



Candies (14pts, 24pts)

Practice Submissions

You have not attempted this problem.

Last updated: May 17 2020, 23:40

Problem

Carl has an array of N candies. The i -th element of the array (indexed starting from 1) is A_i representing *sweetness value* of the i -th candy. He would like to perform a series of Q operations. There are two types of operation:

- Update the sweetness value of a candy in the array.
- Query the *sweetness score* of a subarray.

The sweetness score of a subarray from index l to r is: $A_l \times 1 - A_{l+1} \times 2 + A_{l+2} \times 3 - A_{l+3} \times 4 + A_{l+4} \times 5 \dots$

More formally, the sweetness score is the sum of $(-1)^{i-l} A_i \times (i - l + 1)$, for all i from l to r inclusive.

For example, the sweetness score of:

- [3, 1, 6] is $3 \times 1 - 1 \times 2 + 6 \times 3 = 19$
- [40, 30, 20, 10] is $40 \times 1 - 30 \times 2 + 20 \times 3 - 10 \times 4 = 0$
- [2, 100] is $2 \times 1 - 100 \times 2 = -198$

Carl is interested in finding out the total sum of sweetness scores of all queries. If there is no query operation, the sum is considered to be 0. Can you help Carl find the sum?

Input

The first line of the input gives the number of test cases, T . T test cases follow. Each test case begins with a line containing N and Q . The second line contains N integers describing the array. The i -th integer is A_i . The j -th of the following Q lines describe the j -th operation. Each line begins with a single character describing the type of operation (U for update, Q for query).

- For an update operation, two integers X_j and V_j follow, indicating that the X_j -th element of the array is changed to V_j .
- For a query operation, two integers L_j and R_j follow, querying the sweetness score of the subarray from the L_j -th element to the R_j -th element (inclusive).

Output

For each test case, output one line containing **Case #x:** y , where x is the test case number (starting from 1) and y is the total sum of sweetness scores of all the queries.

Limits

Time limit: 20 seconds per test set.

Memory limit: 1GB.

$1 \leq T \leq 100$.

$1 \leq A_i \leq 100$, for all i .

$1 \leq N \leq 2 \times 10^5$ and $1 \leq Q \leq 10^5$ for at most 6 test cases.

For the remaining cases, $1 \leq N \leq 300$ and $1 \leq Q \leq 300$.

If the j -th operation is an update operation, $1 \leq X_j \leq N$ and $1 \leq V_j \leq 100$.

If the j -th operation is a query operation, $1 \leq L_j \leq R_j \leq N$.

Test set 1

There will be at most 5 update operations.

Test set 2

There are no special constraints.

Sample

Input	Output
2	
5 4	
1 3 9 8 2	
Q 2 4	
Q 5 5	
U 2 10	Case #1: -8
Q 1 2	Case #2: -3
3 3	
4 5 5	
U 1 2	
U 1 7	

In sample case #1:

- The first query asks for the sweetness score of [3, 9, 8] which is $3 \times 1 - 9 \times 2 + 8 \times 3 = 9$.
 - The second query asks for the sweetness score of [2] which is $2 \times 1 = 2$.
 - The third query asks for the sweetness score of [1, 10] which is $1 \times 1 - 10 \times 2 = -19$.
- Thus, the final output should be $9 + 2 - 19 = -8$.

In sample case #2:

- The first and only query asks for the sweetness score of [7, 5] which is $7 \times 1 - 5 \times 2 = -3$.
- Thus, the final output should be -3.