



หมวก (hats)

มีคน N คน เรียกเป็นคนที่ $0, 1, \dots, N - 1$ แต่ละคนใส่หมวก 1 ใบ หมวกแต่ละใบมีสีหนึ่งสี คือดำ (แทนด้วย 0) หรือขาว (แทนด้วย 1) เป็นไปได้ที่คนสองคนจะมีหมวกสีซ้ำกัน แต่ละคนจะเห็นสีของหมวกของทุกคน แต่จะไม่เห็นสีของหมวกของตนเอง

คุณต้องการให้คนหนึ่งคนหาให้ได้ว่าตนเองมีหมวกสีอะไร (อย่างน้อยหนึ่งคน แต่คนเดียวก็เพียงพอ)

ในการหาว่ามีหมวกสีอะไร เกรดเดอร์จะทยอยเรียกฟังก์ชัน `think` สำหรับแต่ละคนไปเรื่อย ๆ โดยจะเรียก `think` ทั้งหมด N รอบ แต่ละรอบจะเรียกคนที่ $0, 1, \dots$ ไปจนถึงคนที่ $N - 1$ แต่ละคน (โปรแกรม) จะไม่สามารถสื่อสารอะไรกันได้นอกจากที่ระบุไว้ และจะไม่สามารถสื่อสารอะไรกับโปรแกรมเองได้ระหว่างรอบด้วย ถ้ามีคนใดคนหนึ่งทราบว่าตนเองมีหมวกสีอะไร ก็สามารถเรียกฟังก์ชัน `answer` เพื่อตอบได้ เมื่อมีคนใดคนหนึ่งตอบ การทำงานจะจบลง

แน่นอนว่า มันไม่มีวิธีที่จะตอบได้อย่างถูกต้องตลอด เพราะเราไม่ให้สื่อสารกันเลย อย่างไรก็ตาม คุณต้องการตกลงกับคนทั้ง N คน (โดยการเขียนโปรแกรมแทนยุทธวิธีในการตอบ) ที่ทำให้ **ความน่าจะเป็นที่ตอบถูก** นั้นมีค่ามากที่สุดเท่าที่เป็นไปได้ โดยสามารถคิดได้ว่าความน่าจะเป็นที่แต่ละคนจะได้หมวกสีขาวหรือดำคือ $\frac{1}{2}$ ทุกคน

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้:

```
int think(int N, int id, vector<int> hats)
```

- ในการตรวจหนึ่งครั้งฟังก์ชันนี้จะถูกเรียกซ้ำ ๆ จำนวน N รอบพอดี ไม่จำเป็นว่าการเรียกหลายครั้งจะต้องทำภายในการ execute โปรแกรมเพียงครั้งเดียว
- ฟังก์ชันจะระบุ
 - N แทนค่า N
 - id หมายเลขของคนที่เรียก มีค่าระหว่าง 0 ถึง $N - 1$
 - `hats` เป็นอาร์เรย์ของสีของหมวกของคนอื่น ๆ ทั้งหมด อาร์เรย์จะมี N ช่อง แต่ละช่องจะระบุค่าภายในเซต $\{-1, 0, 1\}$ ช่องที่ i จะระบุค่าแทนสีของหมวกที่คนที่ i ใส่ สำหรับ i ตั้งแต่ 0 ถึง $N - 1$ ยกเว้นช่อง id จะเป็นช่องเดียวที่ระบุค่า -1
- ระหว่างการทำงานหลาย ๆ รอบของฟังก์ชัน `think` คุณจะต้องไม่สื่อสารหรือเก็บข้อมูลอะไรเพื่อส่งถึงกันระหว่างการเรียกได้

ฟังก์ชันดังกล่าวจะสามารถคืนค่า -1 เพื่อตอบว่า "ไม่รู้" คืนค่า 0 เพื่อตอบว่า "ฉันใส่หมวกสีดำ" หรือคืนค่า 1 เพื่อตอบว่า "ฉันใส่หมวกสีขาว"

ภาพถัดไปแสดงตัวอย่างของคนที่คืนค่า -1



(จริง ๆ แล้ว เขาใส่หมวกสีดำ ดังภาพถัดไป)



เงื่อนไข

- $N \in \{3, 7\}$

ปัญหาย่อย

1. (25 คะแนน) $N = 3$.
2. (75 คะแนน) $N = 7$.

