Zeď

Je-Ní-Ček staví zeď z cihel stejné velikosti. Zeď se skládá z n sloupců cihel, které jsou očíslovány od 0 do $n-1$ zleva doprava. Sloupce mohou být různě vysoké a jejich výšky určuje počet cihel ve sloupci.

Je-Ní-Ček při stavbě postupuje následovně. Na začátku není v žádném sloupci žádná cihla. Stavba se skládá z k kroků, z nichž každý je buď *přidání*, nebo *odebrání* cihel. V každém kroku je zadán interval sloupců, výška h, a typ prováděného kroku. Je-Ní-Ček na základě těchto informací vykoná následující:

- Pokud je typ kroku *přidání*, přidá cihly do těch sloupců z daného intervalu, které obsahují méně než h cihel tak, aby v nich bylo přesně h cihel. S ostatními sloupci neudělá nic.
- Pokud je typ kroku *odebrání*, odebere cihly z těch sloupců z daného intervalu, které obsahují více než h cihel tak, aby v nich bylo přesně h cihel. S ostatními sloupci neudělá nic.

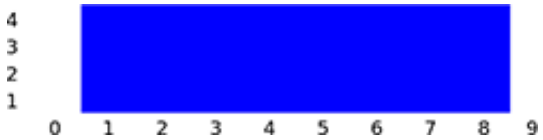
Vaším úkolem je určit výslednou podobu zdi.

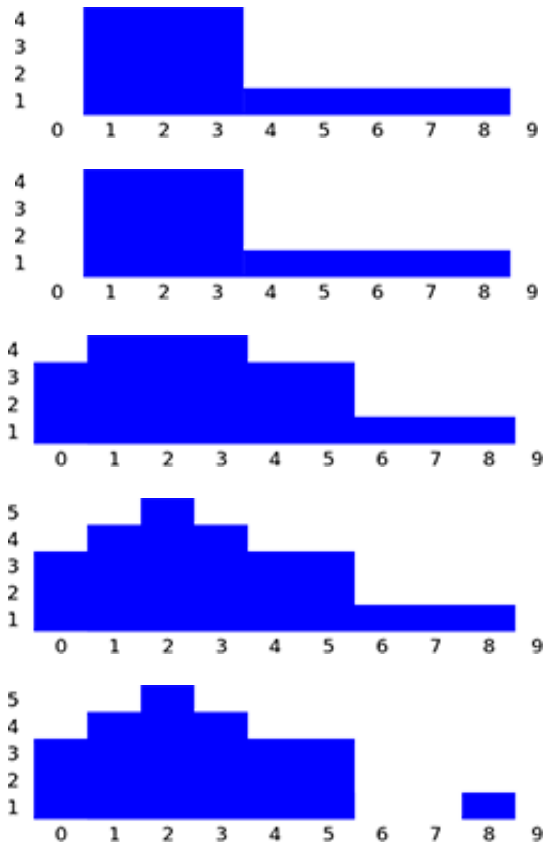
Příklad

Uvažujme 10 sloupců a 6 kroků. Všechny intervaly v následující tabulce zahrnují i oba krajní sloupce. Níže je znázorněn tvar zdi po každém kroku.

krok	typ	interval	výška
0	přidání	sloupce 1 až 8	4
1	odebrání	sloupce 4 až 9	1
2	odebrání	sloupce 3 až 6	5
3	přidání	sloupce 0 až 5	3
4	přidání	sloupec 2	5
5	odebrání	sloupce 6 až 7	0

Jelikož všechny sloupce jsou na začátku prázdné, po kroku 0 obsahuje každý ze slouců 1 až 8 právě 4 cihly. Sloupce 0 až 9 zůstanou prázdné. Po kroku 1 zbyde v každém ze sloupců 4 až 8 jen jedna cihla, sloupec 9 zůstane prázdný. Sloupce 0 až 4, které jsou mimo interval, jsou nezměněny. Krok 2 nezpůsobí žádné změny, neboť žádný ze sloupců 3 až 6 neobsahuje více než 5 cihel. Po třetím kroku počet cihel ve sloupcích 0, 4 a 5 vzroste na 3. Po kroku 4 ve sloupci 2 bude 5 cihel. V kroku 5 Je-Ní-Ček odebere všechny cihly ze sloupců 6 a 7.





Úloha

Je dán popis k kroků stavby zdi. Spočtěte, kolik cihel bude v každém sloupci po vykonání všech kroků. Za tímto účelem implementujte funkci `buildWall`.

- `buildWall(n, k, op, left, right, height, finalHeight)`
  - `n`: počet sloupců.
  - `k`: počet kroků.
  - `op`: pole délky k; `op[i]` je typ kroku i: 1 pokud se jedná o přidání, 2 pokud se jedná o odebrání, pro $0 \leq i \leq k-1$.
  - `left` a `right`: pole délky k; interval sloupců použitý v i-tém kroku začíná sloupcem číslo `left[i]` a končí sloupcem číslo `right[i]` (včetně obou krajních sloupců), pro $0 \leq i \leq k-1$. Vždy bude platit `left[i]` $\leq$ `right[i]`.
  - `height`: pole délky k; `height[i]` je určení výšky v i-tém kroku, pro $0 \leq i \leq k-1$.
  - `finalHeight`: pole délky n; na pozici i zapište výšku i-tého sloupce po dokončení stavby, pro $0 \leq i \leq n-1$.

Podúlohy

Pro všechny podúlohy jsou všechny zadané výšky nezáporná celá čísla menší nebo rovna $100,000$.

podúloha	počet bodů	n	k	další omezení

podúloha	počet bodů	n	k	další omezení
1	8	$1 \leq n \leq 10,000$	$1 \leq k \leq 5,000$	žádná
2	24	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	nejprve se provádějí všechny kroky typu přidání a až poté všechny kroky typu odebrání
3	29	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	žádná
4	39	$1 \leq n \leq 2,000,000$	$1 \leq k \leq 500,000$	žádná

Upřesnění implementace

Odevzdejte právě jeden soubor pojmenovaný `wall.c`, `wall.cpp` nebo `wall.pas`. Tento soubor implementuje funkci popsanou výše s následujícími parametry. Pro jazyk C/C++ nezapomeňte vložit hlavičkový soubor `wall.h`.

Program v C/C++

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

Program v Pascalu

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

Ukázkový vyhodnocovač

Ukázkový vyhodnocovač čte vstup v následujícím formátu:

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