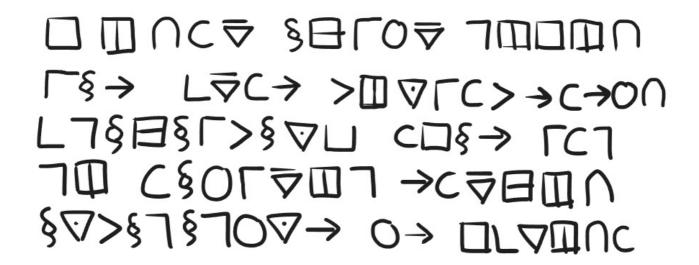


# ภาษาเอเลี่ยน



จากการสำรวจระหว่างจักรวาล (intergalactic exploration) นักผจญภัย ได้ค้นพบสิ่งมีชีวิตมหัศจรรย์อันน่าทึ่ง มี นามว่า MARK (Mysteriously Anomalous Reactive Kind) ผู้สื่อสารด้วยภาษาเอเลี่ยน ในภาษานี้ จะมีส่วน ประกอบที่ย่อยที่สุดคือ "คำ" (word) แต่ละคำจะเขียนด้วยอักขระรูปภาพที่แตกต่างกันออกไป แต่ความพิสดารของ ภาษาเอเลี่ยนนี้ คือเอเลี่ยนสามารถนำคำสองคำมา "ผสม" กันได้ กลายเป็นคำใหม่ หรือสามารถนำคำมา "หักลบ" กันได้ กลายเป็นคำใหม่อีกเช่นกัน โดยไวยากรณ์ของภาษา จะมีเงื่อนไขดังนี้

- 1. หาก a และ b เป็นคำ เราสามารถขอให้เอเลี่ยนรวมคำ a กับ b ได้ โดยเขียนผลรวมเป็น a+b
- 2. หาก a และ b เป็นคำ เราสามารถขอให้เอเลี่ยนหักลบคำ b ออกจาก a ได้ โดยเขียนผลลบเป็น a-b
- 3. หาก a และ b เป็นคำ จะได้ว่า a+b=b+a เสมอ
- 4. หาก a,b และ c เป็นคำ จะได้ว่า a+(b+c)=(a+b)+c เสมอ
- 5. มีคำพิเศษ เขียนแทนด้วย  $\emptyset$  โดยที่ สำหรับคำ a ใดๆ ก็ตาม  $\emptyset+a=a+\emptyset=a$  เสมอ
- 6. สำหรับคำ a ใด ๆ จะได้ว่า  $\emptyset (\emptyset a) = a$  เสมอ
- 7. สำหรับคำ a ใด ๆ จะได้ว่า  $a-a=\emptyset$  เสมอ
- 8. มีคำพิเศษอีกคำหนึ่ง คือคำ s ที่ทุกคำ a จะสามารถเขียนในรูป  $s+s+s+\cdots+s$  (บวกกันจำกัด ครั้ง) โดยเขียนได้แบบเดียวเท่านั้น

เพื่อความง่าย เราใช้สัญลักษณ์ [n]a แทนการบวกกันของ a จำนวน n ครั้ง กล่าวคือ

$$[n]a := \underbrace{a + a + a + \cdots + a}_{n}$$

และให้  $[0]a=\emptyset$  สำหรับทุกคำ a

ไอซี่ นักเคมีแห่งคณะสำรวจจักรวาล ต้องการจะศึกษาภาษาเอเลี่ยนนี้เอง จึงลองคิดที่จะสร้างคำขึ้นมาคำหนึ่ง เรียก ว่าคำ C โดยทราบมาว่า C=[p]s อย่างไรก็ตาม เขาง่วงมาก จึงจำได้แค่ว่า เอเลี่ยนเคยคุยกับเขาแค่สองคำ คือ A

(แปลว่า "มาแล้ว") กับ B (แปลว่า "แพะ") และรู้ข้อมูลมาอีกด้วยว่า A=[n]s และ B=[m]s

โชคดี ที่อารยธรรมของเอเลี่ยนนั้นช่างสุดยอด เอเลี่ยนนั้นได้สร้างเครื่องจักรคำ (word machine) เอาไว้สอง ประเภท คือ

