A. Watering Plants (wateringplants)

There is a tall building in Cesenatico with N floors and a single resident living on each floor. The floors are numbered from 0 to $N-1$ from bottom to top, and resident r lives on floor r.

Each floor has a balcony where residents enjoy the sun and grow their own plants. From there, they can also admire the plants on the balcony directly below. Since all the plants need to be watered once a day, the residents have decided to help each other with watering duties. Each resident can help water the plants on the balcony that is one floor below theirs.

Every morning, at time 0, all residents leave the building. Initially, resident r comes home at time t_r. If resident r comes home strictly before the resident one floor below them, that is $t_r < t_{r-1}$, then resident r waters the plants for resident $r-1$. (Otherwise, resident $r-1$ will water their own plants themselves.) At the end of each day, *exactly one* of the following event types happens:

Type `!` A resident r updates the time at which they will come home, starting from the next day.

Type `?` A resident r asks how many times they have already watered the plants for resident $r-1$.

Note that resident 0 does not water the plants for anyone else and that the plants of resident $N-1$ are never watered by anyone else.

Your task is to help the residents answer all events of type `?`.

Input

The first line contains two integers N and D, the number of residents and the number of days to track.

The next line contains N integers $t_0, t_1, ..., t_{N-1}$, the initial times at which each resident comes home.

Then D lines follow, where the ith of the D lines describes the event at the end of day i.

Each event is in one of the following two formats:

`! r x` Resident r ($0 \le r \le N-1$) comes home at time x, starting from the next day, that is, the value of t_r becomes x. Note that it is possible for x to be the same as the current t_r.

`? r` Ask how many times resident r ($1 \le r \le N-1$) has watered the plants for resident $r-1$ since the beginning of day 0.

It is guaranteed that there is at least one `?` event.

Output

For each `?` event, output one line with a single integer: the number of times resident r has watered the plants for resident $r-1$ since the beginning of day 0.

Note that in this problem, you should **not** consider the number of times a resident waters their own plants.

Constraints

- $2 \le N \le 200\,000$.
- $1 \le D \le 200\,000$.

- $1 \leq t_r \leq 10^9$ initially and after each change.

Scoring

Your program will be tested on several test cases grouped into subtasks. To obtain the score for a subtask, you must correctly solve all the tests it contains.

- **Subtask 0** [**0 points**]: Examples.
- **Subtask 1** [**9 points**]: $D = 1$, i.e. there is exactly one event, which is of type `?`.
- **Subtask 2** [**12 points**]: All events are of type `?`.
- **Subtask 3** [**13 points**]: $N = 2$.
- **Subtask 4** [**18 points**]: $N \leq 2000$ and $D \leq 2000$.
- **Subtask 5** [**21 points**]: Every resident changes their return time at most once.
- **Subtask 6** [**27 points**]: No additional constraints.

Examples

stdin	stdout
3 4 7 7 5 ? 2 ? 1 ? 2 ? 2	1 0 3 4
2 5 5 7 ! 1 4 ? 1 ! 0 4 ! 1 6 ? 1	1 2
4 6 13 9 15 2 ! 1 18 ? 3 ! 0 12 ! 2 1 ? 1 ? 2	2 1 5
3 6 5 2 4 ? 1 ! 1 8 ! 0 10 ! 1 3 ? 1 ? 2	1 4 2

Explanation

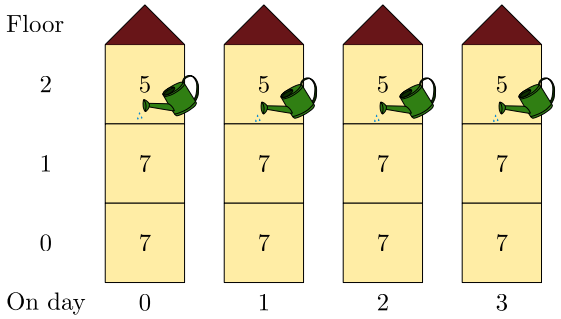

Figure 1: Example 1. The watering can indicates that the resident waters the plants for the resident below them.

The first example is valid for subtasks 2, 4, 5, and 6. Since schedules are never updated, resident 2 comes home before resident 1 and waters their plants every day. After day 0, resident 2 has watered the plants for their neighbour once. As residents 0 and 1 come home at the same time, resident 1 does not water the plants for resident 0. After day 1, resident 1 has not watered the plant for their neighbour. After day 2, resident 2 has watered the plants for their neighbour three times. After day 3, resident 2 has watered the plants for their neighbour four times.

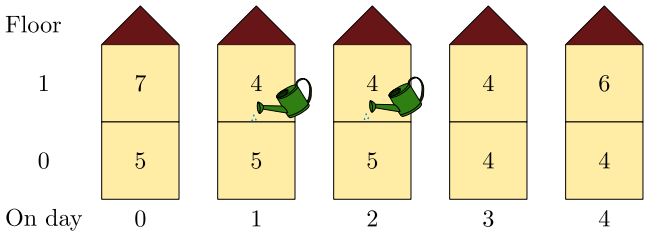

Figure 2: Example 2.

The second example is valid for subtasks 3, 4, and 6. On day 0, resident 1 does not water the plants for their neighbour. After day 0, the schedule of resident 1 is updated. Since they come home earlier than their neighbour on day 1, they water their neighbour's plants. After day 1, resident 1 has watered the plants for their neighbour once. On day 2, resident 1 waters their neighbour's plants again. After day 4, resident 1 has watered their neighbour's plants twice in total.

The third example is valid for subtasks 4, 5, and 6. Note that there is no figure for this example.

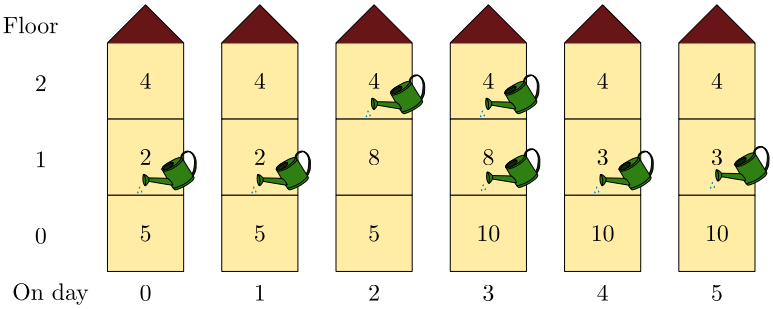

Figure 3: Example 4.

The fourth example is valid for subtask 4 and 6. After day 0, resident 1 has watered their neighbour's plants once. After day 4, resident 1 has watered their neighbour's plants four times (on days 0, 1, 3, and 4). Resident 2 has watered their neighbour's plants twice in total (on days 2 and 3).