



## Librărie

Fișier intrare	librarie.in
Fișier ieșire	librarie.out
Clasele	VII-VIII

Andrei este managerul unei librării care dorește să mențină o evidență exactă a prețurilor cărților din inventar. Emanuel, administratorul librării, implementează planuri de marketing care implică modificări repetate ale prețurilor (operații de modificare de preț), iar deciziile sale se pot schimba ulterior. Pentru a maximiza profitul obținut în funcție de cerere, în fiecare zi, Emanuel efectuează o serie de operații (modificări, anulări și refaceri), iar la sfârșitul fiecărei zile, Andrei trebuie să vadă starea finală a prețurilor din librărie.

## Date de intrare

Fișierul `librarie.in` conține:

- **Linia 1:** Un număr natural  $n$ , numărul de cărți din librărie (numerotate de la 1 la  $n$ ).
- **Linia 2:**  $n$  numere naturale, prețurile inițiale ale cărților.
- **Linia 3:** Un număr natural  $Q$ , numărul de zile.
- **Pentru fiecare din cele  $Q$  zile:**
  - O linie cu un număr natural  $t$ , numărul de operații din ziua respectivă.
  - Următoarele  $t$  linii, fiecare descriind o operație.

## Tipuri de operații:

- **Modificare preț:** `op index1 index2 număr`  
`op` este fie `+` (adaugă număr la prețurile cărților de la `index1` la `index2` inclusiv), fie `-` (scade număr din prețurile cărților de la `index1` la `index2` inclusiv).
- **Undo:** `u` — anulează ultima operație de modificare neanulată (dacă există).
- **Redo:** `r` — reface ultima operație anulată care nu a fost refăcută (dacă există).

## Date de ieșire

Fișierul `librarie.out` va conține  $Q$  linii. Fiecare linie va conține, starea finală a prețurilor cărților la sfârșitul zilei respective, separate prin spații.

## Restricții

- $1 \leq n \leq 10^5$
- $1 \leq Q \leq 100$
- $0 \leq \text{prețurile inițiale} \leq 10^9$
- Pentru fiecare zi,  $1 \leq t < 10^5$
- $-10^4 \leq \text{număr} \leq 10^4$
- $1 \leq \text{index1} \leq \text{index2} \leq n$

## Observații

- Este garantat că toate operațiile vor fi valide.
- Prețurile pot deveni negative în urma operațiilor.
- Starea se acumulează de la o zi la alta: ziua  $i + 1$  începe cu prețurile rezultate la finalul zilei  $i$ , iar istoricul operațiilor (pentru undo/redo) se menține între zile.
- În cazul în care se efectuează o nouă operație de modificare după una sau mai multe comenzi  $u$ , istoricul operațiilor se păstrează, permițând astfel ca operațiile anulate să poată fi refăcute ulterior prin comanda  $r$ .

## Punzare

- Pentru teste în valoare de **16** puncte, numărul de operații de tipul undo/redo este 0.
- Pentru alte teste în valoare de **15** puncte,  $Q = 1$ .
- Pentru alte teste în valoare de **27** de puncte,  $L = 1, R = N$ , pentru fiecare operație de modificare de preț.
- Pentru alte teste în valoare de **42** de puncte, nu există restricții suplimentare.

## Exemple

librarie.in	librarie.out
<pre> 5 10 20 30 40 50 2 3 + 1 3 5 - 2 5 10 + 4 5 2 3 u u u </pre>	<pre> 15 15 25 32 42 10 20 30 40 50 </pre>
<pre> 10 4 8 5 9 1 2 3 7 11 12 2 4 + 2 5 2 - 1 3 2 + 1 7 3 + 1 7 3 5 u r + 3 8 4 - 4 10 1 r </pre>	<pre> 8 14 11 17 9 8 9 7 11 12 8 14 15 20 12 11 12 10 10 11 </pre>

## Explicații

### Exemplul 1

#### Ziua 1:

Inițial: [10, 20, 30, 40, 50].

1. + 1 3 5: se adaugă 5 la pozițiile 1–3, rezultând [15, 25, 35, 40, 50].
2. - 2 5 10: se scade 10 din pozițiile 2–5, rezultând [15, 15, 25, 30, 40].
3. + 4 5 2: se adaugă 2 la pozițiile 4–5, rezultând [15, 15, 25, 32, 42].

La finalul zilei 1, se obține starea: 15 15 25 32 42.

#### Ziua 2:

Inițial: [15, 15, 25, 32, 42].

Operațiile sunt:

- u: anulează ultima operație (+ 4 5 2), rezultând [15, 15, 25, 30, 40].
- u: anulează următoarea operație (care a fost - 2 5 10), rezultând [15, 25, 35, 40, 50].
- u: anulează următoarea operație (care a fost + 1 3 5), rezultând [10, 20, 30, 40, 50].

La finalul zilei 2, se obține starea: 10 20 30 40 50.

### Exemplul 2

#### Ziua 1:

Inițial: [4, 8, 5, 9, 1, 2, 3, 7, 11, 12].

1. + 2 5 2: se adaugă 2 la pozițiile 2–5, rezultând [4, 10, 7, 11, 3, 2, 3, 7, 11, 12].
2. - 1 3 2: se scade 2 din pozițiile 1–3, rezultând [2, 8, 5, 11, 3, 2, 3, 7, 11, 12].
3. + 1 7 3: se adaugă 3 la pozițiile 1–7, rezultând [5, 11, 8, 14, 6, 5, 6, 7, 11, 12].
4. + 1 7 3: se adaugă încă 3 la pozițiile 1–7, rezultând [8, 14, 11, 17, 9, 8, 9, 7, 11, 12].

La finalul zilei 1, starea este: 8 14 11 17 9 8 9 7 11 12.

#### Ziua 2:

Inițial: [8, 14, 11, 17, 9, 8, 9, 7, 11, 12].

Operațiile sunt:

- u: anulează ultima operație (+ 1 7 3), rezultând [5, 11, 8, 14, 6, 8, 9, 7, 11, 12].
- r: refă operația anulată, revenind la [8, 14, 11, 17, 9, 8, 9, 7, 11, 12].
- + 3 8 4: se adaugă 4 la pozițiile 3–8, rezultând [8, 14, 15, 21, 13, 8, 9, 11, 11, 12].
- - 4 10 1: se scade 1 din pozițiile 4–10, rezultând [8, 14, 15, 20, 12, 8, 9, 10, 10, 11].
- r: nu există nicio operație anulată la acest moment (comanda nu are efect), deci starea rămâne [8, 14, 15, 20, 12, 8, 9, 10, 10, 11].

La finalul zilei 2, starea este: 8 14 15 20 12 8 9 10 10 11.