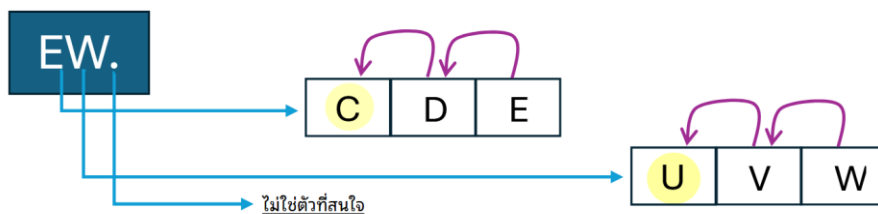


D04 – Caesar (คร่าวของครัสซัส)

Editorial by พ้อยากนอนวันละ 8 ชม.

ข้อนี้เป็นข้อหนึ่งที่น้อง ๆ จะได้เล่นกับ String หรือพวก ASCII Character มากยิ่งขึ้น นั่นคือการทำโปรแกรมเพื่อถอด/เข้ารหัสข้อความหรือที่มีอีกชื่อหนึ่งในภาษาอังกฤษว่า “Cryptography” นั่นเอง โดยในข้อนี้เพื่อความง่ายจึงให้น้อง ๆ ทำเพียงแค่การ Decryption (การถอดรหัส) เท่านั้น

โดยวิธีการเข้า/ถอดรหัสที่พี่ได้นำมาใช้ในการออกโจทย์ครั้งนี้ก็คือ “Caesar Cipher” ซึ่งมีหลักในการถอดคือเราจะต้องมีข้อความที่ต้องการถอด พร้อมกับจำนวนครั้งที่ตัวอักษรนั้นถูกเลื่อนออกไปจากตำแหน่งเดิมเสียก่อน ยกตัวอย่างเช่น เรามีตัวอักษรก่อนถอดรหัสว่า “EW” และทราบว่าคุณข้อความชุดนี้ถูกเลื่อนไปจากเดิม 2 ตำแหน่งเมื่อเราย้อนตำแหน่งกลับไป 2 ครั้งจะได้คำว่า “CU” นั่นเองดังในภาพประกอบข้างต้น



วิธีการแก้ไขข้อนี้ก็มียากหลายมาก ๆ ถึงขั้นไม่ต้องใช้ If-else ก็ย่อมได้ ในที่นี้ขออธิบายโดยใช้เนื้อหา If-else เข้ามาช่วยเพราะส่วนตัวพี่มองว่าวิธีนี้ดูเข้าใจง่ายและน่าจะตรงไปตรงมาที่สุดแล้วครับ

ประการแรก เราต้องรับค่าข้อความที่ถูกเข้ารหัสกับจำนวนเต็ม 1 ค่าที่ใช้สำหรับถอยหลังตัวอักษร (หรือใช้เพื่อถอดรหัส) เข้ามาเสียก่อน สิ่งที่ต้องระมัดระวังในข้อนี้คือมีการรับค่าโดยการเว้นวรรคสมอยู่ด้วย จึงแนะนำให้ปฏิบัติตามคำแนะนำของไฟล์ Statement เป็นอย่างยิ่ง

ประการที่สอง เราจะต้องทำการเข้าถึงตัวอักษรในข้อความที่รับเข้ามาในแต่ละตัว เพื่อใช้ในการถอดรหัสข้อความที่ได้รับ การ Input เข้ามา โดยจะทำการใช้ตัวแปรที่ถูกเข้าถึงค่าของอักขระโดยตรงก็ได้หรือจะสร้างตัวแปรมาคอยรับค่าก็ได้ ในตัวอย่างโค้ดของพี่จะใช้เป็นวิธีสร้างตัวแปรเป็นถึงคอยรับค่าแล้วกระทำการใด ๆ ในตัวแปรที่สร้างใหม่ข้างต้น

```
for (int i=0; i<encodeText.size(); i++) {  
    // Handle each character  
    char s = encodeText[i];  
  
    cout << s;  
}
```

