



## Sein

Jian-Jia ehitab seinu, ladudes üksteise otsa ühesuuruseid kive. Sein koosneb  $n$  tulbast kividest, mis on nummerdatud vasakult paremale  $0 \dots n - 1$ . Tulpadel kõrgused võivad olla erinevad. Tulba kõrguseks on kivide arv selles tulbas.

Jian-Jia ehitab seinu järgneval viisil: Alguses pole üheski tulbas ühtki kivi. Seejärel läbib Jian-Jia  $k$  kivide *lisamise* või *eemaldamise* etappi. Ehitamine lõpeb, kui kõik  $k$  etappi on läbitud. Igas etapis saab Jian-Jia lõigu järjestikuseid kivitulpasid ja kõrguse  $h$  ning sooritab järgmise protseduuri:

- *Lisamise* etapis lisab Jian-Jia etteantud lõigus kive nendesse tulpadesse, kus on vähem kui  $h$  kivi, nii et neis oleks täpselt  $h$  kivi. Tulpadega, kus juba on  $h$  või rohkem kivi, ei tee ta midagi.
- *Eemaldamise* etapis eemaldab Jian-Jia etteantud lõigus kive nendest tulpadest, kus on rohkem kui  $h$  kivi, nii et alles jääb  $h$  kivi. Tulpadega, kus juba on  $h$  või vähem kivi, ei tee ta midagi.

Sinu ülesanne on leida seinu lõplik kuju.

## Näide

Olgu meil on 10 tulpa ja 6 etappi. Kõik tabelis toodud lõigud sisaldavad oma otspunkte. Järgnevad diagrammid näitavad, milline on sein iga etapi järel.

etapp	tüüp	lõik	kõrgus
0	lisa	tulbad 1 kuni 8	4
1	eemalda	tulbad 4 kuni 9	1
2	eemalda	tulbad 3 kuni 6	5
3	lisa	tulbad 0 kuni 5	3
4	lisa	tulp 2	5
5	eemalda	tulbad 6 kuni 7	0

Kuna kõik tulpad on alguses tühjad, on pärast etappi 0 tulpades 1 kuni 8 igaühes neli kivi. Tulpad 0 ja 9 jäävad tühjaks.

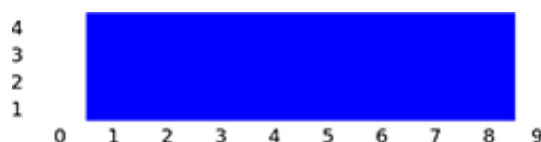
Etapis 1 eemaldatakse tulpadest 4 kuni 8 kive, kuni igaühes neist on üks kivi, ning tulp 9 jääb endiselt tühjaks. Tulpad 0 kuni 3, mis pole selles lõigus, jäävad muutumatuks.

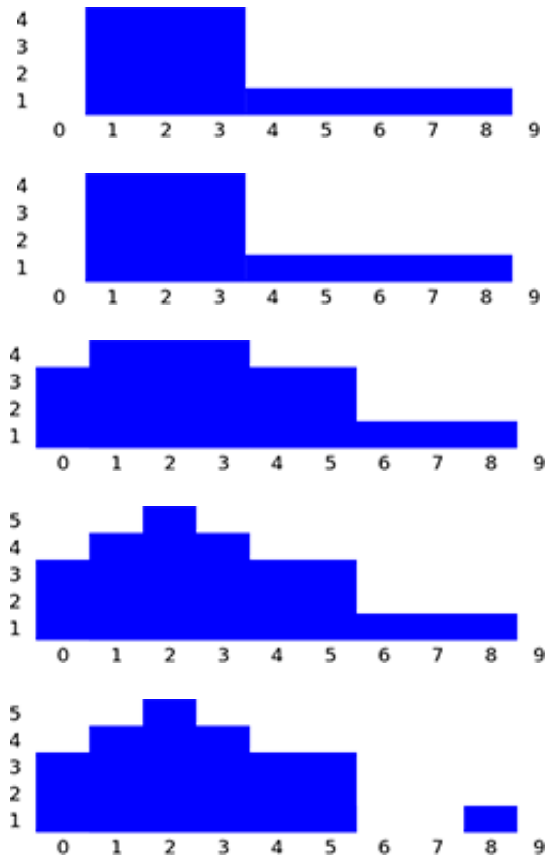
Etapp 2 ei muuda olukorda, sest tulpades 3 kuni 6 pole rohkem kui viis kivi.

Pärast etappi 3 suureneb kivide arv tulpades 0, 4 ja 5 kolmeni.

Pärast etappi 4 on tulbas 2 viis kivi.

Etapp 5 eemaldab kõik kivid tulpadest 6 ja 7.





## Ülesanne

Kui antud on  $k$  etapi kirjeldused, arvutada kivide arv igas tulpas pärast kõigi etappide lõppu. Realiseerida funktsioon `buildWall`.

- `buildWall(n, k, op, left, right, height, finalHeight)`
  - $n$ : tulpade arv seinas;
  - $k$ : etappide arv;
  - `op`: massiiv pikkusega  $k$ , kus `op[i]` on iga  $0 \leq i \leq k - 1$  korral etapi  $i$  tüüp: 1 tähistab lisamist ja 2 eemaldamist;
  - `left` ja `right`: massiivid pikkusega  $k$ , kus iga  $0 \leq i \leq k - 1$  korral tulpade lõik etapis  $i$  algab tulpaga `left[i]` ning lõpeb tulpaga `right[i]` (sisaldades otspunkte `left[i]` ja `right[i]`) ning alati kehtib `left[i] ≤ right[i]`;
  - `height`: massiiv pikkusega  $k$ , kus `height[i]` on iga  $0 \leq i \leq k - 1$  korral etapi  $i$  kõrgusparameeter;
  - `finalHeight`: massiiv pikkusega  $n$ ; arvutuse tulemus tuleb tagastada, seades iga  $0 \leq i \leq n - 1$  korral `finalHeight[i]` väärtuseks kivide arvu tulpas  $i$ .

## Alamülesanded

Igas alamülesandes on kõrguse parameeter mittenegatiivne täisarv, mis ei ületa 100 000.

alamülesanne	punkte	$n$	$k$	märkused
1	8	$1 \leq n \leq 10\,000$	$1 \leq k \leq 5\,000$	täiendavaid piiranguid pole
2	24	$1 \leq n \leq 100\,000$	$1 \leq k \leq 500\,000$	kõik lisamisetapid on enne kõiki eemaldamisetappe
3	29	$1 \leq n \leq 100\,000$	$1 \leq k \leq 500\,000$	täiendavaid piiranguid pole
4	39	$1 \leq n \leq 2\,000\,000$	$1 \leq k \leq 500\,000$	täiendavaid piiranguid pole

## Realisatsioon

Esitada tuleb täpselt üks fail nimega `wall.c`, `wall.cpp` või `wall.pas`. Selles failis peab olema eelpool kirjeldatud alamprogramm järgmise signatuuriga. C ja C++ programmides tuleb kaasata ka päisfail `wall.h`.

### C ja 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);
```

### Näidishindaja

Näidishindaja loeb sisendit järgmises vormingus:

- Esimesel real: arvud  $n$  ja  $k$ .
- Real  $2 + i$ , kus  $0 \leq i \leq k - 1$ : arvud `op[i]`, `left[i]`, `right[i]`, `height[i]`.