Colors

0.3 s/256 MiB

Linda likes to change her hair color from time to time, and would be pleased if her boyfriend Archie would notice the difference between the previous and the new color. Archie always comments on Linda's hair color if and only if he notices a difference—so Linda always knows whether Archie has spotted the difference or not.

There is a new hair dye series in the market where all available colors are numbered by integers from 1 to N such that a smaller difference of the numerical values also means less visual difference.

Linda assumes that for these series there should be some *critical color difference* C ($1 \le C \le N$) for which Archie will notice color difference between the current color $color_{\text{new}}$ and the previous color $color_{\text{prev}}$ if $|color_{\text{new}} - color_{\text{prev}}| \ge C$ and will not if $|color_{\text{new}} - color_{\text{prev}}| < C$.

Now she has bought N sets of hair dye from the new series—one for each of the colors from 1 to N, and is ready to set up an experiment. Linda will change her hair color on a regular basis and will observe Archie's reaction—whether he will notice the color change or not. Since for the proper dye each set should be used completely, each hair color can be obtained no more than once.

Before the experiment, Linda was using a dye from a different series which is not compatible with the new one, so for the clearness of the experiment Archie's reaction to the first used color is meaningless.

Her aim is to find the precise value of C in a limited number of dyes. Write a program which finds the value of C by experimenting with the given N colors and observing Archie's reactions to color changes.

Communication

This is an interactive task. At the beginning the input contains a single integer—the value of N ($1 < N \le 10^{18}$). The value of C is kept secret by the grading system.

Then your program should make queries writing output in the following format: "`?` P", where P is an integer ($1 \le P \le N$) denoting the next color used. For each query the grading system gives an answer in the next line of the input. The answer is 1 if Archie notices the color difference between the last two colors and 0 otherwise. No two queries should have the same P value.

When your program determines C, it should output its value in the following format: "`=` C" and stop the execution. The grading system will not respond to this output and will not accept further queries.

Notes

To establish proper communication between your program and the grading system, you should flush the output stream after each query (Table 1).

Language	Command
C++	`std::cout << std::endl;`[1]
Java	`System.out.flush();`
Python	`sys.stdout.flush()`

Table 1: Flush commands

[1] `std::endl` writes a newline and flushes the stream.

It is possible to receive an "Output isn't correct" outcome even after printing a correct answer, if task constraints were violated during the communication. Violating the communication protocol itself may result in an "Execution killed" outcome.

Submitting user tests requires specifying an input file with the testcase parameters. The format of the input file is "N C" on a single line.

Example

Input	Output	Comments
`7`		$N = 7$
	`? 2`	
`1`		Answer to the first query is meaningless (can also be `0`)
	`? 7`	
`1`		$C \leq 5$
	`? 4`	
`0`		$3 < C \leq 5^{\dagger}$
	`? 1`	
`0`		$3 < C \leq 5$
	`? 5`	
`1`		$3 < C \leq 4$. Therefore, $C = 4$.
	`= 4`	

†It would be wise to check difference 4. However, this can not be done in the next query since $4 + 4 = 8$ and $4 - 4 = 0$ both are outside the allowed interval $1 \leq P \leq 7$.

Grading

Your program may use **at most 64** queries "**?**" to find the correct value of C.
Subtasks:

1. (9 points) $N \leq 64$
2. (13 points) $N \leq 125$
3. (21 points) $N \leq 1000$
4. (24 points) $N \leq 10^9$
5. (33 points) No further constraints

Mixture

2.0 s/256 MiB

Serge, the chef of the famous restaurant "Salt, Pepper & Garlic" is trying to obtain his first Michelin star. He has been informed that a secret expert plans to visit his restaurant this evening.

Even though the expert's name hasn't been disclosed, Serge is certain he knows which dish from the menu will be ordered as well as what the taste preferences of the expert are. Namely, the expert requires an extremely precise proportion of salt, pepper and garlic powder in his dish.

Serge keeps a set of bottles with mixtures of salt, pepper and garlic powder on a special shelf in the kitchen. For each bottle, he knows the exact amount of each of the ingredients in kilograms. Serge can combine any number of bottled mixtures (or just use one of them directly) to get a mixture of particular proportions needed for a certain dish.

Luckily, the absolute amount of a mixture that needs to be added to a dish is so small that you can assume that the amounts in the bottles will always be sufficient. However, the numeric values describing the proportions may be quite large.

Serge would like to know whether it is possible to obtain the expert's favourite mixture from the available bottles, and if so—what is the smallest possible number of bottles needed to achieve that.

Furthermore, the set of bottles on the shelf may change over time as Serge receives new ones or lends his to other chefs. So he would like to answer this question after each such change.

For example, assume that expert's favorite mixture is 1 : 1 : 1 and there are three bottles of mixtures on the shelf (Table 1):

Mixture	The amount of an ingredient in a bottle, kg		
	Salt	Pepper	Garlic powder
1	10	20	30
2	300	200	100
3	12	15	27

Table 1: Bottles on the shelf

To obtain the desired mixture it is enough to use an equivalent amount of mixtures from bottles 1 and 2. If bottle 2 is removed, then it is no longer possible to obtain it.

Write a program that helps Serge to solve this task!

