



## Gondola

Mao-Kong Gondola je poznata turistička atrakcija u Taipei. Sistem gondola čini jedna kružna tračnica, jedna stanica, i  $n$  gondola numerisanih redom brojevima od 1 do  $n$  koje se kreću datom tračnicom u jednom nepromjenjivom smjeru. Na samom početku, nakon što gondola  $i$  prodje kroz stanicu, sljedeća gondola koja će proći stanicu će biti gondola  $i + 1$  ako je  $i < n$ , ili gondola 1 ako je  $i = n$ .

Gondola se mogu pokvariti. Srećom imamo na raspolaganju neograničen broj rezervnih gondola, numerisanih sa  $n + 1$ ,  $n + 2$ , i tako dalje. Kada se jedna gondola pokvari zamijenimo je sa (na istom mjestu na tračnici) sa prvom dostupnom rezervnom gondolom, to jest, dostupnom rezervnom gondolom sa najmanjim brojem. Na primjer, ako imamo ukupno 5 gondola i gondola 1 se pokvari, onda će ona biti zamijenjena sa gondolom 6.

Uživajte da stojite na stanici i posmatrate kako prolaze gondole. Niz od  $n$  brojeva na gondolama koje prolaze kroz stanicu se naziva *sekvenca gondola*. Moguće je da se jedna ili više gondola pokvare (i koje će biti zamijenjene) prije nego što vi stignete, ali nijedna gondola se ne može pokvariti dok posmatrate.

Primjetimo da ista konfiguracija gondola na tračnici može dati više različitih *sekvenci gondola*, u zavisnosti od toga koja gondola prodje prva nakon što vi dodjete na stanicu. Na primjer, ako se nije pokvarila ni jedna gondola onda su i (2, 3, 4, 5, 1) kao i (4, 5, 1, 2, 3) moguće *sekvence gondola*, ali (4, 3, 2, 5, 1) to sigurno nije (ovdje se gondole pojavljuju u pogrešnom redosljedu).

Ukoliko se gondola 1 pokvari, moguće je da primjetimo ovaj niz (4, 5, 6, 2, 3) kao *sekvencu gondola*. Ukoliko je gondola 4 sljedeća koja se pokvari, zamijenimo je sa gondolom 7 i onda je moguće da primjetimo (6, 2, 3, 7, 5) kao *sekvencu gondola*. Ukoliko se sada pokvari gondola 7, biće zamijenjena sa gondolom 8 tako da je moguće da primjetimo *sekvencu gondola* (3, 8, 5, 6, 2).

pokvarena gondola	nova gondola	moguća sekvenca gondola
1	6	(4, 5, 6, 2, 3)
4	7	(6, 2, 3, 7, 5)
7	8	(3, 8, 5, 6, 2)

Sekvenca brojeva gondola koje su se pokvarile se naziva *zamjenska sekvenca* u onom redosljedu u kojem su se pokvarile. U prethodnom primjeru *zamjenska sekvenca* je (1, 4, 7).

Zamjenska sekvenca  $r$  koja daje *sekvencu gondola*  $g$  ako je, nakon što se gondole kvare redom prema zamjenskoj sekvenci  $r$ , moguće primjetiti *sekvencu gondola*  $g$ .

## Provjera Sekvence Gondola

U prva tri podzadatka morate provjeriti da li je ulazni niz *sekvenca gondola*. Pogledajte tabelu ispod za primjere koji jesu i koji nisu *sekvence gondola*. Morate implementirati funkciju `valid`.

- `valid(n, inputSeq)`
  - $n$ : dužina ulaznog niza, odnosno broj gondola koje rade.
  - `inputSeq`: niz dužine  $n$ ; `inputSeq[i]` je element  $i$  ulazne sekvence, za  $0 \leq i \leq n - 1$ .
  - Funkcija treba vratiti 1 ako je ulazni niz *sekvenca gondola*, ili 0 u suprotnom.

## Podzadaci 1, 2, 3

podzadatak	poeni	$n$	<code>inputSeq</code>
1	5	$n \leq 100$	ima svaki broj od 1 do $(n)$ tačno jednom
2	5	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq n$
3	10	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 250,000$

## Primjeri

podzadatak	<code>inputSeq</code>	izlazna veličina	primjedba
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	1 se ne može pojaviti neposredno prije 5
1	(4, 3, 2, 1)	0	4 se ne može pojaviti neposredno prije 3
2	(1, 2, 3, 4, 5, 6, 5)	0	dvije gondole sa brojem 5
3	(2, 3, 4, 9, 6, 7, 1)	1	*zamjenska sekvenca (5, 8)
3	(10, 4, 3, 11, 12)	0	4 se ne može pojaviti neposredno prije 3

## Zamjenska sekvenca

U sljedeća tri podzadatka morate odrediti moguću *zamjensku sekvencu* koja se dobija datom *sekvencom gondola*. Morate implementirati funkciju `replacement`.

