



Gondola(vagón)

La rueda Mao-Kong es una famosa atracción en Taipei. La rueda consiste en un carril circular, una estación, y n vagones numerados consecutivamente de 1 a n a lo largo del carril en una dirección fija. Inicialmente después de que el vagón i pasa la estación, el siguiente vagón a pasar será el vagón $i + 1$ si $i < n$, o el vagón 1 si $i = n$.

Los vagones se pueden averiar. Afortunadamente tenemos un suministro infinito de vagones de repuesto, el cual está numerado $n + 1$, $n + 2$, y así sucesivamente. Cuando un vagón se avería, lo reemplazamos (en la misma posición en el carril) con el primer vagón de repuesto disponible, es decir, el vagón con el número menor. Por ejemplo, si hay cinco vagones y el vagón uno 1 se avería, entonces lo reemplazaremos con el vagón 6.

A usted le gusta pararse en la estación y ver los vagones pasar. Una *secuencia de vagones válida* es una secuencia de n números de vagones que pasan la estación. Es posible que uno o más vagones se hayan averiado (y fueron reemplazados) antes de que usted llegara, pero ningún vagón se avería mientras usted está viendo.

Note que la misma configuración de vagones en el riel puede dar múltiples secuencias de vagones válidas, dependiendo de cual vagón pasa primero cuando usted llega a la estación. Por ejemplo, si ningún vagón se ha averiado, entonces ambas secuencias (2, 3, 4, 5, 1) y (4, 5, 1, 2, 3) son posibles secuencias de vagones válidas, pero (4, 3, 2, 5, 1) no lo es (porque los vagones aparecen en el orden equivocado).

Si el vagón 1 se avería, entonces podemos observar la secuencia de vagones válida (4, 5, 6, 2, 3). Si el vagón cuatro se avería después, lo reemplazamos con el vagón 7 y podremos observar la secuencia de vagones válida (6, 2, 3, 7, 5). Si el vagón 7 se avería después, lo reemplazamos con el vagón 8 y veremos la secuencia de vagones válida (3, 8, 5, 6, 2).

Vagón averiado	Nuevo vagón	Posible secuencia de vagones válida
1	6	(4, 5, 6, 2, 3)
4	7	(6, 2, 3, 7, 5)
7	8	(3, 8, 5, 6, 2)

Una secuencia de reemplazamiento es una secuencia consistente en el número de los vagones que se han averiado, en el orden en que se averiaron. En el ejemplo anterior, la secuencia de reemplazamiento es (1, 4, 7). Una secuencia de reemplazamiento r produce una secuencia de vagones válida g si, después de que los vagones se averían, de acuerdo a la secuencia de reemplazamiento r , la secuencia de vagones g puede ser observada.

Chequeo de secuencias de vagones

En las primeras tres subtarefas usted debe verificar si una secuencia de entrada es una secuencia de

vagones válida. Vea las tablas abajo por ejemplos de secuencias que son y que no son secuencias de vagones válidas. Usted necesita implementar la función `valid`.

- `valid(n, inputSeq)`
 - n : Tamaño de la secuencia de entrada
 - `inputSeq`: arreglo de tamaño n ; `inputSeq[i]` es el elemento i de la secuencia de entrada, para $0 \leq i \leq n - 1$.
 - La función debe retornar 1 si la secuencia de entrada es una secuencia de vagones válida, 0 de otro modo.

Subtareas 1, 2, 3

Subtarea	Puntos	n	<code>inputSeq</code>
1	5	$n \leq 100$	Tiene cada número de 1 a n exactamente una vez
2	5	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq n$
3	10	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 250,000$

Ejemplos

Subtarea	<code>inputSeq</code>	Valor de retorno	Nota
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 no puede aparecer justo antes de 5
1	(4, 3, 2, 1)	0	4 no puede aparecer justo antes de 3
2	(1, 2, 3, 4, 5, 6, 5)	0	Dos vagones numerados 5
3	(2, 3, 4, 9, 6, 7, 1)	1	Secuencia de reemplazamientos (5, 8)
3	(10, 4, 3, 11, 12)	0	4 no puede aparecer justo antes que 3

Secuencia de reemplazamiento

En las siguientes tres subtareas usted debe construir una posible secuencia de reemplazamientos que produce una secuencia de vagones dada. Cualquier secuencia de reemplazamiento válida será aceptada. Usted necesita implementar una función `replacement`.

