

2559_2_String_Q3

ข้อนี้คล้ายกับการทำงานแบบ rot-13 ในสไลด์ โดยในการถอดรหัสข้อความลับ จะต้องอ่าน “ระยะห่างของตัวอักษร” (offset) และ “จำนวนตัวอักษรที่ต้องการใช้ระยะห่างค่านี้” (c_num) จากในข้อความ โดยค่าระยะห่างของตัวอักษร และจำนวนตัวอักษรจะต้องมากกว่า 0 แต่ไม่เกิน 9 เช่น หากข้อความรหัสข้อความลับเป็น "35Ebiil" ตัวเลขตัวแรกจะแทน “ระยะห่างของตัวอักษร” ซึ่งมีค่าเท่ากับ 3 ตัวเลขตัวถัดไป จะแทนจำนวนตัวอักษรที่ต้องการใช้ระยะห่างค่านี้ ซึ่งมีค่าเท่ากับ 5 แปลว่า ใน 5 ตัวอักษรต่อไปนี้จะใช้ระยะห่างของตัวอักษรเป็น 3 ทำให้ถอดรหัสได้เป็น

1. E->H (EFGH)
2. b->e (bcde)
3. i->l (ijkl)
4. i->l (ijkl)
5. l->o (lmno)

ซึ่งการถอดรหัสแต่ละชุด สามารถนำมาต่อกันได้ เช่น "35Ebiil25Umpjb" จะถอดรหัสได้เป็น "HelloWorld" สำหรับตัวอักษรตัวพิมพ์เล็กให้ถอดรหัสเป็นตัวพิมพ์เล็ก ตัวพิมพ์ใหญ่ให้ถอดรหัสเป็นตัวพิมพ์ใหญ่ ส่วนตัวเลขให้ถอดรหัสเป็นตัวเลข เช่น "15483912578909" จะถอดได้เป็น "5940290121" ส่วนเครื่องหมายวรรคตอนต่าง ๆ ให้คงไว้เหมือนเดิม หากข้อความต้นฉบับเป็นตัวอักษรท้าย ๆ เช่น XYZ จะต้องนับวนเหมือนแบบ rot-13 คือ วนกลับไป ABC ซึ่งตัวเลขก็เช่นกัน จะวนจาก 789 กลับไปยัง 012 เช่น หาก offset มีค่าเป็น 2 แล้ว X->Z, Y->A, Z->B, 7->9, 8->0, และ 9->1 ตามลำดับ

นิสิตสามารถใช้โปรแกรมด้านล่างนี้เพื่อเป็นแนวทางในการเขียนโปรแกรมได้

```
s = input().strip()
rot_str = _____ # ใช้เพื่อหาตัวอักษรที่อยู่ตามระยะห่างตามที่กำหนด
count = 0 # ใช้เพื่อนับว่าตัวอักษรในชุดนั้นถูกถอดรหัสครบแล้วหรือไม่
out = ''
offset = -1 # ในกรณีที่ offset และ c_num เป็น -1 หมายถึงเป็นการเริ่มต้นของชุดตัวอักษรที่ใช้ระยะห่างชุดนี้
c_num = -1
for c in s:
    if offset == -1:
        _____ # ในกรณีที่ offset เป็น -1 แสดงว่าตัวอักษร c ต้องเป็นค่า offset
    elif c_num == -1:
        _____ # ในกรณีที่ offset ไม่เป็น -1 แต่ c_num เป็น -1 แสดงว่าตัวอักษร c ต้องเป็นค่า c_num
    else:
        count += 1
        if c not in rot_str:
            out += c # กรณีที่ตัวอักษรเป็นเครื่องหมายวรรคตอน
        else:
            _____ # กรณีที่ตัวอักษรตัวที่ต้องถอดรหัส
            _____
            if _____ # เมื่อครบตัวอักษรหนึ่งชุด ให้กำหนดค่าตัวแปรต่าง ๆ เป็นค่าเริ่มต้นใหม่
                count = _____
                offset = _____
                c_num = _____
print(out)
```

ข้อมูลนำเข้า

เป็นข้อความหนึ่งบรรทัด เป็นข้อความที่ต้องการถอดรหัส

กรณีทดสอบแบ่งเป็น

- มีระยะห่างและจำนวนตัวอักษรเพียงชุดเดียวเช่น "35**ebiil**" และเป็นตัวพิมพ์เล็กทั้งหมด คิดเป็น 10%
- มีระยะห่างและจำนวนตัวอักษรเพียงชุดเดียวเช่น "35**ebiil**" และเป็นตัวพิมพ์เล็กปนกันกับตัวพิมพ์ใหญ่ คิดเป็น 10%
- มีระยะห่างและจำนวนตัวอักษรเพียงชุดเดียวเช่น "1548391" และเป็นตัวเลขทั้งหมด คิดเป็น 10%
- มีระยะห่างและจำนวนตัวอักษรเพียงชุดเดียวเช่น "1748391**ab**" และมีตัวเลข ปนกับตัวพิมพ์ใหญ่และตัวพิมพ์เล็ก 10%
- มีระยะห่างและจำนวนตัวอักษรหลายชุด และมีตัวพิมพ์เล็ก ตัวพิมพ์ใหญ่ และตัวเลข ปนกัน 60%

ข้อมูลส่งออก

ให้แสดงข้อความที่ถอดรหัสแล้ว

ตัวอย่าง

| input (จากแป้นพิมพ์) | output (ทางจอภาพ) | หมายเหตุ |
|--------------------------------------|-------------------|--|
| 35 Ebiil | Hello | |
| 35 Ebiil 25 Umpjb | HelloWorld | |
| 36 Ebiil 25 Umpjb | Hello World | ให้คงตัวเว้นวรรคไว้ |
| 35 Ebiil 26 Umpjb | Hello World | ให้คงตัวเว้นวรรคไว้ |
| 1548391 2578909 | 5940290121 | 5 ตัวแรกใช้ระยะห่างเป็น 1 5 ตัวถัดมาใช้ระยะห่างเป็น 2 |
| 44 ez: 1548391 2578909 | id: 5940290121 | ให้คงตัวเว้นวรรคและ ":" ไว้ |