- `replacement(n, gondolaSeq, replacementSeq)`
  - $n$  je dužina *sekvenci gondola*.
  - `gondolaSeq`: niz dužine  $n$ ; `gondolaSeq` garantuje da će biti *sekvenca gondola* `gondolaSeq[i]` je element  $(i)$  sekvence, za  $(0 \leq i \leq n - 1)$ .
  - Funkcija treba da vrati  $(l)$ , dužinu *zamjenske sekvence*.
  - `replacementSeq`: niz koji je dovoljne veličine za čuvanje *zamjenske sekvence*; trebalo bi vratiti svoj niz postavljanjem elementa  $(i)$  vaše *zamjenske sekvence* u `replacementSeq[i]`, za  $0 \leq i \leq l - 1$ .

## Podzadaci 4, 5, 6

podzadatak	poeni	$n$	<code>gondolaSeq</code>
4	5	$n \leq 100$	$1 \leq \text{gondolaSeq}[i] \leq n + 1$
5	10	$n \leq 1,000$	$1 \leq \text{gondolaSeq}[i] \leq 5,000$
6	20	$n \leq 100,000$	$1 \leq \text{gondolaSeq}[i] \leq 250,000$

## Primjer

podzadatak	<code>gondolaSeq</code>	izlazna veličina	<code>replacementSeq</code>
4	(3, 1, 4)	1	(2)
4	(5, 1, 2, 3, 4)	0	( )
5	(2, 3, 4, 9, 6, 7, 1)	2	(5, 8)

## Prebrojavanje zamjenskih sekvence

U sljedeća četiri podzadanka izračunajte broj mogućih *zamjenskih sekvenci* koje proizvode date sekvence (koje mogu ili ne mogu biti *sekvence gondola*), po modulu **100000009** \*. Implementirajte funkciju `countReplacement`.

- `countReplacement(n, inputSeq)`
  - $n$ : dužina ulazne sekvence.
  - `inputSeq`: niz dužine  $n$ ; `inputSeq[i]` je element  $i$  ulazne sekvence, za  $0 \leq i \leq n - 1$ .
  - Ako je ulazna sekvenca je niz koji predstavlja *sekvencu gondola*, a zatim izbrojite zamjene sekvence koje proizvode ove gondole sekvence (koji može biti izuzetno velika),  $i$  vratiti ovaj broj po modulu **100000009** . Ako ulazna sekvenca nije sekvenca gondola, funkcija treba vratiti 0. Ako je ulazni niz sekvenca gondola, ali se ni jedna gondola nije pokvarila, funkcija treba vratiti 1.

## Podzadaci 7, 8, 9, 10

podzadatak	poeni	$n$	<code>inputSeq</code>
7	5	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq n + 3$
8	15	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq 100$ , $i$ najmanje $n - 3$ od početnih gondola $1, \dots, n$ se nije pokvarilo.
9	15	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 250,000$
10	10	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 1,000,000,000$

## Primjeri

podzadatak	inputSeq	return value	zamjenska sekvenca
7	(1, 2, 7, 6)	2	(3, 4, 5) or (4, 5, 3)
8	(2, 3, 4, 12, 6, 7, 1)	1	(5, 8, 9, 10, 11)
9	(4, 7, 4, 7)	0	inputSeq nije sekvenca gondola
10	(3, 4)	2	(1, 2) or (2, 1)

## Detalji implementacije

Treba da predate tačno jedan file, koji se naziva `gondola.c`, `gondola.cpp` ili `gondola.pas`. Ovaj file treba da implementira opisanu funkciju koristeći sljedeće signature. Pored toga, morate uključiti header file `holiday.h` u slučaju C/C++ implementacije.

### C/C++ programs

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

### Pascal programs

```
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;
```

## Sample grader

Grader čita ulazne podatke u sljedećem formatu:

- line 1:  $T$ , the subtask number your program intends to solve ( $1 \leq T \leq 10$ ).
- line 2:  $n$ , the length of the input sequence.
- line 3: If  $T$  is 4, 5, or 6, this line contains `inputSeq[0], ..., inputSeq[n-1]`. Otherwise this line contains `gondolaSeq[0], ..., gondolaSeq[n-1]`.