



## Stena (Wall)

Ja-n'ko stavia stenu z tehličiek. Všetky tehličky majú rovnakú veľkosť. Stena sa skladá z  $n$  stĺpcov tehličiek, ktoré sú zľava doprava očíslované od 0 do  $n - 1$ . Tieto stĺpce môžu mať rôznu výšku. Výškou stĺpca rozumieme počet tehličiek v ňom.

Ja-n'ko stavia stenu nasledovne: Na začiatku sú všetky stĺpce prázdne. Potom Ja-n'ko postupne vykoná  $k$  krokov pridávania a odoberania tehličiek. (Kroky číslujeme od 0 po  $k - 1$ .) Celá stavba steny bude kompletná, až keď sa vykoná všetkých  $k$  krokov. V každom kroku Ja-n'ko dostáva interval čísel stĺpcov a veľkosť  $h$  a na základe toho vykoná nasledovné:

- V *pridávacom* kroku Ja-n'ko pridá tehličky do všetkých stĺpcov zadaného intervalu, ktoré majú menej ako  $h$  tehličiek, tak, aby mali výšku presne  $h$ . Nerobí nič so stĺpcami, ktoré majú  $h$  alebo viac tehličiek.
- V *odoberacom* kroku Ja-n'ko odoberie tehličky zo všetkých stĺpcov zadaného intervalu, ktoré majú viac ako  $h$  tehličiek, tak, aby mali výšku presne  $h$ . Nerobí nič so stĺpcami, ktoré majú  $h$  alebo menej tehličiek.

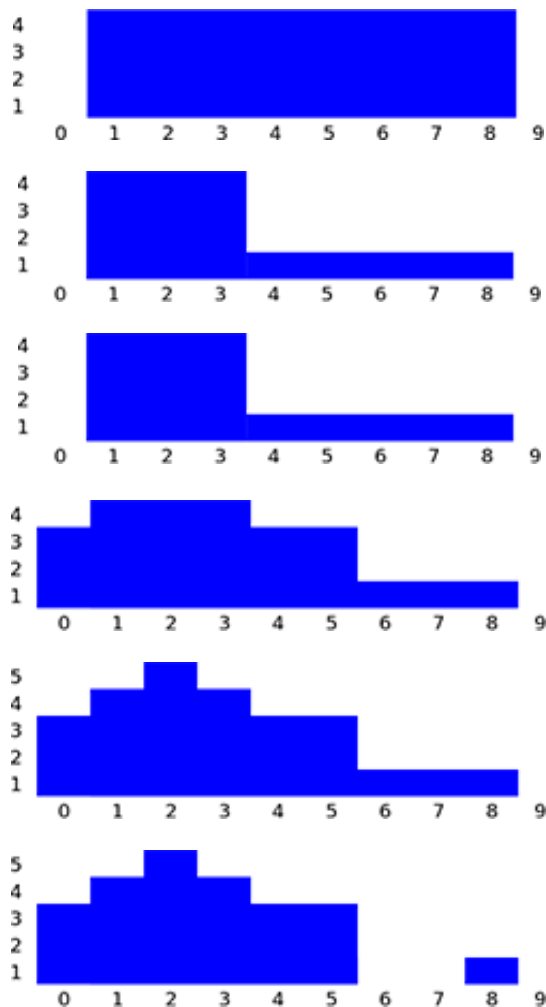
Tvojou úlohou bude zistiť výsledný tvar steny.

