



## Ściana (Wall)

Jian-Jia buduje ścianę za pomocą jednakowych cegieł. Ściana składa się z  $n$  kolumn cegieł, ponumerowanych od 0 do  $n - 1$ , patrząc od lewej do prawej. Kolumny mogą mieć różne wysokości, przy czym wysokością kolumny nazywamy liczbę cegieł w tej kolumnie.

Jian-Jia buduje ścianę w następujący sposób. Początkowo w żadnej kolumnie nie ma żadnych cegieł. Następnie Jian-Jia wykonuje  $k$  faz, z których każda polega na *dodawaniu* lub *usuwaniu* cegieł. Proces budowania kończy się, gdy wszystkie  $k$  faz zostaje ukończonych. W każdej fazie Jian-Jia ma wybrany spójny ciąg kolumn oraz wysokość  $h$  i wykonuje następujące czynności:

- W fazie *dodawania* Jian-Jia w każdej kolumnie o wysokości mniejszej niż  $h$  w zadanym ciągu kolumn umieszcza dodatkowo tyle cegieł, by miała ona wysokość dokładnie  $h$ . Kolumny o wysokości nie mniejszej niż  $h$  pozostawia bez zmian.
- W fazie *usuwania* Jian-Jia z każdej kolumny o wysokości większej niż  $h$  w zadanym ciągu kolumn usuwa tyle cegieł, by miała ona wysokość dokładnie  $h$ . Kolumny o wysokości co najwyżej  $h$  pozostawia bez zmian.

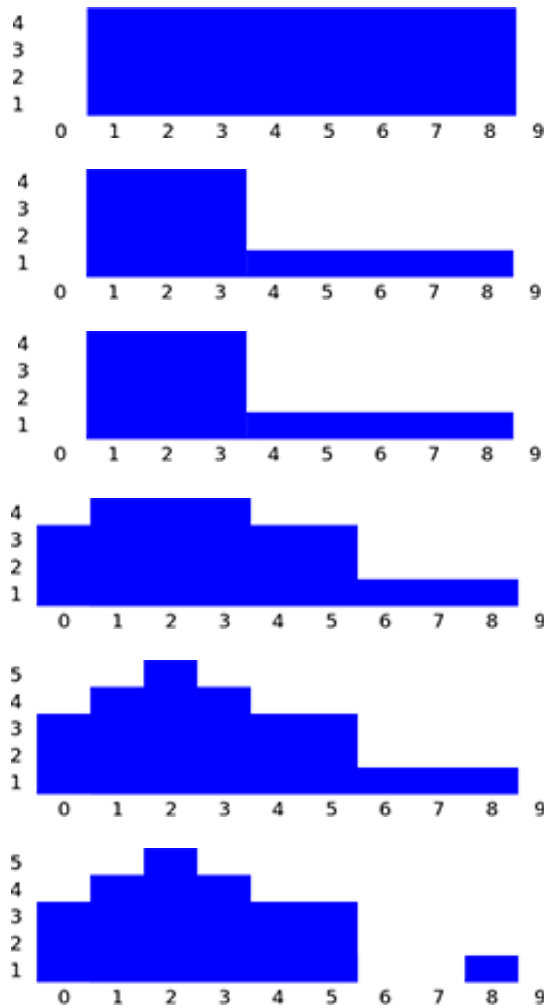
Twoim zadaniem jest wyznaczyć ostateczny kształt ściany.

### Przykład

Założmy, że ściana składa się z 10 kolumn cegieł, a nasz bohater wykonuje 6 faz. Wszystkie zakresy wymienione w poniższej tabeli zawierają końce. Rysunki poglądowe ściany po wykonaniu kolejnych faz są umieszczone poniżej.

faza	typ fazy	zakres	wysokość
0	dodawanie	kolumny od 1 do 8	4
1	usuwanie	kolumny od 4 do 9	1
2	usuwanie	kolumny od 3 do 6	5
3	dodawanie	kolumny od 0 do 5	3
4	dodawanie	kolumna 2	5
5	usuwanie	kolumny od 6 do 7	0

Ponieważ początkowo kolumny są puste, to po fazie 0 każda z kolumn od 1 do 8 zawiera po 4 cegły. Kolumny 0 i 9 pozostają puste. W fazie 1, cegły zostają usunięte z kolumn od 4 do 8 tak, że każda z tych kolumn zawiera po 1 cegle, natomiast kolumna 9 nadal jest pusta. Kolumny od 0 do 3, które znajdują się poza przedziałem usuwania, pozostają nietknięte. W fazie 2 nic się nie zmienia, ponieważ każda z kolumn od 3 do 6 składa się z nie więcej niż 5 cegieł. Po fazie 3 liczba cegieł w każdej z kolumn 0, 4 i 5 wzrasta do 3. Po fazie 4 kolumna 2 zawiera 5 cegieł. W fazie 5 z kolumn 6 i 7 zostaną usunięte wszystkie cegły.



## Zadanie

Mając dany opis  $k$  faz, wyznacz liczbę cegieł w poszczególnych kolumnach po wykonaniu wszystkich faz. Powinieneś napisać funkcję `buildWall`.

- `buildWall(n, k, op, left, right, height, finalHeight)`
  - $n$ : liczba kolumn.
  - $k$ : liczba faz.
  - `op`: tablica rozmiaru  $k$ ; `op[i]` określa typ fazy  $i$ : 1 dla dodawania i 2 dla usuwania, dla  $0 \leq i \leq k - 1$ .
  - `left` oraz `right`: tablice rozmiaru  $k$ ; ciąg kolumn rozważany w fazie  $i$  rozpoczyna się kolumną `left[i]` i kończy się kolumną `right[i]` ( $i$  zawiera oba końce: `left[i]` oraz `right[i]`), dla  $0 \leq i \leq k - 1$ . Zawsze będzie zachodzić `left[i] ≤ right[i]`.
  - `height`: tablica rozmiaru  $k$ ; `height[i]` określa parametr wysokości w fazie  $i$ , dla  $0 \leq i \leq k - 1$ .
  - `finalHeight`: tablica rozmiaru  $n$ ; obliczona przez Twój program ostateczna wysokość kolumny  $i$  powinna znaleźć się w polu `finalHeight[i]`, dla  $0 \leq i \leq n - 1$ .

## Podzadania

We wszystkich podzadaniach parametry wysokości w każdej fazie są nieujemnymi liczbami całkowitymi nie większymi niż **100,000**.

podzadanie	liczba punktów	$n$	$k$	dodatkowe ograniczenia
1	8	$1 \leq n \leq 10,000$	$1 \leq k \leq 5,000$	brak
2	24	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	wszystkie fazy dodawania występują przed wszystkimi fazami usuwania
3	29	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	brak
4	39	$1 \leq n \leq 2,000,000$	$1 \leq k \leq 500,000$	brak

## Implementacja

Powinieneś zgłosić dokładnie jeden plik o nazwie `wall.c`, `wall.cpp` lub `wall.pas`. W pliku powinna znaleźć się implementacja funkcji podanej powyżej o następującej sygnaturze. W przypadku programu w C/C++ powinieneś także załączyć (*include*) plik nagłówkowy `wall.h`.

### Programy w C/C++

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

### Programy w Pascalu

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

### Przykładowy program sprawdzający

Przykładowy program sprawdzający wczytuje dane w następującym formacie:

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