



เดาบ้านนักพนัน

ถนนเส้นหนึ่งซึ่งมีบ้านเรียงอยู่บนถนนทั้งหมด N บ้านแต่ละหลังกำกับด้วยหมายเลข 0 ถึง $N - 1$ จากซ้ายไปขวา ในบ้านเหล่านี้จะมีอยู่หนึ่งหลังพอดีที่ล็อกเปิดเว็บไซต์พนันบอลออนไลน์อยู่ เราต้องการตรวจจับบ้านหลังดังกล่าว

เพื่อให้ไม่ให้เจ้าของบ้านรู้ตัว เรามีอุปกรณ์ตรวจวัดพิเศษอยู่เรียกว่าอุปกรณ์สะท้อนสัญญาณนั้น เราสามารถใช้อุปกรณ์นี้ได้โดยการบอกรายการของบ้านที่ต้องการตรวจวัด และระบุรูปแบบในการตรวจวัดของบ้านแต่ละหลัง เรามีวิธีการตรวจวัดอยู่สองรูปแบบ คือ แบบแดง และ แบบน้ำเงิน เมื่อเราใช้อุปกรณ์นี้ในแต่ละครั้ง อุปกรณ์นี้จะให้ผลลัพธ์ที่เป็นไปได้อยู่ 4 รูปแบบคือ

- ผลลัพธ์เป็นสีแดง ซึ่งจะเกิดขึ้นเมื่อบ้านที่เปิดเว็บไซต์พนันออนไลน์อยู่ในรายการบ้านที่เราตรวจวัดในครั้งนั้นด้วยรูปแบบสีแดง
- ผลลัพธ์เป็นสีน้ำเงิน ซึ่งจะเกิดขึ้นเมื่อบ้านที่เปิดเว็บไซต์พนันออนไลน์อยู่ในรายการบ้านที่เราตรวจวัดในครั้งนั้นด้วยรูปแบบสีน้ำเงิน
- ผลลัพธ์เป็นสีเหลือง ซึ่งจะเกิดขึ้นเมื่อบ้านที่เปิดเว็บไซต์พนันออนไลน์อยู่ห่างจากบ้านในรายการบ้านที่เราตรวจวัดในครั้งนั้น 1 หน่วยพอดี (กล่าวคือ อยู่ติดกันนั่นเอง)
- ในกรณีอื่น ๆ ผลลัพธ์จะเป็นสีดำ

จงเขียนโปรแกรมเพื่อหาว่าบ้านที่เปิดเว็บไซต์พนันคือบ้านหมายเลขที่เท่าไร คะแนนที่ได้จะขึ้นอยู่กับจำนวนครั้งที่ใช้ในการเรียกใช้อุปกรณ์ตรวจวัดพิเศษนี้

นอกจากนี้ ในบางชุดข้อมูลทดสอบ อุปกรณ์ของเราอาจจะโดนจำกัดให้มีรูปแบบการตรวจวัดเพียงสีแดงสีเดียวก็เป็นได้

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้:

```
int find_answer(int N, int K)
```

- N : จำนวนบ้าน
- K : จำนวนรูปแบบการตรวจวัดที่สามารถใช้ได้
- ฟังก์ชันนี้ จะถูกเรียกใช้เพียงครั้งเดียวพอดีและจะต้องคืนค่าหมายเลขบ้านที่เปิดเว็บไซต์พนัน

ฟังก์ชัน `find_answer` สามารถเรียกใช้ฟังก์ชันต่อไปนี้ได้

```
char ask(vector<pair<int, char>> query)
```

- `query` เป็นรายการของคู่อันดับ X, Y โดยที่ X คือหมายเลขบ้านที่ต้องการตรวจวัด และ Y คือรูป

แบบการตรวจวัด

- ในกรณีที่ $K = 1$ นั้น Y ต้องมีค่าเป็น 'R' เท่านั้น
- ในกรณีที่ $K = 2$ นั้น Y ต้องมีค่าเป็น 'R' หรือ 'B' เท่านั้น ซึ่งแทนการตรวจวัดด้วยรูปแบบสีแดงและน้ำเงินตามลำดับ
- ในการเรียกแต่ละครั้งนั้น X ต้องไม่ซ้ำกันเลยและเรียงจากน้อยไปมาก
- คุณไม่สามารถเรียกฟังก์ชันนี้ได้เกิน Q ครั้ง
- หากการเรียก ask ผิดเงื่อนไขข้างต้นนี้ โปรแกรมจะหยุดทำงาน
- ฟังก์ชันนี้จะคืนค่าผลลัพธ์เป็นอักขระดังต่อไปนี้
 - 'R' เมื่อผลลัพธ์เป็นสีแดง
 - 'B' เมื่อผลลัพธ์เป็นสีน้ำเงิน
 - 'C' เมื่อผลลัพธ์เป็นสีเหลือง
 - 'X' เมื่อผลลัพธ์เป็นสีดำ

ปัญหาย่อย

1. (5 คะแนน) $N = 1024, K = 1, Q = 1024$
2. (9 คะแนน) $N = 1024, K = 1, Q = 10$
3. (22 คะแนน) $N = 729, K = 1, Q = 6$
4. (6 คะแนน) $N = 1024, K = 2, Q = 512$
5. (15 คะแนน) $N = 729, K = 2, Q = 6$
6. (43 คะแนน) $N = 1024, K = 2, Q = 5$

ตัวอย่าง

พิจารณาการเรียกฟังก์ชันต่อไปนี้

```
find_answer(5, 2)
```

มีบ้านอยู่ 5 หลัง สมมติให้บ้านหลังที่เปิดพนันบอลคือบ้านหมายเลข 4

ฟังก์ชัน `find_answer` เรียกใช้ `ask({{0, 'R'}}, {1, 'B'})` เพื่อตรวจสอบบ้านหมายเลข 0 ด้วยรูปแบบแดง และบ้านหมายเลข 1 ด้วยรูปแบบน้ำเงิน เนื่องจากทั้งสองบ้านนั้นไม่ได้เป็นบ้านที่เปิดพนันและไม่ได้อยู่ติดกับบ้านที่เปิดพนัน จึงได้ผลลัพธ์เป็น 'X'

หลังจากนั้นฟังก์ชัน `find_answer` เรียกใช้ `ask({{2, 'R'}}, {3, 'B'}, {4, 'B'})` เพื่อตรวจสอบบ้านหมายเลข 2 ด้วยรูปแบบแดง และบ้านหมายเลข 3 และ 4 ด้วยรูปแบบน้ำเงิน เนื่องจากบ้านหมายเลขสี่เป็นบ้านที่เปิดพนันจึงได้ผลลัพธ์เป็น 'B'

หลังจากนั้นฟังก์ชัน `find_answer` เรียกใช้ `ask({{3, 'R'}})` เพื่อตรวจสอบบ้านหมายเลข 3 ด้วยรูปแบบแดง เนื่องจากบ้านหมายเลข 3 ไม่ได้เปิดพนันแต่อยู่ติดกับบ้านที่เปิดพนัน จึงได้ผลลัพธ์เป็น 'C'

ฟังก์ชัน `find_answer` ทราบว่าผลลัพธ์คือบ้านหมายเลข 4 จึงคืนค่า 4 และโปรแกรมจะหยุดทำงาน

เกร็ดเดอร์ดตัวอย่าง

- บรรทัดที่ 1 : N, K, Q

เกรดเดอร์จะแสดงผลว่าคำตอบถูกหรือไม่และจำนวนครั้งที่เรียกฟังก์ชัน `ask`

ข้อจำกัด

- Time limit: 0.5 second
- Memory limit: 128 MB