Croatian Olympiad in Informatics

May 29th 2022

Tasks

Task	Time Limit	Memory Limit	Score
Mađioničar	5 seconds	512 MiB	100
Mensza	5 seconds	512 MiB	100
Povjerenstvo	3 seconds	512 MiB	100
Vinjete	3 seconds	512 MiB	100
Total			400

Task Mađioničar

You might have heard that in his free time, Mr. Malnar does magic. His recent appearance in the famous TV show *Penn & Teller: Fool Us* took the world by storm. He introduced himself as *The Magical Mr. Malnar*, pulled off an incredible mentalist trick, and swept everyone off their feet.

He started off by calling up an eager volounteer from the audience and asking them to think of any string of their choice that consists of exactly N letters. He then proceeded to entertain the audience, occasionally glancing at the volounteer, and at the end he declared: "the longest sub-palindrome[1] of your string has length L". After the volounteer confirmed this is indeed correct, the audience was stunned.

However, observant viewers and close friends of Mr. Malnar suspect this was not mind reading, but a clever selection of words that, when combined with excellent reading of facial expressions, gives away enough information to pull off the trick.

While it seemed like Mr. Malnar was merely fooling around, from time to time he would mention some interval of numbers $[l, r]$, where $1 \leq l \leq r \leq N$ and briefly glance at the volounteer. There are rumors saying he is able to determine whether or not the substring of the volounteer's string that consists of the l-th through the r-th letter (inclusive) is a palindrome, based on their facial expression alone.

You need to write a program which will confirm that Mr. Malnar, if the rumors are true, was able to gather enough information to determine the longest sub-palindrome of the secret string choosen by the volounteer.

Interaction

This is an interactive task. Your program must communicate with a program made by the organizers which simulates the behaviour of the volounteer from the task description.

Before interaction, your program should read an integer N, the length of the secret string, from the standard input task statement.

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 "`?` l r", where $1 \leq l \leq r \leq N$ 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 substring $[l, r]$ is a palindrome, or 0 if it's not. **Your program can make at most** 200 000 **such queries.**

After your program has deduced the length of the longest sub-palindrome, it should write a line to the standard output in the form "`!` L", where L is the said length. 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 performs the interaction correctly, including the output flush.

[1] A *palindrome* is a string that reads the same backward or forward. A *substring* of a string is a string made up from the l-th through the r-rh letter of that string, for some $1 \leq l \leq r \leq N$. A *sub-palindrome* is a substring which is also a palindrome.

Scoring

Subtask	Score	Constraints
1	13	$1 \leq N \leq 7\,500$
2	25	$1 \leq N \leq 65\,000$
3	25	$1 \leq N \leq 100\,000$, the secret string consists of letters `a` and `b` only
4	37	$1 \leq N \leq 100\,000$

Interaction Example

Output	Input	Comment
	5	The secret string has length 5. In this example, the volounteer chose the string `neven`.
`? 1 1`	1	Substring `n` is a palindrome.
`? 2 3`	0	Substring `ev` isn't a palindrome.
`? 2 4`	1	Substring `eve` is a palindrome.
`? 3 5`	0	Substring `ven` isn't a palindrome.
`? 1 5`	1	Substring `neven` is a palindrome.
`! 5`		Correct, the longest sub-palindrome is the whole string `neven`.

Task Mensza

Mr. Malnar recently founded *Mensza* – the greatest, most prestigious and the only international association of highly-intelligent gastronomy enthusiasts. As you might have guessed, not anyone can join the association, only special candidates that have excelled on an entrance exam. Of course, the exam consists of an IQ-related part, and a food-related part. We will showcase an example of an IQ-related part in this task, and the contestants will have a chance to attempt the food-related part after the contest.

The entrance exam candidates in this story are Alojzije, Benjamin and Cecilija. Mr. Malnar sized them up, invented a worthy problem, and first spoke to Alojzije:

"Alojzije, you'll be the first to enter my office, where I'll show you the integer A. Then, you'll write an array of integers $a = (a_1, a_2, \ldots, a_{l_a})$ on a piece of paper, but it's length l_a must not exceed L."

