



Vagoniņi

Mao-Kong vagoniņi ir slavēta Taipejas izklaide. Vagoniņu sistēma sastāv no cikliska gaisa trosu ceļa ar vienu staciju un n vagoniņiem, kas secīgi sanumurēti no 1 līdz n . Vagoniņi tiek pārvietoti vienā noteiktā virzienā. Sākumā, pēc tam kad vagoniņš ar numuru i pabrauks garām stacijai, tam sekos vagoniņš ar numuru $i + 1$, ja $i < n$, vai vagoniņš ar numuru 1, ja $i = n$.

Vagoniņi mēdz salūzt. Par laimi, mums ir bezgalīgi daudz rezerves vagoniņu, ar kuriem aizstāt salūzušos vagoniņus. Tie sanumurēti ar $n + 1$, $n + 2$, utt. Kad kāds vagoniņš salūst, tas tiek aizstāts (tajā pašā pozīcijā) ar rezerves vagoniņu ar mazāko neizmanto numuru. Piemēram, ja sistēma sastāv no pieciem vagoniņiem un vagoniņš nr.1 salūst, tad tas tiks aizstāts ar vagoniņu nr.6.

Jums patīk stāvēt stacijā un vērot, kā vagoniņi pabrauc garām stacijai. *Vagoniņu virkne* ir n vagoniņu numuru virkne, kas pabrauc garām stacijai. Iespējams, ka, pirms Jūs sākat vērot vagoniņu kustību, viens vai vairāki vagoniņi jau bija salūzuši (un ir jau aizvietoti ar rezerves vagoniņiem), bet neviens no vagoniņiem nesalūst, kamēr Jūs tos vērojat.

Ievērojiet, ka viens un tas pats vagoniņu izvietojums var tikt aprakstīts ar vairākām vagoniņu virknēm, atkarībā no tā, kurš vagoniņš brauc garām stacijai, kad Jūs tur ieradāties. Piemēram, ja neviens vagoniņš nav salūzis, tad abas virknes (2, 3, 4, 5, 1) un (4, 5, 1, 2, 3) ir korektas vagoniņu virknes, bet (4, 3, 2, 5, 1) nav, jo vagoniņi parādās nepareizā secībā.

Ja salūst vagoniņš nr.1, tad varētu novērot vagoniņu virkni (4, 5, 6, 2, 3). Ja pēc tam salūst vagoniņš nr.4, tas tiek aizstāts ar vagoniņu nr.7 un varētu novērot vagoniņu virkni (6, 2, 3, 7, 5). Ja vēl pēc tam salūst vagoniņš nr.7, tas tiek aizstāts ar vagoniņu nr.8 un tagad var novērot vagoniņu virkni (3, 8, 5, 6, 2).

Salūzušā vagoniņa nr.	Jaunā vagoniņa nr.	Korektas vagoniņu virknes piemērs
1	6	(4, 5, 6, 2, 3)
4	7	(6, 2, 3, 7, 5)
7	8	(3, 8, 5, 6, 2)

Aizstāšanu virkne ir salūzušo vagoniņu numuru virkne tieši tādā secībā, kādā tie salūza. Iepriekšējā piemērā aizstāšanu virkne ir (1, 4, 7). No aizstāšanu virknes r var iegūt vagoniņu virkni g , ja pēc vagoniņu aizvietošanas atbilstoši aizstāšanu virknei r ir iespējams novērot vagoniņu virkni g .

Vagoniņu virknes pārbaude

Pirmajos trijos apakšuzdevumos Jums jāpārbauda, vai ievadā dotā virkne var būt korekta vagoniņu virkne. Korektu un nekorektu vagoniņu virkņu piemēri doti zemāk esošajā tabulā. Jums jārealizē funkcija `valid`.

- `valid(n, inputSeq)`
 - n : ievada virknes garums.
 - `inputSeq`: n elementu masīvs; `inputSeq[i]` satur ievada virknes i -to elementu, $0 \leq i \leq n - 1$.
 - Funkcijas rezultātam jābūt 1, ja ievadā dotā virkne ir korekta vagoniņu virkne, vai 0, ja nav.

