

## แทนที่ข้อความแบบซับซ้อน (complex replace)

เขียนฟังก์ชัน `complex_replace(s,k_strs,r_strs)` เพื่อแทนที่ข้อความจากลิสต์ของข้อความที่ต้องการค้นหาและแทนที่ โดยฟังก์ชันที่รับอินพุตสามอย่าง คือ ข้อความ (s) ลิสต์ของข้อความที่ต้องการค้นหา (k\_strs) และลิสต์ของข้อความที่ต้องการแทนที่ (r\_strs) เมื่อมีการแทนที่ ให้ใส่เครื่องหมาย '<' และ '>' ครอบข้อความที่ได้รับการแทนที่ โดยหากพบหลายที่ให้แทนที่เฉพาะข้อความที่จุดเริ่มต้นอยู่ตำแหน่งซ้ายสุดเท่านั้น โดยรับประกันว่า จะไม่มีข้อความที่พบซึ่งมีจุดเริ่มต้นอยู่ในตำแหน่งเดียวกัน ตัวอย่างเช่น

```
complex_replace('java world',['war','wor','java'],['War','Wor','C'])
```

ฟังก์ชันนี้จะค้นหาว่าในข้อความ 'java world' มีคำว่า 'war', 'wor' และ 'java' อยู่หรือไม่ ถ้ามีก็แทนด้วยคำจากลิสต์ r\_strs ที่มีตำแหน่งตรงกับที่ค้นพบใน k\_strs จะเห็นได้ว่ามีข้อความ java และ wor ปรากฏอยู่ ให้เลือกแทนที่คำว่า java เนื่องจากมาก่อนคำว่า wor ทำให้ได้ผลลัพธ์เป็น '<C> world'

ตัวอย่างเช่น

Input	k_strs	r_strs	ผลลัพธ์
<code>print(complex_replace('java world',['a','o'],['A','O']))</code>			j<A>va world
<code>print(complex_replace('java world',['j','w'],['J','W']))</code>			<J>ava world
<code>print(complex_replace('ten cars',['car','ten'],['c','t']))</code>			<t> cars
<code>print(complex_replace('java world',['car','ten'],['c','t']))</code>			java world

## งานของคุณ

เขียนฟังก์ชัน `complex_replace(s,k_strs,r_strs)` ตามรูปแบบด้านล่าง

```
def complex_replace(s,k_strs,r_strs):  
    ...  
  
# ห้ามลบหรือแก้ไขบรรทัดด้านล่างนี้  
exec(input())
```