รถระเบิด (TNT)

มีเมือง N เมือง ($2 \leq N \leq 1024$) เรียกเป็นเมืองที่ 0 ถึงเมืองที่ $N-1$ โดยเมืองที่ i จะมีถนนแบบทิศทางเดียวเชื่อมไปยังเมืองที่ $(i+1) \bmod N$ ลักษณะของถนนจะเชื่อมเมืองทั้ง N เมืองเป็นวงรอบ

มีรถระเบิดหนึ่งคันที่เริ่มต้นที่เมือง A รถคันนี้มีความเร็ว B หน่วย ($0 \leq B < N$) กล่าวคือถ้าในเวลาปัจจุบันรถอยู่ที่เมืองที่ j เวลาถัดไป 1 หน่วย รถจะเดินทางไปตามถนนไปอยู่ที่เมืองที่ $(j+B) \bmod N$

คุณต้องการทราบข้อมูลว่ารถดังกล่าวเริ่มออกเดินทางที่เมืองใดและมีความเร็วเท่าใด โดยสามารถสั่งงานชุดอุปกรณ์ตรวจจับสารระเบิดที่ติดตั้งไว้ที่เมืองต่าง ๆ การเตรียมการสั่งงานแต่ละครั้งจะใช้เวลา 1 หน่วย โดยคุณจะสามารถเลือกรายการของเมืองจากเมืองทั้งหมดที่ต้องการให้อุปกรณ์ทำงานได้ เมื่ออุปกรณ์ทำงานคุณจะได้รับคำตอบว่าที่เวลาที่ถามนั้น รถระเบิดอยู่เมืองใดเมืองหนึ่งในรายการหรือไม่ คุณสามารถถามได้ Q ครั้ง ให้ถือว่ารถเคลื่อนที่ในแต่ละหน่วยเวลาเร็วมาก อุปกรณ์จะไม่สามารถตรวจสอบรถที่เคลื่อนที่ได้แต่จะตรวจได้ที่เมืองที่ไปถึงปลายทางแล้วเท่านั้น

พิจารณาตัวอย่างต่อไปนี้ที่ $N = 7$ สมมติว่า $A = 3$ และ $B = 4$ ตำแหน่งของรถในเวลาต่าง ๆ แสดงในตาราง

เวลา	0	1	2	3	4	5	6	7	8
ตำแหน่ง	3	0	4	1	5	2	6	3	0

สมมติว่าคุณสั่งให้อุปกรณ์ตรวจสอบตำแหน่งรถทำงานไป 4 ครั้ง ที่เวลา $1, 2, 3, 4$ ตัวอย่างของรายการเมืองที่สั่งและผลลัพธ์ที่ได้แสดงในตารางด้านล่าง

เวลา	ตำแหน่งรถระเบิด	รายการเมือง	ผลลัพธ์
1	0	0,1,2	`true`
2	4	2,3,6	`false`
3	1	5,6	`false`
4	5	0,1,2,3,4,5,6	`true`

คุณจะต้องใช้อุปกรณ์ลักษณะนี้ในการหาค่า A และ B ให้ได้

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้

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

- ฟังก์ชันนี้จะถูกเรียกหนึ่งครั้ง และจะต้องคืนค่าอาร์เรย์ที่มีขนาด 2 ช่องแรกเป็นค่า A ที่หาได้ และช่องที่สองเป็นค่า B ที่หาได้

ฟังก์ชันจะเรียกใช้ฟังก์ชันด้านล่างจากเกรดเดอร์ จะเรียกได้ไม่เกิน Q ครั้ง

```
bool find_tnt(vector<int> city)
```

- คุณจะต้องใส่รายการของหมายเลขเมืองที่ต้องการให้อุปกรณ์ทำงาน ในรายการ `city` ห้ามระบุเมืองซ้ำกัน
- ถ้าคุณเรียกใช้คำสั่งนี้มากกว่า Q ครั้งหรือหมายเลขเมืองนอกขอบเขต หรือระบุเมืองซ้ำ โปรแกรมจะจบการทำงานโดยอัตโนมัติ

เงื่อนไข

- $2 \leq N \leq 1\,024$
- $0 \leq A < N, 0 \leq B < N$

ปัญหาย่อย

1. (9 คะแนน) $2N^2 \leq Q$ และ $2 \leq N \leq 50$
2. (12 คะแนน) $2N \leq Q$ และ $2 \leq N \leq 100$
3. (16 คะแนน) $Q = 20, 2 \leq N \leq 1\,000$ และมี A หรือ B เป็น 0 อย่างน้อยหนึ่งจำนวน
4. (11 คะแนน) $Q = 200$ และ $2 \leq N \leq 1\,000$
5. (16 คะแนน) $Q = 22$ และ $2 \leq N \leq 1\,000$
6. (14 คะแนน) N มีค่าเป็น 2^K เมื่อ $1 \leq K \leq 10$ และ $Q = 2K$
7. (22 คะแนน) $Q = 20$ และ $2 \leq N \leq 1\,000$

ตัวอย่างการทำงาน

จากตัวอย่างด้านบน เกรดเดอร์จะเรียก

```
find_truck(7)
```

ฟังก์ชันดังกล่าว เรียก `find_tnt` ที่เวลา 1

```
find_tnt([0,1,2])
```

ฟังก์ชันจะคืนค่า `true` เพราะรถอยู่ที่ตำแหน่ง 0 ต่อมาที่เวลา 2 เรียก

```
find_tnt([2,3,6])
```

ฟังก์ชันจะคืนค่า `false` เพราะรถอยู่ที่ตำแหน่ง 4 ต่อมาที่เวลา 3 เรียก

```
find_tnt([5,6])
```

ฟังก์ชันจะคืนค่า `false` เพราะรถอยู่ที่ตำแหน่ง 1 ต่อมาที่เวลา 4 เรียก

```
find_tnt([0,1,2,3,4,5,6])
```

ฟังก์ชันจะคืนค่า `true` เพราะรถอยู่ที่ตำแหน่ง 5

ฟังก์ชัน `find_truck` ที่ถูกต้องจะต้องคืนค่า

```
[3,4]
```

Sample Grader

จะมีทั้งหมด 1 บรรทัด:

- บรรทัด 1: N Q A B

เกรดเดอร์ตัวอย่างจะแสดงผลว่าสามารถหาคำตอบได้ภายใน Q ครั้งหรือไม่ ถ้าคำตอบไม่ถูกต้องจะแสดงผล Wrong Answer หากโปรแกรมถูกจบการทำงานจากการผิดเงื่อนไขจะแสดงผล Runtime Error

Limits

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