



## Gondola

Mao-Kong, Taipeiň belli (meşhur) atraksionydyr. Gondola sistemasy şulardan ybaratdyr: halka şeklindäki demir ýol, ýekeje stansiýa, we belli bir ýolda gityän 1 bilen  $n$  aralygynda belgilenen gondolalar. Başlangyçda,  $i$ -nji gondola stansiýany geçenden soň, indiki geçmeli gondola  $i+1$  bolar (eger-de  $i < n$  bolsa), ýa-da 1 belgili gondola bolar (eger-de  $i = n$  bolsa).

Gondolalar döwülip bilýärler. Hudaýa şükür, zapasda tükeniksiz gondolalar bar, olar  $n+1$ ,  $n+2$ , ... şeklinde belgilenen. Eger-de haýsydam bolsa bir gondola döwülse biz ony şol öňki ýerinde zapasda başda duran başka gondola bilen çalyşarys, ýagny zapasdaky kiçi nomerli gondola bilen. Mysal üçin, eger-de 5 gondola bar bolsa we 1 belgili gondola döwülen bolsa, biz ony 6 belgili gondola bilen çalyşarys.

Sen stansiýada durup gondolalaryň geçişine seretmegi gowy görýäň. Gondola zygiderliligi, stansiýadan geçýän  $n$  sany gondolanyň seriýasydyr. Sen gelmäňkäň birnäçe gondola döwülüp çalyşylan bolmagy ähtimal, emma seniň seredip duran wagtyň hiç bir gondola döwülenok.

Üns ber, relsdaky şolbir gondolalaryň konfigurasiýasy birnäçe gondola zygiderliligini döredip bilýär. Bu, seniň stansiýa gelen wagtyň stansiýadan haýsy gondolanyň birinji geçeneine bagly. Mysal üçin, eger-de hiçbir gondola döwülmedik bolsa, onda (2, 3, 4, 5, 1)-em (4, 5, 1, 2, 3)-em mümkin zygiderliliklerdir, emma (4, 3, 2, 5, 1) däl (sebäbi gondolalar ýalňyş tertipde).

Eger-de 1 belgili gondola döwülse (4, 5, 6, 2, 3) zygiderliligini görüp bileris. Ondan soň 4 belgili gondola döwülse we 7 belgili gondola bilen çalyşylsa, onda biz (6, 2, 3, 7, 5) zygiderliligini görüp bileris. Ondan soňam 7 belgili gondola döwülse we 8 belgili gondola bilen çalyşylsa, onda (3, 8, 5, 6, 2) zygiderliligini görüp bileris.

döwülen gondola	täze gondola	mümkin gondola zygiderliligi
1	6	(4, 5, 6, 2, 3)
4	7	(6, 2, 3, 7, 5)
7	8	(3, 8, 5, 6, 2)

Çalyşma zygiderliligi döwülen gondolalaryň zygiderliligidir (döwüliş tertibine görä). Häлки mysalda çalyşma zygiderliligi (1, 4, 7)-dir. Çalyşma zygiderliligi ("r" bilen görkezilen) gondola zygiderliligini (g) şu ýagdaýda döretýär: eger-de çalyşma zygiderliligine görä gondolalar döwülenden soň gondola zygiderliligini görüp bolýan bolsa.

## Gondola Zygiderliligini Barlamak

Başdaky 3 ýumuşjukda girizilen zygiderliligiň gondola zygiderliligi bolup bolmadygyny barlamaly.

Yzygiderlilikleriň gondola zygiderliligi ýa-da dældigini görmek üçin aşakdaky tablisa seret. Siz valid atly funksiýany ýazmaly.

- `valid(n, inputSeq)`
  - `n`: girizilen zygiderliliğiň uzynlygy.
  - `inputSeq`: `n` uzynlygyndaky massiw; `inputSeq[i]`  $0 \leq i \leq n - 1$  üçin girizilen zygiderliliğiň  $i$ -nji elementi.
  - Eger girizilen zygiderlilik gondola zygiderliliği bolsa funksiýa 1 gaýtarmaly, däl bolsa 0 gaýtarmaly.

## Subtasks (ýumuşjuklar) 1, 2, 3

subtask	ballar	$n$	<code>inputSeq</code>
1	5	$n \leq 100$	1-den $n$ -e çenli her bir san diňe bir gezek bar
2	5	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq n$
3	10	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 250,000$

## Mysallar

subtask	<code>inputSeq</code>	gaýtarylýan san	bellik
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, 5-den öň gelip bilenok
1	(4, 3, 2, 1)	0	4, 3-den öň gelip bilenok
2	(1, 2, 3, 4, 5, 6, 5)	0	iki gondola 5 belgli
3	(2, 3, 4, 9, 6, 7, 1)	1	çalyşma zygiderliliği (5, 8)
3	(10, 4, 3, 11, 12)	0	4, 3-den öň gelip bilenok

## Çalyşma Zygiderliliği

Indiki 3 ýumuşjukda siz, berilen gondola zygiderliliğini döredip biljek mümkin çalyşma zygiderliliklerini gurmaly. Şonuň ýaly çalyşma zygiderliliğiň islendigi kabul bolýar. Siz `replacement` atly funksiýa ýazmaly.

