



ขบวนแมวโคราช

(1 วินาที, 512 MB)

ในงานเกษตรสุนารีที่จัดขึ้นโดยมหาวิทยาลัยเทคโนโลยีสุนารี (มทส.) นอกจากจะมีการจัดแสดงนิทรรศการผลงานวิจัย นวัตกรรม และสิ่งประดิษฐ์เด่นของ มทส. แล้ว ยังมีกิจกรรมการประกวดแมวโคราช หรือที่เรียกกันว่า แมวสีสวาด หรือแมวมาเลศ ซึ่งจัดเป็นหนึ่งในแมวมงคล 17 ชนิดตามตำราแมวคำกลอน และเป็นแมวไทยพันธุ์โบราณพื้นเมืองจังหวัดนครราชสีมา

คุณต้องการนำแมวโคราช จำนวนอย่างน้อย $\lceil N/2 \rceil$ ตัว มาเรียงแถวเป็นขบวนในพิธีเปิด (โดย $\lceil N/2 \rceil$ นิยามถึงจำนวนเต็มที่น้อยที่สุดซึ่งมีค่าไม่ต่ำกว่ากึ่งหนึ่งของจำนวนแมวทั้งหมดในฟาร์ม) ซึ่งคุณจะเลือกมาจากฟาร์มที่มีแมวอยู่ทั้งหมด N ตัว ทั้งนี้ แมวแต่ละตัวถูกกำหนด **ระดับความสวยงาม** เป็นเลขจำนวนเต็มตั้งแต่ 1 ถึง N ไม่ซ้ำกัน โดยที่แมวหมายเลข i จะมีระดับความสวยงามเท่ากับ $P[i]$ เมื่อ $i = 0, 1, \dots, N - 1$

เพื่อความสวยงามของขบวนแมว คุณต้องการให้ลำดับของแมวอย่างน้อย $\lceil N/2 \rceil$ ตัวที่เลือกมานั้นมีระดับความสวยงามเรียงต่อเนื่องกันทั้งหมด กล่าวคือ แมวแต่ละตัวในลำดับที่เลือกมาจะต้องมีระดับความสวยงาม **เพิ่มขึ้นทีละ 1 หรือ ลดลงทีละ 1** ตลอดทั้งขบวน หรือกล่าวอย่างเป็นทางการได้ว่า ถ้าเลือกแมวมา M ตัว และลำดับหมายเลขของแมวที่เลือกมาเป็น $c[0], c[1], \dots, c[M - 1]$ แล้ว จะได้ว่า

- $P[c[k]] = P[c[k - 1]] + 1$ โดยที่ $k = 1, 2, \dots, M - 1$

หรือ

- $P[c[k]] = P[c[k - 1]] - 1$ โดยที่ $k = 1, 2, \dots, M - 1$

อย่างไรก็ดี เจ้าของฟาร์มจำไม่ได้ว่าแมวแต่ละตัวมีระดับความสวยงามเป็นเท่าใด แต่สามารถบอกคุณได้เพียงว่าสำหรับแมวหมายเลข i และ j ใด ๆ แมวคู่นั้นมีระดับความสวยงามแตกต่างกันเท่าใด กล่าวอีกอย่างได้ว่า สำหรับแมวหมายเลข i และ j ใด ๆ เจ้าของฟาร์มสามารถบอกค่า $|P[i] - P[j]|$ ได้ และคุณสามารถถามข้อมูลนี้จากเจ้าของฟาร์มได้ไม่เกิน Q ครั้ง

งานของคุณคือการเขียนโปรแกรมเพื่อสร้างขบวนแมวหรือลำดับหมายเลขของแมวอย่างน้อย $\lceil N/2 \rceil$ ตัว ตามเงื่อนไขดังกล่าว

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้ในการจัดขบวนแถว

```
vector<int> select(int N)
```

- N : จำนวนแถวทั้งหมด
- ฟังก์ชันนี้จะคืนค่าเวกเตอร์ c ซึ่งเป็นลำดับหมายเลขของแถวที่ถูกเลือก

โดยระบบตรวจจะถือว่าคำตอบถูกต้องเมื่อ c มีจำนวนสมาชิกเป็น $M \geq \lceil N/2 \rceil$ และ

- $P[c[k]] = P[c[k-1]] + 1$ โดยที่ $k = 1, 2, \dots, M-1$

หรือ

- $P[c[k]] = P[c[k-1]] - 1$ โดยที่ $k = 1, 2, \dots, M-1$

นอกจากนี้ ฟังก์ชัน `select` ของคุณยังสามารถเรียกใช้ฟังก์ชันต่อไปนี้ได้

```
int ask(int i, int j)
```

- i, j : หมายเลขของแถว ($0 \leq i, j \leq N-1$)
- ฟังก์ชันนี้จะคืนค่าเป็น $|P[i] - P[j]|$
- คุณสามารถเรียกใช้ฟังก์ชันนี้ได้ไม่เกิน Q ครั้ง หากเรียกเกิน Q ครั้ง จะได้ผลการตรวจเป็น Not correct

