



Wall

Jian-Jia esta contruyendo un muro (wall) apilando ladrillos del mismo tamaño. Estos muros consisten de n columnas que estan numeradas de 0 a $n - 1$ de izquierda a derecha. Las columnas pueden tener diferentes alturas. La altura de una columna es el numero de ladrillos en ella.

Jian-Jia contruye el muro como sigue: Inicialmente no hay ladrillos en ninguna columna. Luego, Jian-Jia pasa por k fases de *adding* o *removing* ladrillos. El proceso de contruccion completa cuando todas las k fases estan terminadas. En cada fase, Jian-Jia da un rango de columnas consecutivas y una altura h , y realiza el siguiente procedimiento:

- En un *adding* phase(fase), Jian-Jia añade ladrillos a las columnas del rango dado que tienen menos de h ladrillos, asi ellas tendran exactamente h ladrillos terminada la fase. No se hace nada para columnas que tienen h o mas ladrillos.
- En un *removing* phase(fase), Jian-Jia elimina ladrillos a las columnas del rango dado que tienen mas de h ladrillos, asi ellas tendran exactamente h ladrillos terminada la fase. No se hace nada para las columnas que tienen h o menos ladrillos.

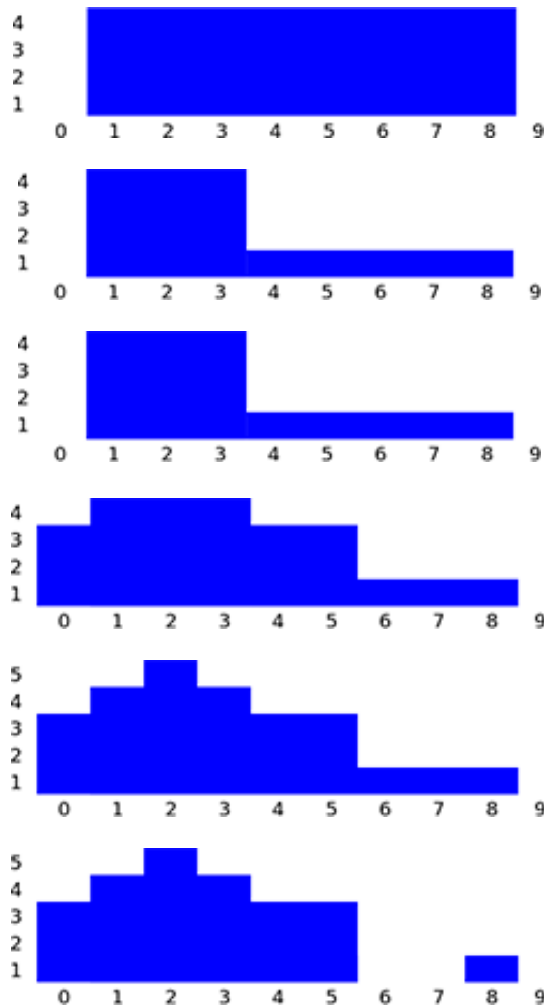
Tu tarea es determinar la forma final del muro.

Example

Asumiremos que hay 10 columnas de ladrillos y 6 fases de contruccion. Todos los rangos en la siguiente tabla son inclusivos. Los diagramas del muro despues de cada fase son mostrados abajo.

phase	type	rango	altura
0	add	columns 1 to 8	4
1	remove	columns 4 to 9	1
2	remove	columns 3 to 6	5
3	add	columns 0 to 5	3
4	add	column 2	5
5	remove	columns 6 to 7	0

Todas las columnas inicialmente estan vacias, despues de la fase 0 la columna 1 a la columna 8 tendran 4 ladrillos. La columna 0 y 9 se mantendran vacias. Despues de la fase 1, los ladrillos son eliminados desde la columna 4 a la 8 hasta que cada una tenga 1 ladrillo, y la columna 9 aun esta vacia. Las columnas de 0 a 3 las cuales estan fuera del rango no cambian. Despues de la fase 2 no se hara cambios puesto que ninguna tiene mas de 5 ladrillos. Despues de la fase 3 el numero de ladrillos en las columnas 0, 4 y 5 se incrementan en 3. Hay 5 ladrillos en la columna 2 despues de la fase 4. despues de la fase 5 se eliminan todos los ladrillos desde la columna 6 y 7.



Task

Dada la descripción de k fases, debes calcular el número de ladrillos en cada columna después de que todas las fases terminen. Necesitas implementar la función `buildWall`.

- `buildWall(n, k, op, left, right, height, finalHeight)`
 - n : Número de columnas en el muro.
 - k : Número de fases.
 - `op`: array de tamaño k ; `op[i]` es el tipo de fase i : 1 para adding phase and 2 para removing phase, para $0 \leq i \leq k - 1$.
 - `left` y `right`: arrays de longitud k ; el rango de columnas en i inicia con la columna `left[i]` y termina con la columna `right[i]` (inclusivo `left[i]` y `right[i]`), para $0 \leq i \leq k - 1$. Siempre tendrás `left[i] ≤ right[i]`.
 - `height`: array de tamaño k ; `height[i]` es el parametro de la altura de la fase i , para $0 \leq i \leq k - 1$.
 - `finalHeight`: array de tamaño n ; tu deberas retornar tus resultados finales del numero de ladrillos en cada columna i en `finalHeight[i]`, for $0 \leq i \leq n - 1$.

Subtasks

For all subtasks the height parameters of all phases are nonnegative integers less or equal to **100,000**.

subtask	points	n	k	note
1	8	$1 \leq n \leq 10,000$	$1 \leq k \leq 5,000$	no additional limits
2	24	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	all adding phases are before all removing phases
3	29	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	no additional limits
4	39	$1 \leq n \leq 2,000,000$	$1 \leq k \leq 500,000$	no additional limits

Implementation details

You have to submit exactly one file, called `wall.c`, `wall.cpp` or `wall.pas`. This file implements the subprogram described above using the following signatures. You also need to include a header file `wall.h` for C/C++ program.

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

The sample grader reads the input in the following format:

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