30. HRVATSKA INFORMATIČKA OLIMPIJADA

ZAGREB, 28. TRAVNJA 2024.

Croatian Olympiad in Informatics

April 28th 2024

Tasks

Task	Time Limit	Memory Limit	Score
CERN	4 seconds	512 MiB	100
Koreografija	8 seconds	1024 MiB	100
Ministarstvo	1 second	1024 MiB	100
Sirologija	1 second	1024 MiB	100
Total			400

Task CERN

CERN is an international institution focused on nuclear research and particle physics. The particle accelerator system at CERN is used to conduct experiments involving the collision of particles at high speeds.

We consider N particles arranged in a sequence. Each particle is defined by its *type* v_i, represented by a positive integer between 1 and N.

In the latest research, it is necessary to conduct Q experiments. In the i-th experiment, we observe all particles from the l_i-th to the r_i-th in the sequence ($l_i < r_i$). Among the observed particles, we can choose any two particles of different types and collide them in the accelerator, causing both particles to be destroyed. We repeat this collision process as long as there are two particles of different types among the observed particles. The experiment ends either when all observed particles are destroyed or when there are some particles of the same type remaining. Of course, depending on the order in which we collide the particles, it is possible to end up with various types of particles at the end.

Since particle collision is not cheap, you have decided to conduct experiments only in theory. Now, for each experiment, you are interested in how many types of particles exist such that it is possible to end the experiment with some remaining particles of that type.

Input

The first line contains two positive integers N and Q, the number of particles and the number of experiments, respectively.

The next line contains a sequence of N numbers $v_1, \ldots, v_N$, representing the types of particles.

In each of the following Q lines, there is a pair of two positive integers l_i and r_i ($1 \leq l_i < r_i \leq N$), representing the interval of observed particles in the i-th experiment.

Output

For each of the Q experiments, print in a separate line the requested number of types of particles with which it is possible to end the experiment.

Scoring

In all subtasks, $2 \leq N \leq 500,000$ and $1 \leq Q \leq 500,000$.

Subtask	Score	Constraints
1	13	$v_i \leq 10$ for each $i = 1, \ldots, N$.
2	19	There are at most two particles of each type.
3	17	$N, Q \leq 2000$
4	19	$N, Q \leq 100,000$
5	32	There are no additional constraints.

Example

input

```
11 5
2 4 2 3 4 4 3 1 4 4 4
1 4
2 8
6 9
8 10
8 11
```

output

```
1
4
1
1
1
```

Explanation of the first example:

In the first experiment, we can collide particles of types 3 and 4, leaving two particles of type 2 remaining. There is no way to end up with any other type of particles.

In the second experiment, it is possible to end up with some remaining particles of each type.

In the fourth and fifth experiments, regardless of the choice of collisions, some particles of type 4 will remain at the end.

Task Koreografija

Jura: *And Tvrtko, how was the show yesterday?*

Tvrtko: *It was great. The best part was when 1000 dancers lined up from left to right and started performing the choreography. Each of them had a number written on their costume between 1 and 1000, and all those numbers were different. But I have to admit, when I saw them in a line, I didn't like their order.*

Jura: *What do you mean?*

Tvrtko: *I saw some consecutive interval of dancers in the line and counted how many pairs of dancers there were such that the dancer at the lower position had a higher number than the dancer at the higher position. I like it when the number of such pairs is an odd number.*

Jura: *Oh Tvrtko, you have to see the big picture. I'll handle it. But tell me, how did their numbers go in order?*

Tvrtko: *Hm... I've already forgotten. But I can tell you for each consecutive interval of dancers whether I liked it or not.*

Jura: *So be it. We have no choice but to try to determine their numbers based on that.*

Interaction

This is an interactive task. Your program needs to establish a dialogue with the program made by the organizers that responds to the queries asked.

Your program can send queries by writing to standard output. Each query should be printed in a separate line and should have the form "`?` a b", where a and b are positive integers satisfying $1 \leq a \leq b \leq 1000$. Numbers a and b represent the positions of the dancers that define the observed interval.

After each printed query, your program should *flush* the output and read the *response* to the query from standard input – a number from the set $\{0, 1\}$ which represents Tvrtko's opinion on the given interval. The number 1 indicates that Tvrtko liked that interval, while 0 indicates he didn't.
Your program may send at most $500\,000$ **such queries.**

Once your program has reconstructed the numbers on the dancers' costumes, it should print in a separate line to standard output the symbol `!` followed by printing the requested sequence of numbers as they appear from left to right.

After that, your program should again *flush* the output and terminate execution.

Scoring

Let Q be the maximum number of queries your program sends in all test cases.

If $Q > 500\ 000$, your program will score 0 points.

