Waterflow

The weather is getting better and Mouse Binna plans to finally fill her swimming pool. Her swimming pool is special, as it does not only have a single tap and a single drain but a total of n taps and drains that Binna can open and close as she likes. For each tap and each drain, Binna knows the speed with which water flows through (this is given in liters per minute). Note that a positive number indicates water flowing in from a tap while a negative number indicates water flowing out through a drain.

Mouse Binna opens one of the taps and goes inside. Mouse Stofl comes along and decides to prank Binna by messing with her pool. He starts to open and close the taps and drains arbitrarily. When he senses Binna coming back, he quickly turns off all the taps, closes all the drains and acts like he just came by. Binna then wonders why there are so many splashes of water all around and why her tap is turned off.

Stofl wants to confess to Binna what he did. He also wants to know the maximum amount of water that was in the pool at any point in time while Binna was in the house. Can you help him find this out?

For this problem, we are not concerned about the maximum amount of water fitting into Binna's pool, and you can assume its capacity to be infinite. Of course there cannot be a negative amount of water in the pool i.e. if the pool is empty and a drain is open, there is simply no water flowing out.

Input

The input starts with a line that contains a single integer m denoting the number of times Stofl tampers with the taps and drains.

This is followed by m lines, each line containing three integers s_i, e_i and c_i for $0 \leq i \leq m-1$. They indicated a tap or drain with water flowing at c_i liters per minute was turned on at minute s_i and off before minute e_i (water is flowing in minutes s_i to $e_i - 1$ but not in minute e_i). It is guaranteed that the first line starts at time $s_0 = 0$ and is a tap i.e. $c_0 > 0$.

Output

Print a line with a single integer representing the maximum amount of water in the pool at any time.

Limits

There are 4 subtasks. For each subtask, we have $1 \leq m \leq 10^5$, $0 \leq s_i < e_i \leq 10^8$ and $-10^4 \leq c_i \leq 10^4$.

- In subtask 1, worth 20 points, we have $c_i \geq 0$ for i.e. there are no drains, $1 \leq m \leq 1000$ and $0 \leq s_i < e_i \leq 10^4$.
- In subtask 2, worth 20 points, we have $1 \leq m \leq 1000$ and $0 \leq s_i < e_i \leq 10^4$.
- In subtask 3, worth 20 points, we have $0 \leq s_i < e_i \leq 10^6$.
- In subtask 4, worth 40 points, there are no further restrictions.

Examples

Input	Output
4 0 3 2 1 2 1 3 4 4 1 4 1	14

In minute 0, we have 2 liters flowing into the pool so we have 2 liters in the pool afterwards. Then in minute 1 we have two additional taps opening, leading to 4 liters flowing into the pool giving a total of 6 liters in the pool. In minute 2 one of the taps is closed again, so 3 liters flow in in this minute giving 9 liters in the pool. Then the first tap is closed an a new one opens, leagin to an inflow of 5 liters in minute 3 so the pool is filled with 14 liters after this. In minute 4 all taps are closed again. The maximum is obtained in minute 3 with 14 liters.

Input	Output
6 0 2 1 1 3 -3 2 5 3 2 3 -4 0 4 1 3 5 2	11

In minute 0 we have two taps opening, with a total throughput of 2 liters/min. Thus, after minute 0 we have 2 liters in the pool. In minute 1 a drain is opened, as its throughput of -3 is larger than the incoming waterflow, we have 1 liter per minute leaving the pool at this point. So after minute 1 we only have 1 liter left. In minute 2 Stofl opens an additional tap and an additional drain and one tap closes, resulting in a total throughput of -3 liters per minute. After minute 2 the pool is empty. In minute 3 the drains have beeen closed and a new tap is opened. At this point we have 6 liters per minute flowing into the pool, so after minute 3 there are 6 liters in the pool. In minute 4 one of the taps has been closed, so we have 5 liters/min flowing in giving a total of 11 liters. Afterwards all taps are closed. The maximum is obtained at minute 4 with 11 liters.

