



Wall

Jian-Jia bygger en mur ved å stable murstein av lik størrelse oppå hverandre. Denne muren består av n søyler med murstein, som er nummerert 0 til $n - 1$ fra venstre til høyre. Disse søylene kan ha forskjellige høyder. Høyden til en søyle er antallet murstein i den.

Jian-Jia bygger muren på følgende måte. Til å begynne med er det ingen murstein i noen av søylene. Deretter går Jian-Jia gjennom k faser med *pålegging* eller *fjerning* av murstein. Byggeprosessen er ferdig når alle de k fasene er ferdige. I hver fase får Jian-Jia oppgitt en sammenhengende sekvens med søyler og en høyde h , og utfører følgende prosedyre:

- I en *påleggingsfase* legger Jian-Jia mursteiner oppå de søylene i den gitte sekvensen som har en høyde som er mindre enn h , slik at de får eksakt h murstein i seg. Han gjør ingenting med søylene som har h eller flere mursteiner i seg fra før av.
- • I en *fjerningsfase* fjerner Jian-Jia mursteiner fra de søylene i den gitte sekvensen som har mer enn h mursteiner i seg, slik at de har eksakt h mursteiner igjen i seg. Han gjør ingenting med søylene som har h eller færre mursteiner i seg fra før av.

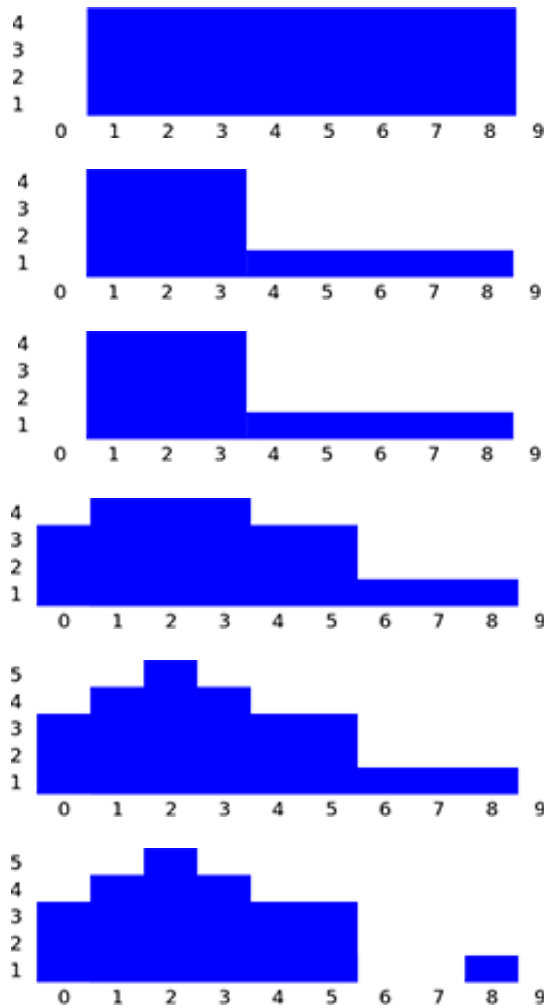
Opgaven din er å avgjøre hvilken form muren får til slutt.

Eksempel

Vi antar at et er 10 mursteins søyler og 6 murbygningsfaser. Alle sekvensene i den følgende tabellen er inklusive. Illustrasjoner av muren etter hver fase er vist nedenfor.

phase	type	range	height
0	legg på	søyle 1 til og med 8	4
1	fjern	søyle 4 til og med 9	1
2	fjern	søyle 3 til og med 6	5
3	legg på	søyle 0 til og med 5	3
4	legg på	søyle 2	5
5	fjern	søyle 6 til og med 7	0

Siden alle søylene i utgangspunktet er tomme, vil søyle 1 til og med 8 ha 4 mursteiner i seg etter fase 0. Søyle 0 og 9 forblir tomme. I fase 1 blir søylene fjernet fra og med søyle 4 til og med 8 inntil hver av dem har 1 murstein, og søyle 9 forblir tom. Søyle 0 til og med 3, som er utenfor den gitte sekvensen, forblir uendret. Fase 2 gjør ingen endringer siden ingen av søylene fra og med 3 til og med 6 har flere enn 5 mursteiner. Etter fase 3 øker antallet mursteiner i søyle 0, 4 og 5 til 3. Det er 5 mursteiner i søyle 2 etter fase 4. Fase 5 fjerner alle mursteinene fra søyle 6 og 7.



Oppgave

Gitt beskrivelsen av de k fasene, regn ut antallet murstein i hver søyle etter at alle fasene er ferdige. Du må implementere funksjonen `buildWall`.

- `buildWall(n, k, op, left, right, height, finalHeight)`
 - n : antall søyler i muren.
 - k : antall faser.
 - `op`: array av lengde k ; `op[i]` er typen til fase i : 1 for en påleggingsfase og 2 for en fjerningsfase, for $0 \leq i \leq k - 1$.
 - `left` og `right`: arrayer av lengde k ; kolonnesequensen i fase i starter med søyle `left[i]` og slutter med søyle `right[i]` (inklusive begge endepunktene `left[i]` og `right[i]`), for $0 \leq i \leq k - 1$. Du vil alltid ha `left[i] ≤ right[i]`.
 - `height`: array av lengde k ; `height[i]` er høydeparameteret til fase i , for $0 \leq i \leq k - 1$.
 - `finalHeight`: array av lengde n ; du skal returnere resultatene dine ved å legge det endelige antallet murstein i søyle i inn i `finalHeight[i]`, for $0 \leq i \leq n - 1$.

Deloppgaver

For alle deloppgavene er høydeparametrene til alle fasene ikke-negative heltall som er mindre enn eller lik **100,000** (kommaet er et tusenskille, ikke et desimaltegn).

subtask	points	n	k	note
1	8	$1 \leq n \leq 10,000$	$1 \leq k \leq 5,000$	ingen ytterligere begrensninger
2	24	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	alle påleggingsfasene kommer før alle fjerningsfasene
3	29	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	ingen ytterligere begrensninger
4	39	$1 \leq n \leq 2,000,000$	$1 \leq k \leq 500,000$	ingen ytterligere begrensninger

Implementasjonsdetaljer

Du må sende inn eksakt én fil, som må hete `wall.c`, `wall.cpp` eller `wall.pas`. Denne filen skal implementere funksjonen som er beskrevet ovenfor med den følgende signaturen. Du må inkludere headerfilen `wall.h` hvis du programmerer i C/C++.

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);
```

Eksempel-grader

Eksempel-graderen leser input på det følgende formatet:

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