



Wall

Jian-Jia construit un mur en superposant des briques de même taille. Ce mur est constitué de n colonnes de briques numérotées de 0 à $n - 1$ de la gauche vers la droite. Les colonnes peuvent avoir des hauteurs différentes. La hauteur d'une colonne est le nombre de briques qui la constituent.

Jian-Jia construit le mur comme suit : initialement, les colonnes ne contiennent aucune brique. Ensuite, Jian-Jia applique k phases d'*ajout* ou *retrait* de briques. La construction finale du mur aura lieu quand toutes les k phases sont réalisées. Dans chaque phase, Jian-Jia a un intervalle donné de colonnes et une hauteur h , et il exécute le procédé suivant :

- Pour la phase *ajout*, Jian-Jia ajoute des briques aux colonnes de l'intervalle donné qui ont moins de h briques, ainsi elles auront à la fin exactement h briques. Il ne touche pas les colonnes ayant h briques ou plus.
- Pour la phase *retrait*, Jian-Jia retire des briques des colonnes de l'intervalle donné qui ont plus que h briques, ainsi elles auront à la fin exactement h briques. Il ne touche pas les colonnes ayant h briques ou moins.

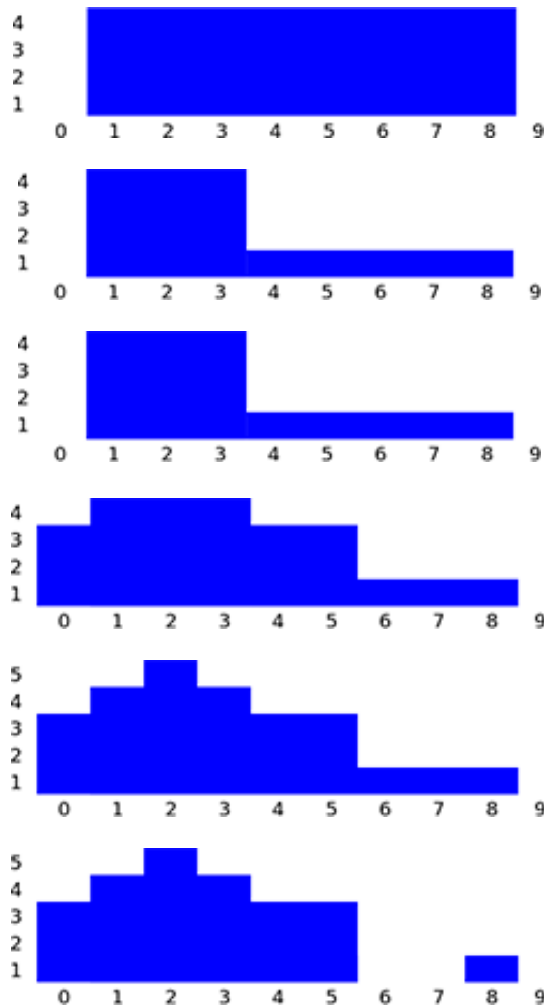
Votre tâche est de déterminer la forme finale du mur.

Exemple

On suppose qu'il y a 10 colonnes de briques et 6 phases de construction du mur. Tous les intervalles dans la table suivante sont inclusifs. La forme du mur, après chaque phase, est montrée ci-après.

phase	type	intervalle	hauteur
0	ajouter	colonnes 1 à 8	4
1	retirer	colonnes 4 à 9	1
2	retirer	colonnes 3 à 6	5
3	ajouter	colonnes 0 à 5	3
4	ajouter	colonne 2	5
5	retirer	colonnes 6 à 7	0

Comme toutes les colonnes sont initialement vides, après la phase 0, chacune des colonnes de 1 à 8 aura 4 briques. Les colonnes 0 et 9 restent vides. Lors de la phase 1, les briques sont retirées de la colonne 4 à la colonne 8 jusqu'à ce que chacune ait exactement 1 brique et la colonne 9 reste vide. Les colonnes 0 à 3, qui ne sont pas dans l'intervalle donné restent inchangées. La phase 2 n'opère aucun changement vu que les colonnes 3 à 6 n'ont pas plus que 5 briques. Après la phase 3, le nombre de briques dans les colonnes 0, 4 et 5 atteint 3. Il y a 5 briques dans la colonne 2 après la phase 4. Lors de la phase 5, toutes les briques des colonnes 6 et 7 sont retirées.



Tâche

Étant donné la description des k phases, calculer le nombre de briques dans chaque colonne après la réalisation de toutes les phases. Vous devez implémenter la fonction `buildWall` :

- `buildWall(n, k, op, left, right, height, finalHeight)`
 - n : le nombre de colonnes du mur.
 - k : le nombre de phases.
 - `op` : tableau de taille k ; `op[i]` est le type de la phase i : 1 pour une phase d'ajout et 2 pour une phase de retrait, pour $0 \leq i \leq k - 1$.
 - `left` et `right` : tableaux de taille k ; l'intervalle des colonnes de la phase i commence par la colonne `left[i]` et se termine par la colonne `right[i]` (incluant les deux extrémités de l'intervalle `left[i]` et `right[i]`), pour $0 \leq i \leq k - 1$. Vous aurez toujours `left[i] ≤ right[i]`.
 - `height` : tableau de taille k ; `height[i]` est la hauteur relative à la phase i , pour $0 \leq i \leq k - 1$.
 - `finalHeight` : tableau de taille n ; vous devez retourner les résultats en plaçant le nombre final de briques de la colonne i dans `finalHeight[i]`, pour $0 \leq i \leq n - 1$.

Sous-Tâches

Pour l'ensemble des sous-tâches les paramètres de la hauteur (height) de toutes les phases seront des entiers positifs ou nuls, inférieurs ou égaux à **100000**.

sous-tâche	points	n	k	note
1	8	$1 \leq n \leq 10000$	$1 \leq k \leq 5000$	Aucune limite additionnelle
2	24	$1 \leq n \leq 100000$	$1 \leq k \leq 500000$	Toutes les phases d'ajout sont avant les phases de retrait
3	29	$1 \leq n \leq 100000$	$1 \leq k \leq 500000$	Aucune limite additionnelle
4	39	$1 \leq n \leq 2000000$	$1 \leq k \leq 500000$	Aucune limite additionnelle

Détails d'implémentation

Vous devez soumettre un seul fichier nommé `wall.c`, `wall.cpp` ou `wall.pas`. Ce fichier implémente la fonction décrite précédemment en utilisant une des signatures suivantes. Vous devez inclure (`#include`) l'entête `wall.h` pour les programmes en C/C++

Programme C/C++

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

Program Pascal

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

Évaluateur

L'évaluateur lit l'entrée en suivant ce format :

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