

การเรียงสับเปลี่ยน (Permutation)

เหล่าฟาโรห์ใช้การเคลื่อนไหวจีนฮัมพัทธ์และแรงโน้มถ่วงของดาวเคราะห์เพื่อที่จะเร่งยานอวกาศของพวกเขา สมมติว่ายานอวกาศจะผ่านเข้าใกล้ดาวเคราะห์จำนวน n ดวง ที่มีความเร็ววงโคจร $p[0], p[1], \dots, p[n-1]$ ตามลำดับ สำหรับดาวเคราะห์แต่ละดวง นักวิทยาศาสตร์ฟาโรห์สามารถเลือกได้ว่าจะใช้ดาวเคราะห์ดวงนั้นในการเร่งความเร็วหรือไม่ เพื่อประหยัดพลังงาน หลังจากการเร่งด้วยดาวเคราะห์ที่มีความเร็ววงโคจร $p[i]$ แล้ว ยานอวกาศจะไม่สามารถเร่งความเร็วโดยใช้ดาวเคราะห์อื่นที่มีความเร็ววงโคจร $p[j] < p[i]$ กล่าวอีกทางหนึ่งก็คือ ดาวเคราะห์ที่เลือกจะต้องทำให้ลำดับของความเร็ววงโคจรเรียงเป็น ลำดับย่อยที่เพิ่มขึ้น ของ $p[0], p[1], \dots, p[n-1]$ ลำดับย่อยของ p คือลำดับที่ได้มาจาก p โดยการลบสมาชิกศูนย์ตัวหรือมากกว่านั้นออกจาก p ยกตัวอย่างเช่น $[0]$, $[], [0, 2]$, และ $[0, 1, 2]$ ล้วนแต่เป็นลำดับย่อยของลำดับ $[0, 1, 2]$ แต่ $[2, 1]$ นั้นไม่ใช่

นักวิทยาศาสตร์ได้ค้นพบว่ารวมแล้วมีรูปแบบในการเลือกเซตของดาวเคราะห์จำนวน k รูปแบบในการเร่งความเร็วยานอวกาศ แต่พวกเขาได้สูญเสียข้อมูลเกี่ยวกับความเร็ววงโคจรไปจนหมดสิ้น (กระทั่งค่า n) อย่างไรก็ตาม พวกเขาจำได้ว่า $(p[0], p[1], \dots, p[n-1])$ เป็นการเรียงสับเปลี่ยน (permutation) ของ $0, 1, \dots, n-1$ การเรียงสับเปลี่ยนคือลำดับที่ประกอบด้วยจำนวนเต็มที่มีค่าระหว่าง 0 ถึง $n-1$ ตัวละหนึ่งครั้ง งานของคุณคือการหาลำดับการเรียงสับเปลี่ยน $p[0], p[1], \dots, p[n-1]$ ที่เป็นไปได้โดยต้องการให้มีขนาดที่สั้นพอ

คุณจะต้องแก้ปัญหาสำหรับยานอวกาศจำนวน q ลำ สำหรับยานอวกาศลำที่ i คุณจะได้อันดับเต็ม k_i แทนจำนวนรูปแบบที่แตกต่างกันในการเลือกเซตของดาวเคราะห์เพื่อใช้เร่งความเร็วยานอวกาศ งานของคุณคือการหาลำดับของความเร็ววงโคจรของดาวเคราะห์ที่มีความยาว n_i ที่มีค่าไม่มากเกินไป โดยมีจำนวนวิธีในการเลือกลำดับย่อยที่เพิ่มขึ้นนั้นมีจำนวนเท่ากับ k_i พอดี

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้:

```
int[] construct_permutation(int64 k)
```

- k : จำนวนของลำดับย่อยที่เพิ่มขึ้นที่ต้องการ
- ฟังก์ชันนี้จะต้องคืนอาร์เรย์ที่มีจำนวนสมาชิก n ค่า โดยที่สมาชิกแต่ละตัวจะมีค่าระหว่าง 0 ถึง $n-1$ (รวมค่าขอบด้วย)
- อาร์เรย์ที่คืนมาจะต้องเป็นการเรียงสับเปลี่ยน (permutation) ที่ถูกต้องและจะต้องมีลำดับเพิ่มขึ้นจำนวน k ลำดับพอดี
- ฟังก์ชันนี้จะถูกเรียกรวมทั้งสิ้น q ครั้ง แต่ละครั้งสามารถพิจารณาได้ว่าเป็นคนละสถานการณ์กัน

เงื่อนไข

- $1 \leq q \leq 100$
- $2 \leq k_i \leq 10^{18}$ (สำหรับทุก ๆ $0 \leq i \leq q - 1$)

ปัญหาย่อย

1. (10 points) $2 \leq k_i \leq 90$ (สำหรับทุก ๆ $0 \leq i \leq q - 1$) ถ้าทุก ๆ การเรียงสับเปลี่ยนที่คุณใช้มีความยาวไม่เกิน 90 และถูกต้อง คุณจะได้คะแนน 10 คะแนน ไม่เช่นนั้นจะได้ 0
2. (90 points) ไม่มีเงื่อนไขเพิ่มเติมจากโจทย์ สำหรับปัญหาย่อยนี้ ให้ m แทนความยาวของการเรียงสับเปลี่ยน (permutation) ที่คุณใช้ในแต่ละสถานการณ์ใด ๆ คะแนนของคุณจะถูกคำนวณตามตารางด้านล่างนี้:

เงื่อนไข	คะแนน
$m \leq 90$	90
$90 < m \leq 120$	$90 - \frac{(m-90)}{3}$
$120 < m \leq 5000$	$80 - \frac{(m-120)}{65}$
$m > 5000$	0

ตัวอย่าง

ตัวอย่าง 1

พิจารณาการเรียกฟังก์ชันต่อไปนี้:

```
construct_permutation(3)
```

ฟังก์ชันนี้จะต้องคืนการเรียงสับเปลี่ยนที่มีลำดับย่อยที่เพิ่มขึ้นจำนวน 3 ลำดับพอดี คำตอบหนึ่งที่เป็นไปได้คือ $[1, 0]$, ที่มี $[]$ (ลำดับย่อยว่าง), $[0]$ และ $[1]$ เป็นลำดับย่อยที่เพิ่มขึ้น

ตัวอย่าง 2

พิจารณาการเรียกฟังก์ชันต่อไปนี้:

```
construct_permutation(8)
```

ฟังก์ชันนี้จะต้องคืนการเรียงสับเปลี่ยนที่มีลำดับย่อยเพิ่มขึ้นจำนวน 8 ลำดับพอดี คำตอบหนึ่งที่เป็นไปได้คือ $[0, 1, 2]$

เกรดเดอร์ตัวอย่าง

เกรดเดอร์ตัวอย่างจะอ่านข้อมูลนำเข้าในรูปแบบต่อไปนี้:

- บรรทัดที่ 1: q

- บรรทัดที่ $2 + i$ ($0 \leq i \leq q - 1$): k_i

สำหรับแต่ละ k_i เกรดเดอร์ตัวอย่างจะพิมพ์ผลลัพธ์หนึ่งบรรทัดแทนค่าที่คืนจากฟังก์ชัน `construct_permutation` หรือข้อความแสดงความผิดพลาดถ้าเกิดขึ้น