Γόνδολα

Οι γόνδολες του Mao-Kong είναι ένα από τα αξιοθέατα της Taipei. Το σύστημα με τις γόνδολες αποτελείται από μία κυκλική διαδρομή, ένα μοναδικό σταθμό, και n γόνδολες, αριθμημένες από 1 έως και n, που κινούνται κατά μήκος της κυκλικής διαδρομής, όλες κατά την ίδια κατεύθυνση. Αρχικά, αφού η γόνδολα i περάσει από το σταθμό, η επόμενη γόνδολα που θα περάσει από το σταθμό θα είναι η γόνδολα $i+1$ αν $i<n$, ή η γόνδολα 1 αν $i=n$.

Οι γόνδολες μερικές φορές χαλάνε. Ευτυχώς υπάρχει ένα άπειρο πλήθος από περισσευούμενες γόνδολες, που είναι αριθμημένες από $n+1, n+2$, κ.λπ. Όταν μία γόνδολα χαλάσει, την αντικαθιστούμε (στην ίδια θέση πάνω στην κυκλική διαδρομή) με την πρώτη διαθέσιμη περισσευούμενη γόνδολα, δηλαδή, αυτή με το μικρότερο αριθμό. Για παράδειγμα, αν υπάρχουν πέντε γόνδολες και η γόνδολα 1 χαλάσει, την αντικαθιστούμε με τη γόνδολα 6.

Σας αρέσει να στέκεστε στο σταθμό και να βλέπετε τις γόνδολες να περνούν. Μία *ακολουθία γονδολών* είναι μία ακολουθία αποτελούμενη από n αριθμούς των γονδολών που περνούν από το σταθμό. Είναι πιθανό μία ή περισσότερες γόνδολες να έχουν χαλάσει (και αντικατασταθεί) πριν εσείς φτάσετε στο σταθμό, καμία όμως γόνδολα δε χαλάει την ώρα που εσείς παρακολουθείτε.

Προσέξτε ότι η ίδια διάταξη των γονδολών στην κυκλική διαδρομή μπορεί να δώσει πολλές διαφορετικές ακολουθίες γονδολών, ανάλογα ποια γόνδολα θα περάσει πρώτη από το σταθμό την ώρα που εσείς θα φτάσετε. Για παράδειγμα, αν καμία από τις γόνδολες δεν έχει χαλάσει, τότε οι (2, 3, 4, 5, 1) και (4, 5, 1, 2, 3) είναι δυνατές ακολουθίες γονδολών, αλλά η (4, 3, 2, 5, 1) δεν είναι (γιατί οι γόνδολες εμφανίζονται με λανθασμένη σειρά).

Αν η γόνδολα 1 χαλάσει, τότε είναι πιθανό να παρατηρήσετε την ακολουθία γονδολών (4, 5, 6, 2, 3). Αν η γόνδολα 4 χαλάσει, θα αντικατασταθεί με τη γόνδολα 7 και είναι πιθανό να παρατηρήσετε την ακολουθία γονδολών (6, 2, 3, 7, 5). Αν τώρα η γόνδολα 7 χαλάσει, την αντικαθιστούμε με τη γόνδολα 8 και είναι τώρα πιθανό να παρατηρήσετε την ακολουθία γονδολών (3, 8, 5, 6, 2).

χαλασμένη γόνδολα	νέα γόνδολα	πιθανή ακολουθία γονδολών
1	6	(4, 5, 6, 2, 3)
4	7	(6, 2, 3, 7, 5)
7	8	(3, 8, 5, 6, 2)

Μία *ακολουθία αντικαταστάσεων* είναι μία ακολουθία που αποτελείται από τους αριθμούς των γονδολών που χαλάνε, με τη σειρά που αυτό συμβαίνει. Στο προηγούμενο παράδειγμα, η ακολουθία αντικαταστάσεων είναι (1, 4, 7). Μία ακολουθία αντικαταστάσεων r *παράγει* μία ακολουθία γονδολών g αν, αφού οι γόνδολες χαλάσουν με τη σειρά που καθορίζεται από την ακολουθία r, η ακολουθία γονδολών g μπορεί να παρατηρηθεί.

