



الحائط

يقوم جيان جيا ببناء حائط عن طريق تكديس (البلوك) المتماثل في الحجم. هذا الحائط يتألف من n عموداً من البلوك، الأعمدة مرقمة من 0 وحتى $n-1$ من اليسار إلى اليمين، يمكن للأعمدة أن تكون مختلفة الارتفاعات حيث أن ارتفاع العمود هو عدد البلوكات الموجودة في هذا العمود.

يقوم جيان جيا ببناء الحائط على الشكل التالي: في البداية تكون جميع الأعمدة فارغة ولا تحوي أي بلوكه. بعد ذلك تمر عملية بناء الحائط بعدة مراحل من "الإضافة" أو "الإزالة" للبلوكات. تكتمل عملية البناء عندما تنتهي جميع المراحل. في كل مرحلة يعطى جيان جيا نطاقاً متصلاً من الأعمدة وارتفاعاً معيناً، ويجب عليه أن يقوم بالعملية التالية:

■ في مرحلة الإضافة، يضيف جيان البلوكات إلى الأعمدة الموجودة ضمن النطاق المعطى والتي يكون ارتفاعها أقل من h بلوكه، وهكذا تصبح هذه الأعمدة تحوي تماماً h بلوكه، لا يقوم جيان بأي شيء للأعمدة التي تحوي h أو أكثر من البلوكات أي ان عملية الإضافة ستجري فقط على الأعمدة التي بها عدد البلوكات أقل h .

■ في مرحلة الحذف، يزيل جيان البلوكات من الأعمدة الموجودة ضمن النطاق المعطى والتي تحوي أكثر من h بلوكه، وهكذا تصبح هذه الأعمدة تحوي h بلوكه، لا يقوم جيان بأي شيء للأعمدة التي تحوي h بلوكه أو أقل.

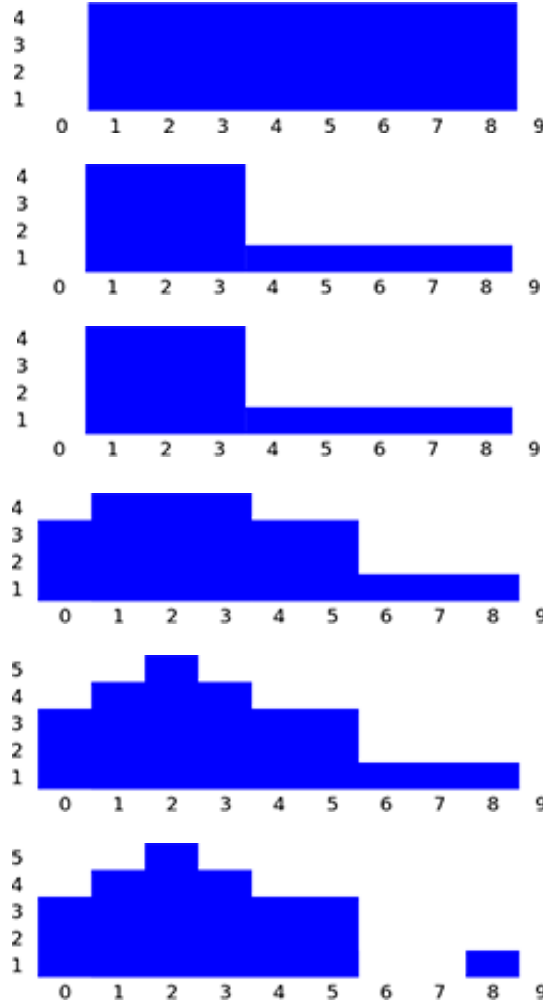
مهمتك هي تحديد الشكل النهائي للحائط بحساب العدد النهائي للبلوكات في كل عمود بعد اخر مرحلة.

مثال:

لنفرض أن هناك 10 أعمدة من البلوك وهناك 6 مراحل بناء. كل النطاقات في الجدول التالي موضحة في الجدول كما يظهر في الشكل في الأسفل شكل الحائط بعد كل مرحلة من المراحل. مع ملاحظة add تعني اضافة و remove تعني حذف. و columns تعني أعمدة.

