Croatian Olympiad in Informatics

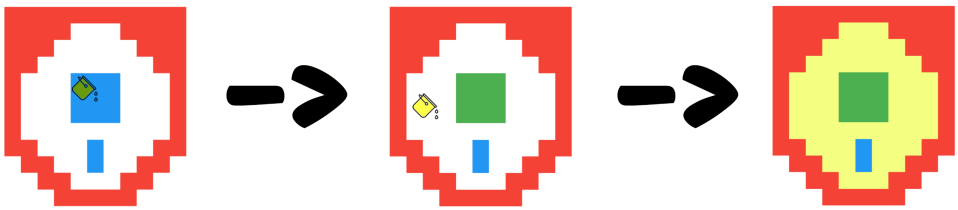
October 3rd 2020

Tasks

Task	Time Limit	Memory Limit	Score
Paint	3 seconds	512 MiB	100
Pastiri	1 second	512 MiB	100
Semafor	4 seconds	512 MiB	100
Zagrade	3 seconds	512 MiB	100
Total			400

Task Paint

We will represent the drawing area of *MS Paint* as a rectangular grid of unit squares divided into R rows and S columns. Each square of the grid represents a single pixel that can be colored in one of the 10^5 different colors. When the user applies the so called *bucket tool* with color A on a pixel (r, s) which is colored in the color B, then all pixels in the *monochrome neighborhood* of pixel (r, s) change their color to A. Monochrome neighborhood of a pixel (r, s) is a set of pixels that are reachable by *walking* from (r, s) in the four general directions (up, down, left and right) without changing the color of the pixel along the way. Note that the pixel (r, s) is itself a part of its monochrome neighborhood.



You are given a starting image drawn in *MS Paint* along with Q instructions that should be executed in the given order. Each instruction tells you on which pixel should you apply the bucket tool and with what color. Your task is to how the image looks like after all instructions are executed.

Input

The first line contains integers R and S from the task description.

Each of the next R lines contains S non-negative integers less than 100 000 that represent the starting image drawn in *MS Paint*. More precisely, the j-th number in the i-th row of the image represents the color of the pixel (i, j).

The next line contains an integer Q from the task description.

The i-th of the next Q lines contains integers r_i, s_i and c_i ($1 \leq r_i \leq R, 1 \leq s_i \leq S, 0 \leq c_i < 100\,000$), which represent the i-th instruction that tells you to use the bucket tool with color c_i on the pixel (r_i, s_i).

Output

You should output the final state of the image in the same format as it was given in the input.

Scoring

Subtask	Score	Constraints
1	8	$1 \leq R \cdot S \leq 10\,000$, $1 \leq Q \leq 10\,000$
2	9	$R = 1$, $1 \leq S \leq 200\,000$, $1 \leq Q \leq 100\,000$
3	31	$1 \leq R \cdot S \leq 200\,000$, $1 \leq Q \leq 100\,000$ Each pixel will in every moment be colored either in color 0 or color 1.
4	52	$1 \leq R \cdot S \leq 200\,000$, $1 \leq Q \leq 100\,000$

Examples

input

```
12 11
1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 0 0 0 1 1 1 1
1 1 1 0 0 0 0 0 1 1 1
1 1 0 0 0 0 0 0 0 1 1
1 0 0 0 2 2 2 0 0 0 1
1 0 0 0 2 2 2 0 0 0 1
1 0 0 0 2 2 2 0 0 0 1
1 0 0 0 0 0 0 0 0 0 1
1 1 0 0 0 2 0 0 0 1 1
0 1 1 0 0 2 0 0 1 1 0
0 0 1 1 0 0 0 1 1 0 0
0 0 0 1 1 1 1 1 0 0 0
2
5 5 3
6 2 4
```

output

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

input

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

output

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

input

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

output

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

Clarification of the first example: The figure from the task description corresponds to the input of the first example. White color corresponds to number 0, red color corresponds to number 1, blue color corresponds to number 2, green color corresponds to number 3 and yellow color corresponds to number 4.