- 1. เครื่องจักรบวก รับคำ a และคำ b เข้ามา แล้วส่งออกคำ a+b
- 2. เครื่องจักรลบ รับคำ a และคำ b เข้ามา แล้วส่งออกคำ a-b

ปกติแล้ว นักเคมีไอซี่ ก็จะเดินไปถามเอเลี่ยน (MARK) ตรง ๆ ว่า C คืออะไร แต่น่าเสียดายที่เอเลี่ยน (MARK) นั้นยุ่ง มากเพราะกำลังศึกษากลศาสตร์ควอนตัมอยู่ เอเลี่ยน (MARK) จึงปล่อยเครื่องจักรบวกกับเครื่องจักรลบมาให้ จำนวน  $5\,000\,000\,$  ชิ้น ให้ไอซี่ทำการต่อเครื่องจักรนี้เข้ากันเป็นวงจร เพื่อที่จะรับคำ A กับ B ไปแล้วคืนค่าคำ C ออกมา

แน่นอนว่า ไอซี่ นักเคมีอัจฉริยะผู้นี้ ก็คงไม่อยากจะใช้เครื่องจักรอย่างสิ้นเปลือง จึงขอให้คุณ ซึ่งเป็นผู้เชี่ยวชาญ ด้านการต่อวงจร ทำการออกแบบวงจรที่จะทำให้ได้คำตอบออกมา (ดูตัวอย่างประกอบเพื่อดูวิธีการต่อวงจร)

โอ้ว ลืมบอกไป! บางครั้ง สิ่งที่ไอซื่อยากรู้อยากเห็น ก็ไม่สามารถหาได้ (กล่าวคือ ไม่สามารถต่อวงจรที่รับ A กับ B แล้วส่งออก C ได้) หากเกิดกรณีนี้ขึ้น ให้ไอซื่ทำการตะโกนเสียงดัง ๆ ออกมาว่า

-1

## รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันดังต่อไปนี้

int make\_circuit(int n, int m, int p)

- ฟังก์ชันนี้จะถูกเรียกเพียงครั้งเดียว
- คุณสามารถเรียกใช้ฟังก์ชัน add\_machine และฟังก์ชัน minus\_machine ได้รวมกันไม่เกิน 5 000 000 ครั้ง
- ullet เริ่มต้นจะมีคำ  $\emptyset$  อยู่ที่ตำแหน่ง 0, คำ A อยู่ที่ตำแหน่ง 1 และมีคำ B อยู่ที่ตำแหน่ง 2
- หากมีวิธีที่จะต่อวงจรได้ ฟังก์ชันนี้จะต้องคืนค่าตำแหน่งของคำ C แต่หากไม่มีวิธีที่ต่อวงจรได้ ฟังก์ชันนี้จะ ต้องคืนค่า -1

ฟังก์ชันที่สามารถเรียกใช้งานได้

int add machine(int src1, int src2)

- ฟังก์ชันนี้จะรับค่าตำแหน่ง src1 และตำแหน่ง src2 หลังจากนั้น จะนำค่าที่ได้จากตำแหน่ง src1 มาบวก กับค่าที่ได้จาก src2 ใส่ในตำแหน่งผลลัพธ์
- ฟังก์ชันนี้จะคืนค่าตำแหน่งของผลลัพธ์ออกมา

int minus machine(int src1, int src2)

- ฟังก์ชันนี้จะรับค่าตำแหน่ง src1 และตำแหน่ง src2 หลังจากนั้น จะนำค่าที่ได้จากตำแหน่ง src1 มาลบ กับค่าที่ได้จาก src2 ใส่ในตำแหน่งผลลัพธ์
- ฟังก์ชันนี้จะคืนค่าตำแหน่งของผลลัพธ์ออกมา

## ข้อจำกัด

- $1 \le n, m, p \le 100000$
- ด้วยข้อจำกัดของเครื่องจักรบวกและลบ ข้อมูลนำเข้าจะต้องเป็นคำศัพท์ภายในเซต  $\{[n]s\colon n\in\mathbb{Z}; -1\,000\,000\,000\le n\le 1\,000\,000\,000\}$  เท่านั้น

# ปัญหาย่อย

หมายเหตุ. สำหรับทุกปัญหาย่อย จะมีการให้คะแนนบางส่วน ดูในส่วนของการให้คะแนน สำหรับเงื่อนไขการให้ คะแนน