Embroideries

When Mouse Stofl and Mouse Binna found out that they qualified for IOI in Hungary, they decided that they would not only prepare incredibly well for the contest but also learn something about Hungarian culture in advance. During their research, they came across pictures of Hungarian traditional costumes. Both the women's and men's costumes are richly embroidered with colorful flowers and other motifs.

After hours of looking at pictures of these embroideries, Mouse Stofl and Mouse Binna agree that they want to have some on their T-shirts for the IOI. On one traditional costume, they find several patterns on the collar that they particularly like. In fact, each of these patterns somewhat resembles a sequence of opening and closing brackets. Good programmers as they are, they of course immediately realize that these bracket patterns are not all regular. They agree that they can't adopt a pattern like this on their T-shirts (it would be a bit embarrassing to have non-regular parentheses on the T-shirt at IOI).

Because of this, they now want to know how different subpatterns they can find in a pattern so that the bracket sequence shown is regular. (A bracket sequence is regular if it can be obtained by writing down a mathematical expression with only the letters '`(`', '`)`', '+' and '1', and then subsequently deleting all characters '+' and '1'.)

Take the pattern "`((())())(`" as an example. This pattern has 8 subpatterns:

"`((())())`", "`(())()`", "`((()))`", "`(()())`", "`(())`", "`()()`", "`()`", and "".

Note that each subpattern is counted only once, even if it occurs in multiple places, and that the empty substring is counted as regular.

Since the number of such subpatterns can be very large, mouse Stofl and mouse Binna want the result to be modulo $10^9 + 7$.

Input

The first line of the input contains n, the length of the pattern. The second line contains the pattern.

Output

Output the number of its different subpatterns that are regular bracket sequences modulo $10^9 + 7$.

Limits

There are four subtasks, each worth 25 points.

- In subtask 1, we have $1 \le n \le 16$.
- In subtask 2, we have $1 \le n \le 36$.
- In subtask 3, we have $1 \le n \le 200$.
- In subtask 4, we have $1 \le n \le 3\,000$.

Example

Input	Output
9 `(()())()(`	8

Mega jump

Mouse Stofl stands at the origin of a rectangular coordinate system. Stofl can make some number of moves: He can walk by one unit in horizontal or vertical direction, and he can also walk diagonally, i.e. by one unit in both horizontal and vertical direction. In other words, he can reach all 8 positions surrounding his current position in one move. Additionally, he can make a *mega jump*, where he jumps by k units horizontally and l units vertically.

Can you find the minimum number of moves needed for Stofl to reach a particular position?

Input

On the first line, there are three integers k, l and q. q lines follow, the i-th lines contains integers x_i and y_i.

Output

Output q lines. The i-th line should contain an integer d_i.

d_i should be the minimal number of moves to get from $(0, 0)$ to (x_i, y_i). In each move, Stofl can move from (x, y) to $(x \pm 1, y)$ or $(x, y \pm 1)$ or $(x \pm 1, y \pm 1)$ or $(x \pm k, y \pm l)$.

Limits

There are 4 subtasks, in all of them $q = 100$, $0 \leq k \leq 1000$, $0 \leq l \leq 1000$, $0 \leq x \leq 10^9$, $0 \leq y \leq 10^9$.

- In subtask 1, worth 10 points, we have $k = 1$, $l = 1$.
- In subtask 2, worth 20 points, we have $1 \leq k \leq 1000$, $l = 0$.
- In subtask 3, worth 20 points, we have $0 \leq k \leq 100$, $0 \leq l \leq 100$, $0 \leq x \leq 100$, $0 \leq y \leq 100$.
- In subtask 4, worth 50 points, there are no further constraints.

