the hay stack on the i-th row and j-th column of the table.

Output

In the first line print "YES" (without quotes), if Polycarpus can perform the reorganisation and "NO" (without quotes) otherwise. If the answer is "YES" (without quotes), then in next n lines print m numbers — the heights of the remaining hay stacks. All the remaining non-zero values should be equal, represent a connected area and at least one of these values shouldn't be altered.

If there are multiple answers, print any of them.

Examples

input

```
2 3 35
10 4 9
9 9 7
```

output

```
YES
7 0 7
7 7 7
```

input

```
4 4 50
5 9 1 1
5 1 1 5
5 1 5 5
5 5 7 1
```

output

```
YES
5 5 0 0
5 0 0 5
5 0 5 5
5 5 5 0
```

input

```
2 4 12
1 1 3 1
1 6 2 4
```

output

```
NO
```

Note

In the first sample non-zero values make up a connected area, their values do not exceed the initial heights of hay stacks. All the non-zero values equal 7, and their number is 5, so the total volume of the remaining hay equals the required value $k = 7 \cdot 5 = 35$. At that the stack that is on the second line and third row remained unaltered.

G. Fence Divercity

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

Long ago, Vasily built a good fence at his country house. Vasily calls a fence *good*, if it is a series of n consecutively fastened vertical boards of centimeter width, the height of each in centimeters is **a positive integer**. The house owner remembers that the height of the i-th board to the left is h_i.

Today Vasily decided to change the design of the fence he had built, by cutting his top connected part so that the fence remained good. The cut part should consist of only the upper parts of the boards, while the adjacent parts must be interconnected (share a non-zero length before cutting out of the fence).

You, as Vasily's curious neighbor, will count the number of possible ways to cut exactly one part as is described above. Two ways to cut a part are called distinct, if for the remaining fences there is such i, that the height of the i-th boards vary.

As Vasily's fence can be very high and long, get the remainder after dividing the required number of ways by $1\,000\,000\,007$ (10^9+7).

Input

The first line contains integer n $(1 \le n \le 1\,000\,000)$ — the number of boards in Vasily's fence.

The second line contains n space-separated numbers $h_1, h_2, ..., h_n$ $(1 \le h_i \le 10^9)$, where h_i equals the height of the i-th board to the left.

Output

Print the remainder after dividing r by $1\,000\,000\,007$, where r is the number of ways to cut exactly one connected part so that the part consisted of the upper parts of the boards and the remaining fence was *good*.

Examples

input

```
2
1 1
```

output

```
0
```

input

```
3
3 4 2
```

output

```
13
```

Note

From the fence from the first example it is impossible to cut exactly one piece so as the remaining fence was *good*.

All the possible variants of the resulting fence from the second sample look as follows (the grey shows the cut out part):