การให้คะแนน

ชุดทดสอบทั้งสองชุด สำหรับทั้งสองปัญหาย่อย จะมีข้อมูลทดสอบทั้งหมด 2^N ข้อมูล เฉพาะข้อนี้ จะไม่มี full feedback สำหรับข้อมูลทดสอบแต่ละข้อมูลภายในระบบตัวตรวจ กล่าวคือ ผู้เข้าแข่งขันจะมองเห็นผลลัพธ์การตรวจเพียงข้อมูลทดสอบบางชุด เพื่อป้องกันการ reverse engineer จากการดูหมายเลขประจำข้อมูลทดสอบ อย่างไรก็ตาม จะมี full feedback สำหรับคะแนนที่ได้ในแต่ละปัญหาย่อย ผู้เข้าแข่งขันจึงสามารถมองเห็นคะแนนได้เหมือนปกติ

หากคุณให้คำตอบถูกต้องจำนวน S ข้อมูลทดสอบ (จากทั้งหมด 2^N) คุณจะได้รับคะแนนเป็นอัตราส่วน

$$\max \left(0, 1.2 \log_2 \left(\min \left(\frac{S(N+1)}{2^N N}, 1 \right) \right) + 1 \right)$$

ของคะแนนเต็มของปัญหาย่อยนั้น

ตัวอย่าง

พิจารณาตัวอย่างแรกที่ $N = 3$ ดังนี้ สมมติว่าสีของหมวกของแต่ละคนระบุในอาร์เรย์ C ดังนี้

```
C = [1, 0, 1]
```

เกรตเตอร์จะทำการเรียก

```
think(3, 0, [-1, 0, 1])
```

ในโปรแกรมแรก

เรียก

```
think(3, 0, [1, -1, 1])
```

ในโปรแกรมที่สอง

และเรียก

```
think(3, 0, [1, 0, -1])
```

ในโปรแกรมที่สาม

- คุณสามารถคืนค่า -1 ในโปรแกรมแรก คืนค่า 0 ในโปรแกรมที่สอง และคืนค่า -1 ในโปรแกรมที่สาม ก็จะถือว่าตอบถูกต้อง
- นอกจากนี้ คุณสามารถคืนค่า 1 ในโปรแกรมแรก คืนค่า -1 ในโปรแกรมที่สอง และคืนค่า 1 ในโปรแกรมที่สาม ก็จะถือว่าตอบถูกต้องเช่นกัน
- อย่างไรก็ตาม หากคุณคืนค่า 0 ในโปรแกรมแรก จะถือว่าตอบผิดทันที (ไม่ว่าโปรแกรมที่ 2 และ 3 จะตอบอย่างไรก็ตาม)
- และหากทุกโปรแกรมคืนค่า -1 ก็จะถือว่าตอบผิดเช่นกัน

เกรตเตอร์ตัวอย่าง

เกรตเตอร์ตัวอย่างอ่านข้อมูลนำเข้าในรูปแบบนี้:

- บรรทัดที่ 1: N hats โดย hats จะเป็นบิตสตริงความยาว N โดยช่องที่ i แทนสีของหมวกของคนที่ i สำหรับ i ตั้งแต่ 0 ถึง $N - 1$

โดยเกรตเตอร์ตัวอย่างจะทำงานดังนี้

- เรียกฟังก์ชัน `think` ให้ครบทุกคน จำนวน N รอบพอดี
- บรรทัดแรกส่งออกสตริงความยาว N โดยตัวที่ i ระบุ `_` หากคนที่ i ตอบว่า "ไม่รู้" ระบุ `0` หากคนที่ i ตอบ `0` และระบุ `1` หากคนที่ i ตอบ `1`
- บรรทัดที่สองส่งออกผลตรวจของสตริงดังกล่าว โดยจะส่ง `Correct Answer` หากมีอักขระอย่างน้อย 1 ตัวที่ตรงกับบิตสตริง hats และจะส่ง `Wrong Answer` หากไม่มีการส่งอักขระอื่นใดนอกจาก `_` เลย หรือเมื่อมีอักขระบางช่อง (ที่ไม่ใช่ `_`) ผิดไปจากบิตสตริง

ข้อจำกัด

- Time limit: 1 second
- Memory limit: 64 MB