Array Sorting

Sumit specialises in sorting algorithms, and Abhishek decides to test Sumit's coding skills. An array of 'n' numbers a[0], a[1], a[2], ..., a[n-1] is given. Abhishek gives a sequence of inputs of the form "v i j". Each input is either a query or an update (query if 'v' is 0, update otherwise).

For any input of the form "0 i j" ,Sumit's output should be as follows:
If the subarray a[i], a[i+1], ... a[j] is unsorted, output 0.
If the subarray a[i], a[i+1], ... a[j] is sorted in non-descending order, output 1.
If the subarray a[i], a[i+1], ... a[j] is sorted in non-ascending order, output 2.
If the subarray a[i], a[i+1], ... a[j] is sorted in both non-ascending and non-descending order (i.e, if all the elements in the range are equal), output 3.

Any other input "v i j" (v!=0) should be treated as an update, as follows:
For each element in the subarray a[i], a[i+1], ... a[j], Sumit has to replace the element a[k] with v-a[k].

Sumit is really tired and does not want to write a program. Can you write a program for Sumit, which responds to Abhishek's instructions?

Input

The first line of input contains 2 space separated integers 'n' and 'q'. The second line contains 'n' integers a[0], a[1], ..., a[n-1]. Each of the next 'q' lines contain 3 integers 'v', 'i', 'j'.

Output

For each query, output a single integer 0, 1, 2 or 3, denoting the answer.

Example

Input:

```
4 5
3 -2 -5 1
1 1 3
0 0 3
0 0 2
0 2 3
0 0 1
```

Output:

```
0
1
2
3
```

Constraints:

1 <= n <= 100000
1 <= q <= 100000
-5000 <= a[i] <= 5000
-5000 <= v <= 5000

Explanation

Initial array is {3, -2, -5, 1}. After first update, the array will be {3, 3, 6, 0}. Now, from indices '0' to '3', it is unsorted. From indices '0' to '2', it is sorted in non-descending order. From indices '2' to '3', it is sorted in non-ascending order. Between indices '0' and '1', the values are equal.