

แถวตอนเรียงสี่ร้อย มาหาข้าพเจ้า!

"จัดแถว! เฮ้!" เสียงตะโกนดังออกมาจากสนาม กองพันคอมพิวเตอร์โอลิมปิก กรมส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี แห่งราชอาณาจักรสารสนเทศ กำลังจัดเตรียมการฝึกทหารใหม่ ผลัด 1/66 ที่กำลังเข้ามาในเดือนตุลาคมนี้เอง หลังผ่านการประชุมครั้งที่ 3/66 ของกองบังคับการกองพันที่ผ่านมาเมื่อไม่นานมานี้ วันนี้แหละ ถึงเวลาแล้ว ที่กองพันจะต้องจัดเตรียมทรัพยากรอย่างเร่งด่วนเพื่อการฝึกทหารใหม่!

ผู้พันমনาว ได้ทำการเรียกนายทหารภายในกองพันมารวมตัวกัน ทั้งหมด N นาย (แน่นอนว่า**ผู้พัน**แค่ หายตัวอีกแล้ว เช่นเคย) โดยรวมกำลังพลทั้งหมด จากทั้งกองเตรียมโจทย์ กองช่วยเตรียมโจทย์ และกองเตรียมระบบ รวมอยู่ในสนามเดียวกัน

"แถวตอนเรียง M มาหาข้าพเจ้า!"

เหล่ากำลังพลต่างวิ่งเข้ามาจัดแถวตามคำสั่งของ**ผู้พันমনาว** จนได้เป็นแถวตอนขนาด N แถว M คอลัมน์ โดยที่**ยังไม่ได้เรียงความสูงตามปกติ** และนายทหารในแถวที่ i คอลัมน์ที่ j จะมีความสูง $H_{i,j}$ ไมโครเมตร (สำหรับทุก $0 \leq i < N$ และ $0 \leq j < M$) **ผู้พันমনาว**รีบเร่งการเตรียมโจทย์อย่างด่วนที่สุด **จำป๋ลิ้ม**เข้ารายงานว่าโจทย์ที่เตรียมไว้นั้น **เกิดบั๊ก!** จึงใช้ไม่ได้กระชั้นหัน จึงได้ทำการหารือ**จ่านนท์**และ**จ๋ามาร์ค** เนื่องจากปัญหาทางด้านเวลา ทำให้**ผู้พันমনาว** ลงมาปฏิบัติหน้าที่ด้วยตนเองอีกครั้ง! หลังจากวางมือจากการกระทำภาคปฏิบัติไปแล้วสักพักใหญ่ ๆ

จำป๋ลิ้ม เห็นว่า สภาพกองพันปัจจุบันนั้นดูยุ่งมาก และคงจะทำงานไม่ทันเสียอย่างแน่นอน จึงทำการเรียก **อาสาสมัครไม่เกิน $R - L + 1$ นาย** ออกมาจากแถวที่ L ไปจนถึงแถวที่ R โดย แต่ด้วยเงื่อนไขแปลก ๆ ของกองพัน จึงต้องเลือกช่วงย่อยของแถวที่ติดกัน และเมื่อเรียกแถวออกมาแล้ว จะไม่มีการสลับแถวใด ๆ

ด้วยเงื่อนไขฟิลิปปินส์ของ**จำป๋ลิ้ม** ผู้ไม่ยอมให้ **เซรพัน** **บ้าเย็นใจ** และ**ป๋าม่วง** โกรธกริ้ว จึงต้องการจัดแถวอาสาสมัครให้ **สง่างาม** โดยเราจะกล่าวว่าหมู่อาสาสมัครนั้น **สง่างาม** ก็ต่อเมื่อความสูงเรียงจากมากไปน้อย

ในทางเทคนิคแล้ว กล่าวได้ว่าการเรียกอาสาสมัครจากแถวที่ L ถึงแถวที่ R คือการเลือกค่า l, r โดยที่ $L \leq l \leq r \leq R$ และเลือกฟังก์ชัน $f: \{l, \dots, r\} \rightarrow \{0, \dots, M-1\}$ เพื่อทำการจัดตั้งหมู่อาสาสมัครจำนวน $r - l + 1$ คน โดยคนที่ i ($0 \leq i \leq r - l$) จะมาจากแถวที่ $l + i$ คอลัมน์ที่ $f(l + i)$ และจะกล่าวว่าหมู่อาสาสมัครนั้นสง่างาม ก็ต่อเมื่อ $H_{l,f(l)} \geq H_{l+1,f(l+1)} \geq \dots \geq H_{r,f(r)}$