Task Pastiri

„*I never felt so full that I couldn't eat one more lamb.*" – Mr. Malnar

A flock of K sheep lives in a tree, a simple connected graph without a cycle. The tree contains N nodes denoted with integers from 1 to N. Each node of a tree is a home to at most one sheep. A wise shepherd realized that, sooner or later, wolves will learn how to climb trees.

In order to protect the sheep, we need to place shepherds into some nodes such that each sheep is protected by at least one shepherd. It is known that **each shepherd protects all sheep that are closest to him**, and only them. The distance between some sheep and some shepherd is equal to the number of nodes on a unique path between the node containing the sheep and the node containing the shepherd (inclusive). Additionally, the shepherd can share a node with a sheep. Of course, in that case he protects only that sheep.

Determine **the minimal number of shepherds** that need to placed in the nodes of a tree such that **each sheep is protected by at least** one shepherd. Additionally, determine one such arrangement of shepherds.

Input

The first line contains integers N i K ($1 \leq K \leq N$) from the task description.

Each of the next $N-1$ lines contains two integers a_i and b_i ($1 \leq a_i, b_i \leq N$) which indicate that there is an undirected edge between nodes a_i and b_i.

The next line contains K different integers o_i ($1 \leq o_i \leq N$) that represent nodes which containing a sheep.

Output

In the first line you should output a number X which represents the minimal number of shepherds from the task description.

In the second line you should output X space-separated integers which represent the nodes containing shepherds.

If there are multiple correct solutions, you may output any of them.

Scoring

Subtask	Score	Constraints
1	8	$1 \leq N \leq 500\,000$, every node $x = 1, \ldots, n-1$ is connected with node $x+1$
2	18	$1 \leq K \leq 15$, $1 \leq N \leq 500\,000$
3	23	$1 \leq N \leq 2\,000$
4	51	$1 \leq N \leq 500\,000$

Examples

input

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

output

```
2
1 3
```

input

```
9 5
1 2
2 3
3 4
3 5
1 6
1 7
7 8
8 9
2 5 6 7 9
```

output

```
3
1 4 9
```

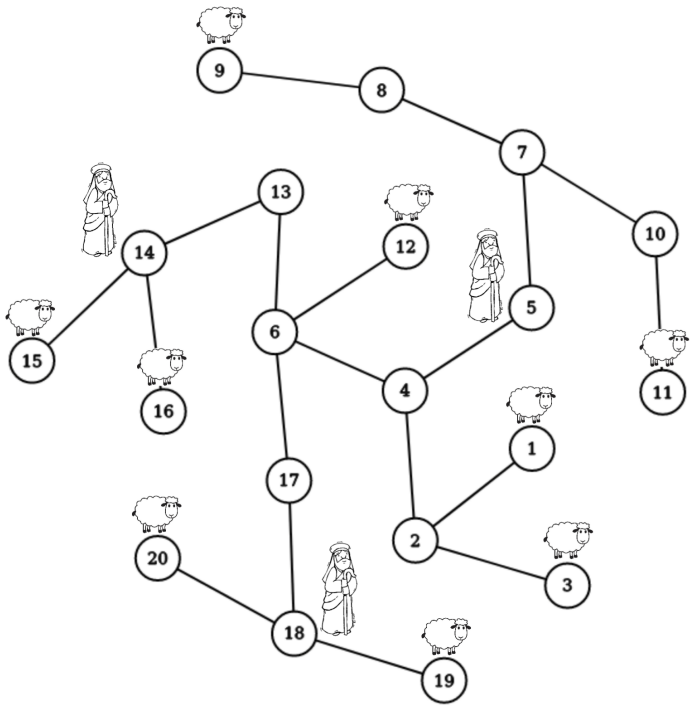
input

```
20 9
1 2
2 3
2 4
4 5
4 6
5 7
7 8
8 9
7 10
10 11
6 12
6 13
6 17
13 14
14 15
14 16
17 18
18 19
18 20
1 3 9 11 12 15 16 19 20
```

