

# I Love Kd-Trees II

Every time someone is preparing to be interviewed, there is someone preparing the interview. While a good guy is preparing for the interview described in <http://www.spoj.com/problems/ILKQUERY/> your boss gives you the job of preparing the input/output for a new task.

But as always, your boss changes his mind so quickly... He will give you an array of integers, but sometime, he will say you that the element at index  $r$  of the original array must toggle his state between active (1) and inactive (0).

Initially all elements start as active elements.

Toggle the state of an element means to change it to active (1) if it was inactive, and change it to inactive (0) if it was active.

Mixed with the toggle operations, he also will give you several queries, represented with three integers,  $i \ l \ k$  meaning he wants to know how many active elements of value  $k$  exist between indexes  $i$  and  $l$  (both inclusive)

## Input

Input consists of one test case.

The first line contains two integers,  $N$  ( $1 \leq N \leq 10^5$ ) and  $Q$  ( $1 \leq Q \leq 10^5$ ).

The next line contains  $N$  integers  $a_i$  ( $-10^9 \leq a_i \leq 10^9$ ).

Then  $Q$  lines follow. Each of them starts with an integer  $q$  which can be 0 or 1. If it's 0, then three integers  $i \ l \ k$  follow ( $0 \leq i < N$  ;  $i \leq l < N$  ;  $-10^9 \leq k \leq 10^9$ ). If it's 1, then an integer  $r$  follows, meaning you have to toggle the state of the element at index  $r$ . ( $0 \leq r < N$ ).

## Output

For each query starting with 0 (in the same order as the input) output a single line with the answer to that query. That is, output the amount of active elements of value  $k$  between the indexes  $i$  and  $l$  (both inclusive).

## Example

**Input:**

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

**Output:**

4

3

0