UTS Open '24 P5 - Parity Challenge

Time limit: 1.0s **Memory limit:** 512M

Steven is learning about even and odd numbers in math class! His teacher, Todd, has written a long expression with N numbers, $a_1, a_2, \ldots, a_N$, on the chalkboard. Between each number is either an addition `+` or multiplication `x` sign.

As part of a learning exercise, Todd has challenged the students to quickly determine whether variations of the expression are equal to an even or odd number. Specifically, Todd has given the students Q events of three types:

1. `V i` - update the value of a_i to $a_i + 1$.
2. `O j` - update the j^{th} operation to `x` if it is currently a `+`, and vice versa.
3. `Q l r` - Todd asks you what the parity of the expression becomes if you place an opening bracket to the left of a_l and a closing bracket to the right of a_r. Recall that calculations inside of brackets are done first, and that multiplication is done before addition. Note that the brackets are not actually added and are not present in future events.

Can you write a program to help Steven conquer the Parity Challenge?

Constraints

$2 \le N, Q \le 2 \times 10^5$
$1 \le a_i \le 10^9$

$1 \le i \le N$
$1 \le j \le N - 1$
$1 \le l \le r \le N$

Subtask 1 [30%]

There are only events of type `Q`.

Subtask 2 [70%]

No additional constraints.

Input Specification

The first line of input contains two integers, N and Q, the number of numbers on the chalkboard and the number of events.

The next line of input contains N space-separated integers, $a_1, a_2, \ldots, a_N$, the numbers initially on the chalkboard.

The third line of input contains $N - 1$ characters, with the i^{th} character being either `+` for addition or `x` for multiplication, indicating the operation initially between a_i and a_{i+1}.

The final Q lines of input contain events in one of the following formats:

1. `V i` - update the value of a_i to $a_i + 1$.
2. `O j` - update the jth operation to `x` if it is currently a `+`, and vice versa.
3. `Q l r` - Determine the parity of the expression if you place an opening bracket to the left of a_l and a closing bracket to the right of a_r.

Output Specification

For each event of type `Q`, output one line containing `even` if the parity of the expression is even after adding the brackets, and `odd` otherwise.

Sample Input

```
5 6
3 1 4 1 5
+++x
Q 1 2
O 2
Q 3 4
Q 5 5
V 5
Q 3 4
```

Sample Output

```
odd
even
even
odd
```

Explanation for Sample

This is the initial expression:
$3 + 1 + 4 + 1 \times 5$

The first event is a query, with brackets around $[1, 2]$:
$(3 + 1) + 4 + 1 \times 5 = 13 \rightarrow$ `odd`

The second event updates operation 2 from `+` to `x`:
$3 + 1 \boxed{\times} 4 + 1 \times 5$

The third event places brackets around $[3, 4]$:
$3 + 1 \times (4 + 1) \times 5 = 28 \rightarrow$ `even`

The fourth event places brackets around $[5, 5]$:
$3 + 1 \times 4 + 1 \times (5) = 12 \rightarrow$ `even`

The fifth event increments a_5 from 5 to $5 + 1 = 6$:
$3 + 1 \times 4 + 1 \times \boxed{6}$

The final event places brackets around $[3, 4]$:
$3 + 1 \times (4 + 1) \times 6 = 33 \rightarrow$ `odd`