output

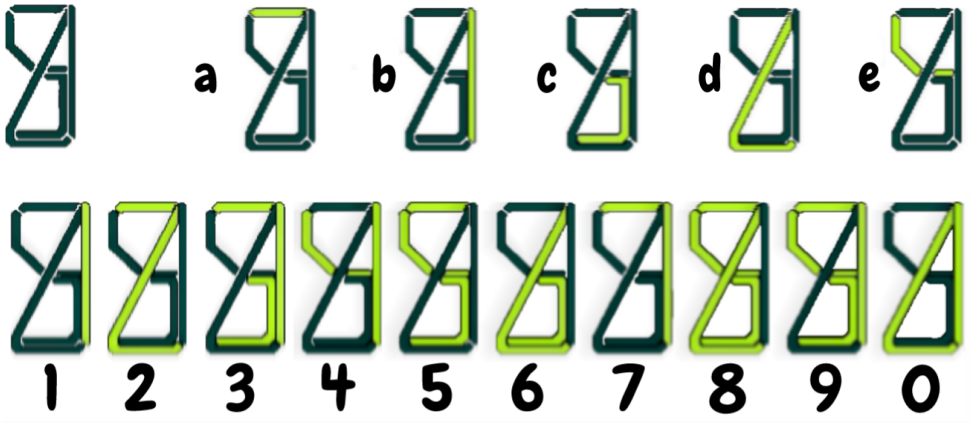
```
3
5 14 18
```

Clarification of the third example:



Task Semafor

You are most likely familiar with a so-called *7-segment* display which is widely used to depict digits on various digital devices, such as watches or calculators. Due to its simplicity, intuitiveness and aestheticism, this design has been accepted across the globe. Still, young Matej argues against the 7-segment design and claims that the same functionality can be obtained more efficiently, using only 5 segments.



5-segment display design – segments are labeled using with letters from `a` to `e`.

Matej decided to make his first entrepreneurial steps in the most prosperous branch of Croatian economy, football. His revolutionary design will be used on a substitution board during the games of 1. HNL.[1] He is currently making a presentation for the board of directors at the Croatian Football Association.

The substitution board consists of M 5-segment displays which, from left to right, represent the digits of a jersey number worn by the player that is about to leave the pitch. At the beginning of Matej's presentation, the substitution board represents the number X, and Matej will make one of the following moves each second:

- Turn on one segment that is currently turned off.
- Turn off one segment that is currently turned on.

Also, Matej will make sure that after each K-th move, the board shows a valid number. The number is valid if each of its digits is correctly shown on the corresponding display (i.e. segments that are turned on show a valid digit). Also, numbers having less than M digits are validly shown if they contain the appropriate number of leading zeros.

For each final state (integer between 0 and $10^M - 1$), Matej is interested in how many different ways could he make his moves during the presentation such that this final state is reached at the end. Of course, he needs to adhere to all limitations presented in the previous chapters. We consider two sequences of moves different if, imagining they are executed on two different boards at the same time, there is a moment at which the two boards represent a different state.

Since the number of different ways can be quite large, you are asked to compute it modulo $10^9 + 7$.

[1] The elite tier of Croatian professional football.

Input

The first line contains integers M, N, K ($1 \leq K \leq N$) and X ($0 \leq X < 10^M$) from the task description.

Output

In the i-th line you should output the number of different ways for the substitution board to show the number $i-1$ at the end of Matej's presentations. The numbers should be printed modulo 10^9+7.

