

To Poland

Little Russell and Mr. Fredricksen are embarking in a new journey with their balloon house. After visiting South America and helping Kevin, the Snipe, they decided to travel to Warsaw, Poland where new specimens of birds might be found.

Their trip is divided into several parts. Due to pressure levels and wind speed, each part of their journey requires that their house is equipped with a minimum number of balloons. As a good boy scout, Russell has already figured out how many balloons are necessary to go through each of the parts. The problem is that atmosphere conditions change and Russell is having trouble determining how many balloons the house should be equipped with to go through continuous parts of the journey, so they don't have to fill up more balloons than necessary.

To help out Russell, you are to develop a program that must deal with the following queries:

- Russell has added or removed balloons from the house
- The minimum number of balloons at certain part has changed
- How many balloons should be added or removed so that the house goes through continuous parts of the journey with the minimum necessary number of balloons?

Input

The first line contains a number **T** ($T \leq 100$), the number of testcases.

Each testcase starts with numbers **N** ($1 \leq N \leq 10^5$) and **M** ($0 \leq M \leq 10^3$) where **N** is the number of parts in the journey and **M** is the initial number of balloons in the house. Then **N** numbers follow, one per line, where each number **N_i** ($0 \leq N_i \leq 10^9$, $0 \leq i < N$) represents the initial number of balloons necessary to go through part **i**.

On the next line there is a number **Q** ($1 \leq Q \leq 10^5$), the number of queries that you should answer. Each of the next **Q** lines contains a query in one of the following formats:

- "**A K**" - where **A** is the actual character 'A' (quotes for clarity) and **K** ($-10^3 \leq K \leq 10^3$) is the number of balloons that have been added or removed from the house. It is assured that the total number of balloons in the house is never negative;
- "**B J K**" - where **B** is the actual character 'B' (quotes for clarity), **J** ($0 \leq J < N$) is the number of the part of the journey (0-indexed) and **K** ($0 \leq K \leq 10^9$) is the new minimum number of balloons required to go through part **J**;
- "**C I J**" - where **C** is the actual character 'C' (quotes for clarity), and **[I, J]**, $0 \leq I \leq J < N$ is the range that Russell would like to query. For each query of this type, print the difference between the number of balloons currently in the house and the **minimum** necessary to go through all the parts between **I** and **J** inclusive. Note that this query doesn't change the number of balloons in the house.

Output

For each testcase print one line with "*Testcase X:*" (quotes for clarity) where **X** is the number of

the testcase (0-indexed). For each query of the type "**C I J**" print one line with the difference between the number of balloons currently in the house and the minimum necessary to go through all the parts between **I** and **J** inclusive.

Print one blank line after each testcase.

Example

Input:

```
2
3 11
5 2 4
1
C 0 2
3 11
16 2 17
5
C 0 2
A -2
C 1 2
B 0 0
C 0 1
```

Output:

Testcase 0:

6

Testcase 1:

6

8

7