



## Gondola

สถานที่ท่องเที่ยวที่มีชื่อเสียงแห่งหนึ่งของไต้หวันคือกระเช้าลอยฟ้าเมาดง ระบบกระเช้าลอยฟ้าประกอบด้วยรางต่อเป็นวงกลม, สถานีจำนวนหนึ่งสถานี และกระเช้าจำนวน  $n$  กระเช้าที่มีหมายเลขกำกับตั้งแต่ 1 ถึง  $n$  กระเช้าเหล่านี้จะวิ่งวนไปตามรางในทิศเดียวกัน หลังจากทีกระเช้าหมายเลข  $i$  วิ่งผ่านสถานี กระเช้าถัดไปที่จะวิ่งผ่านสถานีจะเป็นกระเช้าหมายเลข  $i + 1$  เมื่อ  $i < n$  หรือกระเช้าหมายเลข 1 ถ้า  $i = n$

กระเช้านั้นอาจจะพังก็ได้ แต่โชคดีที่เรามีกระเช้าสำรองจำนวนไม่จำกัด ซึ่งมีหมายเลขกำกับตั้งแต่  $n + 1, n + 2$  ไปเรื่อย ๆ ทันทีที่กระเช้าพังลง เราจะเปลี่ยนกระเช้าดังกล่าวด้วยกระเช้าสำรองแรกสุดที่มีอยู่ ซึ่งก็คือกระเช้าที่มีหมายเลขน้อยสุดนั่นเอง ยกตัวอย่างเช่น ถ้ามีกระเช้าอยู่ 5 กระเช้า และกระเช้าหมายเลข 1 พังลง เราจะเปลี่ยนเอากระเช้าหมายเลข 6 เข้ามาแทน

คุณชอบที่จะยืนอยู่ที่สถานีและมองดูกระเช้าวิ่งผ่านไป ลำดับกระเช้าคือลำดับจำนวน  $n$  ตัวที่ประกอบด้วยหมายเลขของกระเช้าที่วิ่งผ่านสถานี มันเป็นไปได้ที่กระเช้าหนึ่งตัวหรือมากกว่าพังลงและถูกเปลี่ยนไปก่อนที่คุณจะมาถึงสถานี แต่จะไม่มีกระเช้าใดพังเลยในขณะที่คุณกำลังดูอยู่

ให้สังเกตว่าลำดับกระเช้าหลายรูปแบบสามารถเกิดได้จากกระเช้าที่เรียงตัวแบบเดียวกัน โดยขึ้นอยู่กับว่ากระเช้าแรกที่วิ่งผ่านสถานีขณะที่คุณไปถึงคือกระเช้าใด ตัวอย่างเช่น เมื่อไม่มีกระเช้าใดพังเลย ลำดับกระเช้า (2, 3, 4, 5, 1) และ (4, 5, 1, 2, 3) นั้นถือเป็นลำดับกระเช้าที่เป็นไปได้ แต่ (4, 3, 2, 5, 1) ไม่ใช่ลำดับกระเช้าที่เป็นไปได้ (เนื่องจากกระเช้าปรากฏขึ้นมาผิดลำดับ)

ถ้ากระเช้าหมายเลข 1 เกิดพังลง ลำดับกระเช้าลำดับหนึ่งที่เป็นไปได้คือ (4, 5, 6, 2, 3) หลังจากนั้นถ้ากระเช้าหมายเลข 4 พังลง ลำดับกระเช้าลำดับหนึ่งที่เป็นไปได้คือ (6, 2, 3, 7, 5) หลังจากนั้นถ้ากระเช้าหมายเลข 7 พังลง เราจะแทนที่กระเช้าหมายเลข 7 ด้วยกระเช้าหมายเลข 8 ลำดับกระเช้าลำดับหนึ่งที่เป็นไปได้คือ (3, 8, 5, 6, 2)

กระเช้าที่พัง	กระเช้าใหม่	ลำดับกระเช้าที่เป็นไปได้ลำดับหนึ่ง
1	6	(4, 5, 6, 2, 3)
4	7	(6, 2, 3, 7, 5)
7	8	(3, 8, 5, 6, 2)

