



## Wall

Jian-Jia está construindo um muro empilhando blocos do mesmo tamanho. Este muro consiste em  $n$  colunas de tijolos, que estão numeradas de 0 a  $n - 1$  da esquerda para a direita. As colunas podem ter diferentes alturas. A altura de uma coluna é o número de tijolos que estão empilhados nela.

Jian-Jia constrói o muro da maneira seguinte. Inicialmente não existem tijolos em nenhuma coluna. Então, Jian-Jia passa por  $k$  fases *adicionando* ou *removendo* tijolos. O processo de construção completa-se quando todas as  $k$  fases terminam. Em cada fase Jian-Jia recebe um intervalo de colunas consecutivas de tijolos e uma altura  $h$ , e segue o seguinte procedimento:

- Numa fase de *adição*, Jian-Jia adiciona tijolos às colunas do intervalo selecionado que têm menos do que  $h$  tijolos, de maneira a que fiquem com exatamente  $h$  tijolos. Ele não faz nada quanto às colunas que têm  $h$  ou mais tijolos.
- Numa fase de *remoção*, Jian-Jia remove tijolos às colunas do intervalo selecionado que têm mais do que  $h$  tijolos, de maneira a que fiquem com exatamente  $h$  tijolos. Ele não faz nada quanto às colunas que têm  $h$  ou menos tijolos.

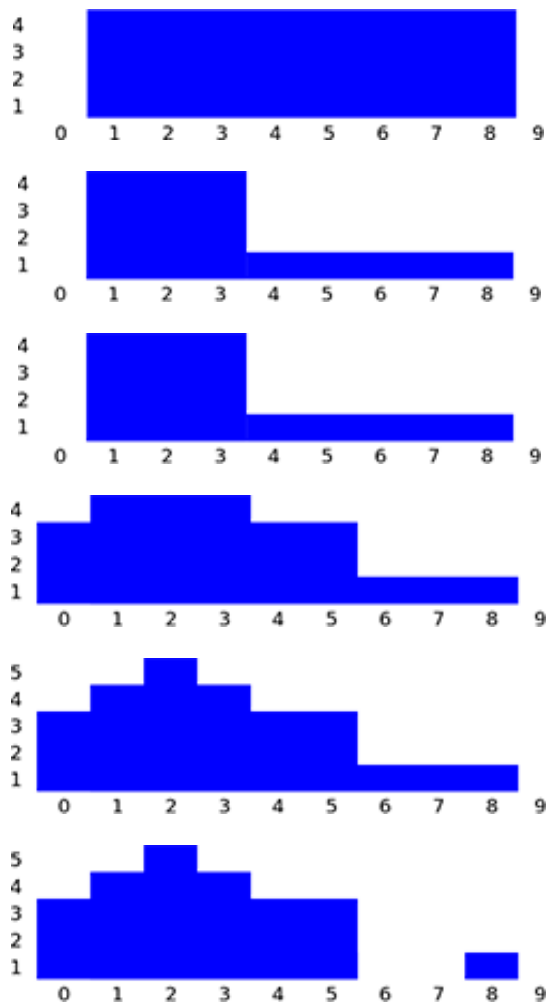
A sua tarefa é determinar o formato final do muro.

## Exemplo

Assumimos que existem 10 colunas de tijolos e 6 fases de construção do muro. Todos os intervalos da tabela seguinte são inclusivos. A seguir à tabela estão diagramas do muro depois de cada fase.

fase	tipo	intervalo	altura
0	adicionar	colunas 1 a 8	4
1	remover	colunas 4 a 9	1
2	remover	colunas 3 a 6	5
3	adicionar	colunas 0 a 5	3
4	adicionar	coluna 2	5
5	remover	colunas 6 a 7	0

Uma vez que todas as colunas estão inicialmente vazias, depois da fase 0 cada uma das colunas de 1 a 8 vai ter 4 tijolos. As colunas 0 e 9 permanecem vazias. Na fase 1, os tijolos são removidos das colunas 4 a 8 até que cada uma delas contenha apenas 1 tijolo, e a coluna 9 permanece vazia. As colunas de 0 a 3, que estão fora do intervalo dado, permanecem inalteradas. A fase 2 não introduz nenhuma mudança, uma vez que as colunas de 3 a 6 não têm mais do que 5 tijolos. Depois da fase 3, o número de tijolos nas colunas 0, 4 e 5 aumentam para 3. Existem 5 tijolos na coluna 2 depois da fase 4. A fase 5 remove todos os tijolos das colunas 6 e 7.



## Tarefa

Dada a descrição de  $k$  fases, por favor calcule o número de tijolos de cada coluna depois de todas as fases terem terminados. Você tem de implementar a função `buildWall`.

- `buildWall(n, k, op, left, right, height, finalHeight)`
  - $n$ : o número de colunas do muro.
  - $k$ : o número de fases.
  - $op$ : vetor de tamanho  $k$ ;  $op[i]$  é o tipo da fase  $i$ : 1 para uma fase de adição e 2 para uma fase de remoção, para  $0 \leq i \leq k - 1$ .
  - $left$  e  $right$ : vetores de tamanho  $k$ ; o intervalo de colunas na fase  $i$  começa com a coluna  $left[i]$  e termina com a coluna  $right[i]$  (incluindo ambos os pontos das extremidades  $left[i]$  e  $right[i]$ ), para  $0 \leq i \leq k - 1$ . Acontece sempre que  $left[i] \leq right[i]$ .
  - $height$ : vetor de tamanho  $k$ ;  $height[i]$  é o parâmetro de altura da fase  $i$ , para  $0 \leq i \leq k - 1$ .
  - $finalHeight$ : vetor de tamanho  $n$ ; você deve retornar os resultados colocando o número final de tijolos de cada coluna  $i$  na posição  $finalHeight[i]$ , para  $0 \leq i \leq n - 1$ .

## Subtarefas

Para todas as subtarefas os parâmetros de altura de todas as fases são inteiros não negativos menores ou iguais a **100,000**.

subtarefa	pontos	$n$	$k$	nota
1	8	$1 \leq n \leq 10,000$	$1 \leq k \leq 5,000$	sem limites adicionais
2	24	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	todas as fases de adição são antes de todas as fases de remoção
3	29	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	sem limites adicionais
4	39	$1 \leq n \leq 2,000,000$	$1 \leq k \leq 500,000$	sem limites adicionais

## Detalhes de implementação

Você tem de submeter exatamente um ficheiro, chamado `wall.c`, `wall.cpp` ou `wall.pas`. Este ficheiro implementa o subprograma descrito anteriormente usando as seguintes assinaturas. Você tem também de incluir um ficheiro de cabeçalho `wall.h` para os programas de C/C++.

### Programa em C/C++

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

### Programa em Pascal

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

### Avaliador exemplo

O avaliador exemplo lê a entrada no seguinte formato:

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