Problem B
Balanced Seesaw Array

Time limit: 3 seconds
Memory limit: 1024 megabytes

Problem Description

Bob likes to play seesaw. He thinks that it would be really funny if the seesaw is in a balanced state. It means that the seesaw is not tilted to the left and right. After playing the seesaw, Bob thinks about a problem related to the balanced seesaw.

Let $A = [a_1, a_2, \ldots, a_m]$ denote an array of length m. Bob thinks that $[a_1, a_2, \ldots, a_m]$ is a *balanced seesaw array* if there exists an integer k between 1 to m such that $\sum_{i=1}^{m}(i-k)a_i = 0$.

Bob gets an array $A = [a_1, a_2, \ldots, a_n]$ as his birthday gift, and he is curious about whether some non-empty subarray is a *balanced seesaw array*. More formally, he is interested in whether $[a_\ell, a_{\ell+1}, \ldots, a_r]$ is a *balanced seesaw array* for some specified pair (ℓ, r) where $1 \le \ell \le r \le n$. Bob also finds that the elements in its array will change by time, it will have the following two types of changes.

1. $a_\ell, a_{\ell+1}, \ldots, a_r$ are increased by x.
2. $a_\ell, a_{\ell+1}, \ldots, a_r$ are changed to x.

For convenience, Bob will give you the array $A = [a_1, a_2, \ldots, a_n]$ first. Then, there are q operations. Each operation will be one of the following three types.

- 1 ℓ r x: means that $a_\ell, a_{\ell+1}, \ldots, a_r$ are increased by x.
- 2 ℓ r x: means that $a_\ell, a_{\ell+1}, \ldots, a_r$ are changed to x.
- 3 ℓ r: means that Bob is curious about whether the subarray $[a_\ell, a_{\ell+1}, \ldots, a_r]$ is a *balanced seesaw array*. You should output "`Yes`" or "`No`" for each operation type 3.

Input Format

The first line of input contains two integers n and q. n is the length of the array, and q is the number of operations. The second line contains n integers a_i to define the array. Each of the following q lines is an operation described in the problem statement.

Output Format

Please output "`Yes`" or "`No`" to indicate whether $[a_\ell, a_{\ell+1}, \ldots, a_r]$ is a *balanced seesaw array* for each type 3 operation.

Technical Specification

- $1 \le n \le 100000$
- $1 \le q \le 1200000$
- $-1000 \le a_i \le 1000$
- $-10000 \le x \le 10000$

- For $1 \leq i \leq n$, you may assume that $|a_i| \leq 1.5 \times 10^9$ after any operation.
- $1 \leq \ell \leq r \leq n$

Sample Input 1

```
3 6
1 2 3
3 1 1
3 1 3
1 1 1 2
3 1 3
2 2 2 0
3 2 3
```

Sample Output 1

```
Yes
No
Yes
Yes
```