ลำดับการเปลี่ยนคือลำดับหมายเลขของกระเช้าที่พังลงโดยเรียงตามเวลาที่พัง ในตัวอย่างข้างต้น ลำดับการเปลี่ยนคือ (1, 4, 7) ลำดับการเปลี่ยน  $r$  นั้นจะทำให้เกิดลำดับกระเช้า  $g$  ถ้าหลังจากที่กระเช้าพังลงตามลำดับการเปลี่ยน  $r$  แล้ว ลำดับกระเช้า  $g$  เป็นลำดับกระเช้าที่เป็นไปได้จากการมองกระเช้า

### การตรวจสอบลำดับกระเช้า

ในงานย่อยสามงานแรก คุณจะต้องตรวจสอบว่าลำดับที่ได้มานั้นเป็นลำดับกระเช้าหรือไม่ ดูตัวอย่างในตารางต่อไปสำหรับตัวอย่างของลำดับที่เป็นและไม่เป็นลำดับกระเช้า คุณจะต้องเขียนฟังก์ชัน **valid**

- **valid(n, inputSeq)**
  - $n$ : ความยาวของลำดับข้อมูลเข้า
  - **inputSeq**: อาร์เรย์ความยาว  $n$  โดยที่ **inputSeq[i]** คือสมาชิกตัวที่  $i$  ของลำดับข้อมูลเข้า สำหรับ  $0 \leq i \leq n - 1$
  - ฟังก์ชันจะต้องคืนค่า 1 ถ้าลำดับข้อมูลเข้านั้นเป็นลำดับกระเช้า และคืนค่า 0 ในกรณีอื่น

### งานย่อย 1, 2, 3

งานย่อย	คะแนน	$n$	inputSeq
1	5	$n \leq 100$	ประกอบด้วยตัวเลขตั้งแต่ 1 ถึง $n$ ทุกตัว และตัวเลขแต่ละตัวปรากฏตัวละ 1 ครั้งเท่านั้น
2	5	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq n$
3	10	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 250,000$

### ตัวอย่าง

งานย่อย	inputSeq	ค่าที่คืน	หมายเหตุ
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 ไม่สามารถปรากฏก่อน 5 ในลำดับที่ติดกันได้
1	(4, 3, 2, 1)	0	4 ไม่สามารถปรากฏก่อน 3 ในลำดับที่ติดกันได้
2	(1, 2, 3, 4, 5, 6, 5)	0	มีกระเช้าหมายเลข 5 สองอัน
3	(2, 3, 4, 9, 6, 7, 1)	1	ลำดับการเปลี่ยนคือ (5, 8)

3	(10, 4, 3, 11, 12)	0	4 ไม่สามารถปรากฏก่อน 3 ในลำดับที่ติดกันได้
---	--------------------	---	--

## ลำดับการเปลี่ยน

ในงานย่อยสามงานถัดมา คุณจะต้องสร้างลำดับการเปลี่ยนที่ทำให้เกิดลำดับกระเช้าที่ให้มา ลำดับการเปลี่ยนใด ๆ ที่ทำให้เกิดลำดับกระเช้าที่ให้มาจะถือว่าใช้ได้ คุณจะต้องเขียนฟังก์ชัน **replacement**

- **replacement(n, gondolaSeq, replacementSeq)**
  - **n**: คือความยาวของลำดับกระเช้า
  - **gondolaSeq**: อาร์เรย์ความยาว  $n$  โดยที่เรารับประกันว่า **gondolaSeq** จะเป็นลำดับกระเช้า และ **gondolaSeq[i]** คือสมาชิกตัวที่  $i$  ของลำดับกระเช้า สำหรับ  $0 \leq i \leq n - 1$
  - ฟังก์ชันจะต้องคืนค่า  $l$  ซึ่งก็คือความยาวของลำดับการเปลี่ยน
  - **replacementSeq**: อาร์เรย์ที่มีขนาดเพียงพอที่จะเก็บลำดับการเปลี่ยน คุณจะต้องคืนลำดับการเปลี่ยนของคุณโดยนำเอาสมาชิกตัวที่  $i$  ของลำดับการเปลี่ยนของคุณลงใน **replacementSeq[i]** สำหรับ  $0 \leq i \leq l - 1$

## งานย่อย 4, 5, 6

งานย่อย	คะแนน	$n$	<b>gondolaSeq</b>
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	$n \leq 100,000$	$1 \leq \text{gondolaSeq}[i] \leq 250,000$

