Gondola

Mao-Kong Gondola es una famosa atracción en Taipei. El sistema de gondola consiste de una riel circular, una sola estación, y n gondolas numeradas consecutivamente de 1 a n circulando alrededor de la riel en dirección fija. Inicialmente después de que la gondola i pase la estación, la siguiente gondola en pasar la estación será la gondola $i+1$ si $i < n$, o la gondola 1 si $i = n$.

Las gondolas pueden descomponerse. Afortunadamente tenemos infinitos suministros de respuesto de gondolas, los cuales son numerados $n+1$, $n+2$, y asi sucesivamente. Cuando una gondola se descompone podemos remplazarla (en su misma posición en la pista) con la primera gondola de repuesto, esto es, la primera con el menor numero. Por ejemplo, sí hay 5 gondolas y la gondola 1 se descompone, entonces podemos reemplazarla con la gondola 6.

Te gusta estar en la estación y mirar las gondolas pasar. Una *secuencia de gondola* es una serie de n números de gondolas que pasan la estación. Es posible que una o más gondolass se descompongan (y fueran reemplazadas) antes que lleges, pero ninguna de las gondolas se descomponen mientras estas observando.

Nota que la misma configuración de gondolas en la riel puede darte múltiples secuencias de gondola, dependiendo cual gondola pase primero cuando tu llegues a la estación. Por ejemplo, si ninguna de las gondolas se han descompuesto entonces (2, 3, 4, 5, 1) y (4, 5, 1, 2, 3) son secuencias de gondola, pero (4, 3, 2, 5, 1) no lo es (porque las gondolas aparecen en orden equivocado).

Sí la gondola 1 se descompone, entonces podriamos observar la secuenia de gondola (4, 5, 6, 2, 3). Si la siguiente gondola en descomponerse es la 4, la reemplazamos con la gondola 7 y podriamos observar la secuencia (6, 2, 3, 7, 5). Si la gondola 7 se descompone despues de esto, podemos reemplazarla con la gondola 8 y observamos la siguiente secuencia de gondola (3, 8, 5, 6, 2).

gondole descompuesta	nueva gondola	posible secuencia de gondola
1	6	(4, 5, 6, 2, 3)
4	7	(6, 2, 3, 7, 5)
7	8	(3, 8, 5, 6, 2)

Una *secuencia de reemplazo* es una secuencia que consiste de números de gondolas que ha sido descompuestas en el orden el cual ellas se descompusieron. En el anterior ejemplo la secuencia de reemplazo es (1, 4, 7). Una secuencia de reemplazo r *produce* una secuencia de gondola g si, las gondolas se descomponen de acuerdo a la secuencia de reemplazo r,la secuencia de gondola g puede ser observada.

Gondola Sequence Checking

En las primeras 3 subtareas debes verificar si una secuencia de entrada es una secuencia de gondola. Vea la tabla de abajo para ejemplos de secuencias que estan y no estan en la secuencia de gondola.

Necesitas implementar una función `valid`.

- `valid(n, inputSeq)`
  - `n`: La longitud de la secuencia.
  - `inputSeq`: array de longitud n; `inputSeq[i]` es el elmento i de la secuencia de entrada, para $0 \leq i \leq n-1$.
  - La función debería retornar 1 si la secuencia de entrada es una gondola de secuencia, 0 en otro caso.

Subtasks 1, 2, 3

subtask	points	n	`inputSeq`
1	5	$n \leq 100$	tiene cada número de 1 a n exactamente una vez
2	5	$n \leq 100,000$	$1 \leq$ `inputSeq[i]` $\leq n$
3	10	$n \leq 100,000$	$1 \leq$ `inputSeq[i]` $\leq 250,000$

Examples

subtask	`inputSeq`	return value	note
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 antes de 5
1	(4, 3, 2, 1)	0	4 no puede aparecer antes de 3
2	(1, 2, 3, 4, 5, 6, 5)	0	2 gondolas con numeradas con 5
3	(2, 3, 4, 9, 6, 7, 1)	1	secuencias reemplazadas (5, 8)
3	(10, 4, 3, 11, 12)	0	4 no puede aparecer antes de 3

Replacement Sequence

En las siguientes 3 subtareas tu debes construir una posible secuencia de reemplazo que produce una secuencia de gondola dada. CUalquier secuencia de reemplazo sera aceptada. Necesitas implementar una función `replacement`.

- `replacement(n, gondolaSeq, replacementSeq)`
  - `n` es la longitud de la secuencia de gondola.
  - `gondolaSeq`: array de longitud n; `gondolaSeq` se garantiza que es una secuencia de gondola, y `gondolaSeq[i]` es elemento i de la secuencia, para $0 \leq i \leq n-1$.
  - La función debe retornar l, la longitud de la secuencia de reemplazo.

- `replacementSeq`: Un array que es suficientemente largo para almacenar la recuencia de reemplazo; tu debes retornar tu secuencia colocando el elemento i de tu secuencia de reemplazo dentro `replacementSeq[i]`, para $0 \leq i \leq l-1$.

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

subtask	`gondolaSeq`	return value	`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)

Count Replacement Sequences

En las siguientes cuatro subtareas tu debes contar el número de posibles secuencias de reemplazo que produce una secuencia dada (la cual puede o no ser ua secuencia de gondola), modulo **1,000,000,009**. Necesitas implementar una función `countReplacement`.

- `countReplacement(n, inputSeq)`
  - `n`: la longitud de la secuencia de entrada.
  - `inputSeq`: array de longitud 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 gondola, entonces contar el número de secuencias de reemplazo que produce esta secuencia de gondola (el cual podria ser extremadamente largo), y retornar este número modulo **1,000,000,009**. Si la secuencia de entrada no es una secuencia de gondola, la función debe retornar 0. Si la secuencia de entrada es secuencia de gondola pero no gondolas descompuestas, la función debe retornar 1.

Subtasks 7, 8, 9, 10

subtask	points	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]` ≤ 100, and at least $n-3$ of the initial gondolas $1, \ldots, n$ did not break down.

subtask	points	n	inputSeq
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

subtask	inputSeq	return value	replacement sequence
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 is not a gondola sequence
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;
```

Sample grader

The sample grader reads the input in the following format:

- line 1: T, the subtask number your program intends to solve ($1 \leq T \leq 10$).
- line 2: n, the length of the input sequence.
- line 3: If T is 4, 5, or 6, this line contains inputSeq[0], ..., inputSeq[n-1]. Otherwise this line contains gondolaSeq[0], ..., gondolaSeq[n-1].