Task 2. Majorant

Majorant of a multiset is an element which occurs more frequently than all other elements combined. Some multisets do not have *majorant*.

Given is an array containing n positive integers a[1], a[2], ..., a[n]. A subarray of array a is the sequence a[l], a[l+1] … a[r], where 1 ≤ l ≤ r ≤ n.

We consider m queries of two types:

1) Update p q. At this query a[p] becomes equal to q

2) Query p q, where p ≤ q. Let b[1]=a[p], b[2]=a[p+1] … b[q-p+1]=a[q]. At this query, you have to calculate by modulo 998244353 the sum, composed of summands of the form: i*(number of subarrays of b for which i is the *majorant*) for all different values of i, each i is a majorant for a subarray of b.

Input

First line of the standard input contains the number n. The second line of the standard input contains n integers – the element of the given array. The third line of the standard input contains the number m. From each of the next m lines read 3 numbers: l, r and t - the query in an *encrypted* form.

To decrypt the query: Let last_output be the last number on the standard output produced by your program (or 0, if there are no such)

Compute type=((t+last_output) mod 2) +1

If type=1 the query is "Update" with p=((l+last_output) mod n)+1, q=((r+last_output) mod 100 000 000)+1

If type=2, the query is "Query" with p=((l+last_output) mod n)+1, q=((r+last_output) mod n)+1

Output

For every query of type 2, output on a separate line the answer to the query.

Constraints

1 ≤ n ≤ 200 000

For every number x of the array a, 1 ≤ x ≤ 100 000 000

1 ≤ m ≤ 100

1 ≤ l, r, t ≤ 100 000 000

Subtasks

Subtask	Points	Additional constraints
1	10	n ≤ 100, m ≤ 50
2	15	n ≤ 1 000, m ≤ 50
3	10	n ≤ 10 000, m ≤ 50
4	10	n ≤ 50 000, m ≤ 5
5	20	n ≤ 65 000, m ≤ 50
6	35	There are no additional constraints.

Points for each subtask will be received only if the program solves correctly all the test cases given for that particular subtask.

Sample input

4

1 2 2 1

3

4 3 1

2 99999990 2

4 2 1

Sample output

12

6

Explanation of the example

After decrypting, the first query becomes “query, p=1, q=4”. There are 2 subarrays with *majorant* 1 and 5 with *majorant* 2 so the answer is 2*1+5*2=12.

The second query becomes “update, p=3, q=3”. After that, the array becomes: 1, 2, 3, 1.

The third query is: “query, p=1, q=3”. There is 1 subarray with *majorant* 1, 1 with *majorant* 2 and 1 with *majorant* 3.