

AlgoNTUA Day 1: Το κρυμμένο Αμπερόμετρο

Time limit: 1.0s Memory limit: 64M

Το κρυμμένο Αμπερόμετρο

Προκειμένου να αυξήσει το ηθικό των φοιτητών, η σχολή ΗΜΜΥ του ΕΜΠ αποφάσισε να δημιουργήσει ένα παιχνίδι κρυμμένου θησαυρού. Μέσα σε ένα εργαστήριο έκρυψαν το μόνο λειτουργικό αμπερόμετρο της σχολής, ανάμεσα σε πολλά μη λειτουργικά αμπερόμετρα. Προκειμένου να μην κλέψει κανείς αυτό το πολύτιμο αμπερόμετρο, το έχουν ασφαλίσει σε συγκεκριμένη θέση.

Οι κανόνες του κρυμμένου θησαυρού είναι οι ακόλουθοι: Οι φοιτητές δοκιμάζουν την τύχη τους σειριακά. Με βάση τον χρόνο που χρειάστηκε κάθε φοιτητής, οι πίνακες τερματισμού δίνουν τη θέση του στη γενική κατάταξη. Ο πρώτος φοιτητής για παράδειγμα θα έχει αναγκαστικά θέση 1. Ο δεύτερος 1 ή 2 και ο N -οστός οποιαδήποτε θέση από 1 έως N .

Πρόβλημα:

Να αναπτύξετε ένα πρόγραμμα το οποίο αφού διαβάσει τη θέση που παίρνει ο κάθε φοιτητής στη γενική κατάταξη μέχρι εκείνη τη στιγμή, θα υπολογίζει την τελική θέση κάθε φοιτητή μετά το τέλος του αγώνα.

Είσοδος:

Η δομή της εισόδου είναι η ακόλουθη: Η πρώτη γραμμή έχει έναν ακέραιο αριθμό N , το πλήθος των φοιτητών που λαμβάνουν μέρος στον αγώνα. Η δεύτερη γραμμή περιέχει N ακέραιους αριθμούς χωρισμένους ανά δύο με ένα κενό διάστημα: τις θέσεις στη γενική κατάταξη που έχει μέχρι εκείνη τη στιγμή ο αντίστοιχος φοιτητής.

Έξοδος:

Η έξοδος θα πρέπει να έχει την ακόλουθη δομή: Έχουν μία γραμμή που περιέχει N ακέραιους αριθμούς, όσες και οι φοιτητές, χωρισμένους ανά δύο με ένα κενό διάστημα. Κάθε αριθμός πρέπει να είναι ο αύξων αριθμός του φοιτητή που θα καταλάβει την αντίστοιχη θέση στην τελική κατάταξη.

Παραδείγματα Εισόδου - Εξόδου:

1ο

STDIN

```
10
1 2 3 4 5 6 7 8 9 10
```

STDOUT

```
1 2 3 4 5 6 7 8 9 10
```

2ο

STDIN

```
10
1 1 1 1 1 1 1 1 1 1
```

STDOUT

```
10 9 8 7 6 5 4 3 2 1
```

3ο

STDIN

```
7
1 1 3 2 3 1 5
```

STDOUT

```
6 2 4 5 7 1 3
```

Εξήγηση: Στο 1ο παράδειγμα, κάθε φοιτητής που τερματίζει κάνει χειρότερο χρόνο από τους προηγούμενους και κατατάσσεται τελευταίος, ενώ στο 2ο κάθε φοιτητής κάνει καλύτερο χρόνο από τους προηγούμενους και κατατάσσεται πρώτος.

Περιορισμοί:

- Για περιπτώσεις ελέγχου συνολικής αξίας 50%, θα είναι:
 $1 \leq N \leq 10.000$
- Για περιπτώσεις ελέγχου συνολικής αξίας 100%, θα είναι:
 $1 \leq N \leq 500.000$

Προσοχή! Φροντίστε να διαβάζετε την είσοδο και να εκτυπώνετε την έξοδο αποδοτικά, ειδικά αν προγραμματίζετε σε C++ ή Java.

- **Μορφοποίηση:** Στην είσοδο αλλά και στην έξοδο, όλες οι γραμμές τερματίζουν με ένα χαρακτήρα `newline`.
- **Μέγιστος χρόνος εκτέλεσης:** 1 sec.
- **Μέγιστη διαθέσιμη μνήμη:** 64 MB.