## ตัวอย่าง

งานย่อย	<b>gondolaSeq</b>	ค่าที่คืน	<b>replacementSeq</b>
4	(3, 1, 4)	1	(2)
4	(5, 1, 2, 3, 4)	0	( )
5	(2, 3, 4, 9, 6, 7, 1)	2	(5, 8)

## นับจำนวนลำดับการเปลี่ยน

ในงานย่อยอีก 4 งานถัดมา คุณจะต้องนับจำนวนของลำดับการเปลี่ยนที่เป็นไปได้ทั้งหมดที่ทำให้เกิดลำดับที่ให้มา (ซึ่งอาจจะเป็นลำดับกระเช้าหรือไม่เป็นก็ได้) โดยให้หารเอาเศษด้วย **1,000,000,009** คุณจะต้องเขียนฟังก์ชัน **countReplacement**

- **countReplacement(n, inputSeq)**
  - **n**: คือความยาวของลำดับข้อมูลเข้า
  - **inputSeq**: อาร์เรย์ความยาว  $n$  **inputSeq[i]** คือสมาชิกตัวที่  $i$  ของลำดับข้อมูลเข้า สำหรับ  $0 \leq i \leq n - 1$
  - ถ้าลำดับข้อมูลเข้าเป็นลำดับกระเช้า ให้นับจำนวนของลำดับการเปลี่ยนทั้งหมดที่ทำให้เกิดลำดับกระเช้านี้ (ซึ่งอาจจะมีขนาดใหญ่มาก) และให้คืนจำนวนดังกล่าวหารเอาเศษด้วย **1,000,000,009** ถ้าลำดับข้อมูลเข้าไม่ใช่ลำดับกระเช้า ฟังก์ชันจะต้องคืนค่า 0 ถ้าข้อมูลนำเข้าเป็นลำดับกระเช้าแต่ไม่มีกระเช้าใดพังเลยให้คืนค่า 1

งานย่อย 7, 8, 9, 10

งานย่อย	คะแนน	$n$	<b>gondolaSeq</b>
7	5	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq n + 3$
8	15	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq 100$ และอย่างน้อยกระเช้าจำนวน $n - 3$ กระเช้าจากกระเช้าเริ่มต้นหมายเลข $1, \dots, n$ ไม่พัง
9	15	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 250,000$
10	10	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 1,000,000,000$

ตัวอย่าง

งานย่อย	<b>inputSeq</b>	ค่าที่คืน	<b>replacementSeq</b>
7	(1, 2, 7, 6)	2	(3, 4, 5) หรือ (4, 5, 3)
8	(2, 3, 4, 12, 6, 7, 1)	1	(5, 8, 9, 10, 11)
9	(4, 7, 4, 7)	0	<b>inputSeq</b> ไม่ใช่ลำดับกระเช้า
10	(3, 4)	2	(1, 2) หรือ (2, 1)

## รายละเอียดการเขียนโปรแกรม

คุณจะต้องส่งแฟ้มหนึ่งแฟ้มในชื่อ `gondola.c`, `gondola.cpp` หรือ `gondola.pas` โดยที่แฟ้มนี้จะต้องเขียนโปรแกรมย่อยดังที่กล่าวไว้ข้างบนทั้งหมด (ถึงแม้ว่าคุณตั้งใจจะทำเพียงแค่บางงานย่อยก็ตาม) โดยใช้รูปแบบดังต่อไปนี้ และคุณจำเป็นต้อง include header `gondola.h` ในแฟ้มดังกล่าวด้วยถ้าคุณใช้ภาษา C/C++

### โปรแกรมภาษา C/C++

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

### โปรแกรมภาษา 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;
```

### เกรตเตอร์ตัวอย่าง

เกรตเตอร์ตัวอย่างจะอ่านข้อมูลนำเข้าดังรูปแบบต่อไปนี้

- บรรทัดที่ 1:  $T$  หมายเลขงานย่อยที่โปรแกรมของคุณจะต้องคำนวณ ( $1 \leq T \leq 10$ )
- บรรทัดที่ 2:  $n$  ความยาวของลำดับข้อมูลเข้า
- บรรทัดที่ 3: ถ้า  $T$  มีค่าเป็น 4, 5 หรือ 6 บรรทัดนี้จะมีข้อมูล `gondolaSeq[0], ..., gondolaSeq[n-1]` ในกรณีที่  $T$  มีค่าอื่น ๆ บรรทัดนี้จะมีข้อมูล `inputSeq[0], ..., inputSeq[n-1]`