Input

The first row contains three non-negative integers S_f, P_f and G_f ($0 \leq S_f, P_f, G_f$; $0 < S_f + P_f + G_f \leq 10^6$) describing the amount of salt, pepper and garlic powder in the expert's favourite mixture. For any real $\alpha > 0$, $(\alpha S_f, \alpha P_f, \alpha G_f)$ also is an expert's favourite mixture.

In the second row, there is a positive integer N (number of changes on the shelf, $N \leq 100\,000$). You should assume that initially the shelf is empty.

Each of the next N rows describes a single change on the shelf:

- If a new bottle is added, the row contains capital letter A followed by three non-negative integers S_i, P_i and G_i ($0 \leq S_i, P_i, G_i$; $0 < S_i + P_i + G_i \leq 10^6$) describing the amount of salt, pepper and garlic powder in the added bottle. Added bottles are numbered consecutively by unique integers starting from 1, that is, the i-th bottle corresponds to the i-th added bottle in the input data.

- If a particular bottle is removed from the shelf, the row contains capital letter R followed by the integer—the bottle number r_i. All values r_i in the removals are unique, r_i never exceeds total number of bottles added thus far.

Output

Output N rows. The j-th row ($1 \leq j \leq N$) should contain the number x_j, the smallest number of bottles needed to prepare a mixture with the expert's favourite proportions of salt, pepper and garlic powder using the bottles available after the first j changes on the shelf, or 0 if it is not possible.

Example

Input	Output
`1 2 3`	`0`
`6`	`2`
`A 5 6 7`	`0`
`A 3 10 17`	`2`
`R 1`	`1`
`A 15 18 21`	`1`
`A 5 10 15`	
`R 3`	

Pay attention that bottles 1 and 3 contain the same proportions of salt, pepper and garlic powder.

Grading

Subtasks:

1. (13 points) $N \leq 50$, $0 < S_i + P_i + G_i \leq 10^2$
2. (17 points) $N \leq 500$, $0 < S_i + P_i + G_i \leq 10^3$
3. (30 points) $N \leq 5000$, $0 < S_i + P_i + G_i \leq 10^4$
4. (40 points) No further constraints

Joker

2.0 s/256 MiB

Joker returns to Gotham City to execute another evil plan. In Gotham City, there are N street junctions (numbered from 1 to N) and M streets (numbered from 1 to M). Each street connects two distinct junctions, and two junctions are connected by at most one street.

For his evil plan, Joker needs to use an odd number of streets that together form a cycle. That is, for a junction S and an **even** positive integer k, there is a sequence of junctions $S, s_1, \ldots, s_k, S$ such that there are streets connecting (a) S and s_1, (b) s_k and S, and (c) s_{i-1} and s_i for each $i = 2, \ldots, k$.

However, the police are controlling the streets of Gotham City. On each day i, they monitor a different subset of all streets with consecutive numbers j: $l_i \leq j \leq r_i$. These monitored streets cannot be a part of Joker's plan, of course. Unfortunately for the police, Joker has spies within the Gotham City Police Department; they tell him which streets are monitored on which day. Now Joker wants to find out, for some given number of days, whether he can execute his evil plan. On such a day there must be a cycle of streets, consisting of an odd number of streets which are not monitored on that day.

Input

The first line of the input contains three integers N, M, and Q ($1 \leq N, M, Q \leq 200\,000$): the number of junctions, the number of streets, and the number of days to be investigated. The following M lines describe the streets. The j-th of these lines ($1 \leq j \leq M$) contains two junction numbers u and v ($u \neq v$), saying that street j connects these two junctions. It is guaranteed that any two junctions are connected by at most one street. The following Q lines contain two integers l_i and r_i, saying that all streets j with $l_i \leq j \leq r_i$ are checked by the police on day i ($1 \leq i \leq Q$).

Output

Your output is to contain Q lines. Line i ($1 \leq i \leq Q$) contains "`YES`" if Joker can execute his plan on day i, or "`NO`" otherwise.

Example

Input	Output
`6 8 2`	`NO`
`1 3`	`YES`
`1 5`	
`1 6`	
`2 5`	
`2 6`	
`3 4`	
`3 5`	
`5 6`	
`4 8`	
`4 7`	

See Figure 1.

Grading

Subtasks:

1. (6 points) $1 \leq N, M, Q \leq 200$

2. (8 points) $1 \le N, M, Q \le 2\,000$
3. (25 points) $l_i = 1$ for $i = 1, \ldots, Q$
4. (10 points) $l_i \le 200$ for $i = 1, \ldots, Q$
5. (22 points) $Q \le 2\,000$
6. (29 points) No further constraints

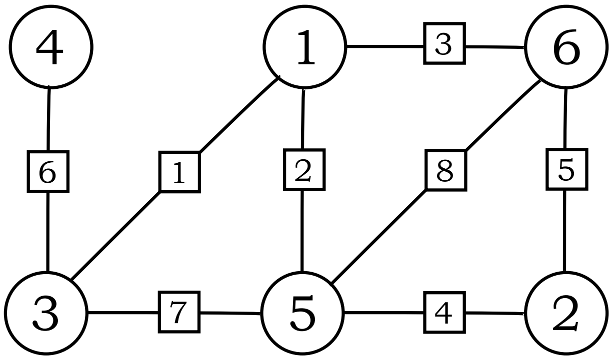

Figure 1: Example