Έλεγχος ακολουθιών γονδολών

Στα τρία πρώτα υποπροβλήματα πρέπει να ελέγχετε αν η ακολουθία εισόδου είναι μία πιθανή ακολουθία γονδολών. Δείτε τον παρακάτω πίνακα για παραδείγματα ακολουθιών που είναι και που δεν είναι ακολουθίες γονδολών. Πρέπει να υλοποιήσετε τη συνάρτηση `valid`.

- `valid(n, inputSeq)`
  - `n`: το μήκος της ακολουθίας εισόδου.
  - `inputSeq`: πίνακας μήκους n, όπου `inputSeq[i]` είναι το στοιχείο i της ακολουθίας εισόδου, για $0 \leq i \leq n-1$.
  - Η συνάρτηση πρέπει να επιστρέφει 1 αν η ακολουθία εισόδου είναι μία ακολουθία γονδολών, διαφορετικά 0.

Υποπροβλήματα 1, 2, 3

υποπρόβλημα	βαθμοί	n	`inputSeq`
1	5	$n \leq 100$	έχει τους αριθμούς από 1 έως και n ακριβώς μία φορά καθέναν
2	5	$n \leq 100{,}000$	$1 \leq$ `inputSeq[i]` $\leq n$
3	10	$n \leq 100{,}000$	$1 \leq$ `inputSeq[i]` $\leq 250{,}000$

Παραδείγματα

υποπρόβλημα	`inputSeq`	τιμή επιστροφής	σημείωση
1	(1, 2, 3, 4, 5, 6, 7)	1	
1	(3, 4, 5, 6, 1, 2)	1	
1	(1, 5, 3, 4, 2, 7, 6)	0	το 1 δεν μπορεί να είναι αμέσως πριν το 5
1	(4, 3, 2, 1)	0	το 4 δεν μπορεί να είναι αμέσως πριν το 3
2	(1, 2, 3, 4, 5, 6, 5)	0	δύο γόνδολες με τον αριθμό 5
3	(2, 3, 4, 9, 6, 7, 1)	1	ακολουθία αντικαταστάσεων (5, 8)
3	(10, 4, 3, 11, 12)	0	το 4 δεν μπορεί να είναι αμέσως πριν το 3

Ακολουθία αντικαταστάσεων

Στα επόμενα τρία υποπροβλήματα πρέπει να κατασκευάσετε μία δυνατή ακολουθία αντικαταστάσεων που παράγει τη δοθείσα ακολουθία γονδολών. Οποιαδήποτε τέτοια ακολουθία αντικαταστάσεων θα γίνεται δεκτή. Πρέπει να υλοποιήσετε τη συνάρτηση `replacement`.

- `replacement(n, gondolaSeq, replacementSeq)`
  - `n`: το μήκος της ακολουθίας γονδολών.

- `gondolaSeq`: πίνακας μήκους n που περιέχει εγγυημένα μία ακολουθία γονδολών, όπου `gondolaSeq[i]` είναι το στοιχείο i της ακολουθίας, για $0 \le i \le n-1$.
- Η συνάρτηση πρέπει να επιστρέφει l, το μήκος της ακολουθίας αντικαταστάσεων.
- `replacementSeq`: πίνακας αρκετά μεγάλος ώστε να χωρέσει την ακολουθία αντικαταστάσεων. Θα πρέπει να τοποθετήσετε το στοιχείο i της ακολουθίας αντικαταστάσεων στο `replacementSeq[i]`, για $0 \le i \le l-1$.

Υποπροβλήματα 4, 5, 6

υποπρόβλημα	**βαθμοί**	n	**gondolaSeq**
4	5	$n \le 100$	$1 \le$ `gondolaSeq[i]` $\le n+1$
5	10	$n \le 1{,}000$	$1 \le$ `gondolaSeq[i]` $\le 5{,}000$
6	20	$n \le 100{,}000$	$1 \le$ `gondolaSeq[i]` $\le 250{,}000$

Παραδείγματα

υποπρόβλημα	**gondolaSeq**	**τιμή επιστροφής**	**replacementSeq**
4	(3, 1, 4)	1	(2)
4	(5, 1, 2, 3, 4)	0	( )
5	(2, 3, 4, 9, 6, 7, 1)	2	(5, 8)

Μέτρηση ακολουθιών αντικαταστάσεων

Στα επόμενα τέσσερα υποπροβλήματα πρέπει να μετράτε το πλήθος των δυνατών ακολουθιών αντικαταστάσεων που παράγουν τη δοθείσα ακολουθία (που μπορεί να είναι ή να μην είναι ακολουθία γονδολών), modulo **1,000,000,009**. Πρέπει να υλοποιήσετε τη συνάρτηση `countReplacement`.

