

## A. Vicini gentili (wateringplants)

C'è un alto palazzo a Cesenatico con  $N$  piani e un singolo residente che vive su ogni piano. I piani sono numerati da 0 a  $N - 1$  dal basso verso l'alto, e il residente  $r$  vive al piano  $r$ .

Ogni piano ha un balcone dove i residenti si godono il sole e curano le proprie piante. Da lì, possono anche ammirare le piante sul balcone direttamente sottostante. Dato che tutte le piante devono essere innaffiate una volta al giorno, i residenti hanno deciso di aiutarsi a vicenda con questo compito. Ogni residente può aiutare ad innaffiare le piante sul balcone che si trova esattamente un piano sotto al proprio.

Ogni mattina, all'istante 0, tutti i residenti escono dal palazzo. Inizialmente, il residente  $r$  torna a casa all'istante  $t_r$ . Se il residente  $r$  torna a casa rigorosamente prima del residente un piano sotto di lui, cioè  $t_r < t_{r-1}$ , allora il residente  $r$  innaffia le piante del residente  $r - 1$ . (Altrimenti, il residente  $r - 1$  innaffierà le proprie piante da solo.) Alla fine di ogni giornata, accade *esattamente uno* dei seguenti tipi di evento:

**Tipo !** Un residente  $r$  aggiorna l'orario in cui tornerà a casa, a partire dal giorno successivo.

**Tipo ?** Un residente  $r$  chiede quante volte ha già innaffiato le piante per il residente  $r - 1$ .

Nota che il residente 0 non innaffia le piante per nessun altro e che le piante del residente  $N - 1$  non vengono mai innaffiate da nessun altro.

Il tuo compito è aiutare i residenti a rispondere a tutti gli eventi di tipo ?.

### Input

La prima riga contiene due interi  $N$  e  $D$ , il numero di residenti e il numero di giorni da monitorare.

La riga successiva contiene  $N$  interi  $t_0, t_1, \dots, t_{N-1}$ , gli orari iniziali in cui ogni residente torna a casa.

Seguono poi  $D$  righe, dove la  $i$ -esima riga descrive l'evento alla fine del giorno  $i$ .

Ogni evento è in uno dei seguenti due formati:

**! r x** Il residente  $r$  ( $0 \leq r \leq N - 1$ ) torna a casa all'istante  $x$ , a partire dal giorno successivo, cioè il valore di  $t_r$  diventa  $x$ . Nota che è possibile che  $x$  sia uguale al  $t_r$  corrente.

**? r** Chiedi quante volte il residente  $r$  ( $1 \leq r \leq N - 1$ ) ha innaffiato le piante per il residente  $r - 1$  dall'inizio del giorno 0.

È garantito che ci sia almeno un evento di tipo ?.

### Output

Per ogni evento di tipo ?, stampa una riga con un singolo intero: il numero di volte in cui il residente  $r$  ha innaffiato le piante per il residente  $r - 1$  dall'inizio del giorno 0.

Nota che in questo problema, **non** devi considerare il numero di volte in cui un residente innaffia le proprie piante.

## Assunzioni

- $2 \leq N \leq 200\,000$ .
- $1 \leq D \leq 200\,000$ .
- $1 \leq t_r \leq 10^9$  inizialmente e dopo ogni modifica.

## Assegnazione del punteggio

Il tuo programma verrà testato su diversi casi di test raggruppati in subtask. Per ottenere il punteggio di un subtask, devi risolvere correttamente tutti i test che contiene.

- **Subtask 0 [ 0 punti]**: Casi d'esempio.
- **Subtask 1 [ 9 punti]**:  $D = 1$ , ovvero c'è esattamente un evento, che è di tipo ?.
- **Subtask 2 [12 punti]**: Tutti gli eventi sono di tipo ?.
- **Subtask 3 [13 punti]**:  $N = 2$ .
- **Subtask 4 [18 punti]**:  $N \leq 2000$  e  $D \leq 2000$ .
- **Subtask 5 [21 punti]**: Ogni residente cambia il proprio orario di rientro al massimo una volta.
- **Subtask 6 [27 punti]**: Nessuna limitazione aggiuntiva.

