Problem 10: Big Boris the CALICOsmonaut
13 Points

Problem ID: `vector`
Rank: `4`

Introduction

Blyat! It's 1991 and the CCCP (CALICO Coding Competition Polity) has collapsed! The imperialist forces of the **USA**CO empire have taken over the entire world, sending it into turmoil and destroying all open-source software. While this happens, Big Boris the CALICOsmonaut is stranded on the *Bir*, the world's first interstellar space station. Determined to create a safe haven for coders across the universe, he primes his engines and sets out for open space. Boris however is not a skilled navigator, due to the lack of dexterity of his clubby paws he can only adjust a continuous range of engine throttles, and having eaten all but two of his boost modulators, he needs your help to manually navigate his ship to new horizons.

Problem Statement

You're given an array of $\mathbf{N}$ integers a_1, a_2, ... $a_{\mathbf{N}}$. Initially, these are assigned values $\mathbf{S_1}$, $\mathbf{S_2}$, ... $\mathbf{S_N}$.

You're then given a sequence of $\mathbf{Q}$ queries that your program must respond to. There are two types of queries:

- *Update*
  - Given $\mathbf{L_i}$, $\mathbf{R_i}$, and $\mathbf{V_i}$, update a_1, a_2, ... $a_{\mathbf{N}}$ by adding $\mathbf{V_i}$ to each value in the range of the indices [$\mathbf{L_i}$, $\mathbf{R_i}$], inclusive.
- *Find*
  - Find the minimum possible sum of some integer array b_1, b_2, ... $b_{\mathbf{N}}$ such that there exists positive integers k, x_1, x_2 satisfying:
    $b_i x_1 = a_i$ for all $1 \le i \le k$
    $b_j x_2 = a_j$ for all $k + 1 \le j \le \mathbf{N}$

Input Format

The first line of input contains a single integer $\mathbf{N}$ denoting the length of the array. The second line of input contains $\mathbf{N}$ space-separated integers $\mathbf{S_1}$ $\mathbf{S_2}$ ... $\mathbf{S_N}$ denoting the initial values of the array.

The third line of input contains a single integer $\mathbf{Q}$ denoting the number of queries that follow. Each query is described in a single line:

- If the i^{th} query is an *Update* query, the line contains `UPDATE` $\mathbf{L_i}$ $\mathbf{R_i}$ $\mathbf{V_i}$ separated by spaces where:
  - `UPDATE` is a string literal.
  - $\mathbf{L_i}$ and $\mathbf{R_i}$ denote the range of indices to perform the update.
  - $\mathbf{V_i}$ denotes the value to add.
- If the i^{th} query is a *Find* query, the line contains only the string `FIND`

Output Format

For each *Find* query, output a single line containing an integer denoting the sum of b

Constraints

Time limit: **1 second**
Memory limit: **256 MB**

$2 \le \mathbf{N} \le 10^5$
$0 \le \mathbf{S_i} \le 10^9$
$1 \le \mathbf{Q} \le 10^4$
$0 \le \mathbf{V_i} \le 10^9$
$1 \le \mathbf{L} \le \mathbf{R} \le \mathbf{N}$
It is guaranteed that the number of *Find* queries in a test file will not exceed 2500.
Note: There is a lot of input/output for this problem, so **fast I/O may be required**.

Sample Test Cases

Sample Input Download

```
5
10 5 15 0 3
6
FIND
UPDATE 3 4 3
FIND
UPDATE 4 5 15
UPDATE 2 2 5
FIND
```

Sample Output Download

```
7
11
5
```

Sample Explanation

The array has a length of $\mathbf{N}$ = 5. Initially, the values given by $\mathbf{S_1}$ $\mathbf{S_2}$ ... $\mathbf{S_N}$ are shown below:

i	1	2	3	4	5
a_i	10	5	15	0	3

There are $\mathbf{Q}$ = 6 queries in total.

The first query is a *Find* query. Let's illustrate the solution using this table:

i	1	2	3	4	5
b_i	2	1	3	0	1
x_1	5	5	5	5	-
x_2	-	-	-	-	3
a_i	10	5	15	0	3

The total sum of the b_i in this case is 7. It can be proved that this is the minimum solution.

The second query is an *Update* query that updates a by adding $\mathbf{V_i}$ = 3 to all values with indices [$\mathbf{L_i}$ = 3, $\mathbf{R_i}$ = 4] which includes a_3 and a_4.

i	1	2	3	4	5
a_i (old)	10	5	15	0	3
$\mathbf{V_i}$	-	-	3	3	-
a_i (new)	10	5	18	3	3

The third query is a *Find* query. Once again, let's illustrate the solution using another table:

i	1	2	3	4	5
b_i	2	1	6	1	1
x_1	5	5	-	-	-
x_2	-	-	3	3	3
a_i	10	5	18	3	3

The total sum of the b_i in this case is 11. It can be proved that this is the minimum solution.

The fourth query is an *Update* query, which adds 15 to a_4 and a_5.

The fifth query is an *Update* query too, which adds 5 to a_2. After both operations, the resulting array a is [10, 10, 18, 18, 18].

The last query is another Find query:

i	1	2	3	4	5
b_i	1	1	1	1	1
x_1	10	10	-	-	-
x_2	-	-	18	18	18
a_i	10	10	18	18	18

The total sum of the b_i in this case is 5. It can be proved that this is the minimum solution.