

แผนการสอนปฏิบัติครั้งที่ 3

- ✚ **วัตถุประสงค์** เข้าใจความสัมพันธ์ระหว่างผังงานกับโปรแกรม/เขียนโปรแกรม/ดัดแปลงโปรแกรม
- ✚ **โจทย์ปัญหา** ให้นำผังงานโปรแกรมทนายตัวเลขที่กำหนดให้ มาเขียนเป็นโปรแกรมภาษาซี
- ✚ **ผลลัพธ์การเรียนรู้**
 - นศ. สามารถแปลงผังงาน มาเป็นคำสั่งโปรแกรมภาษาซีได้
- ✚ **ขั้นตอน**
 - ทำความเข้าใจการเล่นเกมทนายตัวเลข ตามเงื่อนไขที่กำหนด (สอดคล้องกับผังงาน)
 - สุ่มตั้งตัวเลขจำนวนเต็ม 1-100 ให้ทายได้ไม่เกิน 7 ครั้ง
 - ถ้าทายถูกภายใน 7 ครั้ง สรุปว่าชนะ
 - ถ้าครั้งที่ 7 ไม่ถูกสรุปว่าแพ้ และให้เฉลยตัวเลขที่ถูกต้อง
 - ทำความเข้าใจคำสั่งที่สอดคล้องกับผังงาน
 - `srand(time(NULL)); x = rand() %100 + 1 ;` //ตัวอย่างคำสั่งสุ่มค่า x ให้มีค่า 1..100 ต้อง `#include<time.h>`
 - แปลงผังงานเป็นโปรแกรม (ต้องเขียนโปรแกรมให้สอดคล้องกับผังงานที่ออกแบบไว้)
- ✚ **งานที่ต้องส่งให้ดูในคาบ(หรือ upload ส่งใน LEB2)** หน้าจอแสดงผลที่ทดสอบ
 - การตั้งคำถามจะต้องมีการแสดงจำนวนครั้งที่กำลังทายอยู่ (1-7)
 - กรณีป้อนข้อมูลเป็นตัวเลขถูกต้อง 3 กรณี คือ
 - 1. กรณีทายถูกก่อน 7 ครั้ง
 - 2. กรณีทายถูกในครั้งที่ 7 (Cheat ให้พิมพ์ค่า x หลังจากที่ได้ random ก่อน แล้วค่อยลบทิ้ง)
 - 3. กรณีทายผิดครั้งที่ 7 จะต้องมีการเฉลยตัวเลขที่ถูกต้องด้วย
 - 4. หน้าจอกรณีที่มีการข้อมูลผิดหรือเกิน เช่น [120], [5.0], [5 6 7], [5x], [5 x] (ให้เห็นว่าเกิดอะไรขึ้นกับโปรแกรม ต้องรู้ว่าโปรแกรมทำงานผิดพลาดหรือไม่)
- ✚ **รายงานที่ต้องส่งตามหลัง** (ต้องสรุปผลการประเมินตนเองและปัญหาในการทำ Assignment ทุกครั้ง)
 - ดัดแปลงโปรแกรม(User Interface) ให้มีตั้งคำถาม และสรุปคำตอบที่อ่านเข้าใจง่าย
 - รายงานอธิบายการทำงานของโปรแกรมที่สอดคล้องกับผังงานที่เขียน และหน้าจอแสดงผลของทั้ง 3 กรณี
 - โค้ดภาษา C ของโปรแกรมที่สมบูรณ์แล้ว
- ✚ **เฉพาะนศ.ภาควิชาวิศวกรรมคอมพิวเตอร์** ต้องเพิ่มหน้าจอกรณี ป้อนข้อมูลผิดหรือเกิน เช่น [120], [5.0], [5 6 7], [5x], [5 x] ด้วย (สรุปผล/ความผิดพลาดที่ทำให้โปรแกรมทำงานไม่ตรงกับที่คาดไว้)

ลำดับวิธีคิดโปรแกรมทายตัวเลข

ผู้ตั้งคำถาม คือ คอมพิวเตอร์ (Hardware) คนที่ 2 คือ ผู้เล่น (User) ตัวเกมส์ คือ ซอฟต์แวร์ (Software)

