



Wall

Jian-Jia está construyendo un muro apilando 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 en ella.

Jian-Jia construye el muro como se explica a continuación. Inicialmente no hay ladrillos en ninguna columna. Entonces, 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 elige un rango de columnas consecutivas y una altura h , entonces 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.

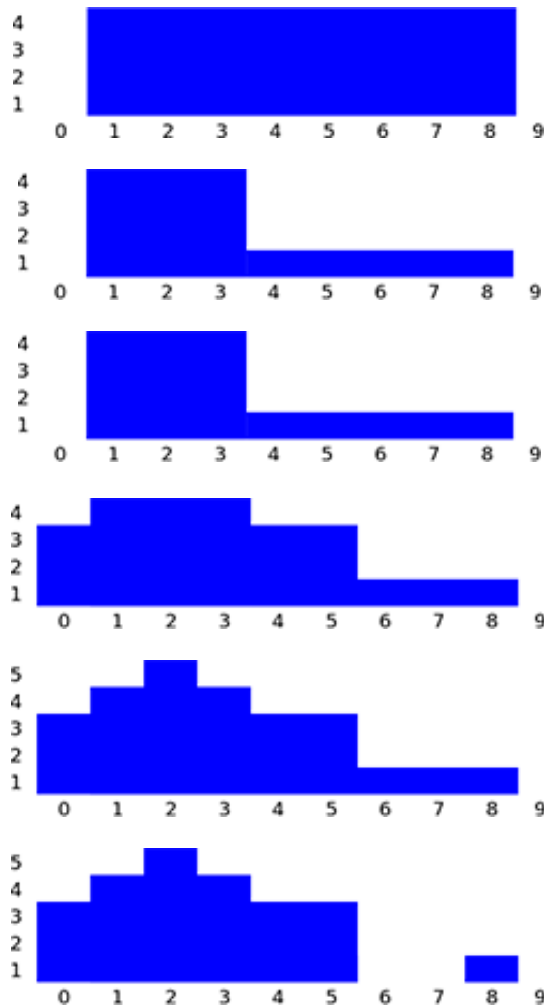
Su tarea es determinar la forma final del muro.

Ejemplo

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

fase	tipo	rango	altura
0	agregar	columnas 1 a la 8	4
1	quitar	columnas 4 a la 9	1
2	quitar	columnas 3 a la 6	5
3	agregar	columnas 0 a la 5	3
4	agregar	columna 2	5
5	quitar	columnas 6 a la 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. 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 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. Habrá 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.



Task

Dada la descripción de las k fases, debes calcular 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 finales `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 $0 \leq i \leq n - 1$.

Subtasks

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 agregan antes de las fases en 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

Implementation details

Debes enviar exactamente un archivo, llamado `wall.c`, `wall.cpp`, o `wall.pas`. Este archivo implementa el subprograma descrito anteriormente usando las siguientes firmas. Debes incluir `wall.h` para programas 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 *grader* de ejemplo debe leer el input en el siguiente formato.

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