



Gondola

A Mao-Kong Gondola Tajpei egyik látványossága, amely egy körgyűrű egy állomással és n gondolával, amiket 1 -től n -ig sorszámozunk. Minden gondola azonos irányban halad, kezdetben sorszámuk szerinti sorrendben egymás után. Mivel körbe mennek, ezért az n . gondolát az 1 . követi.

A tartalék gondolák sorszámai: $n + 1$, $n + 2$, és így tovább. Amikor egy gondola elromlik, a sorrendben következő tartalékra cserélik (a lecserélt pozíciójára kerül). Például, ha 5 gondola esetén az 1 . elromlik, a 6 . gondolára cserélik.

Gondola sorozat: egy adott pillanattól az állomáson megfigyeljük az áthaladó n darab gondolát, a sorszámuk alkotta sorozatot nevezzük *gondola sorozatnak*. A megfigyelés előtt lehetett hibás gondola cseréje, de a megfigyelés közben már nem.

Például, ha nem volt még csere, akkor 5 gondola esetén a $(2, 3, 4, 5, 1)$ és a $(4, 5, 1, 2, 3)$ is gondola sorozat, de a $(4, 3, 2, 5, 1)$ nem (mert a sorrendjük nem jó).

Ha az 1 . gondolát lecseréljük, akkor a $(4, 5, 6, 2, 3)$ gondola sorozat lesz. Ha ezután a 4 . is meghibásodik, a 7 -re cseréljük és így a $(6, 2, 3, 7, 5)$ gondola sorozat lesz. Ha a 7 . romlik el, akkor a 8 -ra cseréljük és a $(3, 8, 5, 6, 2)$ gondola sorozat lesz.

hibás gondola	új gondola	lehetséges gondola sorozat
1	6	$(4, 5, 6, 2, 3)$
4	7	$(6, 2, 3, 7, 5)$
7	8	$(3, 8, 5, 6, 2)$

A *cseresorozat* elromlott gondolák sorszámai sorozata, időrend szerinti sorrendben. Az előző példában cseresorozat az $(1, 4, 7)$. Azt mondjuk, hogy az r cseresorozat a g gondola sorozatot eredményezi, ha az r -beli cseréket egymás után végrehajtva a g gondola sorozat megfigyelhető lesz.

Gondola sorozat ellenőrzés

Az első három részfeladatban ellenőrizned kell, hogy adott sorozat gondola sorozat-e valamely cseresorozat esetén! Ennek megoldására a `valid` függvényt kell megírnod!

- `valid(n, inputSeq)`
 - n : a bemenő sorozat hossza.
 - `inputSeq`: n elemű tömb; `inputSeq[i]` az ellenőrizendő sorozat i . eleme ($0 \leq i \leq n - 1$).
 - A függvény eredménye 1 legyen, ha a sorozat gondola sorozat, egyébként pedig 0 !

1., 2., 3. részfeladat

részfeladat	pont	n	inputSeq
1	5	$n \leq 100$	minden 1 és n közötti szám pontosan egyszer van benne
2	5	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq n$
3	10	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 250,000$

Példák

részfeladat	inputSeq	függvényérték	megjegyzés
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	az 1 nem lehet közvetlenül az 5 előtt
1	(4, 3, 2, 1)	0	a 4 nem lehet közvetlenül a 3 előtt
2	(1, 2, 3, 4, 5, 6, 5)	0	az 5 kétszer fordul elő
3	(2, 3, 4, 9, 6, 7, 1)	1	az (5, 8) cseresorozat esetén
3	(10, 4, 3, 11, 12)	0	a 4 nem lehet közvetlenül a 3 előtt

Cseresorozat

A következő három részfeladatban olyan cseresorozatot kell készítened, amely adott gondola sorozatot eredményez! Ennek megoldására a replacement függvényt kell megírnod!

- `replacement(n, gondolaSeq, replacementSeq)`
 - n a gondola sorozat hossza.
 - `gondolaSeq`: n elemű tömb; `gondolaSeq` biztosan gondola sorozat, `gondolaSeq[i]` a bemenő sorozat i . eleme ($0 \leq i \leq n - 1$).
 - A függvény eredménye a cseresorozat l hossza legyen!
 - `replacementSeq`: a cseresorozatot tartalmazó tömb; `replacementSeq[i]` legyen a cseresorozat i . eleme ($0 \leq i \leq l - 1$).

4., 5., 6. részfeladat

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

Példák

részfeladat	gondolaSeq	függvényérték	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)

Cseresorozatok száma

A következő négy részfeladatban a lehetséges cseresorozatok számát kell kiszámítanod adott sorozathoz (ami nem feltétlenül gondola sorozat), modulo **1,000,000,009**! Ennek megoldására a `countReplacement` függvényt kell megírnod!

- `countReplacement(n, inputSeq)`
 - n : a bemenő sorozat hossza.
 - `inputSeq`: n elemű tömb; `inputSeq[i]` a sorozat i . eleme ($0 \leq i \leq n - 1$).
 - Ha a bemenő sorozat gondola sorozat, akkor a függvény eredménye azon cseresorozatok száma legyen modulo **1,000,000,009**, amelyek ezt a gondolasorozatot eredményezhetik! Ha a bemenő sorozat nem gondola sorozat, akkor a függvény értéke 0 legyen! Ha a bemenő sorozat gondolasorozat, de nem volt csere, akkor a függvény értéke 1 legyen!

7., 8., 9., 10. részfeladat

részfeladat	pont	n	<code>inputSeq</code>
7	5	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq n + 3$
8	15	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq 100$, a kezdeti $1, \dots, n$ gondola sorozatból legalább $n - 3$ nem romlott el.
9	15	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 250,000$
10	10	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 1,000,000,000$

Példák

részfeladat	<code>inputSeq</code>	függvényérték	cseresorozat
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	<code>inputSeq</code> nem gondola sorozat
10	(3, 4)	2	(1, 2) vagy (2, 1)

Megvalósítás

A `gondola.c`, `gondola.cpp` vagy `gondola.pas` fájlt kell beküldened! Ebben kell megvalósítanod a kért függvényeket! Include-old a `gondola.h`-t!

C/C++ program

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

Pascal program

```
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;
```

Minta értékelő

A minta értékelő a bemenetet az alábbi formában várja:

- **1.** sor: T , részfeladat sorszáma ($1 \leq T \leq 10$).
- **2.** sor: n , a bemenő sorozat hossza.
- **3.** sor: Ha T értéke 4, 5, vagy 6, akkor ez a sor az `inputSeq[0], ..., inputSeq[n-1]` értékeket tartalmazza. Egyébként ez a sor a `gondolaSeq[0], ..., gondolaSeq[n-1]` értékeket tartalmazza.