จับคู่ตัวเลข (nummatching)

มีอาร์เรย์ A ของจำนวนเต็ม N ตัว (N เป็นจำนวนเต็มคู่) ที่คุณ**ไม่ทราบข้อมูลภายใน** เป้าหมายของข้อนี้คือการพยายามหาวิธีในการจับคู่จำนวนเต็มในอาร์เรย์ให้ครบ $N/2$ คู่ โดยช่องในอาร์เรย์สองช่องจะสามารถจับคู่กันได้ถ้าจำนวนเต็มในช่องทั้งสองมีค่าเท่ากัน การอ้างถึงอาร์เรย์จะอ้างช่องแรกเริ่มที่ 0

คุณทราบเงื่อนไขเกี่ยวกับอาร์เรย์ดังนี้

- ในอาร์เรย์มีจำนวนเต็มระหว่าง 0 ถึง $N/2$ อยู่
- ในอาร์เรย์ไม่มีจำนวนเต็มใด ๆ ปรากฏมากกว่า 2 ครั้ง **ยกเว้น** 0
- ในช่องอาร์เรย์ จะมีอย่างน้อย 1 ช่องที่ไม่ใช่ 0 ที่**ไม่สามารถ**จับคู่กับช่องข้าง ๆ ได้
- มีจำนวนเต็มที่ไม่ใช่ 0 อย่างน้อย 2 ตัว

ในอาร์เรย์ดังกล่าว เราสามารถแปลงช่องที่เป็น 0 ให้เป็นจำนวนเต็ม 1 ถึง $N/2$ ได้ โดยต้องไม่ให้มีจำนวนเต็มใดปรากฏมากกว่า 2 ครั้ง เราอยากหารูปแบบจับคู่จำนวนเต็มในอาร์เรย์ให้ครบ $N/2$ คู่ (โดยที่สุดท้าย เราไม่เห็นค่าในอาร์เรย์ตั้งต้น)

คุณสามารถสอบถามเกี่ยวกับอาร์เรย์ดังกล่าวได้ โดยรูปแบบในการถามจะเป็นดังนี้

- ให้คุณเลือกสับเซตของช่องของอาร์เรย์ จากนั้นคุณสามารถถามได้ว่าจะสามารถจับคู่ช่องของอาเรย์ในสับเซตดังกล่าวได้อย่างน้อยหนึ่งคู่หรือไม่ โดยสมมติให้คุณสามารถเปลี่ยนค่าของช่องอาร์เรย์ที่เป็น 0 ให้เป็นค่าอะไรก็ได้ (ที่ไม่ผิดเงื่อนไขที่ทำให้มีจำนวนเต็มบางจำนวนในช่องในสับเซตปรากฏมากกว่า 2 ครั้ง)

คุณสามารถถามได้ทั้งสิ้นไม่เกิน Q ครั้ง

พิจารณาตัวอย่างต่อไปนี้ ที่ $N = 6$ สมมติว่าอาร์เรย์ A ที่เรามีคือ

```
[0,1,2,1,0,3]
```

พิจารณาการสอบถามอาร์เรย์ดังนี้

- ถ้าเราเลือกสับเซตของหมายเลขช่องในอาร์เรย์ที่ถามคือ $[3, 4, 5]$ นั่นคือถามช่องที่มีค่า $1, 0, 3$ เราจะพบว่าเราสามารถเปลี่ยนค่า 0 ให้เป็น 1 หรือ 3 เพื่อจับคู่กับ 1 หรือ 3 ได้ ดังนั้นคำตอบคือ yes
- ถ้าเราเลือกสับเซตของหมายเลขช่องในอาร์เรย์ที่ถามคือ $[1, 2]$ นั่นคือถามช่องที่มีค่า $1, 2$ เราพบว่าเราไม่มี 0 ให้ปรับค่า ทำให้ไม่สามารถจับคู่ได้เลย ดังนั้นคำตอบคือ no
- ถ้าเราเลือกสับเซตของหมายเลขช่องในอาร์เรย์ที่ถามคือ $[0, 4]$ นั่นคือถามช่องที่มีค่า $0, 0$ เราสามารถเปลี่ยน 0 ทั้งคู่ให้เป็นจำนวนเต็มใดก็ได้ที่ตรงกัน ดังนั้นคำตอบคือ yes
- ถ้าเราเลือกสับเซตของหมายเลขช่องในอาร์เรย์ที่ถามคือ $[1, 2, 3]$ นั่นคือถามช่องที่มีค่า $1, 2, 1$ เราสามารถจับคู่ 1 ได้ ดังนั้นคำตอบคือ yes
- ถ้าเราเลือกสับเซตของหมายเลขช่องในอาร์เรย์ที่ถามคือ $[1, 2, 5]$ นั่นคือถามช่องที่มีค่า $1, 2, 3$ เราพบว่าเราไม่สามารถจับคู่ได้ ดังนั้นคำตอบคือ no

