Fal

Jian-Jia azonos méretű téglákból n oszlopból álló falat épít. Az oszlopokat 0-tól $n-1$-ig sorszámozzuk. Az oszlop magassága a benne levő téglák száma.

Kezdetben minden oszlop üres. Ezután k lépésben, lépésenként vagy hozzáad, vagy elvesz téglákat két oszlop közötti szakaszon. Az oszlop intervallumok mindenhol zártak, azaz beleértjük a két végét is.

- Hozzáadás esetén két oszlop közötti összes olyan oszlopot h magasságúra épít, amely alacsonyabb volt h-nál. A többieket nem változtatja.
- Elvétel esetén két oszlop közötti összes olyan oszlopot h magasságúra bont, amely magasabb volt h-nál. A többieket nem változtatja.

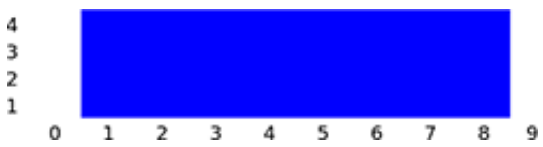
A feladatod: megadni k lépés után minden oszlop magasságát!

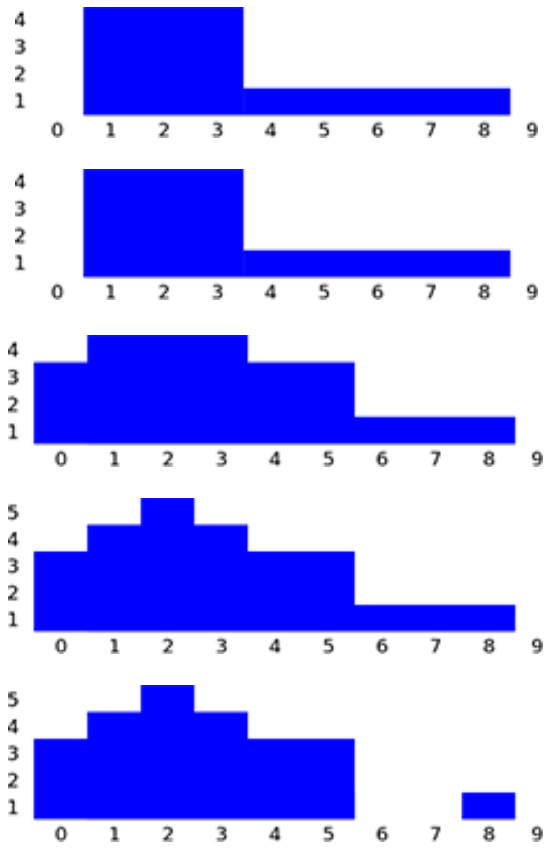
Példa

10 oszlopunk van és 6 lépésünk.

lépés	típus	intervallum	magasság
0	hozzáad	1..8	4
1	elvesz	4..9	1
2	elvesz	3..6	5
3	hozzáad	0..5	3
4	hozzáad	2..2	5
5	elvesz	6..7	0

Kezdetben minden oszlop üres.
A 0. lépés után az 1..8 oszlopok 4 magasságúak lesznek, a 0. és a 9. oszlop üres marad.
Az 1. lépés után a 4..8 oszlopok 1 magasságúra változnak, a 9. oszlop 0 marad, a 0..3 nem változik, mert az intervallumon kívüliek.
A 2. lépés nem változtatja meg a 3..6 oszlopok értékét, mert egyik sem volt 5-nél magasabb.
A 3. lépés után a 0,4,5 oszlopok 3 magasságúra nőnek.
A 4. lépés után a 2. oszlop 5 magasságú lesz.
Az 5. lépés elveszi az összes téglát a 6. és 7. oszlopból.





Feladat

A k lépés leírása alapján számítsd ki, hogy hány tégla lesz az egyes oszlopokban a k lépés után! A `buildWall` függvényt kell megírnod!

- `buildWall(n, k, op, left, right, height, finalHeight)`
  - `n`: az oszlopok száma.
  - `k`: a lépések száma.
  - `op`: k elemű tömb; az `op[i]` az i. lépés típusa: értéke 1 hozzáadásnál, illetve 2 elvételnél ($0 \leq i \leq k-1$).
  - `left` és `right`: k elemű tömbök; az i. lépés intervalluma bal végpontja `left[i]` és jobb végpontja `right[i]` (az intervallum eleme `left[i]` és `right[i]` is, $0 \leq i \leq k-1$, `left[i]` $\leq$ `right[i]`).
  - `height`: k elemű tömb; `height[i]` értéke az i. lépésben kapott h magasság paraméter ($0 \leq i \leq k-1$).
  - `finalHeight`: n elemű tömb; ebben kell megadnod az eredményt, azaz a `finalHeight[i]` értéke legyen az i. oszlop magassága a lépések után ($0 \leq i \leq n-1$).

Részfeladatok

Minden tesztesetben a magasság paraméter legfeljebb $100,000$.

részf.	pont	n	k	megjegyzés
1	8	$1 \leq n \leq 10,000$	$1 \leq k \leq 5,000$	nincs más korlát
2	24	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	az összes hozzáadás megelőzi az összes elvételt
3	29	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	nincs más korlát
4	39	$1 \leq n \leq 2,000,000$	$1 \leq k \leq 500,000$	nincs más korlát

Megvalósítás

A `wall.c`, `wall.cpp` vagy `wall.pas` fájlt kell beadnod! Ebben a kért függvényt kell megvalósítanod! C/C++ esetén include-olnod kell a `wall.h`-t!

C/C++ program

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

Pascal program

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

Minta értékelő

A mintaértékelő az alábbi formában olvassa az adatokat:

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