Scoring

Subtask	Score	Constraints
1	6	$M = 1$, $1 \leq N \leq 12$
2	15	$M = 1$, $1 \leq N \leq 10^{15}$
3	12	$M = 2$, $1 \leq N \leq 1\,500$, $K = N$
4	12	$M = 2$, $1 \leq N \leq 10^{15}$, $1 \leq K \leq 15$
5	15	$M = 2$, $1 \leq N \leq 10^{15}$, $1 \leq K \leq 1\,500$
6	40	$M = 2$, $1 \leq N \leq 10^{15}$

Examples

input

```
1 2 1 5
```

output

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

input

```
1 3 3 8
```

output

```
0
0
0
6
0
13
0
0
0
0
```

input

```
1 4 2 4
```

output

```
24
0
8
0
37
0
4
28
4
24
```

Clarification of the first example: At the beginning of the presentation the (single-digit) substitution board shows the number 5. After each move (due to $K = 1$), the board must show a valid number. Matej is going to make a total of $N = 2$ moves. Therefore, at the end of the presentation, the board can show either number 3 or number 5. Number 3 can be obtained using one way ($5 - 9 - 3$) and number 5 can be obtained using two ways ($5 - 9 - 5$ and $5 - 8 - 5$).

Task Zagrade

It is well known that the Central Intelligence Agency is tasked with gathering, processing and analyzing national security information. It is also suspected that they own quite large collections of commonly-used computer passwords and are developing some quite sophisticated tools that are capable of compromising password-protected computer systems.

The tables have turned, your task is to compromise the security of a CIA server. Good luck!

Naturally, they are well aware of typical patterns that humans produce while coming up with their passwords, so attempts such as `123456`, `password`, `1q2w3e4r` or `welcome` are futile. Luckily, we have uncovered certain pieces of information that might be of use to you.

Namely, their master password consists of exactly N characters, where N is an even number. Exactly half of those characters is equal to the open parenthesis (`'('`), while the other half is equal to the closing parenthesis (`')'`). Additionally, instead of the usual "*forgot your password?*" functionality, their engineers have decided to expose an API to the forgetful administrator. Using the API, an administrator can execute at most Q queries asking „*whether the interval of the password from a-th to the b-th character is mathematically valid*".

The mathematical validity of a sequence of parentheses is defined inductively as:

- `()` is a mathematically valid sequence.
- If A is a mathematically valid sequence, then `(`A`)` is a mathematically valid sequence as well.
- If both A and B are mathematically valid sequences, then AB is also a mathematically valid.

Interaction

This is an interactive task. Your program must communicate with a program made by the organizers which simulates the functionality of a **fictitious** insecure CIA server from the task description.

Before interaction, your program should read an even integer Q from the standard input, which represents the number of queries that your program can make.

After that, your program can send query requests by writing to the standard output. Each query must be printed in a separate line and have the form "`?` a b", where $1 \leq a \leq b \leq N$ holds. After each query has been written, your program should **flush** the output and read the *answer* from the standard input. The answer is a 1 if the interval of the password starting from a-th and ending at the b-th character forms a mathematically valid sequence of parentheses. Otherwise, the answer is 0. Your program can make at most Q such queries.

After your program has deduced the secret password, it should write a line to the standard output in the form "`!` $x_1x_2\ldots x_N$", where characters x_1, x_2, ..., x_N represent the characters of the secret password. After that, your program should *flush* the output once more and gracefully terminate its execution.

Note: You can download the sample source code from the judging system that correctly interact with the CIA server, including the output flush.

Scoring

Subtask	Score	Constraints
1	14	$1 \leq N \leq 1\,000$, $Q = \frac{N^2}{4}$, the whole password is a mathematically valid sequence.
2	7	$1 \leq N \leq 1\,000$, $Q = \frac{N^2}{4}$
3	57	$1 \leq N \leq 100\,000$, $Q = N - 1$, the whole password is a mathematically valid sequence.
4	22	$1 \leq N \leq 100\,000$, $Q = N - 1$

Interaction Example

Input	Output	Comment
	6 9	The secret password is `((()))` of length 6, and a program may ask at most 9 queries.
`? 1 6`	`1`	The whole password is mathematically valid.
`? 1 2`	`0`	`((` isn't mathematically valid.
`? 2 4`	`0`	`(()` isn't mathematically valid.
`? 2 5`	`1`	`(())` is mathematically valid.
`? 3 4`	`1`	`()` is mathematically valid.
`! ((()))`		The password is correctly deduced and CIA is compromised.