



## Wall

Jian-Jia esta construyendo una pared apilando ladrillos del mismo tamaño. Esta pared consiste de  $n$  columnas de ladrillos, las cuales estan numeradas desde 0 hasta  $n - 1$  de izquierda a derecha. Las columnas pueden tener diferentes alturas. La altura de una columna es el numero de ladrillos que esta tiene.

Jian-Jia construye la pared como sigue. Inicialmente no hay ladrillos en ninguna columna. Entonces, Jian-Jia va a traves de  $k$  fases de *adicionar* o *eliminar* ladrillos. El proceso de construccion se completa cuando las  $k$  fases hayan finalizado. En cada fase a Jian-Jia le es dado un rango de columnas consecutivas de ladrillos y una altura  $h$ , y el hace el siguiente procedimiento:

- En una fase de *adicionar* ladrillos, Jian-Jia agrega ladrillos a aquellas columnas en el rango dado que tienen menos de  $h$  ladrillos, de tal manera que estas queden exactamente con  $h$  ladrillos. Jian-Jia no hace nada en las columnas que tienen  $h$  o más ladrillos.
- En una fase de *eliminar* ladrillos, Jian-Jia quita ladrillos de aquellas columnas en el rango dado que tengan más de  $h$  ladrillos, de tal manera que queden con exactamente  $h$  ladrillos. Jian-Jia no hace nada con las columnas que tienen  $h$  ladrillos o menos.

Tu tarea es determinar la forma final de la pared.

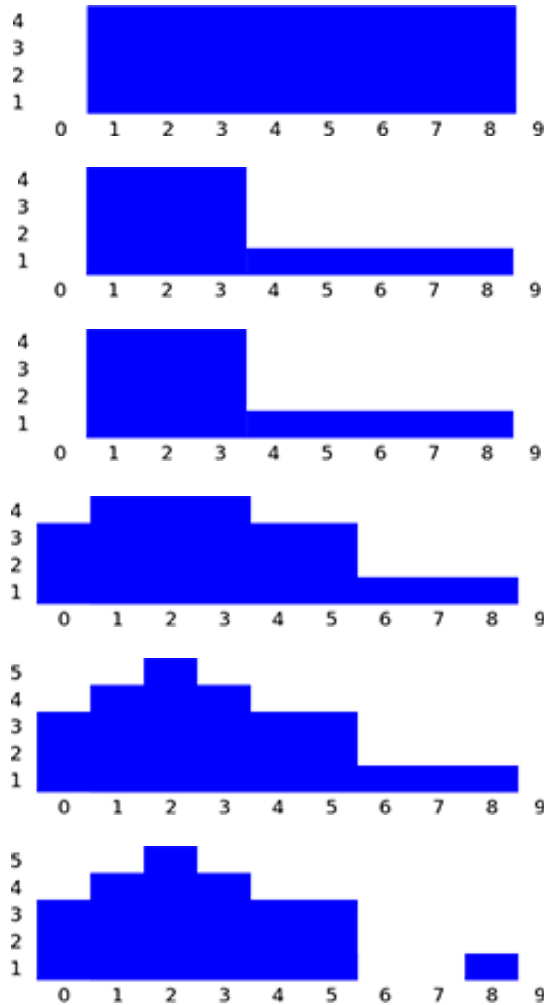
## Ejemplo

Nosotros asumimos que hay 10 columnas de ladrillos y 6 paredes en las fases de construccion. Todos los rangos en la siguiente tabla son inclusivos. Los diagramas de la pared despues de cada fase son mostrados debajo.

fase	tipo	rango	altura
0	adicionar	columnas 1 a 8	4
1	eliminar	columnas 4 a 9	1
2	eliminar	columnas 3 a 6	5
3	adicionar	columnas 0 a 5	3
4	adicionar	columna 2	5
5	eliminar	columnas 6 a 7	0

Ya que todas las columnas estan inicialmente vacias, despues de la fase 0 cada una de las columnas de la 1 a la 8 quedaran con 4 ladrillos. Las columnas 0 y 9 permanecen vacias. En la fase 1, los ladrillos son eliminados desde las columnas de la 4 a la 8 hasta que cada una de ellas tenga 1 ladrillo, y la columna 9 permanece vacia. Las columnas de la 0 a la 3, las cuales estan fuera del rango dado, permanecen sin cambiar. La fase 2 no hace cambios desde las columnas 3 a la 6 pues no tienen mas de 5 ladrillos. Despues de la fase 3 el numero de ladrillos en las columnas 0, 4, y 5 se incrementan a 3. Hay 5 ladrillos en la columna 2 despues de la fase 4. En la fase 5 se eliminan todos los ladrillos de la

columnas 6 y 7.



## Tarea

Dada la descripción de las  $k$  fases, calcule el número de ladrillos en cada columna después que todas las fases finalicen. Usted necesita implementar la función `buildWall`.

- `buildWall(n, k, op, left, right, height, finalHeight)`
  - $n$ : el número de columnas de la pared.
  - $k$ : el número de fases.
  - `op`: arreglo de tamaño  $k$ ; `op[i]` es el tipo de la fase  $i$ : 1 para una fase de adicionar y 2 para una fase de eliminar, para  $0 \leq i \leq k - 1$ .
  - `left` y `right`: arreglos de tamaño  $k$ ; el número de columnas en la fase  $i$  que comienza con la columna `left[i]` y finaliza con la columna `right[i]` (incluyendo ambos extremos `left[i]` y `right[i]`), para  $0 \leq i \leq k - 1$ . Usted siempre tendrá `left[i] ≤ right[i]`.
  - `height`: arreglo de tamaño  $k$ ; `height[i]` es el parámetro de la altura de la fase  $i$ , para  $0 \leq i \leq k - 1$ .
  - `finalHeight`: arreglo de tamaño  $n$ ; usted debe retornar sus resultados ubicando el

numero final de ladrillos de la columna  $i$  en `finalHeight[i]`, para  $0 \leq i \leq n - 1$ .

## Subtaeas

Para todas las subtaeas el parametro de altura en todas las fases son enteros no negativos menores o iguales a **100,000**.

subtarea	puntos	$n$	$k$	notas
1	8	$1 \leq n \leq 10,000$	$1 \leq k \leq 5,000$	sin límites adicionales
2	24	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	todas las fases donde se agregan primero
3	29	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	sin límites adicionales
4	39	$1 \leq n \leq 2,000,000$	$1 \leq k \leq 500,000$	sin límites adicionales

## Detalles de implementacion

Usted tiene que enviar exactamente un archivo, llamado `wall.c`, `wall.cpp`, o `wall.pas`. Este archivo implementa el subprograma descrito antes usando los siguientes prototipos. Usted también necesitará incluir un archivo de cabecera `wall.h` para los programas escritos en C/C++.

### programas en C/C++

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

### programas en Pascal

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

## Ejemplo de calificador

El ejemplo de grader leera la entrada con el siguiente fomato:

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