

ปัญหา ซีซาร์ชิฟต์ 2 [CaesarShift2] (6 หรือ 12 คะแนน)

[ทรัพยากรการคำนวณ: 1 วินาที, 32 MB]

ในสมัยก่อน การเข้ารหัสข้อความแบบเรียบง่ายนับว่าเป็นกลยุทธ์ที่สำคัญในการส่งข่าวต่าง ๆ ระหว่างสงคราม เพราะแม้จดหมายนั้นถูกยึดหรืออ่านโดยฝ่ายศัตรู หากฝ่ายศัตรูไม่สามารถถอดรหัสได้ การข่าวก็จะไม่รู้ว่ไหล ซึ่งหนึ่งในวิธีที่มีใช้ในวงสมัยสองพันปีที่แล้วคือวิธีของ Julius Caesar โดยจะการใช้การเปลี่ยนตัวอักษรเป็นตัวถัดไปเพื่อเข้ารหัส เช่น คำว่า **ANT** เป็น **BOU** ส่วนการถอดรหัสก็คือการเปลี่ยนเป็นตัวอักษรก่อนหน้า ดังนั้นคนอ่านข้อความ **BOU** จะต้องรู้ว่าตัวก่อนหน้า **B** คือ **A** ก่อนหน้า **O** คือ **N** และก่อนหน้า **U** คือ **T** แล้วเปลี่ยนข้อความกลับเป็น **ANT**

อย่างไรก็ตาม วิธีการเปลี่ยนตัวอักษรเป็นตัวติดกันไปมาเช่นนี้นั้นดูยุ่งยากเกินไป ในปัญหานี้ เราจะเปลี่ยนตัวอักษรอื่น ๆ ที่ไม่ต้องติดกันก็ได้ และไม่จำเป็นต้องเปลี่ยนแบบเดียวกันทุกตัวอักษรในข้อความหนึ่ง นั่นคือเราจะกำหนดระยะห่างของการเปลี่ยนให้มิได้หลายค่าในแต่ละข้อความ เช่น ถ้ากำหนดเป็นค่า 2 1 4 จะหมายความว่าตัวอักษรตัวแรกในข้อความจะเปลี่ยนเป็นอักษรที่ถัดไปสองตัว เช่น ข้อความ **ANT** ตัวแรกคือ **A** จะกลายเป็น **C** สำหรับตัวที่สองในข้อความจะเปลี่ยนเป็นอักษรที่ถัดไปหนึ่งตัว ดังนั้นจะเปลี่ยนเป็น **O** และตัวที่สามจะเปลี่ยนเป็นสี่ตัวถัดไป ดังนั้น **T** จะเปลี่ยนเป็น **X**

หากค่าที่กำหนดยังคงเป็น 2 1 4 แต่ข้อความยาวกว่านั้น ตัวที่สี่จะเริ่มวนกลับมาใช้ค่า 2 ตัวที่ 5 จะวนมาใช้ค่า 1 และตัวที่หกจะวนกลับมาใช้ค่า 4 และเป็นเช่นนี้ไปเรื่อย ๆ จนครบทุกตัวอักษรในข้อความ ส่วนกรณีที่มีการเลื่อนอักษรจะทำให้พ้นตัว **Z** ไป ตัวอักษรก็จะวนกลับมาที่ **A** ใหม่ เช่น ถ้าตัวอักษรก่อนเข้ารหัสคือ **X** และต้องเลื่อนไป 4 ตัวอักษรผลลัพธ์ที่ได้จะเป็น **B**

ตัวอย่างเพิ่มเติมเช่น **JDAXZKAYPQWXYZ** ถ้ากำหนดค่าเป็น 3 4 1 2 เราจะได้ผลการเข้ารหัสซีซาร์แบบปรับปรุงเป็น **MHBZCOBASUXZBD**

จงเขียนเมธอดที่ทำการเข้ารหัสข้อความในลักษณะดังกล่าว ซึ่งการเขียนแบบใช้เมธอดเรียกซ้ำ (Recursive Method) จะได้คะแนนมากที่สุดที่ 12 คะแนน ส่วนผู้ที่เขียนแบบธรรมดา (ซึ่งไม่ต้องมีเมธอดใดเพิ่มเติมนอกจาก **main** ในกรณีนี้) จะได้คะแนนสูงสุดที่ 6 คะแนน

แผน ก สำหรับคนที่ไม่ต้องการเขียนแบบเรียกซ้ำ หน้าที่ของคุณคือ “จงเขียนโปรแกรมที่เข้ารหัสตัวอักษรตามข้อกำหนดดังกล่าว” (คุณสามารถเขียนทุกอย่างลงในเมธอด main หรือจะแยกเมธอดออกมาก็ได้)