- `replacement(n, gondolaSeq, replacementSeq)`
 - n es el tamaño de la secuencia de vagones.
 - `gondolaSeq`: Arreglo de tamaño n ; `gondolaSeq` es una secuencia de vagones válida, y `gondolaSeq[i]` es el elemento i de la secuencia, para $0 \leq i \leq n - 1$.

- La función deberá retornar l , la longitud de la secuencia de reemplazamiento.
- `replacementSeq`: Es un arreglo suficientemente largo para almacenar la secuencia de reemplazamiento; usted debe retornar la secuencia reemplazando el elemento i de su secuencia de reemplazamiento en `replacementSeq[i]`, para $0 \leq i \leq l - 1$.

Subtareas 4, 5, 6

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

Ejemplos

Subtarea	<code>gondolaSeq</code>	Valor de retorno	<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)

Contar las secuencias de reemplazamientos

En las siguientes cuatro subtareas usted debe contar el número de posibles secuencias de reemplazamiento que producen una secuencia dada (la cual puede o no puede ser una secuencia de vagones), la respuesta debe darse modulo **1,000,000,009**. Usted necesita implementar una función `countReplacement`.

- `countReplacement(n, inputSeq)`
 - n : Tamaño de la secuencia de entrada
 - `inputSeq`: Arreglo de tamaño n ; `inputSeq[i]` es el elemento i de la secuencia de entrada, para $0 \leq i \leq n - 1$.
- Si la secuencia de entrada es una secuencia de vagones válida, entonces debe contar el número de secuencias de reemplazamiento que producen dicha secuencia de vagones (el cual puede ser extremadamente largo), y *retornar este número módulo 1,000,000,009*. Si la secuencia de entrada no es una secuencia de vagones válida, la función deberá retornar 0. Si la secuencia de entrada es una secuencia de vagones, pero ningún vagón se averió, la función deberá retornar 1.

Subtareas 7, 8, 9, 10

Subtarea	Puntos	n	<code>inputSeq</code>
7	5	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq n + 3$

Subtarea	Puntos	n	$inputSeq$
8	15	$4 \leq n \leq 50$	$1 \leq inputSeq[i] \leq 100$, y al menos $n - 3$ de los vagones iniciales $1, \dots, n$ no se averiaron.
9	15	$n \leq 100,000$	$1 \leq inputSeq[i] \leq 250,000$
10	10	$n \leq 100,000$	$1 \leq inputSeq[i] \leq 1,000,000,000$

Ejemplos

Subtarea	$inputSeq$	Valor de retorno	Secuencia de reemplazamiento
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	$inputSeq$ No es una secuencia de reemplazamiento
10	(3, 4)	2	(1, 2) or (2, 1)

Detalles de implementación

Usted debe enviar exactamente un archivo, llamado `gondola.c`, `gondola.cpp` or `gondola.pas`. Este archivo deberá implementar las funciones descritas, usando los siguientes prototipos. Usted también necesita incluir un archivo de cabecera `gondola.h` para implementaciones en C/C++.

Programas en C/C++

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

Programas en 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;
```

Calificador ejemplo

El calificador de ejemplo lee la entrada en el siguiente formato:

- línea 1: T , el número de subtarea que su programa debe resolver ($1 \leq T \leq 10$).
- línea 2: n , la longitud de la secuencia de entrada .
- línea 3: Si T es 4, 5, or 6, esta línea contiene $gondolaSeq[0], \dots, gondolaSeq[n-1]$. de otro

modo contiene $\text{inputSeq}[0], \dots, \text{inputSeq}[n-1]$.