



Gondola

Mao-Kong Gondola je poznata turistička atrakcija u Taipeiju. Sistem gondola čini jedna kružna staza, jedna stanica, i n gondola numerisanih redom brojevima od 1 do n koje se kreću datom stazom u jednom fiksnom smjeru. Nakon što gondola i prođe kroz stanicu, sljedeća gondola koja će proći stanicu je 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 zamjenjujemo je (na istom mjestu na stazi) sa prvom dostupnom rezervnom gondolom, tj. onom sa najmanjim brojem. Na primjer, ako imamo 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 *sekvencu 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.

Primijetimo da ista konfiguracija gondola na stazi može dati više različitih *sekvenci gondola*, u zavisnosti od toga koja gondola prođe prva nakon što vi dođete 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 uočimo niz (4, 5, 6, 2, 3) kao *sekvencu gondola*. Ukoliko je gondola 4 sljedeća koja se pokvari, zamijenićemo je sa gondolom 7 i onda je moguće da uočimo sekvencu (6, 2, 3, 7, 5). Ukoliko se sada pokvari gondola 7, biće zamijenjena sa gondolom 8 tako da je moguće da uočimo sekvencu (3, 8, 5, 6, 2).

| pokvarena gondola | nova gondola | moguća sekvencu 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 naziva se *zamjenska sekvencu*, i prikazuje se u onom redosljedu u kojem su se gondole kvarile. U prethodnom primjeru *zamjenska sekvencu* je (1, 4, 7). Zamjenska sekvencu r *proizvodi* sekvencu gondola g ako se, nakon što se gondole pokvare u skladu sa zamjenskom sekvencu r , može uočiti sekvencu gondola g .

Provjeravanje sekvenci gondola

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

- `valid(n, inputSeq)`
 - n : dužina ulaznog niza (sekvence), 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 da vrati 1 ako je ulazni niz *sekvenca gondola*, ili 0 u suprotnom slučaju.

Podzadaci 1, 2, 3

| podzadatak | poeni | n | <code>inputSeq</code> |
|------------|-------|------------------|--|
| 1 | 5 | $n \leq 100$ | sadrži 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> | rezultat | komentar |
|------------|-----------------------|----------|--|
| 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 podzadataka morate odrediti jednu od mogućih zamjenskih sekvenci koje proizvode datu sekvencu gondola. Bilo koja zamjenska sekvenca će biti prihvaćena. Morate implementirati funkciju `replacement`.

- `replacement(n, gondolaSeq, replacementSeq)`
 - n je dužina sekvence gondola.
 - `gondolaSeq`: niz dužine n ; `gondolaSeq` je obavezno ispravna sekvenca gondola, `gondolaSeq[i]` je i -ti element 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 da vratite vaš niz postavljanjem elementa i 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> | rezultat | <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 sekvenci

U sljedeća četiri podzadataka morate izračunati broj mogućih zamjenskih sekvenci koje proizvode datu sekvencu (koja može ali i ne mora biti sekvenca gondola), po modulu **1,000,000,009**. Morate implementirati 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 ispravna sekvenca gondola, potrebno je da izračunate broj zamjenskih sekvenci koje proizvode ulaznu sekvencu (koji bi mogao biti izuzetno veliki), i vratiti ovaj broj po modulu **1,000,000,009**. Ako ulazna sekvenca nije ispravna sekvenca gondola, funkcija treba da vrati 0. Ako je ulazna sekvenca ispravna sekvenca gondola, pri čemu se nijedna gondola nije pokvarila, funkcija treba da vrati 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 | $4 \leq n \leq 50$ | $1 \leq \text{inputSeq}[i] \leq 100$, pri čemu se najmanje $n - 3$ od početnog niza gondola $1, \dots, n$ nije kvarilo. |
| 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 | rezultat | zamjenska sekvenca |
|------------|------------------------|----------|--------------------------------|
| 7 | (1, 2, 7, 6) | 2 | (3, 4, 5) ili (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) ili (2, 1) |

Implementacioni detalji

Potrebno je da predate samo jedan fajl sa imenom `gondola.c`, `gondola.cpp` ili `gondola.pas`. Ovaj fajl treba da implementira gore opisane potprograme (funkcije) koristeći sljedeće signature (potpise, zaglavja). Potrebno je da uključite heder fajl `gondola.h` za C/C++ implementaciju.

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;
```

Ocjenjivač

Ocjenjivač čita ulaz u sljedećem formatu:

- linija 1: T , broj podzadatka ($1 \leq T \leq 10$).
- linija 2: n , dužina ulazne sekvence.
- linija 3: Ako je T 4, 5, ili 6, ova linija sadrži `gondolaSeq[0], ..., gondolaSeq[n-1]`. U suprotnom slučaju, linija sadrži `inputSeq[0], ..., inputSeq[n-1]`.