- `countReplacement(n, inputSeq)`
    - `n`: το μήκος της ακολουθίας εισόδου.
    - `inputSeq`: πίνακας μήκους n, όπου `inputSeq[i]` είναι το στοιχείο i της ακολουθίας εισόδου, για $0 \le i \le n-1$.
    - Αν η ακολουθία εισόδου είναι ακολουθία γονδολών, τότε μετρήστε το πλήθος των ακολουθιών αντικαταστάσεων που παράγουν αυτή την ακολουθία γονδολών (το πλήθος τους μπορεί να είναι τεράστιο), *και επιστρέψτε αυτό τον αριθμό modulo **1,000,000,009***. Αν η ακολουθία εισόδου δεν είναι ακολουθία γονδολών, η συνάρτηση πρέπει να επιστρέφει 0. Αν η ακολουθία εισόδου είναι ακολουθία γονδολών αλλά καμία γόνδολα δεν έχει χαλάσει, η συνάρτηση πρέπει να επιστρέφει 1.

Υποπροβλήματα 7, 8, 9, 10

υποπρόβλημα	**βαθμοί**	n	**inputSeq**
7	5	$4 \le n \le 50$	$1 \le$ `inputSeq[i]` $\le n+3$
8	15	$4 \le n \le 50$	$1 \le$ `inputSeq[i]` ≤ 100, και τουλάχιστον $n-3$ από τις αρχικές γόνδολες $1, \ldots, n$ δεν έχουν χαλάσει.
9	15	$n \le 100,000$	$1 \le$ `inputSeq[i]` $\le 250,000$
10	10	$n \le 100,000$	$1 \le$ `inputSeq[i]` $\le 1,000,000,000$

Παραδείγματα

υποπρόβλημα	**inputSeq**	**τιμή επιστροφής**	**ακολουθία αντικαταστάσεων**
7	(1, 2, 7, 6)	2	(3, 4, 5) ή (4, 5, 3)
8	(2, 3, 4, 12, 6, 7, 1)	1	(5, 8, 9, 10, 11)
9	(4, 7, 4, 7)	0	η `inputSeq` δεν είναι ακολουθία γονδολών
10	(3, 4)	2	(1, 2) ή (2, 1)

Λεπτομέρειες υλοποίησης

Πρέπει να υποβάλετε ακριβώς ένα αρχείο, με όνομα `gondola.c`, `gondola.cpp` ή `gondola.pas`. Αυτό το αρχείο πρέπει να υλοποιεί και τα τρία υποπρογράμματα που περιγράφονται παραπάνω (ακόμα και αν προτίθεστε να λύσετε μόνο κάποιο από τα υποπροβλήματα), με τις παρακάτω επικεφαλίδες. Για τις υλοποιήσεις σε C/C++, το αρχείο σας πρέπει να κάνει include το αρχείο επικεφαλίδας `gondola.h`.

Πρόγραμμα C/C++

```
int valid(int n, int inputSeq[]);
int replacement(int n, int gondolaSeq[], int replacementSeq[]);
int countReplacement(int n, int inputSeq[]);
```

Πρόγραμμα Pascal

```
function valid(n: longint; inputSeq: array of longint): integer;
function replacement(n: longint; gondolaSeq: array of longint;
var replacementSeq: array of longint): longint;
function countReplacement(n: longint; inputSeq: array of longint):
longint;
```

Ενδεικτικός βαθμολογητής

Ο ενδεικτικός βαθμολογητής διαβάζει την είσοδο με την ακόλουθη μορφή:

- γραμμή 1: `T`, ο αριθμός του υποπροβλήματος που το πρόγραμμά σας προτίθεται να λύσει ($1 \le T \le 10$).
- γραμμή 2: `n`, το μήκος της ακολουθίας εισόδου.
- γραμμή 3: Αν το `T` είναι 4, 5, ή 6, η γραμμή αυτή περιέχει `gondolaSeq[0]`, ..., `gondolaSeq[n-1]`. Διαφορετικά, η γραμμή αυτή περιέχει `inputSeq[0]`, ..., `inputSeq[n-1]`.