สำคัญ: ในบางชุดทดสอบ ฟังก์ชัน `ask` อาจมีพฤติกรรมแบบ adaptive กล่าวคือ ในการเรียกฟังก์ชัน `ask` แต่ละครั้ง ฟังก์ชันจะเลือกใช้ P ใดก็ได้ที่สอดคล้องกับคำตอบของ `ask` ก่อนหน้าทั้งหมด

ตัวอย่าง

ตัวอย่างที่ 1: กรณีที่ฟังก์ชัน `ask` ไม่มีพฤติกรรมแบบ adaptive

สมมติให้ $N = 5$ พิจารณาสถานการณ์ที่ $P = [3, 5, 4, 1, 2]$ โดยที่ $P[0] = 3, P[1] = 5, P[2] = 4, P[3] = 1, P[4] = 2$ ถ้าเราเรียกฟังก์ชัน `ask` เพื่อถามความแตกต่างระหว่างแถวหมายเลข 0 กับแถวหมายเลข 1, 2, 3 และ 4 ตามลำดับ จะได้ผลดังตารางต่อไปนี้

การเรียกฟังก์ชัน	คำตอบที่ได้
<code>ask(0, 1)</code>	2
<code>ask(0, 2)</code>	1
<code>ask(0, 3)</code>	2
<code>ask(0, 4)</code>	1

ในกรณีนี้ ($N = 5$) โปรแกรมของคุณต้องคืนค่าลำดับหมายเลขของแมวที่ถูกเลือกอย่างน้อย 3 ตัวที่มีระดับความสวยงามเรียงกัน จึงจะถือว่าสร้างขบวนแมวได้ตามเงื่อนไข ทั้งนี้ ตัวอย่างหนึ่งของผลลัพธ์จากการเรียกฟังก์ชัน `select` คือ `[3, 4, 0]` ซึ่งหมายถึงสร้างขบวนแมวโดยเรียงลำดับจากระดับความสวยงามน้อยไปหามาก ($P[3] = 1, P[4] = 2, P[0] = 3$)

ตัวอย่างที่ 2: กรณีที่ฟังก์ชัน `ask` มีพฤติกรรมแบบ adaptive

สมมติให้ $N = 5$ พิจารณาการเรียกฟังก์ชัน `ask` เพื่อถามความแตกต่างระหว่างแมวหมายเลข 0 กับแมวหมายเลข 1, 2 และ 3 ตามลำดับ อาจได้ผลดังตารางต่อไปนี้

การเรียกฟังก์ชัน	คำตอบที่ได้	ตัวอย่าง P ที่เป็นไปได้
<code>ask(0, 1)</code>	2	<code>[1, 3, 2, 4, 5], [1, 3, 4, 5, 2], [2, 4, 1, 5, 3], [3, 5, 4, 1, 2], [3, 1, 2, 5, 4], ...</code>
<code>ask(0, 2)</code>	1	<code>[1, 3, 2, 4, 5], [2, 4, 1, 5, 3], [3, 5, 4, 1, 2], [3, 1, 2, 5, 4], ...</code>
<code>ask(0, 3)</code>	2	<code>[3, 5, 4, 1, 2], [3, 1, 2, 5, 4], ...</code>

จากตาราง

- เมื่อเรียกฟังก์ชัน `ask(0, 1)` จะเห็นว่า หนึ่งในตัวอย่าง P ที่เป็นไปได้คือ `[1, 3, 4, 5, 2]` แต่เมื่อเรียก `ask(0, 2)` แล้ว `[1, 3, 4, 5, 2]` จะไม่ใช่ P ที่เป็นไปได้ เนื่องจาก $|P[0] - P[2]|$ ไม่เท่ากับ 1
- เมื่อเรียกฟังก์ชัน `ask(0, 2)` จะเห็นว่า ตัวอย่าง P ที่เป็นไปได้จะสอดคล้องกับคำตอบของ `ask(0, 1)`
- เมื่อเรียกฟังก์ชัน `ask(0, 3)` จะเห็นว่า ตัวอย่าง P ที่เป็นไปได้จะสอดคล้องกับคำตอบของ `ask(0, 1)` และ `ask(0, 2)`

ขอบเขตของข้อมูล

- $3 \leq N \leq 100,000$

ปัญหาย่อย

ปัญหาย่อย	คะแนน	เงื่อนไขเพิ่มเติม
1	3	$N = 3, Q = 10,000$
2	6	$N = 6, Q = 10,000$
3	14	$N \leq 100, Q = 10,000$
4	18	$P[0] = 1, Q = 125,000$
5	20	$Q = 200,000$
6	21	$Q = 150,000$
7	18	$Q = 125,000$

เกรดเดอร์ตัวอย่าง

ไฟล์เกรดเดอร์ตัวอย่างจะรับข้อมูลนำเข้าในรูปแบบต่อไปนี้

- บรรทัดที่ 1: รับค่า N และ Q
- บรรทัดที่ 2: รับค่า $P[0]$ ถึง $P[N - 1]$ ตามลำดับ

หลังจากนั้นไฟล์เกรดเดอร์ตัวอย่างจะเรียกฟังก์ชัน `select` ของคุณ แล้วแสดงผลลัพธ์ที่ได้จากฟังก์ชันออกทางหน้าจอ