



Gondola

Gondola Mao-Kong je priljubljena znamenitost mesta Taipei. Tvorijo jo krožna proga, ena postaja in n gondol, zaporedoma oštevilčenih od 1 od n , ki se po progi premikajo samo v eni smeri. Velja, da ko gondola i zapusti postajo, bo naslednja gondola, ki prispe na postajo, $i + 1$ (če $i < n$), oziroma gondola 1 (če $i = n$).

Tudi tu lahko gondole zgrmijo na tla. Na srečo obstaja neskončna zaloga nadomestnih gondol, ki so oštevilčene z $n + 1$, $n + 2$, in tako naprej. Ko se neka gondola zruši, jo zamenjamo (na istem položaju na progi) s prvo nadomestno gondolo, to je tisto, ki ima najmanjšo številko. Za primer imejmo pet gondol. Če pade gondola 1, jo zamenja gondola 6.

Rad stojiš na postaji in opazuješ gondole kako gredo mimo. *Zaporedje gondol* je zaporedje n števil gondol, ki gredo mimo postaje. Obstaja možnost, da je ena ali več gondol padlo po tleh (in je bilo zamenjanih) preden si prišel, vendar se nobena gondola ne zruši tekom tvoje opazovanja.

Bodi pozoren na to, da lahko isto postavitve gondol na progi vidimo kot več zaporedij gondol. Odvisno je od tega, katera gondola gre prva mimo nas, ko prispemo na postajo. Na primer če se nobena gondola ni zrušila, potem sta možni npr. zaporedji (2, 3, 4, 5, 1) in (4, 5, 1, 2, 3); (4, 3, 2, 5, 1) pa ni (ker se gondole pojavijo v napačnem zaporedju).

Če bi se zrušila gondola 1, potem bi lahko videli zaporedje gondol (4, 5, 6, 2, 3). Če bi se naslednja zrušila gondola 4, bi jo nadomestili s 7 in lahko bi videli zaporedje gondol (6, 2, 3, 7, 5). Če pa bi naslednja padla gondola 7, bi jo zamenjala gondola 8 in lahko bi videli zaporedje gondol (3, 8, 5, 6, 2).

stara gondola	nova gondola	možno zaporedje gondol
1	6	(4, 5, 6, 2, 3)
4	7	(6, 2, 3, 7, 5)
7	8	(3, 8, 5, 6, 2)

Zaporedje zamenjav je zaporedje sestavljeno iz števil padlih gondol, v vrstnem redu njihove pogube. V zgornjem primeru je zaporedje zamenjav (1, 4, 7). Zaporedje zamenjav r proizvede zaporedje gondol g , če po menjavi gondol v vrstnem redu zaporedja zamenjav r lahko vidimo zaporedje gondol g .

Preverjanje zaporedja gondol

Pri prvih treh podnalogah moraš preveriti ali je vhodno zaporedje veljavno zaporedje gondol. Glej spodnjo tabelo za primere zaporedij, katera so in katera niso veljavna zaporedja gondol. Implementirati moraš funkcijo `valid`.

- `valid(n, inputSeq)`
 - `n`: dolžina vhodnega zaporedja.
 - `inputSeq`: tabela dolžine `n`; `inputSeq[i]` je element `i` vhodnega zaporedja, za $0 \leq i \leq n - 1$.
 - Funkcija naj vrne 1, če je vhodno zaporedje veljavno zaporedje gondol, in 0 sicer.

