Task Name

Interval Power Updates 1

Difficulty

Medium

Problem Description

Start with an array of zeros. Apply `q` inclusive range updates, then print the final values of the array.

Problem Statement

At the start, you have an array `a` of length `n`, and every value is `0`.

You are given `q` operations. For each operation `l r x`, add `x` to all elements:

- `a[l]`
- `a[l + 1]`
- `...`
- `a[r]`

After applying all operations, print the final array.

Input Format

- The first line contains two integers `n` and `q`.
- Each of the next `q` lines contains three integers `l`, `r`, and `x`.

Constraints

- `1 <= n <= 10^5`
- `0 <= q <= 10^4`
- `0 <= l <= r < n`
- `-1000 <= x <= 1000`

Output Format

Print `n` space-separated integers: the final values of the array after all updates.

Example 1

Input

6 3 0 2 4 1 4 -1 3 5 2

Output

4 3 3 1 1 2

Explanation

- Add `4` to positions `0..2` -> `[4, 4, 4, 0, 0, 0]`
- Add `-1` to positions `1..4` -> `[4, 3, 3, -1, -1, 0]`
- Add `2` to positions `3..5` -> `[4, 3, 3, 1, 1, 2]`

Example 2

Input

7 3 2 6 3 0 3 1 4 4 -2

Output

1 1 4 4 1 3 3

Explanation

- Add `3` to positions `2..6`
- Add `1` to positions `0..3`
- Add `-2` to position `4`

The final array is `[1, 1, 4, 4, 1, 3, 3]`.

Tags

Arrays, Prefix Sums, Difference Array

Reference Solution Idea

Use a difference array: - For an update `l r x`, do `diff[l] += x` - If `r + 1 < n`, do `diff[r + 1] -= x` - Take prefix sums over `diff` to rebuild the final array

Complexity

- Time: `O(n + q)`
- Space: `O(n)`

Suggested testcase roles

- `input00/output00` -> Sample, strength `0`
- `input01/output01` -> Sample, strength `0`
- `input02/output02` -> Hidden, strength `50`
- `input03/output03` -> Hidden, strength `50`