1. คอมพิวเตอร์ **สมมุติตัวเลข 1-100** เพื่อใช้เป็นคำตอบ เริ่มต้นที่ 0 (กำหนดค่าเริ่มต้น)
2. คอมพิวเตอร์ เริ่มต้นนับเพิ่มครั้งที่ทาย (คอมพิวเตอร์บวกจำนวนครั้งเพิ่ม 1)
3. คอมพิวเตอร์ **ตั้งคำถาม** ให้ผู้ใช้ตอบตัวเลข (คอมพิวเตอร์ต้องแสดงผลคำถาม)
4. ผู้เล่นคิดตัวเลขแล้วตอบ **โดยป้อนตัวเลขให้คอมพิวเตอร์** (คอมพิวเตอร์ต้องรับตัวเลขไปเก็บ)
5. คอมพิวเตอร์ **เปรียบเทียบ** ตัวเลขของผู้เล่น กับคำตอบที่เตรียมไว้

5.1 ถ้า **เท่ากัน** แสดงว่าตอบถูก **สรุปว่าชนะเพื่อจบเกมส์** (จบเกมส์ได้)

5.2 ถ้า **ไม่เท่า** แสดงว่าตอบผิด ให้คิดต่อว่าทายมากไป หรือน้อยไป

5.2.1 ถ้าตัวเลข **มากกว่า** คำตอบ **สรุปว่าทายมากไป**

5.2.2 หรือถ้าตัวเลข **น้อยกว่า** คำตอบ **สรุปว่าทายน้อยไป**

5.2.3 ตรวจสอบถ้าผิดครบ **7 ครั้ง** ให้สรุปว่าแพ้ และเฉลยคำตอบเพื่อจบเกมส์ (ถ้ายังไม่ครบไม่ต้องทำอะไร)

6. ตรวจสอบการจบเกมส์

ถ้า **ยังไม่จบเกมส์** (ตอบผิด และน้อยกว่า 7 ครั้ง) ให้วนกลับไปทำซ้ำ ใน step 2 คือนับเพิ่มและตั้งคำถาม จนถึงสรุปใน step 5

7. กรณีจบเกมส์แล้ว (ผู้ใช้ตอบถูก หรือทายผิดครบ 7 ครั้ง) ให้จบโปรแกรม

ดัดแปลงโปรแกรมเพิ่มเติม (เมื่อทำโปรแกรมขั้นที่ 1-7 ถูกต้องแล้ว)

1. เพิ่มการแนะนำตัวของโปรแกรมว่า เป็นโปรแกรมทายตัวเลข
 2. เพิ่มคำถาม เพื่อถามชื่อผู้เล่น และมีการทักทายผู้เล่น
 3. ปรับปรุงคำถาม คำตอบ ให้มีการระบุชื่อผู้เล่นด้วย
 4. เมื่อเล่นจบ 1 ครั้ง ให้โปรแกรมเพิ่มคำถามว่าจะเริ่มเล่นใหม่หรือไม่
 - โปรแกรมรับเฉพาะปุ่ม [n] หรือ [y] เท่านั้น ถ้ากดผิดปุ่มให้รอรับใหม่
 - ถ้ากดปุ่ม [y] หรือ [Y] แสดงว่าเล่นต้องการเล่นซ้ำ ให้วนกลับไปสุ่มตัวเลขเริ่มต้นใหม่ แต่ไม่ต้องถามชื่อผู้เล่นซ้ำอีก ถ้าไม่ใช่ให้จบโปรแกรมได้
- กรณีในศ.ภาควิชาวิศวกรรมคอมพิวเตอร์ ต้องมีการเขียนโปรแกรมป้องกันความผิดพลาดในการป้อนตัวเลข ด้วย

```
char name[30];  
printf("Enter you name ");  
scanf("%s",name);  
printf("Hello %s",name);
```

ผังงานเกมส่ายตัวเลข

จองตัวแปรเลขจำนวนเต็ม
int x, y, count ;

```
srand(time(NULL));  
x = rand() %100 + 1 ; // 0-99  
count = 0;
```

การวนกลับเพื่อทำซ้ำ

- ต้องมีจุดเริ่มต้น
- ต้องมีจุดตัดสินใจ

การรอรับคำตอบ
ต้องตั้งคำถามก่อน

การตัดสินใจ

- แยกเป็น 2 ทางเลือก
- ถ้าต้องการ 3 ทางเลือก
ให้ตัดสินใจซ้อน 2 ครั้ง

ทางที่แยกออกจากการตัดสินใจ 2
ทาง ต้องกลับมารวมกัน

