สลับและนับ

มีนักเรียน N คน ($2 \leq N \leq 100\,000$) ยืนเข้าแถวต่อกัน ตำแหน่งในแถวจะเรียกเป็นตำแหน่งที่ 0 ถึงตำแหน่งที่ $N-1$ โดยที่ตำแหน่งที่ 0 คือหัวแถวและตำแหน่งที่ $N-1$ เป็นท้ายแถว

นักเรียนคนที่ i มีความสูงเป็นอันดับที่ $N-i+1$ สำหรับ $1 \leq i \leq N$ นั่นคือนักเรียนคนที่ 1 นั้นเตี้ยที่สุด ส่วนนักเรียนคนที่ N นั้นสูงที่สุด ไม่มีนักเรียนคู่ใดที่สูงเท่ากัน

ในแต่ละขณะเวลา นักเรียนคนที่ i จะสามารถสังเกตได้ว่าคนที่อยู่ด้านหน้าในแถวของเธอมีกี่คนที่สูงกว่าเธอ คุณสามารถสอบถามนักเรียนทั้งกลุ่มถึงผลรวมของค่าดังกล่าว ยกตัวอย่างเช่น สมมติว่า $N = 5$ และถ้าในปัจจุบันนักเรียนเข้าแถวตามลำดับดังนี้

$$4, 1, 3, 5, 2$$

นักเรียนคนที่ 4 จะไม่เห็นคนที่สูงกว่าอยู่ด้านหน้าเลย ส่วนนักเรียนคนที่ 1 จะเห็น 1 คน นักเรียนคนที่ 3 จะเป็น 1 คน นักเรียนคนที่ 5 ไม่เห็นเลย สุดท้ายนักเรียนคนที่ 2 จะเห็นคนที่สูงกว่า 3 คน ดังนั้นเมื่อคุณถามสถานะของการเข้าแถว คุณจะได้คำตอบเป็น $0 + 1 + 1 + 0 + 3 = 5$ เป็นต้น

คุณอยากทราบลำดับว่านักเรียนยืนเรียนกันอย่างไร ด้วยข้อมูลดังกล่าวคุณยังไม่สามารถบอกได้ ยกเว้นได้คำตอบเป็น 0 หรือ $N(N-1)/2$ คุณจึงบอกให้นักเรียนคนที่อยู่ที่ตำแหน่งต่าง ๆ สลับตำแหน่งกันและสอบถามค่าแสดงสถานะของแถวดังกล่าว

ยกตัวอย่างเช่น ถ้าคุณให้นักเรียนที่ยืนที่ลำดับที่ 1 สลับกับนักเรียนตำแหน่งที่ 3 ผลลัพธ์ของนักเรียนในแถวจะเป็น

$$4, 5, 3, 1, 2$$

เมื่อคุณถามค่าสถานะของแถวจะได้คำตอบเป็น $0 + 0 + 2 + 3 + 3 = 8$

ให้เขียนโปรแกรมที่รับ N และค่าสถานะเริ่มต้นของแถว จากนั้นให้คุณให้นักเรียนสลับตำแหน่งกันเพื่อถามค่าสถานะของแถวจนกระทั่งสามารถตอบลำดับการเข้าแถวเริ่มต้นได้ คุณถามได้ไม่เกิน $Q \leq 100\,000$ ครั้ง คะแนนที่คุณได้จะขึ้นกับจำนวนครั้งของการถามของคุณ

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้

```
vector find_permutation(int N, long long V)
```

- ฟังก์ชันนี้จะถูกเรียกหนึ่งครั้ง โดยที่ N ระบุจำนวนนักเรียน และ V แทนค่าสถานะของแถวเมื่อเริ่มต้น
- ฟังก์ชันจะต้องคืนค่าลำดับของแถวตั้งต้น

คุณสามารถเรียกฟังก์ชันต่อไปนี้เพื่อสลับตำแหน่งนักเรียนในแถวและอ่านค่าสถานะของแถวหลังสลับ

```
long long swap_and_report(int i, int j)
```

- ฟังก์ชันนี้จะสลับนักเรียนในตำแหน่งที่ i กับตำแหน่งที่ j
- และจะคืนค่าสถานะของแถวหลังการสลับ
- คุณสามารถเรียกฟังก์ชันได้ไม่เกิน Q ครั้ง
- ในเกรดเดอร์จริง ฟังก์ชันดังกล่าวถูกเขียนให้มีประสิทธิภาพดีพอที่จะรองรับการทำงานให้ทำได้ตามเวลา $O(\log^2 N)$

ขอบเขต

- $2 \leq N \leq 100\,000$
- $N \leq Q \leq 100\,000$

ปัญหาย่อย

1. (5 คะแนน) $N \leq 8, Q = 100\,000$
2. (6 คะแนน) $N \leq 8, Q = 64$
3. (8 คะแนน) $N \leq 8, Q = N$
4. (7 คะแนน) $N \leq 300, Q = 100\,000$
5. (10 คะแนน) $N \leq 300, Q = 1\,000$
6. (11 คะแนน) $N \leq 300, Q = N$
7. (13 คะแนน) $N \leq 50\,000, Q = 100\,000$
8. (14 คะแนน) $N \leq 50\,000, Q = N$
9. (26 คะแนน) $N \leq 100\,000, Q = 100\,000$

ตัวอย่าง

จากตัวอย่างด้านบน ที่ $N = 5$ และนักเรียนเข้าแถวตามลำดับดังนี้

$$4, 1, 3, 5, 2$$

สังเกตว่าค่าสถานะคือ 5 ดังนั้น เกรดเดอร์จะเรียกฟังก์ชัน

```
find_permutation(5, 5)
```

ฟังก์ชันดังกล่าว จะเรียก `swap_and_report` เพื่อสลับตำแหน่งนักเรียนคนที่ 1 กับ 3

```
swap_and_report(1,3)
```

ซึ่งจะได้ค่าคืนกลับมาคือ 8 และลำดับของแถวเปลี่ยนเป็น

$$4, 5, 3, 1, 2$$

ถ้าเรียก

```
swap_and_report(0,4)
```

จะได้ค่าคืนกลับมาคือ $0 + 0 + 1 + 3 + 1 = 5$ และลำดับของแถวเปลี่ยนเป็น

$$2, 5, 3, 1, 4$$

ฟังก์ชัน `find_permutation` ที่ทำงานถูกต้องจะต้องคืนค่า

```
[4, 1, 3, 5, 2]
```

เกรดเดอร์ตัวอย่าง

จะมีทั้งหมด 2 บรรทัด:

- บรรทัด 1: N
- บรรทัด 2: $A[0]\ \ A[1]\ \ A[2]\ \ \ldots\ \ A[N-1]$

เกรดเดอร์ตัวอย่างจะแสดงผลจำนวนครั้งที่เรียก `swap_and_report` ถ้าคำตอบไม่ถูกต้อง จะแสดงผล `Wrong Answer`

ข้อจำกัด

- Time limit: 1 second
- Memory limit: 1 GB