Absurdistans Teaparties

The people in Absurdistan like tea. In fact, it's their favourite drink. But normally, they don't drink their tea alone. They make big teaparties and I've heard the people there drink so much tea their drunk after the parties. The sad side of the story is, that the people there don't do anything else than go to teaparties. They sit home alone waiting until someone sends them an invitation for a teaparty on the national network Teanet(TM). This network has employed you as a developer.

Strangely, the n citizens of Absurdistan have their n houses built on a line with the same distance between each two adjacent houses. To go with the car from one house to the next, one needs exactly c minutes. And in each house, exactly one citizen lives there.

Teanet(TM) wants to create an app for the people to help them plan their teaparties. Suppose a citizen living in building x wants to make a teaparty in t seconds. For this teaparty, he wants to invite all the peaple living in houses between house l and r (l and r included). He wants to know how much time he has left until he has to send the invitiations. We know for every citizen the time he needs to make himself ready. After a citizen receives an invitation, he immediately makes himself ready and then takes the car and goes to the citizen living in building x. We want to know the latest point in time such that x still is able to send the invitations and everyone is here when the party starts. But sometimes, people move out of a building and new people move in, so the time for a citizen in a building to prepare can change.

And since the people return drunk from their parties, it may happen that there are accidents on the streets. If an accident happens, the street from building a up to building b is closed, but only in one direction. The people who don't live in a building between a and b don't care since they can use another street, but the people between a and b need some time more if they want to drive in the same direction as the accident. Because the people in Absurdistan don't really work, the effects of the accident are permanent...

So there are three types of queries:

"invite", followed by four numbers x l r t. x wants to invite all the people between l and r to his teaparty. If he wants to start his team party in t seconds, in how many seconds is the latest point in time when he has to send his invitations? All the citizens in Absurdistan always take the direct way (they don't change direction and go back etc.). Note that the result may be negative if the person who wants to send an invitation is already too late. Important: Since the person who invites already knows that he will be inviting, he doesn't need to prepare himself.

"change" followed by two numbers x v. The citizen in building x moved out and the new resident needs v seconds to prepare.

"accident" followed by 3 numbers and a char: l r v d. There was an accident between l and r. All citizens between l and r need v more time if they travel in direction d. Note that this doesn't affect the people from outside if they want to go to a teaparty at a location between l and r.

Also note that all indices are 0-based and inclusive.

Input

First line: Three integers n q c. n is is the number of citizens / buildings, q the number of queries and c the time you need between two houses.

On the next line, n values follow: the i-th value is the time the i-th citizen (citizen 0 lives in building 0 etc.) needs to prepare.

Then q lines follow, each with a query as in the description.

Output

For each query of type "invite", output the number of seconds citizen x has left to send the invitation.

Constraints

1 <= n, q <= 1e5

0 <= c <= 1e4

The preparation-time for any citizen will never exceed 1e4 and v in query "accident" is also not bigger than 1e4.

Whener we have a range l r, then 0 <= l <= r < n holds.

Example

Sample 0

Input:

```
10 11 2
0 0 0 0 0 0 0 0 0 0
invite 3 6 8 12
accident 6 8 2 l
invite 3 6 8 12
invite 9 6 8 12
change 5 4
invite 3 5 8 16
change 7 6
invite 0 5 8 32
accident 0 9 10 r
change 8 11
invite 5 0 9 50
```

Output:

```
2
0
6
4
10
30
```

Sample 1

Input:

```
10 10 34
10 23 1 43 12 22 84 17 41 24
invite 2 4 8 333
change 2 11
invite 8 4 9 542
accident 0 9 10 l
accident 0 5 11 r
accident 6 7 22 l
invite 4 2 6 233
invite 8 2 6 542
change 3 1
invite 4 3 3 47
```

Output:

```
88
390
49
316
1
```

Description of Sample 0 :

Citizen wants to invite all the people betweeen 6 and 8. Citizen 8 needs 10 seconds to get to citizen 3, so if citizen 3 sends the invitation in any later than 2 seconds, citizen 8 would be too late.
Before the second invitation, there was an accident and Citizen 8 now needs 2 seconds longer because of this.
In the 3rd invitation, the citizens travel to the right and the accident only affects people going to the left. Person 6 needs 6 seconds so if he gets the invitation any later than in 6 seconds, he's too late.
Then, citizen 5 moves ous and a new person moves in. This person needs a bit more time to get ready, and when person 3 makes anther party, he needs to send his invitation even earlier.
Then citizen 0 makes an invitation etc., but then another accident happens, now influencing everyone going to the right. A new person moves in building 8 and citizen 5 makes another party. Person 9 needs 8 seconds to get to citizen 5. Person 8 needs 11 seconds to get ready, 2 second to avoid the (first) accident and 6 seconds to get there giving a total of 19 seconds. But person 0 needs even longer: Person 0 doesn't need to prepare, but since he wants to go to the right, he needs to avoid the (second) accident, giving him 10 extra seconds. But he then needs to go to person 5 and it takes him 10 seconds to get there, giving a total of 20 seconds until he arrives. So

person 5 needs to send his invitation in 30 seconds.

Edit: Feb 12th, 2020: Updated the task description to clarify that the person who invites doesn't need to prepare

Edit: Mar 3th, 2020: Added Constraints