Podnaloge 1, 2, 3

podnaloga	točke	n	<code>inputSeq</code>
1	5	$n \leq 100$	vsebuje vsa števila od 1 do n natanko enkrat
2	5	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq n$
3	10	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 250,000$

Primeri

podnaloga	<code>inputSeq</code>	rezultat	opomba
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 more pojaviti takoj pred 5
1	(4, 3, 2, 1)	0	4 se ne more pojaviti takoj pred 3
2	(1, 2, 3, 4, 5, 6, 5)	0	dve gondoli s številko 5
3	(2, 3, 4, 9, 6, 7, 1)	1	zaporedje zamenjav (5, 8)
3	(10, 4, 3, 11, 12)	0	4 se ne more pojaviti takoj pred 3

Zaporedje zamenjav

Pri naslednjih treh podnalogah moraš sestaviti možno zaporedje zamenjav, ki proizvede podano zaporedje gondol. Sprejeto bo katero koli zaporedje zamenjav. Implementirati moraš funkcijo `replacement`.

- `replacement(n, gondolaSeq, replacementSeq)`
 - `n` je dolžina zaporedja gondol.
 - `gondolaSeq`: tabela dolžine `n`; `gondolaSeq` je zagotovo zaporedje gondol, in `gondolaSeq[i]` je element `i` zaporedja, za $0 \leq i \leq n - 1$.
 - funkcija naj vrne `l`, dolžino zaporedja zamenjav.
 - `replacementSeq`: tabela je dovolj velika za hrambo zaporedja zamenjav; svoje zaporednje vračaš tako, da vstaviš element `i` svojega zaporedja zamenjav v `replacementSeq[i]`, za $0 \leq i \leq l - 1$.

Podnaloge 4, 5, 6

podnaloga	točke	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	$n \leq 100,000$	$1 \leq \text{gondolaSeq}[i] \leq 250,000$

Primeri

podnaloga	<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)

Preštevanje zaporedij zamenjav

Pri naslednjih štirih podnalogah moraš prešteti število možnih zaporedij zamenjav ki proizvedejo podano zaporedje (ki je, ali ni zaporedje gondol), po modulu 1,000,000,009. Implementirati moraš funkcijo `countReplacement`.

- `countReplacement(n, inputSeq)`
 - n : dolžina vhodnega zaporedja.
 - `inputSeq`: tabela dolžine n ; `inputSeq[i]` je element i vhodnega zaporedja, za $0 \leq i \leq n - 1$.
 - Če je vhodno zaporedje zaporedje gondol, potem preštej število zaporedij zamenjav, ki proizvedejo podano zaporedje gondol (ki je lahko zelo veliko), in vrni to število po modulu 1,000,000,009. Če vhodno zaporedje ni zaporedje gondol, naj funkcija vrne 0. Če je vhodno zaporedje zaporedje gondol, a nobena gondola ni bila zamenjana, potem naj funkcija vrne 1.

Podnaloge 7, 8, 9, 10

podnaloga	točke	n	<code>inputSeq</code>
7	5	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq n + 3$
8	15	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq 100$, in vsaj $n - 3$ izmed začetnih gondol $1, \dots, n$ ni bilo zamenjanih.
9	15	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 250,000$
10	10	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 1,000,000,000$

Primeri

podnaloga	inputSeq	vrednost rezultata	zaporedje zamenjav
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 ni zaporedje gondol
10	(3, 4)	2	(1, 2) or (2, 1)

Podrobnosti implementacije

Oddati morate natanko eno datoteko, imenovano `gondola.c`, `gondola.cpp` ali `gondola.pas`. V tej datoteki naj se nahajo implementacije zgoraj opisanih podprogramov, z naslednjimi podpisi. V jeziku C/C++ morate obvezno vključiti tudi header datoteko `gondola.h`.

Programski jezik C/C++

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

Programski jezik 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;
```

Vzorčni ocenjevalnik

Vzorčni ocenjevalnik bere vhod v sledeči obliki:

- vrstica 1: T , številka podnaloge ki jo tvoj program poskuša rešiti ($1 \leq T \leq 10$).
- vrstica 2: n , dolžina vhodnega zaporedja.
- vrstica 3: Če je T 4, 5, ali 6, ta vrstica vsebuje `gondolaSeq[0], ..., gondolaSeq[n-1]`. Sicer ta vrstica vsebuje `inputSeq[0], ..., inputSeq[n-1]`.