

Back To School '18: Function Maximization

Time limit: 1.4s **Memory limit:** 1G

Java: 2.75s

Turing: 11.0s

Evan somehow found an array, a , consisting of N integers in the range $[K, K + 5\,000]$ on the ground. Looking at it, he came up with a problem. However, he has no clue how to solve it. After asking all his friends, no one knew how to solve it, so he comes to you, asking for help.

Evan gives you the following function:

$$f(x) = \begin{cases} x & \text{if } (x-1)! + 1 \equiv 0 \pmod{x} \\ -1 & \text{if } (x-1)! + 1 \not\equiv 0 \pmod{x} \end{cases}$$

where $!$ denotes the factorial operation. In other words, the function returns x if $(x-1)! + 1$ is divisible by x , and -1 otherwise.

Evan will give you Q queries:

- `1 l r` Print the maximum value of $f(k)$ for all elements k in the subarray $[l, r]$ ($1 \leq l \leq r \leq N$).
- `2 l r v` Add v ($1 \leq |v| \leq 5\,000$) to all elements in the subarray $[l, r]$ ($1 \leq l \leq r \leq N$).

Input Specification

The first line will contain 3 integers, N, Q, K ($1 \leq N \leq 10^5, 1 \leq Q \leq 5 \times 10^4, 2 \leq K \leq 10^{14}$).

The second line will contain N integers, a_1, a_2, \dots, a_N ($K \leq a_i \leq K + 5\,000$), the initial array.

The next Q lines will each contain a valid query as defined above.

It is guaranteed that at any point in time, all elements of the array will be in the range $[K, K + 5\,000]$.

Output Specification

For every type 1 query, output the maximum value of $f(k)$ for all elements k in the subarray $[l, r]$ on its own line.

Constraints

Subtask 1 [10%]

$N, Q \leq 100$

$K = 2$

Subtask 2 [30%]

$K \leq 10^5$

Subtask 3 [60%]

No additional constraints.

Sample Input

```
9 9 2
2 3 5 6 8 10 7 11 16
1 1 5
2 1 5 1
1 1 5
1 4 5
1 6 8
1 9 9
2 6 9 -3
1 6 8
1 9 9
```

Sample Output

```
5
7
7
11
-1
7
13
```