Problem F
Flip

Time limit: 3 seconds
Memory limit: 2048 megabytes

Problem Description

Suppose you are given an array of n entries where each array entry is either 0 or 1. For any pair (ℓ, r) such that $1 \leq \ell \leq r \leq n$, $[a[\ell], a[\ell+1], \ldots, a[r]]$ is a subarray of the array $[a[1], a[2], \ldots, a[n]]$. An alternating subarray $[a[\ell], a[\ell+1], \ldots, a[r]]$ of $[a[1], a[2], \ldots, a[n]]$ if $a[\ell] \neq a[\ell+1] \neq \cdots \neq a[r]$. I.e., every entry in the subarray is different from its neighbors in the subarray. Since the definition of alternating subarrays only considers the entries in the subarrays, $[1, 0, 1]$ is still an alterating subarray of $[1, 1, 0, 1, 1]$.

In this problem, two types of operations will be applied to the given array.

- 1 ℓ r: for every $i \in [\ell, r]$, change $a[i]$ into $1 - a[i]$.
- 2 ℓ r: report the total number of pairs (x, y) such that $\ell \leq x \leq y \leq r$ and subarray $[a[x], a[x+1], \ldots, a[y]]$ is an alternating subarray.

Please write a program to maintain the given array. Your program must report the numbers efficiently.

Input Format

The first line contains two integers n and q, indicating the length of the given array and the number of operations. The second line contain n space separated numbers $a[1], a[2], \ldots, a[n]$ representing the given array $[a[1], a[2], \ldots, a[n]]$. Then q lines follow, and the i-th of them contains 3 integers t_i, ℓ_i, r_i where the i-th operation is t_i ℓ_i r_i.

Output Format

For each operation of the second type, output the reported number on one line.

Technical Specification

- $1 \leq n \leq 200000$
- $1 \leq q \leq 200000$
- $a[i] \in \{0, 1\}$ for all $i \in \{1, 2, \ldots, n\}$.
- $t_j \in \{1, 2\}$ for all $j \in \{1, 2, \ldots, q\}$.
- $1 \leq \ell_j \leq r_j \leq q$ for all $j \in \{1, 2, \ldots, q\}$.

Sample Input 1

```
3 1
1 1 0
2 1 3
```

Sample Output 1

```
4
```

Sample Input 2

```
20 20
0 0 1 0 1 0 0 1 1 1 0 1 0 0 0 1 1 1 0 0
1 1 10
2 2 7
1 3 15
2 1 9
1 4 9
2 1 13
1 13 15
2 10 20
1 1 5
2 2 10
1 15 17
2 15 18
1 1 3
2 4 6
1 15 19
2 1 6
1 15 15
2 10 17
1 1 8
2 15 19
```

Sample Output 2

```
16
16
21
14
12
6
4
9
10
8
```