- ถ้าเราเลือกสับเซตของหมายเลขช่องในอาร์เรย์ที่ถามคือ $[0, 1]$ นั่นคือถามช่องที่มีค่า $0, 1$ สังเกตว่าเราสามารถเปลี่ยนค่า 0 เป็นหนึ่ง เพื่อจับคู่กับ 1 ได้ ดังนั้นคำตอบคือ `yes` สังเกตว่าเงื่อนไขการเปลี่ยนค่า 0 ในการตอบคำถามนั้น สนใจเฉพาะส่วนของสับเซตเท่านั้น (เพราะถ้าพิจารณาทั้งอาร์เรย์ เราจะไม่สามารถแก้ 0 เป็น 1 ได้ เนื่องจาก 1 จะปรากฏในอาร์เรย์ 3 ครั้ง)

ถ้าเราถามจนได้ข้อมูลเพียงพอแล้ว เราจะต้องตอบการจับคู่ที่เป็นไปได้ ในกรณีนี้ คำตอบหนึ่งที่เป็นไปได้คือการเปลี่ยน 0 ตัวแรกเป็น 3 และเปลี่ยน 0 ตัวที่สองเป็น 2 ได้ผลลัพธ์เป็นอาร์เรย์ `[3,1,2,1,2,3]` และจับคู่ดัชนีของอาร์เรย์ดังนี้

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

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้:

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

- ฟังก์ชันนี้จะถูกเรียกหนึ่งครั้ง
- ฟังก์ชันจะสามารถเรียกฟังก์ชัน `can_match` ด้านล่างได้ไม่เกิน Q ครั้ง
- ฟังก์ชันจะต้องคืนค่าการจับคู่รูปแบบหนึ่งที่เป็นไปได้ เป็นอาร์เรย์ M ขนาด $N/2$ ที่ระบุว่าช่องที่ $M[0][0]$ จับคู่กับช่อง $M[0][1]$, ช่องที่ $M[1][0]$ จับคู่กับช่อง $M[1][1]$ ไปเรื่อย ๆ จนถึง ช่องที่ $M[N/2][0]$ จับคู่กับช่อง $M[N/2][1]$ ลำดับในรายการจะเป็นอย่างใดก็ได้

ฟังก์ชัน `find_matching` สามารถเรียกใช้ฟังก์ชัน `can_match` ด้านล่างได้ Q ครั้ง

```
bool can_match(vector<int> B)
```

- พารามิเตอร์ B คืออาร์เรย์แทนรายการดัชนีของช่องที่สนใจในสับเซต ค่าในอาร์เรย์อยู่ระหว่าง 0 ถึง $N-1$
- ฟังก์ชันนี้จะคืนค่า `true` ก็ต่อเมื่อ ในสับเซตของช่องในอาร์เรย์ที่ระบุใน B เราสามารถจับคู่ตัวเลขได้อย่างน้อยหนึ่งคู่ โดยสมมติว่าเราสามารถแก้ไขช่องในอาร์เรย์ที่มีค่าเป็น 0 ได้

เงื่อนไข

- $2 \le N \le 1\,000$, N เป็นจำนวนเต็มคู่
- $Q = 15\,000$
- $0 \le A[i] \le N/2$

ปัญหาย่อย

1. (5 points) $N \le 100$, $A[i] \neq 0$, สำหรับทุก ๆ ค่า i
2. (9 points) $N \le 100$
3. (11 points) จำนวนเต็ม 1 ถึง $N/2$ ปรากฏในอาร์เรย์ A จำนวนละ 1 ครั้งพอดี

4. (34 points) $A[i] \neq 0$, สำหรับทุก ๆ ค่า i
5. (12 points) มีช่องที่ $A[i] = 0$ จำนวนไม่เกิน 20 ช่อง
6. (29 points) ไม่มีเงื่อนไขเพิ่มเติม

ตัวอย่าง

จากตัวอย่างด้านบน เกรดเดอร์จะเรียกฟังก์ชัน

```
find_matching(6)
```

ฟังก์ชันดังกล่าว สามารถเรียกฟังก์ชัน `can_match` ได้หลายครั้ง และได้ผลลัพธ์ตามตัวอย่างข้างต้นดังนี้

```
can_match([3,4,5])      // returns true
can_match([1,2])        // returns false
can_match([0,4])        // returns true
can_match([1,2,3])      // returns true
can_match([1,2,5])      // returns false
can_match([0,1])        // returns true
```

เมื่อเรียก `can_match` จนได้ข้อมูลเพียงพอแล้ว จะต้องตอบเป็นรายการของการจับคู่ คำตอบที่เป็นไปได้คำตอบหนึ่งคือ

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

เกรดเดอร์ตัวอย่าง

เกรดเดอร์ตัวอย่างจะอ่านข้อมูลนำเข้าดังนี้

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

ถ้ามีการเรียก `can_match` มากกว่า Q ครั้ง จะพิมพ์แสดงความผิดพลาด ไม่เช่นนั้นจะพิมพ์การจับคู่ที่คืนจาก `find_matching` มา พร้อมกับระบุว่าจับได้ถูกต้องหรือผิดพลาด ถ้าจับถูกต้องจะพิมพ์จำนวนครั้งการเรียก `can_match` ออกมาด้วย

ขีดจำกัด

- Time limit: 0.5 seconds
- Memory limit: 512 MB