



Ask Ask

บนเส้นจำนวนตั้งแต่ $0, \dots, N - 1$ มีระเบิดซ่อนอยู่สองลูกภายในบางพิกัดระหว่าง 0 ถึง $N - 1$

คุณสามารถส่งคำถามได้ โดยส่งเซต S ซึ่งเป็นเซตย่อยของจำนวนเต็มระหว่าง 0 ถึง $N - 1$ ไปถาม อุปกรณ์ช่วยกู้ระเบิดจะทำการตอบคำถามโดยตอบ `true` หากมีระเบิดที่มีพิกัดอยู่ใน S อย่างน้อยหนึ่งลูก และตอบ `false` หากไม่มีระเบิดที่มีพิกัดอยู่ใน S

เนื่องจากคุณเป็นนักคอมพิวเตอร์ที่กำลังทำงานจากบ้าน (work from home) อยู่ คุณจึงไม่สามารถส่งการอุปกรณ์ช่วยกู้ระเบิดได้ทันที สิ่งที่คุณทำได้คือเตรียมรายการคำถามทั้งหมดไว้ก่อน หลังจากนั้นอุปกรณ์กู้ระเบิดจะตอบคำถามทั้งหมดภายในครั้งเดียว

งานของคุณคือเขียนโปรแกรมเพื่อหาว่าระเบิดทั้งสองลูกนั้นอยู่ที่พิกัดใด

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้:

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

- ฟังก์ชันนี้จะถูกเรียก **หลายครั้ง** (รับประกันว่าไม่เกิน 32 ครั้งต่อการตรวจหนึ่งข้อมูลทดสอบ)
- ฟังก์ชันนี้สามารถเรียกฟังก์ชัน `analyse` ได้เพียงครั้งเดียวภายในการเรียกใช้ฟังก์ชันนี้หนึ่งครั้ง
- ฟังก์ชันนี้จะต้องคืนค่าเวกเตอร์ขนาด 2 โดยสมาชิกของเวกเตอร์ระบุพิกัดของระเบิดทั้งสอง เรียงจากน้อยไปมาก

คุณสามารถเรียกฟังก์ชันต่อไปนี้ได้หนึ่งครั้งต่อการเรียกใช้ `find_bombs` หนึ่งครั้ง:

```
vector<bool> analyse(vector<vector<int>>> Z)
```

- Z เป็นเวกเตอร์ของเซต S ที่จะส่งไปถาม โดยฟังก์ชันนี้จะคืนค่าเป็นเวกเตอร์ที่แต่ละตัวระบุคำตอบของแต่ละคำถามตามลำดับของเซตที่ส่งไป
- ขนาดของ Z จะต้องมีค่าไม่เกิน N
- เวกเตอร์ S แต่ละตัวใน Z จะต้องมีขนาดไม่เกิน N สมาชิกแต่ละตัวจะต้องอยู่ระหว่าง 0 ถึง $N - 1$ และห้ามมีสมาชิกที่ซ้ำกันภายใน S

ขอบเขต

- $N \leq 1024$

ปัญหาย่อย

1. (4 คะแนน) $N = 2$, $\tilde{Q} = 1$ และรับประกันว่าระเบิดทั้งสองลูกอยู่ในพิกัดเดียวกัน
2. (6 คะแนน) $N = 4$, $\tilde{Q} = 2$ และรับประกันว่าระเบิดทั้งสองลูกอยู่ในพิกัดเดียวกัน
3. (9 คะแนน) $N = 8$, $\tilde{Q} = 3$ และรับประกันว่าระเบิดทั้งสองลูกอยู่ในพิกัดเดียวกัน
4. (28 คะแนน) $N = 1024$, $\tilde{Q} = 10$ และรับประกันว่าระเบิดทั้งสองลูกอยู่ในพิกัดเดียวกัน
5. (53 คะแนน) $N = 64$

ปัญหาย่อยจะมีการให้คะแนนแบบพิเศษ ดังต่อไปนี้ (โดยให้ Q แทนขนาดของเวกเตอร์ Z ที่ส่งไปถาม)

การให้คะแนนสำหรับปัญหาย่อยที่ 1 ถึง 4

เงื่อนไข	อัตราส่วนของคะแนนที่ได้ต่อคะแนนเต็มของปัญหาย่อยนั้น ๆ
$\tilde{Q} < Q \leq N$	$\frac{N-Q}{N-\tilde{Q}}$
$Q \leq \tilde{Q}$	1

การให้คะแนนสำหรับปัญหาย่อยที่ 5

เงื่อนไข	คะแนนที่ได้
$40 < Q \leq 64$	$85 - \frac{5}{4}Q$
$23 < Q \leq 40$	$18 \left(\frac{40-Q}{17} \right)^2 + 35$
$Q \leq 23$	53

