Array Manipulation

Starting with a 1-indexed array of zeros and a list of operations, for each operation add a value to each the array element between two given indices, inclusive. Once all operations have been performed, return the maximum value in the array.

Example
$n = 10$
$queries = [[1, 5, 3], [4, 8, 7], [6, 9, 1]]$

Queries are interpreted as follows:

```
    a b k
    1 5 3
    4 8 7
    6 9 1
```

Add the values of k between the indices a and b inclusive:

```
index->  1 2 3  4  5 6 7 8 9 10
        [0,0,0, 0, 0,0,0,0,0, 0]
        [3,3,3, 3, 3,0,0,0,0, 0]
        [3,3,3,10,10,7,7,7,0, 0]
        [3,3,3,10,10,8,8,8,1, 0]
```

The largest value is 10 after all operations are performed.

Function Description

Complete the function *arrayManipulation* in the editor below.

arrayManipulation has the following parameters:

- *int n* - the number of elements in the array
- *int queries[q][3]* - a two dimensional array of queries where each *queries[i]* contains three integers, *a*, *b*, and *k*.

Returns

- *int* - the maximum value in the resultant array

Input Format

The first line contains two space-separated integers n and m, the size of the array and the number of operations.
Each of the next m lines contains three space-separated integers a, b and k, the left index, right index and summand.

Constraints

- $3 \leq n \leq 10^7$
- $1 \leq m \leq 2 * 10^5$
- $1 \leq a \leq b \leq n$
- $0 \leq k \leq 10^9$

Sample Input

```
5 3
1 2 100
```

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

Sample Output

```
200
```

Explanation

After the first update the list is `100 100 0 0 0`.
After the second update list is `100 200 100 100 100`.
After the third update list is `100 200 200 200 100`.

The maximum value is 200.