## Esempi di input/output

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

## Spiegazione

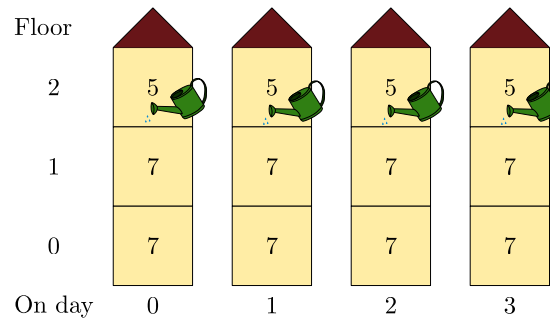


Figura 1: Esempio 1. L'annaffiatoio indica che il residente innaffia le piante per il residente sotto di lui.

Il primo esempio è valido per i subtask 2, 4, 5 e 6. Dato che gli orari non vengono mai aggiornati, il residente 2 torna a casa prima del residente 1 e innaffia le sue piante ogni giorno. Dopo il giorno 0, il residente 2 ha innaffiato le piante per il suo vicino una volta. Poiché i residenti 0 e 1 tornano a casa allo stesso orario, il residente 1 non innaffia le piante per il residente 0. Dopo il giorno 1, il residente 1 non ha innaffiato la pianta per il suo vicino. Dopo il giorno 2, il residente 2 ha innaffiato le piante per il suo vicino tre volte. Dopo il giorno 3, il residente 2 ha innaffiato le piante per il suo vicino quattro volte.

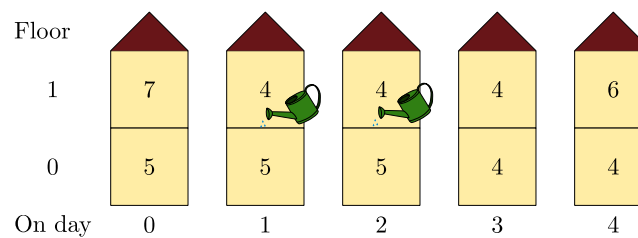


Figura 2: Esempio 2.

Il secondo esempio è valido per i subtask 3, 4 e 6. Il giorno 0, il residente 1 non innaffia le piante per il suo vicino. Dopo il giorno 0, l'orario del residente 1 viene aggiornato. Dato che torna a casa prima del suo vicino il giorno 1, innaffia le piante del suo vicino. Dopo il giorno 1, il residente 1 ha innaffiato le piante per il suo vicino una volta. Il giorno 2, il residente 1 innaffia di nuovo le piante del suo vicino. Dopo il giorno 4, il residente 1 ha innaffiato le piante del suo vicino due volte in totale.

Il terzo esempio è valido per i subtask 4, 5 e 6. Nota che non c'è una figura per questo esempio.

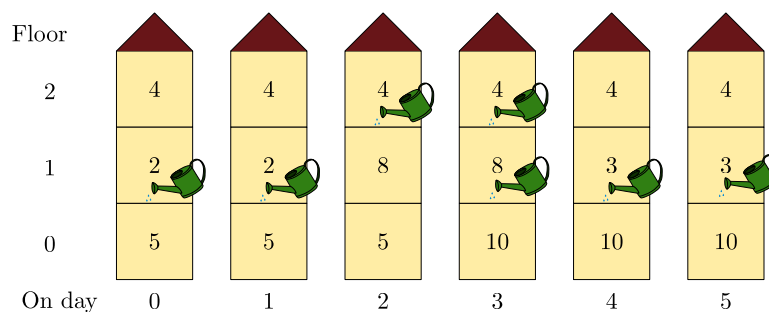


Figura 3: Esempio 4.

Il quarto esempio è valido per i subtask 4 e 6. Dopo il giorno 0, il residente 1 ha innaffiato le piante del suo vicino una volta. Dopo il giorno 4, il residente 1 ha innaffiato le piante del suo vicino quattro volte (nei giorni 0, 1, 3 e 4). Il residente 2 ha innaffiato le piante del suo vicino due volte in totale (nei giorni 2 e 3).