MacroGCD Refinement

It's your first day at your new job working in the MacroGCD Refinement office! You were excited at first, but it turns out the work is not at all what you were expecting. You've been given numerous files, each filled with an absurdly large list of numbers that you're required to sift through. The department head, Mark F., says that you are to pick out subsets of the numbers based on a very particular defining feature: their *greatest common divisors*. You have no idea why you're supposed to do this, and when asked, Mark just rattles off something about the work being "mysterious and important..." After a few hours of computing GCD's by hand, you've just about dropped dead from boredom. To keep what's left of your sanity, you decide to use some of your Advanced Algo knowledge to speed up your work.

Write a program that, given a MacroGCD Refinement file, represented by a list of integers A, will answer queries of the form $GCD(\ell, r)$ by calculating the greatest common divisor of the *range* of numbers included in $A[\ell \dots r]$.

In addition, the files in MacroGCD Refinement are constantly getting updated over seemingly random ranges of numbers. Your program must also support queries of the form $ADD(\ell, r, \delta)$. This query should update $A[\ell \dots r]$ by adding δ to each value in the range.

Input

The first line will contain the size of the file N and the number of queries Q. The next line will contain N space-separate integers—the numbers contained in the file. The following Q lines will each contain a query in one of the two following formats:

- "GCD" followed by integers ℓ and r indicating what range to calculate the greatest common divisor over.
- "ADD" followed by integers ℓ, r, and δ, where δ is the amount to add to each number in the specified range.

Output

For each GCD query, output the greatest common divisor of the numbers included in the range. That is, a $GCD(\ell, r)$ query should output

$$\gcd(A[\ell], A[\ell+1], \dots, A[r]).$$

Sample Input

```
5 6
9 45 15 63 14
GCD 0 2
ADD 1 3 1
GCD 1 3
GCD 0 1
ADD 0 0 1
GCD 0 1
```

Sample Output

```
3
2
1
2
```

Hint: Implementing a *single* data structure that supports both queries in $O(\log N)$ is probably not feasible. You may find the identity

$$\gcd(a_\ell, a_{\ell+1}, \ldots, a_r) = \gcd(a_\ell, a_{\ell+1} - a_\ell, a_{\ell+2} - a_{\ell+1}, \ldots, a_r - a_{r-1})$$

useful.