壁 (Wall)

Jian-Jia は同じ大きさのレンガを積み上げて壁を作っている．壁は n 列からなり，列には左から右へ順に 0 から $n-1$ までの番号がつけられている．ある列の高さとはその列にあるレンガの個数であり，それぞれの列の高さはすべて同じである必要はない．

Jian-Jia は次のようにして壁を作る．はじめはどの列にもレンガはない．そこから Jian-Jia はレンガを**加える** (adding) か**取り除く** (removing) といった操作を k 回行う．k 回の操作がすべて終わると壁の建築は完了する．それぞれの操作において Jian-Jia はある連続した列の範囲と高さ h を指定され，以下のような手続きを行う．

- **加える**操作では，Jian-Jia は指定された列の範囲のうち，レンガの個数が h よりも小さいような列に対してレンガの個数がちょうど h となるようにレンガを加える．レンガの個数が h 以上であるような列に対しては何も行わない．
- **取り除く**操作では，Jian-Jia は指定された列の範囲のうち，レンガの個数が h よりも大きいような列に対してレンガの個数がちょうど h となるようにレンガを取り除く．レンガの個数が h 以下であるような列に対しては何も行わない．

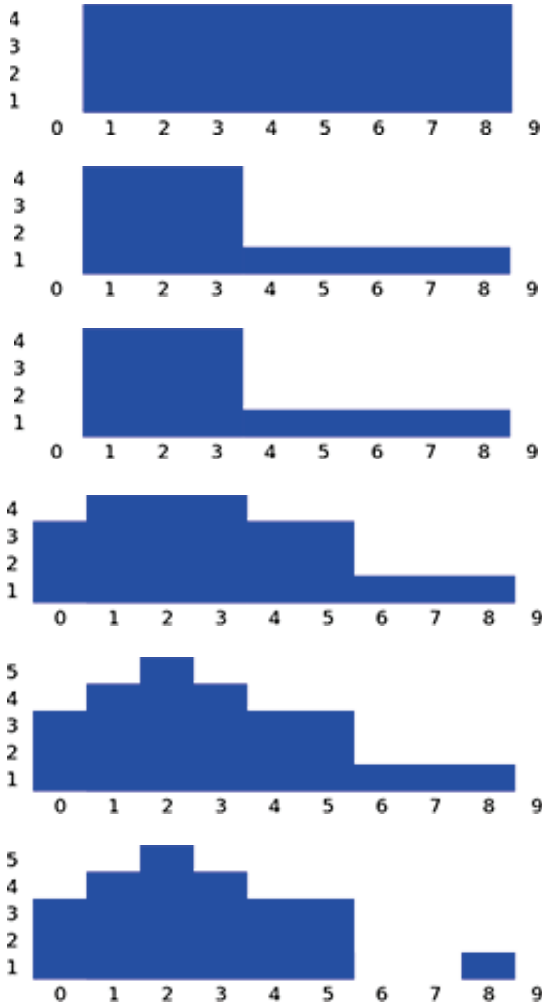
壁の最終的な形を決定するプログラムを作成せよ．

例 (Example)

壁は 10 個の列からなり，6 回の操作が行われるとする．以下の表に示される範囲はすべて両端を含む．図はそれぞれの操作の後の壁の様子を表している．

操作	種類	範囲	高さ
0	加える	列 1 から列 8	4
1	取り除く	列 4 から列 9	1
2	取り除く	列 3 から列 6	5
3	加える	列 0 から列 5	3
4	加える	列 2	5
5	取り除く	列 6 から列 7	0

はじめはどの列にもレンガはないため，操作 0 の後では列 1 から列 8 にはそれぞれ 4 個のレンガが積まれており，列 0 と列 9 にはレンガがないままである．操作 1 では，列 4 から列 8 にあるレンガがそれぞれの列にあるレンガが 1 個になるまで取り除かれ，列 9 はレンガがないままである．また，列 0 から列 3 は指定された範囲の外にあるためレンガの個数は変わらない．操作 2 では，列 3 から列 6 はどの列もレンガの個数が 5 以下であるため何も行われない．操作 3 が終わった後では列 0, 4, 5 のレンガが 3 個に増やされている．操作 4 が終わった後では列 2 に 5 個のレンガがある．操作 5 では列 6 と列 7 にあるレンガがすべて取り除かれる．



課題 (Task)

k 回の操作の情報が与えられたとき，すべての操作が終わった後に壁のそれぞれの列にレンガがいくつあるかを求めよ．関数 `buildWall` を実装せよ．

- `buildWall(n, k, op, left, right, height, finalHeight)`
  - `n`： 壁の列の個数
  - `k`： 操作の回数
  - `op`： サイズ k の配列であり，$0 \leqq i \leqq k-1$ に対し `op[i]` は操作 i の種類を表す．1 は加える操作を，2 は取り除く操作を表す．

- `left` および `right`： どちらもサイズ k の配列であり，$0 \leqq i \leqq k-1$ に対し，操作 i で指定される列の範囲が列 `left[i]` から列 `right[i]` であることを表す (両端の `left[i]` と `right[i]` も含む)．`left[i]` $\leqq$ `right[i]` が常に成り立つ．
- `height`： サイズ k の配列であり，$0 \leqq i \leqq k-1$ に対し `height[i]` は操作 i で指定される高さを表す．
- `finalHeight`： サイズ n の配列であり，あなたは，$0 \leqq i \leqq n-1$ に対し，最終的に列 i にあるレンガの個数を `finalHeight[i]` に代入しなければならない．

小課題 (Subtasks)

すべての小課題において操作の際に指定される高さは 100,000 以下の非負整数である．

小課題	得点	n	k	追加の制約
1	8	$1 \leqq n \leqq 10,000$	$1 \leqq k \leqq 5,000$	(追加の制約なし)
2	24	$1 \leqq n \leqq 100,000$	$1 \leqq k \leqq 500,000$	すべての加える操作はどの取り除く操作よりも前に行われる．
3	29	$1 \leqq n \leqq 100,000$	$1 \leqq k \leqq 500,000$	(追加の制約なし)
4	39	$1 \leqq n \leqq 2,000,000$	$1 \leqq k \leqq 500,000$	(追加の制約なし)

実装の詳細 (Implementation details)

1 つのファイルを提出せよ．提出するファイルの名前は `wall.c`, `wall.cpp`, `wall.pas` のいずれかである．このファイルには課題で指定されたサブプログラムを以下のシグネチャを用いて実装すること．C/C++ のプログラムにおいては，`wall.h` をインクルード (include) する必要がある．

C/C++ プログラム (C/C++ program)

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

Pascal プログラム (Pascal program)

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

採点プログラムのサンプル (Sample grader)

採点プログラムのサンプルは，以下のフォーマットの入力を読み込む：

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