### Príklad

Predpokladáme, že máme 10 stĺpcov tehličiek a stenu staviame na 6 krokov. Všetky intervaly v nasledovnej tabuľke sú uzavreté, patria do nich teda aj obe hranice. Diagramy steny po každom kroku sú zobrazené pod tabuľkou.

krok	typ	interval	výška
0	pridaj	stĺpce 1 až 8	4
1	odober	stĺpce 4 až 9	1
2	odober	stĺpce 3 až 6	5
3	pridaj	stĺpce 0 až 5	3
4	pridaj	stĺpec 2	5
5	odober	stĺpce 6 až 7	0

Keďže na začiatku sú všetky stĺpce prázdne, po 0. kroku budú v každom zo stĺpcov od 1 do 8 po 4 tehličky. Stĺpce 0 a 9 zostávajú prázdne. V kroku 1 sa odoberajú tehličky zo stĺpcov od 4 do 9, až kým v nich nezostane nanajviš 1 tehlička. Stĺpec 9 teda zostáva prázdny a stĺpce od 0 do 3 ostávajú bez zmeny, nakoľko sú mimo rozsahu intervalu. Krok 2 nič nezmení, keďže stĺpce od 3 do 6 nemajú viac ako 5 tehličiek. Po 3. kroku sa počet tehličiek v stĺpcoch 0, 4 a 5 zvýši na 3. Po kroku 4 sa v stĺpci 2 zvýši počet tehličiek na 5. Krok 5 odoberie všetky tehličky zo stĺpcov 6 a 7.



## Úloha

Zisti počet tehličiek v jednotlivých stĺpcoch steny po skončení všetkých  $k$  krokov. Implementuj funkciu `buildWall`:

- `buildWall(n, k, op, left, right, height, finalHeight)`
- $n$ : počet stĺpcov steny.
- $k$ : počet krokov.
- `op`: pole veľkosti  $k$ ; kde pre všetky  $0 \leq i \leq k - 1$ : `op[i]` je typ  $i$ -teho kroku: 1 pre prídavací krok a 2 pre odoberací.
- `left` a `right`: polia veľkosti  $k$ ; označujú interval stĺpcov pre  $i$ -ty krok, teda pre každé  $0 \leq i \leq k - 1$  popisujeme interval od `left[i]` do `right[i]` (vrátane oboch koncových bodov `left[i]` a `right[i]`). Pritom vždy bude platiť, že `left[i] ≤ right[i]`.
- `height`: pole veľkosti  $k$ , kde pre všetky  $0 \leq i \leq k - 1$ : `height[i]` je výška použitá v  $i$ -tom kroku.
- `finalHeight`: pole veľkosti  $n$ ; tu by si mal vrátiť výsledné výšky stĺpcov, teda pre každé  $0 \leq i \leq n - 1$  bude `finalHeight[i]` obsahovať výšku  $i$ -teho stĺpca po skončení všetkých krokov.

## Podúlohy

V každej podúlohe platí, že všetky výskyty parametra `height` sú nezáporné celé čísla menšie alebo rovné **100,000**.

podúloha	body	$n$	$k$	poznámka
1	8	$1 \leq n \leq 10,000$	$1 \leq k \leq 5,000$	bez ďalších obmedzení
2	24	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	najskôr niekoľko volaní pridáva tehličky, potom niekoľko volaní tehličky odoberá
3	29	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	bez ďalších obmedzení
4	39	$1 \leq n \leq 2,000,000$	$1 \leq k \leq 500,000$	bez ďalších obmedzení

## Detaily implementácie

Odvzdávaš presne jeden súbor, nazvaný `wall.c`, `wall.cpp` alebo `wall.pas`. V tomto súbore by mali byť implementovaná vyššie popísaná funkcia `buildWall`. Musí mať hlavičku uvedenú nižšie. Ak programuješ v C/C++, tvoj súbor musí vložiť (`include`) súbor `wall.h`.

### C/C++

```
void buildWall(int n, int k, int op[], int left[], int right[],
int height[], int finalHeight[]);
```

### Pascal

```
procedure buildWall(n, k : longint; op, left, right, height :
array of longint; var finalHeight : array of longint);
```

## Ukážkový grader

Ukážkový grader, ktorý máte k dispozícii, očakáva vstup v nasledovnom formáte:

- riadok 1:  $n, k$ .
- riadok  $2+i$  pre ( $0 \leq i \leq k - 1$ ):  $op[i], left[i], right[i], height[i]$ .