

ปัญหา ลบอักขระที่ซ้ำกันใด ๆ ออกไป (RemoveAnyDup) [5 หรือ 10 คะแนน] [หน่วยความจำ 32 MB, เวลาคำนวณ 1 วินาที]

กำหนดสตริงให้อันหนึ่งความยาวไม่เกิน 500 อักขระ เราต้องการกำจัดอักขระที่ซ้ำออกไป เหลือไว้แต่ตัวที่ปรากฏเป็นครั้งแรก เช่น ถ้าหากสตริงคือ **MONOTONE** เราจะพบว่าตัวอักขระที่มีการซ้ำกันคือ **O** และ **N** โดย **O** มีสามตัว และ **N** สองตัว ซึ่งเราจะเก็บไว้เฉพาะตัวที่ปรากฏเป็นครั้งแรก และผลลัพธ์ที่ต้องการคือ **MONTE**

แผน ก สำหรับคนที่ไม่ต้องการเขียนแบบเรียกซ้ำ (Recursive) จะได้สูงสุด 5 คะแนน และหน้าที่ของคุณคือ “จงเขียนโปรแกรมที่พิมพ์อักขระออกมาในลักษณะตามข้อกำหนดดังกล่าว”

แผน ข สำหรับคนที่เลือกที่จะเขียนแบบเรียกซ้ำ จะได้สูงสุด 10 คะแนน โดยข้อกำหนดคือ

1. คุณจะต้องเขียนเมธอดแบบเรียกซ้ำที่มีรูปแบบเป็น **String removeAnyDup(String str)**
2. เมธอด **removeAnyDup** ไม่มีการรับข้อมูลเข้าใด ๆ เพิ่มเติม และห้ามใช้ลูป
3. เมธอด **removeAnyDup** ไม่มีการอ้างถึงข้อมูลที่อยู่นอกเหนือจากพารามิเตอร์และข้อมูลที่ประกาศไว้ภายในตัวเมธอดเอง
4. เมธอด **removeAnyDup** ไม่มีการพิมพ์ผลลัพธ์ใด ๆ ออกมา แต่จะต้องคืนสตริงที่เป็นผลลัพธ์ไปให้เมธอด **main** และเมธอด **main** จะพิมพ์ผลลัพธ์ดังกล่าวออกมาเอง
5. ตัวโปรแกรมของคุณจะต้องมีลักษณะตามที่ให้ไว้ทางด้านล่างนี้ คุณสามารถแก้ไขได้เฉพาะโค้ดภายในเมธอด **removeAnyDup** เท่านั้น (รวมถึงอาจแก้ไขคลาสได้ ถ้าจำเป็น)

```
public class RemoveAnyDup {
    String removeAnyDup(String str) {
        // ใส่โค้ดของคุณอยู่ที่นี่ และห้ามแก้ไขโค้ดนอกเมธอด ยกเว้นแก้ไขคลาสเพื่อให้ทำงานกับ
        // ระบบตรวจอัตโนมัติได้
    }

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        String str = scan.next();
        RemoveAnyDup remover = new RemoveAnyDup ();
        String result = remover.removeAnyDup(str);
        System.out.print(result);
    }
}
```

ตัวอย่าง

ข้อมูลเข้า	ผลลัพธ์	ข้อมูลเข้า	ผลลัพธ์
SILAPIN	SILAPN	MONOTONE	MONTE
ABCCBD	ABCD	SOURCECODE	SOURCED

[มีคำแนะนำในหน้าถัดไป]

คำแนะนำสำหรับแผน ก

1. คุณไม่จำเป็นต้องคำนวณสตริงผลลัพธ์ออกมา แต่ใช้วิธีพิมพ์อักขระที่ปรากฏเป็นครั้งแรกออกมาทีละตัว ๆ ก็จะทำให้สำเร็จได้
2. ถ้าอยากรู้ว่าอักขระที่กำลังพิจารณาอยู่นั้นเป็นตัวแรกหรือเปล่า ลองใช้ `indexOf` แล้วดูว่าตำแหน่งที่ได้จาก `indexOf` กับตำแหน่งของอักขระที่พิจารณาอยู่นี้เป็นค่าเดียวกันหรือไม่ ถ้าเป็นค่าเดียวกันแสดงว่านี่เป็นอักขระที่ปรากฏเป็นครั้งแรก

คำแนะนำสำหรับแผน ข

สมมติว่าคุณพิจารณาอักขระทางด้านท้ายสุดของ `str` คุณอาจจะพบว่าถ้าคิดตามแนวทางที่ให้ทางด้านล่างนี้อาจจะทำให้ปัญหายุ่งยากนัก

1. เราจะเลือกที่จะตัดอักขระทางท้ายสุดออกหรือเก็บไว้ ดังนี้ (1) ถ้าอักขระทางท้ายสุดคือการปรากฏเป็นครั้งแรก เราจะเก็บมันไว้ (2) แต่ถ้าอักขระท้ายสุดนั้นไม่ใช่การปรากฏครั้งแรกเราจะตัดทิ้ง
2. ไม่ว่าจะเก็บไว้หรือตัดทิ้ง เราจะเรียกเมธอด `removeAnyDup` ซ้ำด้วยการส่งพารามิเตอร์ที่เป็นสตริงที่ยาวน้อยกว่าเดิม 1 อักขระ ซึ่งวิธีที่จะทำให้เราได้สตริงที่สั้นลงหนึ่งอักขระทำได้ด้วยการใช้เมธอด `substring` ที่อยู่ในคลาส `String`

ซึ่งในกรณีนี้เราอาจจะเขียนโค้ดว่า `str.substring(0, n-1)` เพื่อสร้างสตริงใหม่ที่สั้นลงน้อยกว่าเดิม 1 อักขระ เมื่อ `n` คือความยาวของ `str` (อย่าลืมว่าคำสั่งนี้สร้างสตริงใหม่ขึ้นมา ไม่ได้ตัดต่อ `str` แต่อย่างใด)

3. จากข้อ 1 คำว่า “เก็บมันไว้” หมายความว่า เราจะนำอักขระนี้ไปต่อท้ายผลลัพธ์ที่ได้มาจากการเรียกซ้ำ
4. วิธีดูว่าอักขระปรากฏครั้งแรกหรือไม่ ให้ดูคำแนะนำข้อ 2 ในแผน ก
5. อย่าลืมว่าเราต้องเขียนกรณีพื้นฐาน (`base case`) ด้วย ไม่งั้นโปรแกรมเราจะเรียกซ้ำไปเรื่อย ๆ และเมื่อมันตัดสตริงให้สั้นลงกว่าเดิมไม่ได้แล้ว มันก็เกิดเหตุการณ์ติดปรกติขึ้น