Otherwise, the number of points your program will score is based on the following table:

Range	Score
$40\ 000 \leq Q \leq 500\ 000$	$30 + 70 \cdot \frac{1/Q - 1/500\,000}{1/40\,000 - 1/500\,000}$
$Q \leq 40\ 000$	100

Sample Interaction

Although in the task the number of dancers will always be 1000, for illustration purposes we provide an example interaction when the number of dancers is 4.

Let's assume the numbers on the dancers' costumes go in order `2 1 4 3`.

Output	Input	Note
`? 1 2`	`1`	Tvrtko counted one pair.
`? 1 3`	`1`	Tvrtko counted one pair.
`? 1 4`	`0`	Tvrtko counted two pairs.
`? 2 3`	`0`	Tvrtko counted zero pairs.
`? 2 4`	`1`	Tvrtko counted one pair.
`? 3 4`	`1`	Tvrtko counted one pair.
`!`		The numbers have been found, they are printed in order.
`2 1 4 3`		

Task Ministarstvo

After a successful career in a party we won't name, Pero got a job at the Ministry of Tourism. Pero oversees a network of N cities, labeled with numbers from 1 to N, where there is exactly one, one-way road between each pair of cities. In order to increase revenue, he has decided to introduce permits for traffic. Pero would prefer to introduce a special permit for each road, but that would alert his superiors. Therefore, he will introduce K different permits, labeled from 1 to K, and possession of a specific permit will be required to travel on each road.

To still ensure substantial revenue, Pero will settle for the following property.

- For each city v, there is some city u, such that it is not possible to travel from city v to city u with just one permit.

Pero asks for your help to determine the minimum K such that there exists an assignment of permits with the desired property, if such an assignment exists! If no such assignment exists, output `-1`.

Input

The first line contains a positive integer N.

The i-th of the following N lines contains N numbers $a_{i,j}$ where $a_{i,j} = 1$ if there is a road from city i to city j. Note that $a_{i,i} = 0$ and for $i \neq j$, exactly one of the numbers $a_{i,j}$ and $a_{j,i}$ is non-zero.

Output

If there is no assignment with the desired property, output `-1` in the first and only line.

Otherwise, output the minimum positive integer K in the first line.

In the following N lines, output the description of the assignment.

In the i-th line, output N numbers $b_{i,j}$ where if $a_{i,j} = 0$, then $b_{i,j} = 0$, otherwise $1 \leq b_{i,j} \leq K$ indicating which permit is required for traveling on that road.

Scoring

In all subtasks, $2 \leq N \leq 1000$. In each subtask, 15% of the points come from only deciding whether such an assignment exists or not. For these points, if you don't output `-1`, you need to output some assignment, but it doesn't have to satisfy Pero's desired property.

Subtask	Score	Constraints
1	20	$N \leq 5$
2	80	No additional constraints.

Examples

input

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

output

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

input

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

output

```
-1
```

input

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

output

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

Explanation for the third sample test:

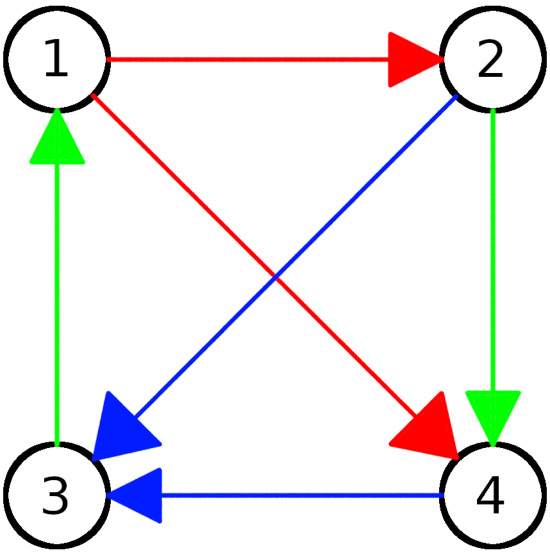
Roads requiring the first permit are marked in red, the second permit in blue, and the third permit in green.

From city 1, it is not possible to reach city 3 using just one permit.

From city 2, it is not possible to reach city 1 using just one permit.

From city 3, it is not possible to reach city 2 using just one permit.

From city 4, it is not possible to reach city 1 using just one permit.



Task Sirologija

You are an ant, but not just any ant – you're an ant obsessed with cheeseology!