Examples

Input	Output

`4 3 2`
`3 3`
`10 1`
	`2`
	`4`

Here, we have $k = 4$, $l = 3$. In the first query, Stofl can make a mega jump from $(0, 0)$ to $(4, 3)$, and then walk a step to $(3, 3)$, for a total of 2 moves. In the second query, Stofl can make two mega jumps $(0, 0)$ – $(4, 3)$ – $(8, 0)$, and then walk two steps $(8, 0)$ – $(9, 1)$ – $(10, 1)$.

Input	Output

`5 0 2`
`6 0`
`4 4`
	`2`
	`4`

Here, we have $k = 5$, $l = 0$. In the first query, Stofl can walk to $(1, 0)$ and make a mega jump to $(6, 0)$. In the second query, Stofl can walk 4 steps $(0, 0) - (1, 1) - \ldots - (4, 4)$. He can't get there faster with mega jumps.

Frenemies

Mouse Binna and Mouse Stofl just arrived at their new informatics school. Their class consists of N other students and as they don't yet know all their names, they decide to number the students from 0 to $N-1$. On the one hand, Mouse Stofl is very shy and does not want to risk talking to two mice that dislike each other. On the other hand, Mouse Binna wants to socialise as much as possible with her new classmates.

Nonetheless, one thing they both want above all is to stay together as they have been Miraculous Mouse Mates for quite some time. Therefore, they need to decide who they will be friends with for the years to come. Based on their previous criteria the friend group needs the following proprieties: No two mice dislike each other within the friend group and the group must be as big as possible.

Unfortunately, they don't yet know who likes and who dislikes whom. To change this, they start listening to the different gossips that goes around in the school. Since they are smart mice they know that the enemy of their enemy is their friend and the friend of their enemy is also their enemy. Furthermore, the friend of their friend is their friend (and also their friends are their friends. . .). For instance, if Mouse 0 is friend with Mouse 1 and Mouse 1 dislikes Mouse 2 then Mouse 0 will also dislike Mouse 2. Note that all relations are symmetrical, this means that if Mouse 0 likes Mouse 1 then Mouse 1 likes Mouse 0.

Finally, Mouse Binna and Mouse Stofl only consider coherent gossips. They listen to the gossips one by one and only believe gossips that are consistent with their current knowledge. (e.g. if they know Mouse 1 likes Mouse 2 but they then hear that Mouse 2 dislikes Mouse 1 they will disregard the latter statement).

Input

On the first line there are two numbers N and Q — the number of mice in the class and the number of gossips they hear.

On each of the following Q lines, there are three numbers s, a, b $(0 \leq s \leq 1, 0 \leq a \neq b \leq N-1)$ representing a gossip. The first one being a 0 means the two mice are friends and being a 1 means they are enemies. The next two numbers represents the two mice which are either friends or enemies.

Output

Firstly, output Q lines. In the i-th line, output "`YES`" if the i-th gossip from the input is coherent with the previous gossips and "`NO`" otherwise.

Then, output a single positive integer, the biggest group possible such that no two mice dislike each other. Note that in this group it is not necessary that everyone knows everyone else.

Limits

There are 3 subtasks. In all subtasks we have $2 \leq N \leq 100\,000$ and $1 \leq Q \leq 350\,000$.

- In subtask 1, worth 30 points, it is guaranteed that $2 \leq N \leq 1000$, $1 \leq Q \leq 2000$ and that, after processing all consistent gosspis, every mouse dislike at most one other mouse.
- In subtask 2, worth 30 points, it is guaranteed that $2 \leq N \leq 1000$ and $1 \leq Q \leq 2000$
- In subtask 3, worth 40 points, there is no additional constraint.

For each subtask you can get **50%** of the points allocated if all the Q first lines are correct but the size of the biggest group is incorrect.

Examples

Input	Output
4 4 0 0 1 1 2 3 1 0 1 1 2 3	YES YES NO YES 3

Input	Output
5 6 0 0 1 1 0 2 0 2 3 0 1 2 1 4 3 1 4 1	YES YES YES NO YES NO 3