



磚牆

建佳正在把相同大小的磚塊堆疊在一起，以建造一座磚牆。這座磚牆是 n 道由左至右依序標示為 0 到 $n-1$ 的牆面所構成。每一道牆面可能有不同的高度。一道牆面的高度定義為這道牆面所包含的總磚塊數。

建佳建造這座磚牆的方式如下：一開始，每道牆面都沒有任何磚塊。接著建佳進行 k 次新增 (adding) 或移除 (removing) 磚塊的動作。當這 k 次動作都完成之後，磚牆就建造完成。在每一次動作當中，建佳會被給定一段連續牆面的範圍以及一個高度參數 h ，接著他會執行以下程序：

- 在新增的動作中，如果在這段連續牆面的範圍裡，有任何牆面所包含的磚塊數不足 h ，那麼建佳會補足這些牆面的磚塊數，使這些牆面每一道都剛好有 h 個磚塊。如果牆面已經有 h 個或更多的磚塊，那麼建佳就不動這道牆面。
- 在移除的動作中，建佳會移除那些在給定的範圍裡，總磚塊數超過 h 的那些牆面所包含的磚塊，使得這些牆面每一道都剛好只剩下 h 個磚塊。如果牆面只有 h 個磚塊或更少，那麼他就不動這道牆面。

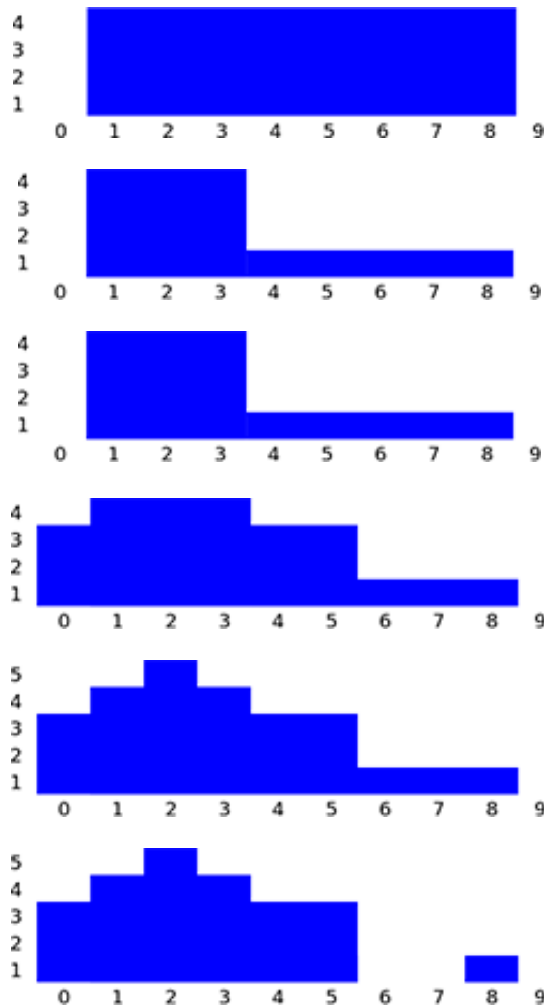
你的任務是確認這座磚牆最後的形狀。

範例

我們假設有 10 道牆面以及 6 次建造的動作。下表所有的範圍都包含本身的數字。這座磚牆經過每次動作後的形狀如下圖所示：

動作	型別	範圍	高度
0	新增	牆面 1 到 8	4
1	移除	牆面 4 到 9	1
2	移除	牆面 3 到 6	5
3	新增	牆面 0 到 5	3
4	新增	牆面 2	5
5	移除	牆面 6 到 7	0

由於所有牆面初始化都是空的，因此在動作 0 之後，從牆面 1 到牆面 8，每一道牆面都將有 4 個磚塊。牆面 0 與牆面 9 則仍然是空的。在動作 1 之後，牆面 4 到牆面 8 的磚塊都會被移除到每道牆面都只剩 1 個磚塊，而牆面 9 則仍然是空的。牆面 0 到牆面 3 由於不在範圍裡，因此也是維持不動。由於牆面 3 到牆面 6 的磚塊數都沒超過 5 個，因此動作 2 不會造成任何改變。在動作 3 之後，牆面 0、牆面 4，以及牆面 5 的磚塊數都增加到 3 個。在動作 4 之後，牆面 2 會有 5 個磚塊。動作 5 會將牆面 6 及牆面 7 的磚塊全部移除。



任務

給定 k 個動作的描述，請計算出在所有動作完成後，每一道牆面所包含的磚塊數。你需要實作函式 `buildWall`.

- `buildWall(n, k, op, left, right, height, finalHeight)`
 - n : 磚牆所包含的牆面個數。
 - k : 動作的個數。
 - `op`: 長度為 k 的陣列; `op[i]` 是動作 i 的型別: 1 表示是新增的動作, 2 表示是移除的動作, $0 \leq i \leq k - 1$ 。
 - `left` 與 `right`: 長度為 k 的陣列; 動作 i 的牆面範圍從牆面 `left[i]` 開始, 到牆面 `right[i]` 結束 (包含牆面 `left[i]` 及牆面 `right[i]`), $0 \leq i \leq k - 1$ 。
◦ `left[i] ≤ right[i]` 永遠成立。
 - `height`: 長度為 k 的陣列; `height[i]` 是動作 i 中, 高度參數的部分, $0 \leq i \leq k - 1$ 。
 - `finalHeight`: 長度為 n 的陣列; 你應該將你的執行結果存放在這個陣列後回傳, 牆面 i 最後的磚塊總數存放在 `finalHeight[i]`, $0 \leq i \leq n - 1$ 。

子任務

所有子任務的所有動作，其高度參數都是非負整數，並且小於或等於 100,000。

子任務	分數	n	k	附註
1	8	$1 \leq n \leq 10,000$	$1 \leq k \leq 5,000$	沒有額外限制
2	24	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	所有新增動作都在所有移除動作之前
3	29	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	沒有額外限制
4	39	$1 \leq n \leq 2,000,000$	$1 \leq k \leq 500,000$	沒有額外限制

實作細節

你只能送出一個檔案，命名為 `wall.c`、`wall.cpp`，或 `wall.pas`。這個檔案實作以上描述的子程式，並使用以下的函式原型。若你是使用 C/C++ 程式語言，你也需要引用標頭檔 `wall.h`。

C/C++ 程式

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

Pascal 程式

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

範例評分程式

範例評分程式讀進的輸入為以下格式：

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