



## Wall

Jian-Jia está construyendo un muro mediante el apilamiento de ladrillos del mismo tamaño. Este muro consiste de  $n$  columnas de ladrillos, las cuales están numeradas de 0 a  $n - 1$  de izquierda a derecha. Las columnas pueden tener diferentes alturas. La altura de una columna es el número de ladrillos que esta tiene.

Jian-Jia construye el muro como se explica a continuación. Inicialmente no hay ladrillos en ninguna columna. Luego, Jian-Jia realiza  $k$  fases en las cuales puede *agregar* o *quitar* ladrillos. El proceso de construcción se completa cuando terminan las  $k$  fases. En cada fase, Jian-Jia tiene un rango de columnas consecutivas y una altura  $h$ , y hace el siguiente procedimiento:

- En una fase en la que *agrega* 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 en la que *quita* 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.

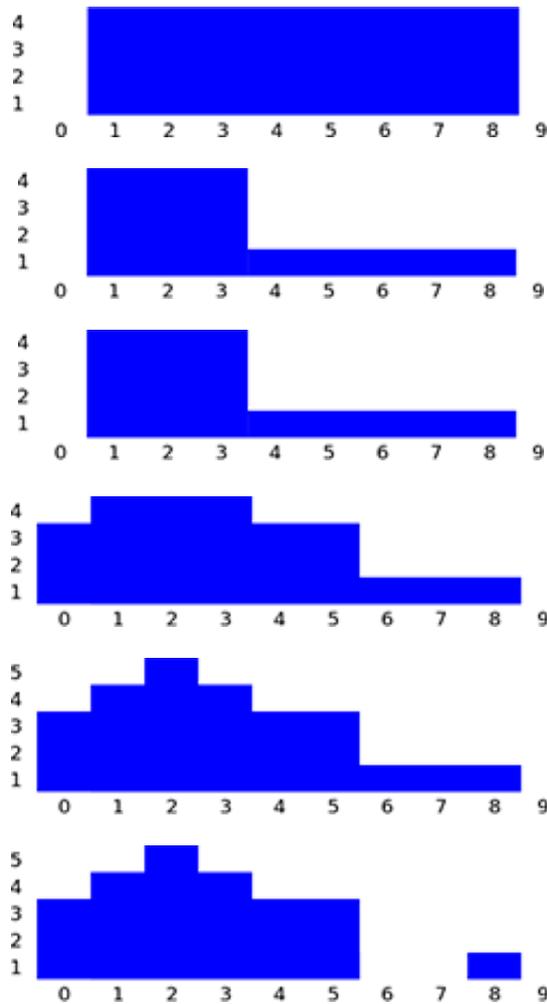
Su tarea es determinar la forma final del muro.

## Ejemplo

Asumiremos que hay 10 columnas de ladrillos y 6 fases de construcción. Todos los rangos en la siguiente tabla son inclusivos. A continuación se muestran diagramas del muro después de cada fase.

fase	tipo	rango	altura
0	agregar	columnas 1 to 8	4
1	quitar	columnas 4 to 9	1
2	quitar	columnas 3 to 6	5
3	agregar	columnas 0 to 5	3
4	agregar	columna 2	5
5	quitar	columnas 6 to 7	0

Como todas las columnas están inicialmente vacías, después de la fase 0 cada columna de la 1 a la 8 quedará con 4 ladrillos. Las columnas 0 y 9 permanecerán vacías. En la fase 1, se quitan ladrillos en las columnas de la 4 a la 8 hasta que cada uno de ellos quede con un ladrillo y la columna 9 permanece vacía. Las columnas de la 0 a la 3, las cuales están fuera del rango dado, permanecen sin cambio. La fase 2 no hace cambios porque las columnas de la 3 a la 6 no tienen más de 5 ladrillos. Después de la fase 3 el número de ladrillos en las columnas 0, 4 y 5 se incrementan a 3. Hay 5 ladrillos en la columna 2 después de la fase 4. En la fase 5 se remueven todos los ladrillos de las columnas 6 y 7.



## Tarea

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

- `buildWall(n, k, op, left, right, height, finalHeight)`
  - $n$ : Número de columnas en el muro.
  - $k$ : número de fases.
  - `op`: arreglo de tamaño  $k$ ; `op[i]` es el tipo de la fase  $i$ : 1 para una fase de agregar y 2 para una fase de quitar, para  $0 \leq i \leq k - 1$ .
  - `left` y `right`: arreglos de tamaño  $k$ ; el rango de columnas en la fase  $i$  empieza con la columna `left[i]` y termina con la columna `right[i]` (incluyendo ambos puntos extremos `left[i]` y `right[i]`), para  $0 \leq i \leq k - 1$ . siempre se tendrá que `left[i] ≤ right[i]`.
  - `height`: arreglo de tamaño  $k$ ; `height[i]` es el parametro de altura en la fase  $i$ , para  $0 \leq i \leq k - 1$ .
  - `finalHeight`: arreglo de tamaño  $n$ ; usted deberá retornar sus resultados poniendo el número final de ladrillos de la columna  $i$  en `finalHeight[i]`, para  $0 \leq i \leq n - 1$ .

## Subtareas

Para todas las subtareas el parámetro de altura en todas las fases serán 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 agrega preceden a todas las fases donde se quita
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 implementación

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 programas escritos en 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);
```

### Sample grader

El sample grader (programa para probar ejemplos propios) lee la entrada en el siguiente formato.

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