Apakšuzdevumi 1, 2, 3

Apakšuzdevums	Punkti	n	<code>inputSeq</code>
1	5	$n \leq 100$	satur katru skaitli no 1 to n tieši vienu reizi
2	5	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq n$
3	10	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 250,000$

Piemēri

Apakšuzdevums	<code>inputSeq</code>	Rezultāts	Piezīmes
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 nevar parādīties tieši pirms 5
1	(4, 3, 2, 1)	0	4 nevar parādīties tieši pirms 3
2	(1, 2, 3, 4, 5, 6, 5)	0	divi vagoniņi ar nr.5
3	(2, 3, 4, 9, 6, 7, 1)	1	aizstāšanu virkne (5, 8)
3	(10, 4, 3, 11, 12)	0	4 nevar parādīties tieši pirms 3

Aizstāšanu virkne

Nākamajos trīs apakšuzdevumos Jums jākonstruē aizstāšanu virkne, no kuras būtu iespējams iegūt doto vagoniņu virkni. Jebkura derīga aizstāšanu virkne tiks akceptēta. Jums jārealizē funkcija `replacement`.

- `replacement(n, gondolaSeq, replacementSeq)`
 - n vagoniņu virknes garums.
 - `gondolaSeq`: n elementu masīvs. Tiek garantēts, ka `gondolaSeq` ir korekta vagoniņu virkne, un `gondolaSeq[i]` ir i -tais vagoniņu virknes elements, $0 \leq i \leq n - 1$.
 - Funkcijai jāatgriež l - aizstāšanu virknes garums.
 - `replacementSeq`: masīvs, kas ir pietiekami garš, lai saglabātu aizstāšanu virkni; atrastās aizstāšanu virknes i -tais elements jāieraksta elementā `replacementSeq[i]`, $0 \leq i \leq l - 1$.

Apakšuzdevumi 4, 5, 6

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

Piemēri

Apakšuzdevums	<code>gondolaSeq</code>	Atgriežamā vērtība	<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)

Aizstāšanu virkņu skaitīšana

Nākamajos četros apakšuzdevumos Jums jāskaita iespējamo aizstāšanu virkņu, no kurām var iegūt doto vagoniņu virkni, skaits. Ievērojiet, ka dotā vagoniņu virkne var nebūt korekta! Atbilde jāizvada pēc moduļa **1,000,000,009**. Jums jārealizē funkcija `countReplacement`.

- `countReplacement(n, inputSeq)`
 - n : ievada virknes garums.
 - `inputSeq`: n elementu masīvs; `inputSeq[i]` satur i -to ievada virknes elementu, $0 \leq i \leq n - 1$.
 - Ja ievadā saņemtā virkne ir korekta vagoniņu virkne, tad saskaitiet iespējamās aizstāšanu virknes, no kurām var iegūt doto vagoniņu virkni, un atgrieziet atbildi pēc moduļa **1,000,000,009**, jo atbilde var būt ļoti liels skaitlis. Ja ievada virkne nav korekta vagoniņu virkne, tad funkcijai jāatgriež 0. Ja ievada virkne ir korekta vagoniņu virkne, bet neviens vagoniņš nav salūzis, tad funkcijai jāatgriež 1.

Apakšuzdevumi 7, 8, 9, 10

Apakšuzdevums	Punkti	n	<code>inputSeq</code>
7	5	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq n + 3$
8	15	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq 100$, un vismaz $n - 3$ no sākotnējiem vagoniņiem $1, \dots, n$ nav salūzuši.
9	15	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 250,000$
10	10	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 1,000,000,000$

Piemēri

Apakšuzdevums	inputSeq	Rezultāts	Aizstāšanu virkne
7	(1, 2, 7, 6)	2	(3, 4, 5) vai (4, 5, 3)
8	(2, 3, 4, 12, 6, 7, 1)	1	(5, 8, 9, 10, 11)
9	(4, 7, 4, 7)	0	inputSeq nav korekta vagoniņu virkne
10	(3, 4)	2	(1, 2) vai (2, 1)

Realizācijas detaļas

Jums jāiesniedz tieši viens fails ar nosaukumu `gondola.c`, `gondola.cpp` vai `gondola.pas`. Šajā failā jārealizē visas trīs apakšprogrammas (pat tad, ja Jūs plānojat risināt tikai daļu no visiem apakšuzdevumiem), kas aprakstītas iepriekš, ar norādītajām signatūrām. C/C++ programmas gadījumā Jums programmas tekstā jāiekļauj `gondola.h`.

C/C++ programma

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

Pascal programma

```
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;
```

Paraugtestētājs

Paraugtestētājs lasa ievadu šādā formātā:

- 1.rinda: T , apakšuzdevuma numurs, kuru jūsu programma mēģina risināt ($1 \leq T \leq 10$).
- 2.rinda: n , ievada virknes garums.
- 3.rinda: Ja T ir 4, 5 vai 6, šī rinda satur `gondolaSeq[0], ..., gondolaSeq[n-1]`. Citādi šī rinda satur `inputSeq[0], ..., inputSeq[n-1]`.