คำถามคือ จะสามารถจัดหาหมู่อาสาสมัครที่จำนวนยอดมากที่สุดได้เท่าไร (หาค่า $r - l + 1$ มากสุดที่เป็นไปได้ที่จะสอดคล้องกับเงื่อนไข)

แน่นอนว่า การเรียกอาสาสมัครเพียงครั้งเดียวนั้น ไม่เพียงพอต่องานที่กองพันกำลังแบกรับอยู่ (โดยเฉพาะกองเตรียมโจทย์ซึ่งกำลังหัวหมุนกันอยู่) **จำป๋ลิ้ม**จึงต้องการเรียกอาสาสมัครทั้งหมดถึง Q ครั้ง โดยแต่ละครั้ง แค่เพียงเรียกออกมา จัดหมู่อาสา ทำการรายงาน**ผู้พันমনาว** ก็จะเป็นที่เรียบร้อย หลังจากการเรียกออกมาแต่ละรอบแล้ว จะให้นายทหารแต่ละนายกลับคืนสู่ตำแหน่งเดิมในแถว จึงทำให้การถามคำถามแต่ละคำถาม ไม่มีความเกี่ยวเนื่องกัน (ถามก่อนหรือถามหลังก็ได้คำตอบเหมือนกัน)

ผู้กองธนะ อยากทราบล่วงหน้าว่าในการเรียกออกมาแต่ละครั้ง จะได้จำนวนยอดสูงสุดกี่นาย เพื่อทำการเก็บสถิติ รายงานกองบังคับการกองพันต่อไป เขาจึงอยากให้คุณ ซึ่งเป็นนักเรียนภายใต้การดูแลของกองพัน ได้ทำการเขียน โปรแกรมหาจำนวนยอดเหล่านี้ออกมา

แต่ก่อนที่คุณจะได้เริ่มเขียนโปรแกรม **จำปัลลิม** ก็ตะโกนคติประจำกองพันอีกครั้งหนึ่ง

"ชาติ เกียรติ วินัย กล้าหาญ อดทน สามัคคี"

ขอให้นายทหารทุกคนอย่าลืม ว่าการต้องออกไปสู้รบกับเอเลี่ยน ใน การสู้รบทางคอมพิวเตอร์ระหว่างประเทศ ครั้งที่ 36 เราทุกคนจงถวายดวงใจ ยอมสละหยาดเหงื่อ เลือดเนื้อ ทุกหยด เพื่อหนึ่งคะแนน ที่จะนำพามนุษยชาติ ให้ได้เปรียบเอเลี่ยนมากขึ้นแม้เพียงเล็กน้อยก็ตาม

กองพัน ขอแสดงความขอบคุณ ที่คุณได้มาเป็นส่วนหนึ่งของกำลังรบอันสำคัญที่จะเป็นอนาคตของมนุษยชาติ ดังนั้น จึงขอให้คุณช่วยตอบคำถามของ **ผู้พันธนะ** ดังที่กล่าวไป

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันดังต่อไปนี้

```
void setup_battalion(int N, int M, int Q, vector<vector<int>> H)
```

- ฟังก์ชันนี้จะถูกเรียกเพียงครั้งเดียว ระบุโครงสร้างแถวเริ่มต้น เป็นแถวตอนเรียง M จำนวน N แถว นายทหารในแถวที่ i หมู่ที่ j จะมีความสูง $H_{i,j}$

```
int count_volunteers(int L, int R)
```

- ฟังก์ชันนี้จะถูกเรียก Q ครั้งพอดี เป็นการถามว่า หากพิจารณาตั้งแต่แถว L ไปจนถึงแถว R โดยเลือกช่วง $[l, r]$ ภายใน $[L, R]$ และหยิบทหารหนึ่งนายออกจากแต่ละแถวภายในช่วงนั้น เพื่อสร้างหมู่อาสาสมัครที่สง่างาม แล้วจะได้จำนวนอาสาสมัครสูงสุดกี่คน กล่าวคือ $r - l + 1$ มีค่ามากที่สุดเป็นเท่าใด

ข้อจำกัด

- $1 \leq M \leq 400$
- $1 \leq N \leq 20\,000$
- $1 \leq Q \leq 500\,000$
- $1 \leq H_{i,j} \leq 2\,000\,000$ สำหรับทุกจำนวนเต็ม $0 \leq i < N$ และ $0 \leq j < M$

