



## 纜車 (Gondola)

貓空纜車 (Mao-Kong Gondola) 是臺北市的一個著名景點。這個纜車系統包括一個環形軌道、一個纜車站和  $n$  個編號為 1 到  $n$  的纜車。這些纜車以固定的方向在軌道上循環運行。在纜車  $i$  經過纜車站之後，下一個經過纜車站的纜車將會是  $i+1$  ( $i < n$  時)，或者是纜車 1 ( $i = n$  時)。

纜車可能會發生故障。幸運的是，我們有無限多個後備的空閒纜車，其編號依次為  $n+1$ ,  $n+2$  等等。當某個纜車發生故障時，我們會在軌道上的同一位置用最前一個空閒纜車替換它，也就是說，編號最小的空閒纜車。舉個例子，如果當前有五輛纜車而纜車 1 發生了故障，那麼我們將用纜車 6 來替換它。

你喜歡去纜車站上觀察纜車過站。一個纜車序列 (gondola sequence) 是由纜車過站次序形成的  $n$  個纜車編號的序列。在你到達纜車站之前，有可能會有一到多個纜車發生故障（並且被替換掉），但是在你觀察過程中是不會有纜車發生故障的。

注意，在軌道上的相同一組纜車，有可能給出多種纜車序列，這取決於當你到纜車站時哪輛纜車最先過站。舉個例子，如果沒有任何纜車發生故障，那麼  $(2, 3, 4, 5, 1)$  和  $(4, 5, 1, 2, 3)$  都可能是纜車序列，但是  $(4, 3, 2, 5, 1)$  不可能是（因為纜車的次序有誤）。

如果纜車 1 發生故障，那麼我們可能會觀察到纜車序列  $(4, 5, 6, 2, 3)$ 。如果接著纜車 4 發生故障而我們用纜車 7 替換它，就有可能會觀察到纜車序列  $(6, 2, 3, 7, 5)$ 。如果纜車 7 在此後發生故障而我們用纜車 8 替換它，那麼現在就有可能會觀察到纜車序列  $(3, 8, 5, 6, 2)$ 。

故障纜車	新纜車	可能的纜車序列
1	6	$(4, 5, 6, 2, 3)$
4	7	$(6, 2, 3, 7, 5)$
7	8	$(3, 8, 5, 6, 2)$

一個替換序列 (replacement sequence) 是一個由故障纜車編號組成的序列，其次序與故障發生次序相同。在前面的例子中，替換序列是  $(1, 4, 7)$ 。如果一個替換序列  $r$  對應的故障發生後，我們由此有可能觀察到纜車序列  $g$ ，就稱替換序列  $r$  生成纜車序列  $g$ 。

### 纜車序列檢查

在前三個子任務中，你必須檢查某個輸入序列是否是一個纜車序列。下表舉例說明了哪些序列是纜車序列而哪些不是。你需要實現一個函數 `valid`。

- `valid(n, inputSeq)`
  - $n$ : 輸入序列的長度。
  - `inputSeq`: 大小為  $n$  的陣列；`inputSeq[i]` 是輸入序列的第  $i$  個元素，其中

$0 \leq i \leq n - 1$ 。

- 當輸入序列是一個纜車序列時，函數應返回 1，否則返回 0。

### 子任務 1, 2, 3

子任務	分值	$n$	inputSeq
1	5	$n \leq 100$	從 1 到 $n$ 的數字恰好各出現一次
2	5	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq n$
3	10	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 250,000$

### 例子

子任務	inputSeq	返回值	備註
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 不能恰好出現在 5 之前
1	(4, 3, 2, 1)	0	4 不能恰好出現在 3 之前
2	(1, 2, 3, 4, 5, 6, 5)	0	有兩個纜車編號都是 5
3	(2, 3, 4, 9, 6, 7, 1)	1	替換序列是 (5, 8)
3	(10, 4, 3, 11, 12)	0	4 不能恰好出現在 3 之前

## 替換序列

在接下來的三個子任務中，你必須構造一個能夠生成給定纜車序列的可能的替換序列。滿足條件的任意替換序列都可以。你需要實現一個函數 replacement。

- replacement( $n$ , gondolaSeq, replacementSeq)
  - $n$  是纜車序列的長度。
  - gondolaSeq: 大小為  $n$  的陣列；gondolaSeq 保證是一個纜車序列，而 gondolaSeq[ $i$ ] 是序列中的第  $i$  個元素，這裡  $0 \leq i \leq n - 1$ 。
  - 函數應返回替換序列的長度  $l$ 。
  - replacementSeq: 一個足夠大的能存下替換序列的陣列；你應當將替換序列中的第  $i$  個元素存放到 replacementSeq[ $i$ ] 做為返回結果，這裡  $0 \leq i \leq l - 1$ 。

### 子任務 4、5、6

子任務	分值	$n$	gondolaSeq
4	5	$n \leq 100$	$1 \leq \text{gondolaSeq}[i] \leq n + 1$
5	10	$n \leq 1,000$	$1 \leq \text{gondolaSeq}[i] \leq 5,000$

子任務	分值	$n$	<code>gondolaSeq</code>
6	20	$n \leq 100,000$	$1 \leq \text{gondolaSeq}[i] \leq 250,000$

## 例子

子任務	<code>gondolaSeq</code>	返回值	<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)

## 替換序列計數

在接下來的四個子任務中，你必須計算所有能夠生成給定序列（有可能是纜車序列，也有可能不是）的可能替換序列的數目，並將其對 **1,000,000,009** 取模。你需要實現一個函數 `countReplacement`。

- `countReplacement(n, inputSeq)`
  - $n$ : 輸入序列的長度。
  - `inputSeq`: 大小為  $n$  的陣列；`inputSeq[i]` 是輸入序列的第  $i$  個元素，這裡  $0 \leq i \leq n - 1$ 。
  - 如果輸入序列是一個纜車序列，則計算能夠生成該纜車序列的可能的替換序列的數目（有可能會非常大），然後將該數值對 **1,000,000,009** 取模做為返回值。如果輸入序列不是一個纜車序列，函數應返回 0。如果輸入序列是一個纜車序列，但是沒有纜車發生故障，函數應返回 1。

## 子任務 7、8、9、10

子任務	分值	$n$	<code>inputSeq</code>
7	5	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq n + 3$
8	15	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq 100$ , 初始纜車 $1, \dots, n$ 中至少有 $n - 3$ 不會發生故障。
9	15	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 250,000$
10	10	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 1,000,000,000$

## 例子

子任務	inputSeq	返回值	替換序列
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 不是一個纜車序列
10	(3, 4)	2	(1, 2) or (2, 1)

## 實現細節

你只能提交一個檔，名為 `gondola.c`、`gondola.cpp` 或者 `gondola.pas`。在該檔中應實現前面所述的三個函數（即便你只想解決其中的部分子任務，也要給出全部函數），並遵循下述命名與介面。對於 C/C++ 程式，你還需要包含標頭檔 `gondola.h`。

### C/C++ 程式

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

### Pascal 程式

```
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;
```

## 評測方式

評測系統將會讀入如下格式的輸入資料：

- 第1行:  $T$ ，你的程式需要解決的子任務編號 ( $1 \leq T \leq 10$ )。
- 第2行:  $n$ ，輸入序列的長度。
- 第3行: 如果  $T$  是 4、5 或者 6，此行包含 `gondolaSeq[0], ..., gondolaSeq[n-1]`。否則此行包含 `inputSeq[0], ..., inputSeq[n-1]`。