



Τοίχος

Ο Jian-Jia κτίζει ένα τοίχο στοιβάζοντας τούβλα ίδιου μεγέθους. Ο τοίχος αυτός αποτελείται από n στήλες από τούβλα, αριθμημένες από 0 μέχρι $n - 1$ από τα αριστερά προς τα δεξιά. Οι στήλες μπορεί να έχουν διαφορετικά ύψη. Το ύψος μιας στήλης είναι το πλήθος των τούβλων που περιέχει.

Ο Jian-Jia κτίζει τον τοίχο ως ακολούθως. Αρχικά δεν υπάρχουν τούβλα σε καμία στήλη. Στη συνέχεια, ο Jian-Jia ακολουθεί k φάσεις από προσθήκες ή διαγραφές τούβλων. Η διαδικασία κτισίματος ολοκληρώνεται όταν έχουν τελειώσει όλες οι k φάσεις. Σε κάθε φάση δίδεται στον Jian-Jia ένα διάστημα διαδοχικών στηλών τούβλων και ένα ύψος h , και ακολουθεί την παρακάτω διαδικασία:

- Σε μία φάση *προσθήκης*, ο Jian-Jia προσθέτει τούβλα σε εκείνες τις στήλες του δοθέντος διαστήματος που έχουν λιγότερα από h τούβλα, έτσι ώστε να έχουν ακριβώς h τούβλα. Δεν πειράζει τις στήλες που έχουν h ή περισσότερα τούβλα.
- Σε μια φάση *διαγραφής*, ο Jian-Jia αφαιρεί τούβλα από τις στήλες του δοθέντος διαστήματος που έχουν περισσότερα από h τούβλα, έτσι ώστε να έχουν ακριβώς h τούβλα. Δεν πειράζει τις στήλες που έχουν h ή λιγότερα τούβλα.

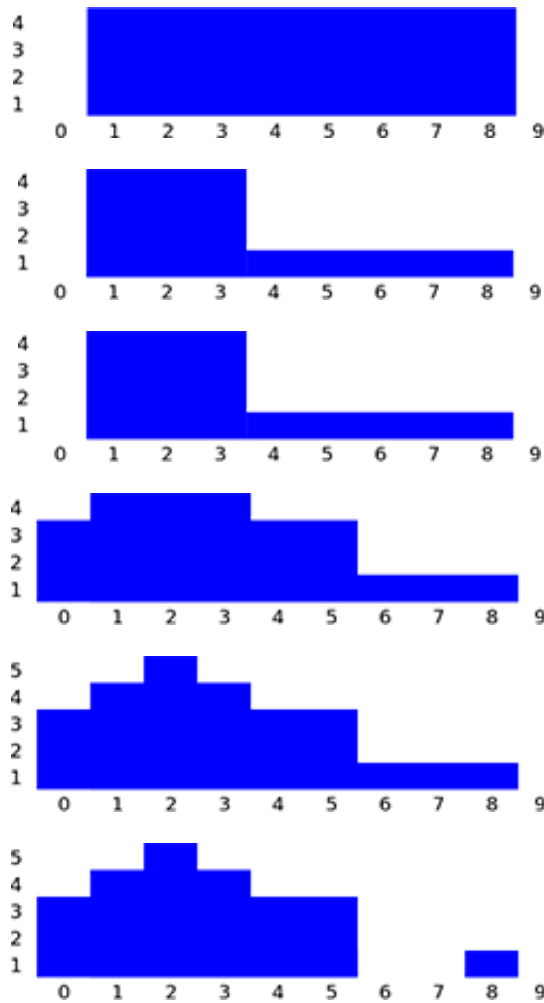
Το πρόβλημά σας είναι να βρείτε το τελικό σχήμα του τοίχου.

Παράδειγμα

Υποθέτουμε ότι υπάρχουν 10 στήλες από τούβλα και 6 φάσεις κτισίματος του τοίχου. Όλα τα διαστήματα στον παρακάτω πίνακα περιλαμβάνουν τα άκρα. Παρακάτω εμφανίζεται το σχήμα του τοίχου μετά από κάθε φάση.