ปัญหาย่อย

- (10 คะแนน) $M = 1$ และ $Q = 1$
- (20 คะแนน) $M = 1$
- (10 คะแนน) $M \leq 5$ และ $Q = 1$
- (20 คะแนน) $Q = 1$

5. (15 คะแนน) $M \leq 5$
6. (10 คะแนน) $L = 0$ สำหรับทุกคำถาม
7. (15 คะแนน) ไม่มีเงื่อนไขเพิ่มเติม

ตัวอย่าง

ตัวอย่างที่ 1

```
setup_battalion(5, 2, 4, [[1, 4], [3, 3], [5, 2], [10, 1], [3, 2]])
```

ทำการตั้งค่าการจัดแถวของกองพัน โดยนายทหารแต่ละนายมีความสูงดังนี้

คอลัมน์ 1	คอลัมน์ 2
1	4
3	3
5	2
10	1
3	2

ต่อมามีคำถาม 4 คำถาม ดังนี้

```
count_volunteers(3, 4)
```

ตอบ 2 เพราะสามารถเลือกแถวที่ 3 คอลัมน์ที่ 1 กับแถวที่ 4 คอลัมน์ที่ 1 (หรือ 2 ก็ได้) ออกไปได้

```
count_volunteers(0, 3)
```

ตอบ 4 เพราะสามารถเลือกคอลัมน์ที่ 2 ทั้งหมดทุกแถว จะได้ 4, 3, 2, 1 ซึ่งเรียงจากมากไปหาน้อยอยู่แล้ว

```
count_volunteers(1, 3)
```

ตอบ 3 เพราะสามารถเลือกคอลัมน์ที่ 2 ทั้งหมดทุกแถว จะได้ 3, 2, 1 ซึ่งเรียงจากมากไปหาน้อยอยู่แล้ว

```
count_volunteers(0, 4)
```

ตอบ 4 เพราะสามารถเลือก $l = 0, r = 3$ คอลัมน์ที่ 2 ทั้งหมด ได้มากที่สุด 4 แถว สังเกตว่าไม่สามารถเลือกทั้ง 5 แถวออกมาได้

ตัวอย่างที่ 2

```
setup_battalion(4, 3, 3, [[1, 7, 3], [4, 1, 8], [6, 5, 7], [6, 2, 9]])
```

```
count_volunteers(0, 2)
```

คืนค่า 2

```
count_volunteers(1, 3)
```

คืนค่า 3

```
count_volunteers(1, 1)
```

คืนค่า 1

ตัวอย่างที่ 3

```
setup_battalion(7, 4, 8, [  
    [196189, 490241, 805661, 1199622],  
    [649691, 229184, 1288874, 150692],  
    [1113012, 974282, 1323486, 89131],  
    [1831337, 334124, 908635, 1993563],  
    [935389, 1888119, 230503, 116277],  
    [1955545, 408897, 1553388, 573314],  
    [444041, 153762, 59384, 1631437]  
])
```

```
count_volunteers(4, 4)
```

คืนค่า 1

```
count_volunteers(2, 2)
```

คืนค่า 1

```
count_volunteers(0 ,1)
```

คืนค่า 2

```
count_volunteers(0, 6)
```

คืนค่า 4

```
count_volunteers(1, 6)
```

คืนค่า 4

```
count_volunteers(0, 2)
```

คืนค่า 3

```
count_volunteers(1, 3)
```

คืนค่า 3

```
count_volunteers(3, 4)
```

คืนค่า 2

เกรตเตอร์ตัวอย่าง

เกรตเตอร์ตัวอย่างจะอ่านข้อมูลดังต่อไปนี้

- บรรทัดที่ 1: $N \ M \ Q$
- บรรทัดที่ $2 + i$ ถึง $2 + N - 1$: $H_{i,0} \ H_{i,1} \ H_{i,2} \ \dots \ H_{i,M-1}$
- บรรทัดที่ $2 + N + i$ ถึง $2 + N + Q - 1$: $L \ R$

เกรตเตอร์ตัวอย่างจะส่งออกค่าที่คืนมาจากฟังก์ชัน `count_volunteers` ในแต่ละรอบ

ขอบเขต

- Time limit: 3.5 seconds
- Memory limit: 512 MB