- 1. (20 คะแนน) n=m=1
- 2. (20 คะแนน) p=1
- 3. (20 คะแนน) n=100 และ m=1
- 4. (40 คะแนน) ไม่มีเงื่อนไขเพิ่มเติม

## การให้คะแนน

สำหรับข้อมูลทดสอบแต่ละข้อมูล จะมีการให้คะแนนบางส่วน ดังนี้

- ullet หากคำ C สามารถหาได้ แต่ผู้เข้าแข่งขันตอบว่า -1 จะได้คะแนน 0%
- หากคำ C สามารถหาได้ และผู้เข้าแข่งขันตอบตำแหน่ง ที่เมื่อนำไปตรวจแล้วได้ค่า C จริง จะได้คะแนนดัง ตารางด้านล่าง
- ullet หากคำ C สามารถหาได้ แต่ผู้เข้าแข่งขันตอบอะไรอย่างอื่นที่ไม่ใช่ -1 ที่เมื่อนำไปตรวจแล้วได้ค่าไม่ตรงกับ ค่า C จะได้คะแนน 15%
- ullet หากคำ C ไม่สามารถหาได้ แต่ผู้เข้าแข่งขันตอบอะไรก็ตามที่ไม่ใช่ -1 จะได้คะแนน 0%
- ullet หากคำ C ไม่สามารถหาได้ และผู้เข้าแข่งขันตอบ -1 จะได้คะแนน 100%

และคะแนนในแต่ละปัญหาย่อยจะเป็น **ค่าน้อยสุด** ของคะแนนในแต่ละข้อมูลทดสอบในปัญหาย่อยนั้น

สำหรับตารางถัดไปนี้ จะเป็นเกณฑ์การให้คะแนนกรณีที่ตอบถูกต้อง โดยให้ M แทนจำนวนเครื่องจักรทั้งหมดที่มีการเรียกใช้

เงื่อนไข	คะแนน (ร้อยละ)
$64 \leq M \leq 5000000$	$20+rac{5120}{M}$
$M \leq 63$	100

#### ตัวอย่าง

#### make\_circuit(2, 3, 5)

#### ตอนแรก คุณมีคำ

- 🛭 อยู่ที่ตำแหน่ง 0
- ullet A=[2]s อยู่ที่ 1
- ullet B=[3]s อยู่ที่ 2

คุณสามารถเรียก minus\_machine (2, 1) เพื่อหาค่า B-A=[3]s-[2]s=[1]s ได้ สมมติว่าฟังก์ชันนี้ คืนค่า 112 เราสามารถเรียก add\_machine (112, 2) ได้ เพื่อหาค่าของ [3]s+[1]s=[4]s สมมติว่า ฟังก์ชันนี้คืนค่า 42 ต่อมาทำการเรียก add\_machine (42, 112) เพื่อหาค่า [4]s+[1]s=[5]s สมมติ ฟังก์ชันนี้คือค่า 25 เราสามารถคืนค่า 25 ได้ทันที เพื่อเป็นการบอกว่า ค่าของคำ C อยู่ที่ตำแหน่ง 25

```
make circuit(4, 6, 1)
```

ฟังก์ชันต้องคืนค่า -1

## เกรดเดอร์ตัวอย่าง

เกรดเดอร์ตัวอย่างจะรับข้อมูลนำเข้าดังนี้

บรรทัดที่ 1: n m p

เกรดเดอร์ตัวอย่างจะส่งออกสองบรรทัด

- บรรทัดที่ 1: หากการเรียกใช้ฟังก์ชันไม่มีปัญหา จะส่งออกค่า output : x เมื่อพบค่า [x]s ในตำแหน่งที่ คืนค่ามาจากฟังก์ชัน หรือ null หากฟังก์ชันคืนค่า -1 หากมีปัญหา จะเป็นการแจ้งว่าผิดเงื่อนไขประเภท ใด
- ullet บรรทัดที่ 2: จะระบุ machines count: ตามด้วยจำนวนเครื่องจักรที่มีการเรียกใช้ทั้งหมด

#### ขอบเขต

Time limit: 2 secondsMemory limit: 512 MB