E. Wall

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

This is a problem from the IOI (International Informatics Olympiad) 2014. Try to solve it using the segment tree!

Jian-Jia is building a wall by stacking bricks of the same size together. This wall consists of n columns of bricks, which are numbered from 0 to $(n-1)$ from left to right. The columns may have different heights. The height of a column is the number of bricks in it.

Jian-Jia builds the wall as follows. Initially there are no bricks in any column. Then, Jian-Jia goes through k phases of adding or removing bricks. The building process completes when all k phases are finished. In each phase Jian-Jia is given a range of consecutive brick columns and a height h, and he does the following procedure:

- In an adding phase, Jian-Jia adds bricks to those columns in the given range that have less than h bricks, so that they have exactly h bricks. He does nothing on the columns having h or more bricks.
- In a removing phase, Jian-Jia removes bricks from those columns in the given range that have more than h bricks, so that they have exactly h bricks. He does nothing on the columns having h bricks or less.

Your task is to determine the final shape of the wall.

Input

The first line contains two integers n and k $(1 \le n \le 2\,000\,000,\ 1 \le k \le 500\,000)$, the number of columns and number of phases.

The next k lines contain descriptions of phases. Each line contains four numbers. First, the number t denotes the type of phase: 1 if it is an adding phase, and 2 if it is a removing phase. The next two numbers l and r $(0 \le l \le r \le n-1)$ define the range of the phase: the the range of columns starts with column l and ends with column r. The fourth number h $(0 \le h \le 100\,000)$ is the height of the action.

Output

Print n lines. In the i-th line print the number of bricks in the $(i-1)$-th column after all phases are finished.

Example

input Copy

```
10 6
1 1 8 4
2 4 9 1
2 3 6 5
1 0 5 3
1 2 2 5
2 6 7 0
```

output Copy

```
3
4
5
4
3
3
0
0
1
0
```

Note

Picture for the example: