

Wild operations (wild)

Ο Φίλιππος θέλει να δοκιμάσει την ικανότητα του Φραγκίσκου να χειρίζεται περίεργες λειτουργίες σε πίνακες, γι' αυτό του έδωσε έναν πίνακα A_0, \dots, A_{N-1} με μήκος N .

Τώρα ο Φίλιππος θα ζητήσει από τον Φραγκίσκο να εκτελέσει κάποιες λειτουργίες στον πίνακα, όπου κάθε λειτουργία μπορεί να είναι είτε:

- αλλαγή της τιμής του A_p σε x , για κάποιον ακέραιο x και έγκυρο δείκτη p .
- διατάραξη του εύρους $[l, r]$, δηλαδή, να θέσει $A_p = \max(A_p, A_{p-1})$ **ταυτόχρονα** για όλα τα $l < p \leq r$.

Ανά πάσα στιγμή, ο Φίλιππος μπορεί να ζητήσει από τον Φραγκίσκο να του πει την τιμή του A_p για κάποιον έγκυρο δείκτη p .

Ο Φραγκίσκος είναι πολύ απασχολημένος, γι' αυτό αποφάσισε να ζητήσει τη βοήθειά σου για να απαντήσει στις ερωτήσεις του Φίλιππου.

Υλοποίηση

Πρέπει να υποβάλεις ένα μόνο αρχείο με επέκταση `.cpp`.



Μεταξύ των συνημμένων για αυτό το πρόβλημα θα βρεις ένα πρότυπο αρχείο `wild.cpp` με ένα παράδειγμα υλοποίησης.

Πρέπει να υλοποιήσεις τις ακόλουθες συναρτήσεις:

C++

```
void init(int N, vector<int> A);
```

- Αυτή η συνάρτηση καλείται μία φορά, στην αρχή της εκτέλεσης του προγράμματός σου.
- Ο ακέραιος N είναι το μήκος του πίνακα.
- Ο πίνακας A , με δείκτες από 0 μέχρι $N - 1$, είναι ο αρχικός πίνακας που επέλεξε ο Φίλιππος.

C++

```
void change(int p, int x);
```

- Αυτή η συνάρτηση καλείται πολλές φορές κατά την εκτέλεση του προγράμματός σου, όταν ο Φίλιππος εκτελεί μια αλλαγή.
- Ο ακέραιος p είναι ο δείκτης της τιμής που αλλάζει στον πίνακα.
- Ο ακέραιος x είναι η νέα τιμή που θα ανατεθεί.

C++

```
void perturb(int l, int r);
```

- Αυτή η συνάρτηση καλείται πολλές φορές κατά την εκτέλεση του προγράμματός σου, όταν ο Φίλιππος διαταράσσει ένα εύρος.
- Ο ακέραιος l είναι το αριστερό άκρο του εύρους που διαταράσσεται από τον Φίλιππο.
- Ο ακέραιος r είναι το δεξί άκρο του εύρους που διαταράσσεται από τον Φίλιππο.

C++

```
int calc(int p);
```

- Αυτή η συνάρτηση καλείται πολλές φορές κατά την εκτέλεση του προγράμματός σου, όταν ο Φίλιππος ρωτά την τιμή ενός στοιχείου του πίνακα.
- Ο ακέραιος p είναι ο δείκτης του στοιχείου που ρώτησε ο Φίλιππος.
- Η συνάρτηση πρέπει να επιστρέψει την τιμή του A_p αφού εφαρμοστούν όλες οι προηγούμενες λειτουργίες.

Grader-Δείγμα

Μια απλοποιημένη έκδοση του grader που χρησιμοποιείται κατά τη διόρθωση είναι διαθέσιμη στον φάκελο που σχετίζεται με αυτό το πρόβλημα. Μπορείς να τη χρησιμοποιήσεις για να δοκιμάσεις τις λύσεις σου τοπικά. Ο grader-δείγμα διαβάζει δεδομένα εισόδου από το `stdin`, καλεί τις συναρτήσεις που πρέπει να υλοποιήσεις, και γράφει στο `stdout` στην ακόλουθη μορφή.

Εστω Q ο συνολικός αριθμός αλλαγών, διαταράξεων και ερωτήσεων που κάνει ο Φίλιππος.

Τότε, το αρχείο εισόδου αποτελείται από $2 + Q$ γραμμές, που περιέχουν:

- Γραμμή 1: οι ακέραιοι N, Q .
- Γραμμή 2: N ακέραιοι A_0, \dots, A_{N-1} , οι αρχικές τιμές του πίνακα.
- Γραμμή $3 + i$ ($0 \leq i < Q$): 2 ή 3 ακέραιοι, σε μία από τις ακόλουθες μορφές:
 - $1 \ p \ x$: σημαίνει ότι ο Filippo αλλάζει το A_p σε x .
 - $2 \ l \ r$: σημαίνει ότι ο Filippo διαταράσσει το εύρος $[l, r]$.
 - $3 \ p$: σημαίνει ότι ο Filippo ρωτά την τιμή του A_p .

Το αρχείο εξόδου αποτελείται από Q_3 γραμμές (όπου Q_3 είναι ο αριθμός των κλήσεων στην `calc`) που περιέχουν τις τιμές που επιστρέφει η συνάρτηση `calc`.

Περιορισμοί

- $1 \leq N \leq 400\,000$.
- $0 \leq Q \leq 400\,000$.
- $1 \leq A_i \leq 10^9$ για όλα τα $0 \leq i < N$.
- $0 \leq p < N$ σε κάθε κλήση της `change` και της `calc`.
- $0 \leq l < r \leq N - 1$ σε κάθε κλήση της `perturb`.
- $1 \leq x \leq 10^9$ σε κάθε κλήση της `change`.

Βαθμολογία

Το πρόγραμμά σου θα δοκιμαστεί σε διάφορα σύνολα ελέγχου (test cases) ομαδοποιημένα σε υποπρόβληματα (subtasks). Για να πάρεις τους βαθμούς για ένα υποπρόβλημα, πρέπει να λύσεις σωστά όλα τα σύνολα ελέγχου που περιέχει.

Εστω Q_1 ο αριθμός των κλήσεων στη συνάρτηση `change` σε ένα σύνολο ελέγχου, τότε:

- **Υποπρόβλημα 0 [0 πόντοι]**: Παράδειγμα.
- **Υποπρόβλημα 1 [15 πόντοι]**: Η συνάρτηση `change` δεν καλείται ποτέ. $l = 0, r = N - 1$ σε κάθε κλήση της `perturb`.
- **Υποπρόβλημα 2 [16 πόντοι]**: $A_i \leq 10$ για όλα τα $0 \leq i < N$ και $x \leq 10$ για όλες τις κλήσεις της `change`.
- **Υποπρόβλημα 3 [13 πόντοι]**: Οι κλήσεις στη συνάρτηση `change` δεν μειώνουν τις τιμές ($x \geq A_p$), $Q_1 \leq 1000$ και $l = 0, r = N - 1$ σε κάθε κλήση της `perturb`.
- **Υποπρόβλημα 4 [22 πόντοι]**: Η συνάρτηση `change` δεν καλείται ποτέ.
- **Υποπρόβλημα 5 [14 πόντοι]**: Οι κλήσεις στη συνάρτηση `change` δεν μειώνουν τις τιμές ($x \geq A_p$), $Q_1 \leq 1000$.
- **Υποπρόβλημα 6 [20 πόντοι]**: Χωρίς επιπλέον περιορισμούς.

Παραδείγματα εισόδου/εξόδου

stdin	stdout
10 28	1
5 1 7 8 3 2 5 6 9 4	3
1 1 1	1
1 0 1	7
2 0 1	8
2 2 6	1
1 6 5	8
2 2 9	3
2 2 5	6
2 4 5	4
1 4 5	9
2 3 8	
1 8 4	
3 0	
1 6 3	
1 4 1	
2 5 7	
1 0 3	
2 4 5	
1 6 3	
3 0	
3 1	
3 2	
3 3	
3 4	
3 5	
3 6	
3 7	
3 8	
3 9	

Εξήγηση

Ξεκινάμε με τον πίνακα $A = [5, 1, 7, 8, 3, 2, 5, 6, 9, 4]$.

- Γεγονός 1: Ο Φίλιππος αλλάζει το A_1 σε 1 (ήταν ήδη 1): ο νέος πίνακας είναι $[5, 1, 7, 8, 3, 2, 5, 6, 9, 4]$.
- Γεγονός 2: Ο Φίλιππος αλλάζει το A_0 σε 1: ο νέος πίνακας είναι $[1, 1, 7, 8, 3, 2, 5, 6, 9, 4]$.
- Γεγονός 3: Ο Φίλιππος διαταράσσει το $[0, 1]$: ο νέος πίνακας είναι $[1, 1, 7, 8, 3, 2, 5, 6, 9, 4]$.
- Γεγονός 4: Ο Φίλιππος διαταράσσει το $[2, 6]$: ο νέος πίνακας είναι $[1, 1, 7, 8, 8, 3, 5, 6, 9, 4]$.

Από το γεγονός 19 και μετά, ο Φίλιππος ρωτά μόνο για τιμές στον πίνακα χωρίς να κάνει αλλαγές ή διαταράξεις. Σε αυτό το σημείο ο πίνακας είναι $[3, 1, 7, 8, 1, 8, 3, 6, 4, 9]$.