المرحلة	النوع	النطاق	الارتفاع
0	add	columns 1 to 8	4
1	remove	columns 4 to 9	1
2	remove	columns 3 to 6	5
3	add	columns 0 to 5	3
4	add	column 2	5
5	remove	columns 6 to 7	0

بما أن كل الأعمدة تكون فارغة في البداية، بعد المرحلة رقم 0 كل الأعمدة من الرقم 1 وحتى الرقم 8 ستحوي 4 بلوكات، العمود رقم 0 ورقم 9 تبقى فارغة. في المرحلة 1، يتم إزالة البلوكات من العمود 4 وحتى العمود 8 إلى أن تصبح كل هذه الأعمدة تحوي بلوكه واحدة فقط، ويبقى العمود 9 فارغاً، كما أن الأعمدة من 0 وحتى 3 لا تتغير لأنها خارج النطاق. المرحلة رقم 2 لا تحدث أي تغيير لأن الأعمدة من 3 وحتى 6 لا تحوي أكثر من 5 بلوكات. بعد المرحلة رقم 3 يزداد عدد البلوكات في الأعمدة 0 و 4 و 5 ليصبح 3 بلوكات. يصبح عدد البلوكات في العمود رقم 2 بعد المرحلة 4 يحتوي 5 بلوكات. بعد المرحلة 5 يتم حذف كل البلوكات من العمودين 6 و 7.



المهمة Task

بعد إعطائك k مرحلة، قم من فضلك بحساب عدد البلوكات في كل عمود بعد انتهاء كل المراحل، يجب عليك أن تكتب الدالة التي تقوم بذلك على الشكل التالي:

`.buildWall`

`'buildWall(n, k, op, left, right, height, finalHeight)`

- n : عدد الأعمدة في الحائط.
- k : عدد المراحل.
- op : مصفوفة طولها k وهي توضح نوع العملية طرح أو إضافة فإذا كانت $op[i] = 1$ فإن العملية عملية إضافة وإذا كانت $op[i] = 2$ فإن العملية عملية حذف
- $left$ and $right$: مصفوفتان طولهما k ; نطاق الأعمدة في المرحلة i التي ستجري عليها العملية يبدأ من العمود $left[i]$ وينتهي بالعمود $right[i]$ (متضمناً نقطتي النهاية $left[i]$ و $right[i]$), من أجل $0 \leq i \leq k - 1$. سيكون دائماً $left[i] \leq right[i]$.
- $height$: مصفوفة طولها k ; $height[i]$ هو معامل الارتفاع في المرحلة i , من أجل $0 \leq i \leq k - 1$.
- $finalHeight$: مصفوفة طولها n ; يجب عليك إعادة النتيجة عن طريق وضع العدد النهائي من الأجزاء ضمن العمود i في $finalHeight[i]$, من أجل $0 \leq i \leq n - 1$.

المهام الجزئية subtasks

في جميع الحالات سيكون h الارتفاع لكل المراحل عدد صحيحاً غير سالب أقل أو يساوي 100,000.

المهمة الجزئية	النقاط	n	k	ملاحظات
1	8	$1 \leq n \leq 10,000$	$1 \leq k \leq 5,000$	لا يوجد شروط إضافية
2	24	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	كل مراحل الإضافة ستكون قبل مراحل الحذف
3	29	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	لا يوجد شروط إضافية
4	39	$1 \leq n \leq 2,000,000$	$1 \leq k \leq 500,000$	لا يوجد شروط إضافية

تفاصيل التنفيذ

يجب عليك إرسال ملف واحد فقط اسمه wall.cpp لمنظومة التصحيح يجب عليك أن تقوم بتضمين المكتبة wall.h داخل البرنامج

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);
```

شكل الدخل

تقوم المنظومة بقراءة الدخل على الشكل التالي:

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