แผน ข สำหรับคนที่เลือกที่จะเขียนแบบเรียกซ้ำ จะได้สูงสุด 12 คะแนน โดยข้อกำหนดคือ

1. คุณจะต้องเขียนเมธอดแบบเรียกซ้ำที่มีรูปแบบเป็น

`String cshift2(String msg, int[] dist, int i)` และ ห้ามใช้ลูป

2. เมธอดนี้ไม่มีการรับข้อมูลเข้าใด ๆ เพิ่มเติม และไม่มีการพิมพ์ข้อความใดในนั้น
3. เมธอด `csift2` ไม่มีการอ้างอิงถึงข้อมูลที่อยู่นอกเหนือจากพารามิเตอร์และข้อมูลที่ประกาศไว้ภายในตัวเมธอดเอง (คือใช้ได้เฉพาะพารามิเตอร์และสามารถประกาศตัวแปรเพิ่มเติมเฉพาะภายในเมธอดเท่านั้น ห้ามสร้างตัวแปรที่เป็นสมาชิกคลาสเพิ่มเติม)
4. เมธอด `csift2` ไม่มีการพิมพ์ผลลัพธ์ใด ๆ ออกมา แต่จะต้องคืนสตริงที่เป็นผลลัพธ์ไปให้เมธอด `main` พิมพ์ผลลัพธ์ดังกล่าวออกมา
5. ตัวโปรแกรมของคุณจะต้องมีลักษณะตามที่ให้ไว้ทางด้านล่างนี้ คุณสามารถแก้ไขโค้ดได้เฉพาะภายในเมธอด `csift2` เท่านั้น (รวมถึงอาจแก้ไขชื่อคลาสได้ ถ้าจำเป็นสำหรับการทำงานกับระบบตรวจอัตโนมัติ)

```
import java.util.Scanner;
public class CaesarShift2 {
    String cshift2(String msg, int[] dist, int i) {
        // You can enter your code here and only here.
    }

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        String msg = scan.next();
        int n = scan.nextInt();
        int[] dist = new int[n];
        for(int i = 0; i < n; ++i)
```

```

        dist[i] = scan.nextInt();
        CaesarShift2 cs = new CaesarShift2();
        String ans = cs.cshift2(msg, dist, 0);
        System.out.println(ans);
    }
}

```

คำแนะนำ เมธอด `substring` ของสตริงเป็นเพื่อนที่ดีของคุณสำหรับข้อนี้

คำอธิบายและคำแนะนำเพิ่มเติม

- พารามิเตอร์ของ `cshift2(String msg, int[] dist, int i)` นั้นตัวแรกคือสตริงหลักที่เราต้องการเข้ารหัส ตัวที่สอง (`dist`) คือระยะความห่างตัวอักษรที่ต้องการเปลี่ยน เนื่องจากมีหลายค่าเราจึงส่งมาเป็นอาร์เรย์ และ `i` คือลำดับในอาร์เรย์ `dist` ที่จะหยิบมาใช้บอกระยะ ในตอนแรกเราต้องการหยิบ `dist[0]` มาใช้เป็นตัวแรก เราจึงใส่ค่านี้เป็น `0` ตอนเรียกใช้ `cshift2` ใน `main`
- การใช้พารามิเตอร์ตัวที่สามของ `cshift2` ควรส่งค่า `i` ที่เพิ่มขึ้นหนึ่งไปเรื่อย ๆ เว้นเสียมันจะไปถึงความยาวของ `dist` เราจึงกำหนดให้ค่านี้วนกลับมาที่ `0` ใหม่

ตัวอย่าง

(เลขในบรรทัดที่สองของข้อมูลเข้าคือจำนวนระยะการเปลี่ยนตัวอักษรที่จะเข้ารหัส)

ข้อมูลเข้า	ผลลัพธ์
ABCDEFGHXYZ 3 1 2 3	BDFEGIHJAZB
JDAXZKAYPQWXYZ 4 3 4 1 2	MHBZCOBASUXZBD
AQZWEUGXUZZPWWMNYTP 1 3	DTCZHXJAXCMSZZIPQBWS

หมายเหตุ สตริงข้อมูลเข้าจะยาวไม่เกิน 500 ตัวอักษร และจะมีข้อมูลเข้า 30% ที่มีจำนวนระยะการเปลี่ยนตัวอักษรเป็น 1 (ข้อมูลเข้าบรรทัดที่สองมีค่าเป็นหนึ่ง แบบนี้ทำให้เราจัดพารามิเตอร์ตัวที่สามให้เป็น 0 ค้างไว้ได้ตลอด) นอกจากนี้ ระยะความห่างในการเปลี่ยนตัวอักษรจะไม่เกิน 5