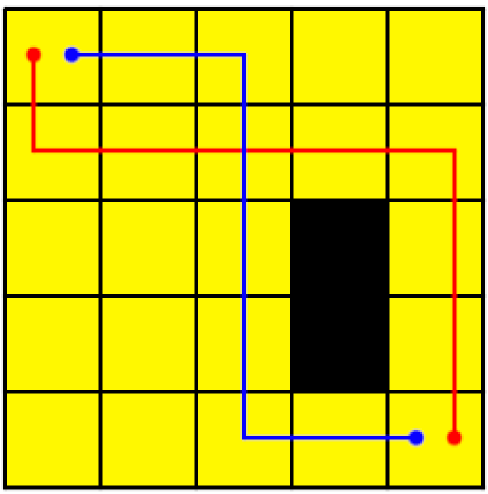
You've discovered a new slice of cheese in the kitchen and want to send as many of your minions as possible to explore it. Imagine the cheese as a table with N rows and M columns, where the rows are labeled from 1 to N from top to bottom, and the columns are labeled from 1 to M from left to right. Some fields contain holes, while others contain cheese. We will denote the field in the r-th row and s-th column as (r, s). The top-left and bottom-right fields will definitely contain cheese.

Let's denote the number of minions as K. Your minions will start their exploration in the top-left field and finish in the bottom-right field. They can only move downwards and to the right. Additionally, their paths must not "cross", meaning we can assign labels from 1 to K to them in such a way that there is no field from which a minion with a lower label exited to the right, and a minion with a higher label exited downwards.

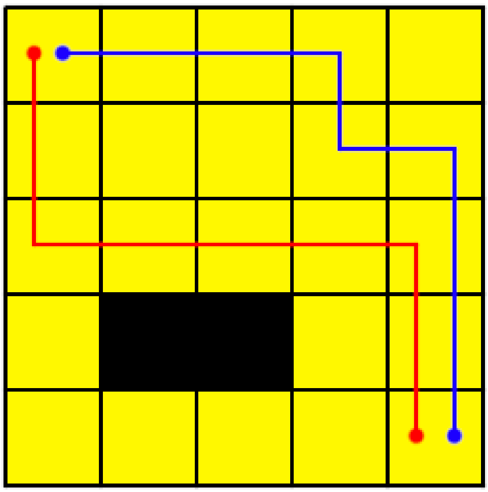
Moreover, you would like these paths to be "different" in some sense, meaning that for every two minions, there exists a field (r, s) containing a hole, such that one of them was at some point in column s and in a row labeled lower than r, while the other was at some point (not necessarily simultaneously) in column s and in a row labeled higher than r. Informally, every pair of minions approached some hole from different sides.

Output the maximum value of K such that there exists a selection of minion paths satisfying the given conditions.

Some examples of paths that do not satisfy the conditions:



(a) Invalid choice of paths - they intersect



(b) Invalid choice of paths - they approach a hole from the same side

Input

The first line contains positive integers N, M.

The next N lines contain descriptions of the table rows. The i-th line contains M characters, where . denotes cheese and # denotes a field containing a hole.

Output

Output the maximum possible value of the number K in a single line.

Scoring

In all subtasks, $2 \leq N, M \leq 2000$.

Subtask	Score	Constraints
1	15	All holes are in the same row.
2	18	$N, M \leq 10$
3	16	$N, M \leq 50$, there are no holes in the first or last row or in the first or last column.
4	18	$N, M \leq 50$
5	16	$N, M \leq 2000$, there are no holes in the first or last row or in the first or last column.
6	17	No additional constraints.

Examples

input

```
5 5
.....
.#...
.....
...#.
.....
```

output

```
3
```

input

```
5 5
....#
....#
.....
.....
#....
```

output

```
1
```

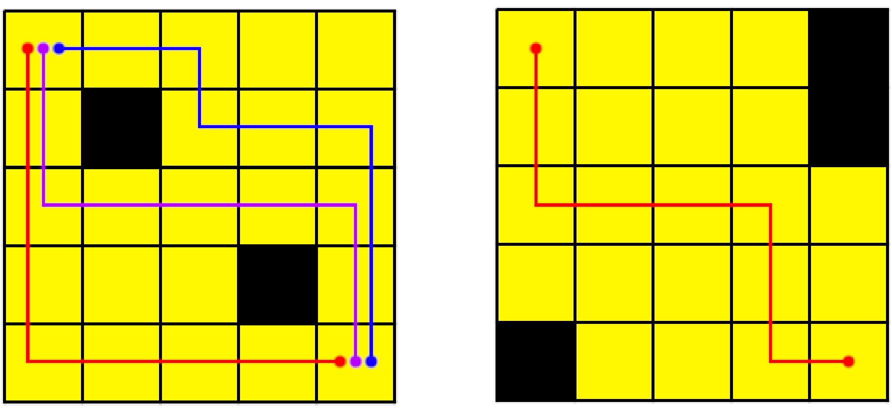
input

```
3 2
.#
#.
..
```

output

```
0
```

Explanation of the first and second example:



(a) Example of paths for the first sample (b) Example of paths for the second sample