ประการที่สาม ทำการลบค่า ASCII โดยตรงไปได้เลยหากค่านั้นถูกถอดแล้วค่าของตัวอักษรไม่ออกจากขอบเขตของ 'A' หรือ 'a' หากมองเป็นจำนวนเต็มก็คือมากกว่า 65 หรือ 97

```
for (int i=0; i<encodeText.size(); i++) {  
    // Handle each character  
    char s = encodeText[i];  
    // Detect it's alphabet or not for calculate  
    if ((s >= 'A' && s <= 'Z')) {  
        if (s - cycle >= 'A') {  
            s -= cycle;  
        } else {  
            // ...  
        }  
    } else if ((s >= 'a' && s <= 'z')) {  
        if (s - cycle >= 'a') {  
            s -= cycle;  
        } else {  
            // ...  
        }  
    }  
    cout << s;  
}
```

ประการที่สี่ ในกรณีที่ตัวอักษรดังกล่าวถูกถอยมาเกินตัว 'A' หรือ 'a' อย่างแน่นอนให้เราสมมติว่ามันเป็นตัว 'Z' หรือ 'z' เสียเลย (เพราะยังไงมันก็ต้องวนกลับมาเริ่มใหม่) แล้วจึงค่อยลบด้วยจำนวนที่ถอยออกไป แต่หากเราหิบบค่านั้นมาตรง ๆ สิ่งที่จะมีปัญหาคือถ้าตัวอักษรที่ต้องถอยไม่ใช่ 'A' หรือ 'a' สิ่งที่เกิดขึ้นหากถอยไปก็จะได้ตำแหน่งตัวอักษรที่ควรจะเป็นเพี้ยนไป ฉะนั้นแล้วเราควรต้องลบโดยการใช้สมการต่อไปนี้

ลำดับตัวอักษร Z - (จำนวนที่ถอยตัวอักษร - (ตัวอักษรข้อความปัจจุบัน - ตัวอักษรเริ่มต้น)) + 1 = ตัวอักษรที่ถูกถอด

โดยการที่นำ (ตัวอักษรข้อความปัจจุบัน - ตัวอักษรเริ่มต้น) ผลลัพธ์ที่เกิดขึ้นคือจำนวนที่ควรจะถอยหลังต่อไปหลังจากเข้าสู่ตัว 'Z' หรือ 'z' แล้ว ยกตัวอย่างให้จำนวนที่ใช้ในการถอยคือ 3 การถอยด้วยสมการข้างต้นจะได้ว่า

ตัวอักษรก่อนถอด	สมการที่ใช้ในการถอด	ตัวอักษรหลังถอด
C	$90 - (3 - (67-65)) + 1$	Z (90)
B	$90 - (3 - (66-65)) + 1$	Y (89)
A	$90 - (3 - (65-65)) + 1$	X (88)

ผลสุดท้ายจะได้ Code เป็นดังนี้

```
for (int i=0; i<encodeText.size(); i++) {  
    // Handle each character  
    char s = encodeText[i];  
    // Detect it's alphabet or not for calculate  
    if ((s >= 'A' && s <= 'Z')) {  
        if (s - cycle >= 'A') {  
            s -= cycle;  
        } else {  
            // We're cycling with character 'a' between 's' for calculate distance from last alphabet  
            s = 'Z' - (cycle - (s-'A')) + 1;  
        }  
    } else if ((s >= 'a' && s <= 'z')) {  
        if (s - cycle >= 'a') {  
            s -= cycle;  
        } else {  
            s = 'z' - (cycle - (s-'a')) + 1;  
        }  
    }  
    cout << s;  
}
```

ที่เหลือหลังจากนี้ก็จะเป็นเรื่องของการแสดงผลตัวอักษรแล้ว ขอให้ลองโค๊ดดูนะคะ <3