φάση	τύπος	διάστημα	ύψος
0	προσθήκη	στήλες 1 έως 8	4
1	διαγραφή	στήλες 4 έως 9	1
2	διαγραφή	στήλες 3 έως 6	5
3	προσθήκη	στήλες 0 έως 5	3
4	προσθήκη	στήλη 2	5
5	διαγραφή	στήλες 6 έως 7	0

Εφόσον όλες οι στήλες είναι αρχικά κενές, μετά την φάση 0 κάθε μία από τις στήλες 1 έως 8 θα έχει 4 τούβλα. Οι στήλες 0 και 9 παραμένουν κενές. Στη φάση 1, διαγράφονται τούβλα από τις στήλες 4 έως 8 μέχρι η κάθε μία να έχει ένα τούβλο, και η στήλη 9 παραμένει κενή. Οι στήλες 0 έως 3, που είναι εκτός του δοθέντος διαστήματος, παραμένουν αναλλοίωτες. Η φάση 2 δεν επιφέρει καμία αλλαγή εφόσον οι στήλες 3 έως 6 δεν έχουν περισσότερα από 5 τούβλα. Μετά τη φάση 3 το πλήθος των τούβλων στις στήλες 0, 4, και 5 αυξάνει σε 3. Μετά τη φάση 4, υπάρχουν 5 τούβλα στη στήλη 2. Η φάση 5 διαγράφει όλα τα τούβλα από τις στήλες 6 και 7.



Πρόβλημα

Δοθείσης της περιγραφής των k φάσεων, υπολογίστε το πλήθος των τούβλων σε κάθε στήλη μετά το πέρας όλων των φάσεων. Πρέπει να υλοποιήσετε την συνάρτηση `buildWall`.

- `buildWall(n, k, op, left, right, height, finalHeight)`
 - n : το πλήθος των στηλών του τοίχου.
 - k : το πλήθος των φάσεων.
 - op : πίνακας μεγέθους k , όπου $op[i]$ είναι ο τύπος της φάσης i : 1 για φάση προσθήκης και 2 για φάση διαγραφής, για $0 \leq i \leq k - 1$.
 - $left$ και $right$: πίνακες μεγέθους k , όπου το διάστημα των στηλών της φάσης i αρχίζει από τη στήλη $left[i]$ και τελειώνει με τη στήλη $right[i]$ (συμπεριλαμβανομένων των άκρων $left[i]$ και $right[i]$), για $0 \leq i \leq k - 1$. Θα ισχύει πάντα $left[i] \leq right[i]$.
 - $height$: πίνακας μεγέθους k , όπου $height[i]$ είναι η παράμετρος του ύψους της φάσης i , για $0 \leq i \leq k - 1$.
 - $finalHeight$: πίνακας μεγέθους n , όπου θα πρέπει να επιστρέψετε τα αποτελέσματά σας, θέτοντας το τελικό πλήθος των τούβλων της στήλης i στο $finalHeight[i]$, για $0 \leq i \leq n - 1$.

Υποπροβλήματα

Για όλα τα υποπροβλήματα οι παράμετροι ύψους όλων των φάσεων είναι μη αρνητικοί ακέραιοι, μικρότεροι ή ίσοι του **100,000**.

υποπρόβλημα	βαθμοί	n	k	σημείωση
1	8	$1 \leq n \leq 10,000$	$1 \leq k \leq 5,000$	κανένας επιπλέον περιορισμός
2	24	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	όλες οι φάσεις προσθήκης προηγούνται όλων των φάσεων διαγραφής
3	29	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	κανένας επιπλέον περιορισμός
4	39	$1 \leq n \leq 2,000,000$	$1 \leq k \leq 500,000$	κανένας επιπλέον περιορισμός

Λεπτομέρειες υλοποίησης

Πρέπει να υποβάλετε ακριβώς ένα αρχείο, με όνομα `wall.c`, `wall.cpp` ή `wall.pas`. Αυτό το αρχείο πρέπει να υλοποιεί το υποπρόγραμμα που περιγράφεται παραπάνω, το οποίο πρέπει να έχει μία από τις παρακάτω επικεφαλίδες. Για τις υλοποιήσεις σε C/C++, το αρχείο σας πρέπει να κάνει `include` το αρχείο επικεφαλίδας `wall.h`.

Πρόγραμμα C/C++

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

Πρόγραμμα Pascal

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

Ενδεικτικός βαθμολογητής

Ο ενδεικτικός βαθμολογητής διαβάζει την είσοδο με την παρακάτω μορφή:

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