After that, he turned over to Benjamin:

"Benjamin, you'll be the next to enter my office, where I'll show you the integer $B \neq A$. Then, you'll write an array of integers $b = (b_1, b_2, \ldots, b_{l_b})$ on a piece of paper, but it's length l_b must not exceed L."

Lastly, he addressed Cecilija:

"Cecliija, you'll be the last to enter my office, where I'll show you the array of integers c which I've determined based on arrays a and b. More precisely, for each number appearing in the arrays a and b, I'll append to the array c number of times that number occurs in the union of a and b. Also, I'll present the array c to you in a non-decreasing order. For example, if $a = (1, 2, 4)$ and $b = (1, 1, 2, 3)$, I'll show you $c = (1, 1, 2, 3)$ because numbers 3 and 4 appear once, number 2 twice, and number 1 thrice. After I show you the array c, you should tell me which of the integers A and B is greater."

He once more addressed all of the candidates:

"You have 60 minutes to think of a strategy and then we'll proceed with the exam. After that, you're not allowed to communicate any more. We'll repeat the whole procedure a couple of times until ~~I confirm you're not just lucky~~ the food arrives."

Your task is to think of a strategy that would allow Alojzije, Benjamin and Cecilija to pass the IQ-part of the exam.

Input

The first line contains an integer L from the task description.

The second line contains an integer Q representing the number of scenarios you must process. Each scenario corresponds to some interaction happening in the office of Mr. Malnar.

The i-th of the next Q lines describes the i-th scenario. The line will begin with either `alojzije`, `benjamin` or `cecilija`, depending on which candidate was summoned to the office.

If the i-th line begins with the word `alojzije`, then it will also contain the integer A from the task description.

If the i-th line begins with the word `benjamin`, then it will also contain the integer B from the task description.

If the i-th line begins with the word `cecilija`, then it will be continued with l_c (length of an array c), followed by the elements of c in non-decreasing order ($c_1 \leq c_2 \leq \ldots \leq c_{l_c}$).

Output

The i-th of the next Q lines should contain an answer to the i-th input scenario.

If the i-th input scenario was of the form `alojzije` A, then you should first output the number $0 \leq l_a \leq L$ (length of a), followed by the elements of a representing the array that Alozije would write on the piece of paper after being presented with the number A in Mr. Malnar's office. Elements of a need to be between 0 and 10^9 inclusive.

If the i-th input scenario was of the form `benjamin` B, then you should first output the number $0 \leq l_b \leq L$ (length of b), followed by the elements of b representing the array that Benjamin would write on the piece of paper after being presented with the number B in Mr. Malnar's office. Elements of b need to be between 0 and 10^9 inclusive.

If the i-th input scenario was of the form `cecilija` l_c c_1 ... c_{l_c}, then you need to output "`A`" if Ceclija would determine that $A > B$ based on array c. Conversely, you need to output "`B`", of Cecilija would determine that $A < B$ based on array c. You can assume that an array c will be generated based on arrays a and b that your program has produced when processing scenarios `alojzije` A and `benjamin` B. It is possible that your program has produced the arrays a and b in a previous run.

Scoring

Your solution will be tested in two runs. First it will be ran on a test case containing only scenarios of the form `alojzije` A or `benjamin` B. Assuming your program processes all scenarios and produces outputs in a valid format, it will be ran a second time. In the second run, all scenarios will begin with the word `cecilija`, and the corresponding arrays c will be generated based on various combinations of the arrays a and b your program has produced in the first run. The value of the input parameter L will be the same in both runs. If your program correctly answers all scenarios in the second run, it will be considered correct.

The execution time of your submission is the sum of execution times of both runs.

