Problem Weirdtree

C++ header `weirdtree.h`

Azusa, the witch of the highlands, has discovered a garden full of weird trees! Therefore, together with her friend, Laika, she decided to spend some time there taking care of the garden.

The garden can be viewed as a sequence of N trees, where the trees are indexed from 1 to N. Each tree has a certain non-negative integer height. Azusa will then spend her time according to a schedule containing Q entries, which can be of several types:

1. A tree cutting phase, characterised by three integers l, r, and k. In this phase, Azusa will spend the next k days cutting trees. Each day she finds the tallest tree whose index is between l and r and decreases its height by 1. In case there are several trees of this maximal height, she chooses the leftmost one. If the tallest tree has height 0, then nothing happens on that day.
2. A magic phase, characterised by two integers i and x. In this phase, Azusa changes the tree with index i so that it has height x.
3. A tree inspection phase, characterised by two integers l and r. In this phase, Azusa will find the sum of the heights of the trees with indices between l and r.

(Note that "between" is meant inclusively; e.g. $1, 2, 3, 4, 5$ are "between" 1 and 5.)

Azusa is curious what the results of the tree inspection phases will be, and wants to know them without having to go through the entire schedule. Can you help her?

Interaction Protocol

The contestant must implement the following four functions:

```
void initialise(int N, int Q, int h[]);
void cut(int l, int r, int k);
void magic(int i, int x);
long long int inspect(int l, int r);
```

The function `initialise` is given N (the number of trees), Q (the number of entries in the schedule), and an array h, where $h[i]$ is the height of tree i, for $1 \leq i \leq N$. This function is called by the committee's code exactly once, before any of the other three functions are called. The `cut`, `magic` and `inspect` functions represent tree cutting, magic and tree inspecting phases respectively, and are characterised by their respective parameters. The contestant's implementation of the `inspect` function must return the sum of the heights of the trees with indices between l and r.

The contestant should not implement the `main` function. This will be implemented in the committee's `grader.cpp` file; you will be given a sample `grader.cpp` in the attachments. Our `main` function will read N, Q, the sequence of N initial heights, and the Q schedule entries. The three types of schedule entries (`cut(l, r, k)`, `magic(i, x)` and `inspect(l, r)`) are encoded as `1 l r k`, `2 i x` and `3 l r` respectively. This is the input format that will be used in the examples below.

Note that the contestant is allowed to use global variables, additional functions, methods and classes.

Restrictions

- $1 \leq N, Q \leq 300\,000$
- It is guaranteed that the `cut`, `magic` and `inspect` functions will be called exactly Q times in total.
- $1 \leq i \leq N$
- $0 \leq x, k, h[i] \leq 1\,000\,000\,000$
- $1 \leq l \leq r \leq N$

#	Points	Restrictions
1	5	$N \leq 1\,000, Q \leq 1\,000, k = 1$
2	8	$N \leq 80\,000, Q \leq 80\,000, k = 1$
3	8	$N \leq 1\,000, Q \leq 1\,000$, there are no `magic` operations.
4	19	There are no `magic` operations.
5	10	$l = 1, r = N$
6	21	$N \leq 80\,000, Q \leq 80\,000$
7	29	No further restrictions

Examples

Input file	Output file
6 10 1 2 3 1 2 3 1 1 6 3 3 1 6 1 1 3 3 3 1 6 1 1 3 1000 3 1 6 2 1 1000 3 1 6 1 1 3 999 3 1 5	9 6 5 1005 4

Explanation

In the first phase, after each of the 3 days of tree cutting, the heights of the trees are $1, 2, 2, 1, 2, 3$; $1, 2, 2, 1, 2, 2$; and $1, 1, 2, 1, 2, 2$. The sum of these values is 9, which is the answer to the inspection in the second phase.

In the third phase, after each of the 3 days of tree cutting, the heights of the trees are $1, 1, 1, 1, 2, 2$; $0, 1, 1, 1, 2, 2$; and $0, 0, 1, 1, 2, 2$. The sum of these values is 6, which is the answer to the inspection in the fourth phase.

In the fifth phase, after each of the 1000 days of tree cutting, the heights of the trees are $0, 0, 0, 1, 2, 2$. This is because a tree with height 0 cannot be cut. The sum of these values is 5, which is the answer to the inspection in the sixth phase.

In the seventh phase, the first tree is grown to height 1000, giving us tree heights $1000, 0, 0, 1, 2, 2$. The sum of these values is 1005, which is the answer to the inspection in the eighth phase.

In the ninth phase, each of the 999 days of tree cutting reduces the height of the first tree by 1. This gives us tree heights $1, 0, 0, 1, 2, 2$ at the end of the phase. The sum of the first five of these values is 4, which is the answer to the inspection in the tenth and final phase.