

Motivation:

This is a small algorithmic test. Beyond giving a correct solution, please also highlight possible algorithm design tradeoffs and elaborate, why you choose your approach and maybe also what other solutions are possible.

We prefer solutions in Java, since we run a Java backend. You don't need to use any build tools - just a small executable main class would be fine as well.

Task:

You get a list (initialized to 0) of size X . You need to perform Y operations on it and print the maximum value of all final values of all the Y elements in the list. For each operation, 3 integers are given - i, j and k and you have to add value k to all the elements in the range of index i to j (both inclusive).

Input Format:

First line: two integers X and Y separated by one whitespace.

Next Y lines contain three numbers i, j and k separated by one whitespace.

Numbers in the list are numbered from 1 to X .

Constraints:

$$3 \leq X \leq 10^7$$

$$1 \leq Y \leq 2 \cdot 10^5$$

$$1 \leq i \leq j \leq X$$

$$0 \leq k \leq 10^9$$

Output Format:

A single line containing the maximum value in the list, after all operations are applied.

Sample Input:

```
5 3
1 2 100
2 5 100
3 4 100
```

Sample Output:

```
200
```

Explanation

After first update list will be 100 100 0 0 0. After second update list will be 100 200 100 100 100. After the third update list will be 100 200 200 200 100.

So the required answer will be 200.