If we denote with N the maximum value of numbers A and B in all test cases of a particular subtask, your solutions will be scored according to the following table:

Subtask	Score	Constraints
1	11	$N = 100$, $L = 200$, $1 \leq Q \leq 10\,000$
2	23	$N = 1\,000$, $L = 110$, $1 \leq Q \leq 1\,000\,000$
3	66	$N = 500\,000$, $L = 20$, $1 \leq Q \leq 1\,000\,000$

Example

First Run

Input	Output	Comment
200		Arrays that Alojzije and Benjamin are writing down can have at most 200 elements.
3		You must process 3 scenarios.
`alojzije 1`	`1 23`	Alojzije is writing down $a = (23)$ based on number 1.
`benjamin 2`	`1 42`	Benjamin is writing down $b = (42)$ based on number 2.
`alojzije 3`	`2 11 11`	Alojzije is writing down $a = (11, 11)$ based on number 3.

Second Run

Input	Output	Comment
200		Arrays that Alojzije and Benjamin are writing down can have at most 200 elements.
2		You must process 2 scenarios.
`cecilija 2 1 1`	`B`	Array $c = (1, 1)$ was generated based on $a = (23)$ and $b = (42)$, so $A < B$.
`cecilija 2 1 2`	`A`	Array $c = (1, 2)$ was generated based on $a = (11, 11)$ and $b = (42)$, so $A > B$.

Task Povjerenstvo

Do you know how hard it is to choose a set of people for the problem selection committee? No? Well do you know who does? Mr. Malnar, of course. By observing human interactions, the all-knowing Mr. Malnar has decided what the ideal choice should look like.

A total of N people are being considered for the committee and M relations between them have been recorded. A relation is described by an ordered pair (a, b) representing the fact that person a dislikes person b. Mr. Malnar defines a *dislike circle* to be a sequence of distinct people $x_1, x_2, \ldots, x_k$ such that person x_i dislikes person x_{i+1}, for each $1 \leq i \leq k$ (it is assumed that $x_{k+1} = x_1$). Mr. Malnar noticed a peculiar property regarding the set of people in question: **there is no dislike circle consisting of an odd number of people**.

To minimize dissatisfaction with the choice of committee, Mr. Malnar is looking for a committee such that everyone within the committee agrees with each other and everyone outside of the committee is glad not to be in it. More precisely:

- There must not be two people within the committee such that one person dislikes the other.
- For each person outside the committee there should be someone in the committee who they dislike.

Can you find such a set of people?

Input

The first line contains positive integers N and M, the number of people and number of relations between them, respectively.

The i-th of the following M lines contains an ordered pair of positive integers a_i and b_i $(1 \leq a_i, b_i \leq N)$, representing the fact that person a_i dislikes the person b_i. It holds that $a_i \neq b_i$ for all $i = 1, 2, \ldots M$ and no ordered pair is listed twice.

The given relations will be such that there is no dislike circle consisting of an odd number of people.

Output

If it is not possible to choose a set of people satisfying the given conditions, in the only line print `-1`.

Otherwise, in the first line print a positive integer K $(1 \leq K \leq N)$, the number of people in the committee. In the next line print K distinct positive integers $p_1, p_2, \ldots p_K$ $(1 \leq p_i \leq N)$, the indices of the people which make up the committee.

If there is more than one solution, output any one of them.

Scoring

In all subtasks, it holds that $1 \leq N \leq 500\,000$ and $0 \leq M \leq 500\,000$.

Subtask	Score	Constraints
1	13	There is no dislike circle.
2	21	There is no sequence of people of odd length $x_1, x_2, \ldots x_k$ such that one of x_i or x_{i+1} dislikes the other, for all $1 \leq i \leq k$.
3	33	$N, M \leq 5000$
4	33	No additional constraints.

Examples

input

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

output

```
2
4 1
```

input

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

output

```
2
1 3
```

input

```
8 11
1 2
2 1
3 4
4 5
5 6
6 3
7 8
8 7
3 2
7 3
8 1
```

output

```
3
1 3 5
```

Explanation of the examples:

The set of chosen people is shown in the output of each test case.