สำหรับการตรวจในข้อนี้ ในแต่ละปัญหาย่อยจะมีหลายข้อมูลทดสอบ โดยในแต่ละข้อมูลทดสอบอาจมีการเรียกใช้ฟังก์ชัน `find_bombs` หลายครั้ง (รับประกันว่าไม่เกิน 32 ครั้ง) ผลลัพธ์ Q ของแต่ละข้อมูลทดสอบคือค่า **มากที่สุด** ระหว่าง Q จากการเรียก `find_bombs` แต่ละครั้ง และคะแนนของแต่ละปัญหาย่อยคือคะแนน **ต่ำสุด** ระหว่างคะแนนของแต่ละข้อมูลทดสอบ

ตัวอย่าง

หมายเหตุ: ตัวอย่างนี้ไม่สอดคล้องกับข้อมูลทดสอบปัญหาย่อยใดเลย เป็นเพียงตัวอย่างคร่าว ๆ เท่านั้น

เริ่มต้น เกรดเดอร์ทำการเรียก `find_bombs` ดังนี้ (สมมติว่าระเบิดซ่อนอยู่ที่พิกัด 0 และ 3)

```
find_bombs(4)
```

หลังจากนั้น วิธีการหนึ่งที่เป็นไปได้คือ ผู้เข้าแข่งขันส่งคำถามโดยการเรียกใช้ฟังก์ชันดังนี้

```
analyse([[0], [1], [2], [3]])
```

โดยฟังก์ชันจะคืนค่าเวกเตอร์ `[true, false, false, true]` หลังจากนั้นผู้เข้าแข่งขันจึงทราบว่าระเบิดอยู่ที่ตำแหน่ง 0 และ 3 และคืนค่า `[0, 3]` มาในรูปเวกเตอร์

ตัวอย่างต่อมาสอดคล้องกับข้อมูลทดสอบในปัญหาย่อยที่ 2 ดังนี้ เริ่มต้น เกรดเดอร์ทำการเรียก `find_bombs` ดังนี้ (สมมติว่าระเบิดซ่อนอยู่ที่พิกัด 1 และ 1)

```
find_bombs(4)
```

หลังจากนั้น วิธีการหนึ่งที่เป็นไปได้คือ ผู้เข้าแข่งขันส่งคำถามโดยการเรียกใช้ฟังก์ชันดังนี้

```
analyse([[0, 2, 3]])
```

ฟังก์ชันจะคืนค่า `[false]` ทำให้ผู้เข้าแข่งขันทราบว่าไม่มีระเบิดอยู่ในตำแหน่ง 0, 2 และ 3 จึงทำให้สรุปได้ว่าระเบิดทั้งสองอยู่ที่พิกัด 1 อย่างแน่นอน ผู้เข้าแข่งขันจึงทำการคืนค่า `[1, 1]` มาในรูปเวกเตอร์

เกรดเดอร์ตัวอย่าง

เกรดเดอร์ตัวอย่างอ่านข้อมูลในรูปแบบดังต่อไปนี้:

- บรรทัดที่ 1: $N \ x_1 \ x_2$

(เมื่อ x_1 และ x_2 แทนพิกัดของระเบิดลูกที่หนึ่งและสองตามลำดับ)

- หากเงื่อนไขของการถามถูกต้อง และคำตอบถูกต้อง เกรดเดอร์ตัวอย่างจะส่งออกผลลัพธ์ `Correct Answer: Q` เมื่อ Q แทนขนาดของเวกเตอร์ที่ส่งไปถาม
- หากเงื่อนไขของการถามถูกต้องแต่คำตอบผิด
 - หากคำตอบไม่ได้มีขนาด 2 เกรดเดอร์ตัวอย่างจะส่งออก `Wrong Answer: output size is not 2`
 - หากคำตอบมีขนาด 2 เกรดเดอร์ตัวอย่างจะส่งออกผลลัพธ์ `Wrong Answer: expected x_1 x_2 but found y_1 y_2` เมื่อ x_1 กับ x_2 แทนพิกัดของระเบิดจริง และ y_1 กับ y_2 แทนพิกัดที่ได้รับมาจากการคืนค่าของฟังก์ชัน `find_bombs`
- หากเงื่อนไขของการถามผิด
 - หากคำถามมี $Q > N$ เกรดเดอร์ตัวอย่างจะส่งออกผลลัพธ์ `Invalid Query: too many queries`
 - หากคำถามมีเวกเตอร์ S ที่มีขนาดมากกว่า N หรือมีสมาชิกที่มีค่าซ้ำกัน หรือมีสมาชิกบางตัวนอกขอบเขต $0, \dots, N - 1$ เกรดเดอร์ตัวอย่างจะส่งออกผลลัพธ์ `Invalid Query: invalid query`

ข้อจำกัด

- Time limit: 1 second
- Memory limit: 512 MB