Wall

Jian-Jia bouwt een muur door bakstenen van hetzelfde formaat op elkaar te stapelen. Een muur bestaat uit n kolommen bakstenen, die van links naar rechts genummerd zijn van 0 tot en met $n-1$. De hoogte van elke kolom kan anders zijn. De hoogte van een kolom wordt uitgedrukt als het aantal bakstenen in de kolom.

Jian-Jia bouwt de muur als volgt. Aan het begin zijn er geen stenen in de kolommen. Hierna doorloopt Jian-Jia k een bouwproces van fasen van telkens *toevoegen* of *verwijderen* van bakstenen. Het bouwproces is af zodra alle k fasen zijn doorlopen. In elke fase werkt Jian-Jia met een subrij van aaneengesloten kolommen en een hoogte h. Hij voert dan de volgende procedure uit:

- In een *toevoeg* fase voegt Jian-Jia bakstenen toe aan de kolommen met minder dan h bakstenen, totdat ze precies h bakstenen bevatten. Met de kolommen die h of meer bakstenen bevatten doet hij niets.
- In een *verwijder* fase, verwijdert Jian-Jia bakstenen van de kolommen met meer dan h bakstenen, totdat ze precies h bakstenen bevatten. Met de kolommen die h of minder bakstenen bevatten doet hij niets.

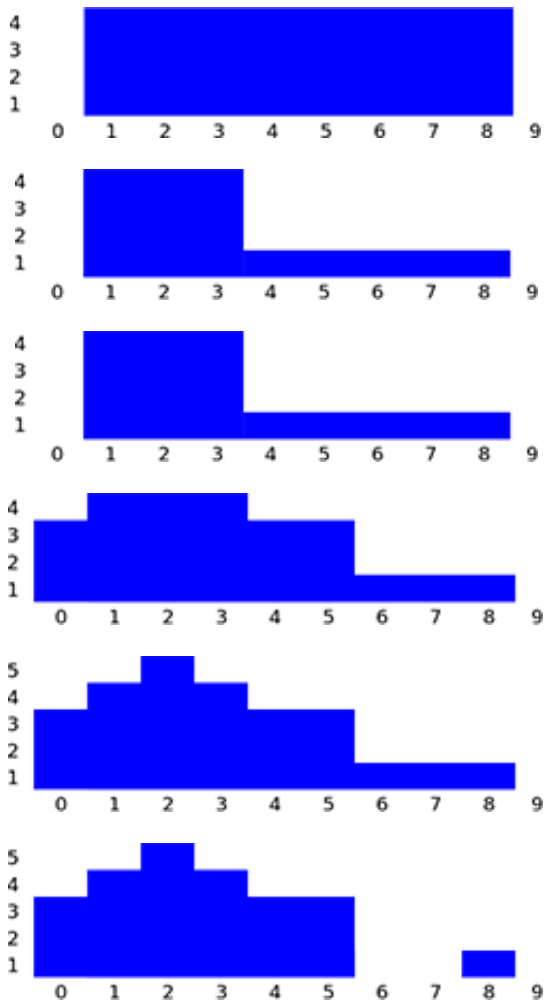
Bepaal het eindresultaat van de muur.

Voorbeeld

We gaan uit een muur die uit 10 kolommen bestaat. Er zijn 6 fases. Een diagram met hoe de muur eruit ziet na elke fase zie je ook.

fase	type	range	h
0	toevoegen	kolommen 1 tot en met 8	4
1	verwijderen	kolommen 4 tot en met 9	1
2	verwijderen	kolommen 3 tot en met 6	5
3	toevoegen	kolommen 0 tot en met 5	3
4	toevoegen	kolom 2	5
5	verwijderen	kolommen 6 tot en met 7	0

In het begin zijn alle kolommen leeg. Na fase 0 liggen er in elke kolom 1 tot met 8 precies 4 bakstenen. Kolommen 0 en 9 blijven leeg. In fase 1 worden de bakstenen van kolommen 4 tot en met 8 verwijderd totdat er op deze kolommen precies 1 baksteen over is. De kolommen 0 tot en met 3 veranderen niet. In fase 2 gebeurt er niets omdat geen van de kolommen 3 tot en met 6 meer dan 5 bakstenen heeft. Na fase 3 wordt het aantal bakstenen in kolommen 0, 4 en 5 verhoogd tot 3. Na fase 4 zijn er 5 bakstenen in kolom 2. Tenslotte verwijdert fase 5 alle bakstenen uit kolommen 6 en 7.



Opdracht

Je krijgt de beschrijving van de k fases. Bereken het aantal bakstenen dat in elke kolom is overgebleven nadat alle fases zijn uitgevoerd.

Je moet de functie `buildWall` implementeren.

- `buildWall(n, k, op, left, right, height, finalHeight)`
  - `n`: het aantal kolommen in de muur.
  - `k`: het aantal fasen.
  - `op`: array van lengte k; `op[i]` is het type fase i: 1 staat voor toevoegen; en 2 voor verwijderen, met $0 \leq i \leq k-1$.
  - `left` en `right`: arrays van lengte k; de rij van kolommen in fase i loopt vanaf `left[i]` tot en met `right[i]` (dus de eerste en laatste kolommen `left[i]` en `right[i]` zijn inbegrepen), met $0 \leq i \leq k-1$. Er geldt altijd `left[i]` $\leq$ `right[i]`.
  - `height`: array van lengte k; `height[i]` is de hoogte parameter van fase i, met $0 \leq i \leq k-1$.
  - `finalHeight`: array van lengte n; je moet je antwoord aanleveren door het uiteindelijke aantal bakstenen in kolom i op te slaan in `finalHeight[i]`, met $0 \leq i \leq n-1$.

Subtasks

Voor alle subtaken geldt dat de hoogte parameter in alle fases een niet-negatieve integer is kleiner dan of gelijk aan 100000.

subtask	punten	n	k	opmerkingen
1	8	$1 \leq n \leq 10000$	$1 \leq k \leq 5000$	geen aanvullende limieten
2	24	$1 \leq n \leq 100000$	$1 \leq k \leq 500000$	alle toevoeg fasen zijn voor de verwijder fasen
3	29	$1 \leq n \leq 100000$	$1 \leq k \leq 500000$	geen aanvullende limieten
4	39	$1 \leq n \leq 2000000$	$1 \leq k \leq 500000$	geen aanvullende limieten

Implementatie details

Stuur één bestand `wall.c`, `wall.cpp` of `wall.pas` in.
Dit bestand implementeert een subprogramma zoals hierboven beschreven op basis van de hieronder beschreven interface. Voor een C/C++ programma moet je ook een header-bestand `wall.h` insturen.

C/C++ programma

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

Pascal programma

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

Voorbeeld grader

De voorbeeld grader leest de invoer in het volgende formaat:

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