The first example is a valid test case for the first subtask and for the second subtask.

The second example is not a valid test case for the first subtask, but it is valid for the second subtask.

The third example is not a valid test case for the first subtask nor for the second subtask.

Task Vinjete

After two years of being online, the International Olympiad in Informatics (IOI) is going to be held live. The ISC and ITC are stressed as usual, the competitors are excited, parents proud and nervous, but the person most excited about the event being held on-site is Mr. Malnar. Once more he will taste the early-morning grape juice at the Zagreb airport, once more he will taste the finest Asian recipes, once more he will enjoy the daily excursions.

More experienced among you will ask themselves: "What excursions?! Mr. Malnar almost never takes the organized excursions with the rest of the delegations.". You're right, he doesn't, he plans his own excursions months ahead of the event.

First he solves all the car rental logistics, then he makes a short list of N cities he'd like to visit. He circles these cities on the map and connects each pair of cities that are directly connected via a highway. Interestingly, this year he drew exactly $(N-1)$ connections, and realized there exists a path between each pair of cities using these highways.

That's not all, it looks like there are M different types of vignettes that you're able to purchase in Asia. For each highway, it is known what subset of vignette types you need to have. Mr. Malnar immediately indexed all the different vignette types using integers from 1 to M. Interestingly, he managed to index them in such a way that in order to travel via i-th highway, you need buy all vignettes with indices greater or equal l_i and smaller or equal r_i.

Similarly, he indexed all the cities with integers from 1 to N such that Yogyakarta, a city in Indonesia hosting the olympiad, is denoted with 1.

To better plan his routes, he decided to ask you to write a program that will compute, for each city, what is the smallest number of vignettes he has to purchase in order to travel from Yogykarta to that city.

Input

The first line contains integers N and M from the task description.

The i-th of the next $N-1$ lines contains a_i, b_i, l_i and r_i meaning that the i-th highway connects cities with indices a_i and b_i ($1 \leq a_i, b_i \leq N, a_i \neq b_i$), and that travelling via that highway entails buying vignettes with indices from an interval $[l_i, r_i]$ ($1 \leq l_i \leq r_i \leq M$).

The highways are such that they connect each pair of the N cities.

Output

The i-th of the $N-1$ lines should contain the smallest number of vignettes Mr. Malnar has to buy in order to travel from Yogykarta (city with index 1) to city with index $(i+1)$.

Scoring

Subtask	Score	Constraints
1	11	$1 \leq N \leq 1\,000$, $1 \leq M \leq 1\,000$
2	13	$1 \leq N \leq 1\,000$, $1 \leq M \leq 10^9$
3	16	$1 \leq N \leq 50\,000$, $1 \leq M \leq 50\,000$
4	29	$1 \leq N \leq 100\,000$, $1 \leq M \leq 100\,000$
5	31	$1 \leq N \leq 100\,000$, $1 \leq M \leq 10^9$

Examples

input

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

output

```
3
4
4
5
4
```

input

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

output

```
1
2
3
5
```

Explanation of the first example:

In order to travel to city with index 2 you can buy vignettes with indices $(2, 3, 4)$.
In order to travel to city with index 3 you can buy vignettes with indices $(1, 2, 3, 4)$.
In order to travel to city with index 4 you can buy vignettes with indices $(2, 3, 4, 5)$.
In order to travel to city with index 5 you can buy vignettes with indices $(2, 3, 4, 5, 6)$.
In order to travel to city with index 6 you can buy vignettes with indices $(1, 2, 3, 4)$.

Explanation of the second example:

In order to travel to city with index 2 you can buy a vignette with index 2.
In order to travel to city with index 3 you can buy vignettes with indices $(2, 3)$.
In order to travel to city with index 4 you can buy vignettes with indices $(1, 2, 3)$.
In order to travel to city with index 5 you can buy vignettes with indices $(1, 2, 3, 4, 5)$.