Task 4: RMQ

Kraw the Krow has N cows. Each cow wears a tag containing a **unique** number from 0 to $N-1$. The cows are lined up in a row in some unknown order. The positions of the cows are labelled from 0 to $N-1$ in order.

For reasons that we will probably never know, Kraw found the need to answer an embarrassingly large number of Range Minimum Queries (RMQs). RMQs are questions of the form, "What is the smallest tag number of the cows standing at or between positions L and R?"

Kraw is very lazy, so he paid Coco the Code Monkey less than minimum wage to solve the RMQs for him. Coco completed all of them four minutes before the deadline, and Kraw was very pleased.

That was in 2013. Now, Kraw is making a big loss in his Cow Tagging conglomerate and is starting to doubt the authenticity of Coco's work. For all he knew, Coco could have just randomly generated answers to all the RMQs.

Unfortunately, after so many years, all the cows have dispersed and Coco is suspisciously uncontactable. Help Kraw check if there exists any ordering of cows such that all of Coco's answers are valid.

Input format

Your program should read the input from standard input. The input consists of:

- one line with two integers N ($1 \leq N \leq 100\,000$), the number of cows, and Q ($1 \leq Q \leq 100\,000$), the number of RMQs;
- Q lines, with the i-th line containing three integers: L_i and R_i ($0 \leq L_i \leq R_i < N$), the left and right bounds of the i-th RMQ, and A_i ($0 \leq A_i < N$), Coco's answer to the RMQ.

Output format

Output one line containing N space-separated integers: any possible ordering of cows such that A_i is the correct answer to the RMQ $[L_i, R_i]$ for all i, and no two cows share the same tag number.

If there does not exist any such ordering of cows, fill all N integers with -1.

Subtasks

The maximum execution time on each instance is 1.0s. Your program will be tested on sets of input instances that satisfy the following restrictions:

Subtask	Marks	N	Q
1	23	$N \leq 10$	$Q \leq 10$
2	44	$N \leq 1\,000$	$Q \leq 1\,000$
3	33	No additional constraints.	

For each test case, if your program's output is not a permutation (that is, it contains repeated tag numbers) but satisfies all the RMQs, you will still be eligible for **30%** of that subtask's score.

Sample Testcase 1

Input	Output
5 3 0 2 1 1 3 0 1 4 0	1 4 3 0 2

Note that this is not the only output that would be accepted.

Sample Testcase 1 Explanation

First, we note that our proposed ordering $[1, 4, 3, 0, 2]$ is indeed a permutation (that is, every integer from 0 to 4 appears exactly once). Then, we check the RMQs:

- The first RMQ ($L = 0$, $R = 2$) refers to the subarray $[1, 4, 3]$. The smallest tag number within that range is 1.
- The second RMQ refers to the subarray $[4, 3, 0]$. The smallest tag number within that range is 0.
- The third RMQ refers to the subarray $[4, 3, 0, 2]$. The smallest tag number within that range is 0.

Since all answers coincide with Coco's answers, $[1, 4, 3, 0, 2]$ is indeed a valid solution.

Sample Testcase 2

Input	Output
3 2 0 1 1 1 2 1	-1 -1 -1

Sample Testcase 2 Explanation

If the 0-th cow were in position 0 or 1, the answer to the first RMQ would be 0 instead of 1. If the 0-th cow were in position 2, the answer to the second RMQ would be 0 instead of 1. Thus, it is impossible to order the cows such that all RMQs are satisfied.

Note that the following output would still be valid for 30% of the subtask's score:

```
1 1 1
```