- `replacement(n, gondolaSeq, replacementSeq)`
  - `n` gondola zygiderliliğiň uzynlygy.
  - `gondolaSeq`: `n` uzynlygyndaky massiw; `gondolaSeq` hökman gondola zygiderliliği bolar, we `gondolaSeq[i]` zygiderliliğiň  $i$ -nji elementi ( $0 \leq i \leq n - 1$  üçin).
  - Funksiýa çalyşma zygiderliliğiň uzynlygy `l`-ni gaýtarmaly ("l"-bir däl, "L" harpyň kiçisidir).
  - `replacementSeq`: çalyşma zygiderliliğini saklap biljek derejedäki uly massiwdir; siz zygiderliliğini  $i$ -nji elementi `replacementSeq[i]`-a goýup gaýtarmaly ( $0 \leq i \leq l - 1$

üçin).

## Subtasks 4, 5, 6

subtask	points	$n$	$gondolaSeq$
4	5	$n \leq 100$	$1 \leq gondolaSeq[i] \leq n + 1$
5	10	$n \leq 1,000$	$1 \leq gondolaSeq[i] \leq 5,000$
6	20	$n \leq 100,000$	$1 \leq gondolaSeq[i] \leq 250,000$

## Examples (Mysallar)

subtask	$gondolaSeq$	gaýtarylýan san	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)

## Çalyşma Yzygiderlilikini Sanamak

Indiki dört ýumuşjukda berilen zzygiderliliği (gondola zzygiderlili bolubam biler bolmanam biler) döredip biljek mümkin çalyşma zzygiderliliklerini sanamaly, mod **1,000,000,009** görä. Siz `countReplacement` atly funksiýa ýazmaly.

- `countReplacement(n, inputSeq)`
  - $n$ : girizilen zzygiderliliğiň uzynlygy.
  - `inputSeq`:  $n$  uzynlygyndaky massiw; `inputSeq[i]` girizilen zzygiderliliğiň  $i$ -nji elementi, for  $0 \leq i \leq n - 1$ .
  - Eger girizilen zzygiderlilik gondola zzygiderliliği bolsa, onda bu zzygiderliliği döredip biljek çalyşma zzygiderliliğini sanaň we **1,000,000,009** görä moduny (modul) alyp ugradyň. Eger girizilen zzygiderlilik gondola zzygiderliliği däl bolsa, funksiýa 0 gaýtramly. Eger girizilen zzygiderlilik gondola zzygiderliliği bolsa emma hiçir gondola döwülmeşe onda 1 gaýtarylmaly.

## Subtasks 7, 8, 9, 10

subtask	ballar	$n$	$inputSeq$
7	5	$4 \leq n \leq 50$	$1 \leq inputSeq[i] \leq n + 3$
8	15	$4 \leq n \leq 50$	$1 \leq inputSeq[i] \leq 100$ , and at least $n - 3$ of the initial gondolas $1, \dots, n$ did not break down. (başlangyçdaky $1, \dots, n$ gondolalaryň iň azyndan $n+3$ sanysy döwülmedi)
9	15	$n \leq 100,000$	$1 \leq inputSeq[i] \leq 250,000$
10	10	$n \leq 100,000$	$1 \leq inputSeq[i] \leq 1,000,000,000$

## Examples (Mysallar)

subtask	inputSeq	gaýtarmaly san	çalyşma yzygiderliligi
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 gondola yzygiderliligi däl
10	(3, 4)	2	(1, 2) or (2, 1)

## Implementation details

You have to submit exactly one file, called `gondola.c`, `gondola.cpp` or `gondola.pas`. This file should implement the subprograms described above, using the following signatures. You also need to include a header file `gondola.h` for C/C++ implementation.

### 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;
```

## Mysaly greýder

Greýder girişi aşakdaky formatda okaýar:

- 1-nji setir:  $T$ , siziň programmaňyzyň çözmeli ýumuşjuk sany ( $1 \leq T \leq 10$ ).
- 2-nji setir:  $n$ , girizilýän yzygiderliligiň uzynlygy.
- 3-nji setir: Eger de  $T$  4, 5, ýa 6 bolsa, bu setirde `inputSeq[0], ..., inputSeq[n-1]` bolar; beýleki ýagdaýda bu setirde `gondolaSeq[0], ..., gondolaSeq[n-1]` bolar.