

## การทดลองที่ 3

# ตัวแปรพื้นฐาน คำสั่งรับค่าจากแป้นพิมพ์ และการดำเนินการทางคณิตศาสตร์

### วัตถุประสงค์

1. เข้าใจการประกาศตัวแปรพื้นฐานชนิดต่าง ๆ
2. เข้าใจการใช้คำสั่ง scanf ในการรับค่าจากแป้นพิมพ์
3. เข้าใจวิธีการ ขั้นตอน ในการคำนวณทางคณิตศาสตร์ของภาษาซี

### ทฤษฎีโดยย่อ

#### ก. ตัวแปรพื้นฐาน

ตัวแปรในภาษาซีโดยทั่วไปมีหน้าที่เก็บค่าต่าง ๆ เพื่อใช้ในการประมวลผล หรือเก็บค่าที่เป็นผลลัพธ์จากการประมวลผล ตัวแปรแต่ละตัวจะเก็บข้อมูลได้ตามชนิดข้อมูลที่ได้ประกาศไว้ตอนที่สร้างตัวแปรนั้น หากให้ตัวแปรเก็บค่าที่ไม่สอดคล้องกับชนิดข้อมูลที่ตัวแปรนั้นรองรับก็จะทำให้เกิดความผิดพลาดกับข้อมูลได้

การประกาศตัวแปรในภาษาซีมีรูปแบบดังนี้

```
type var_name
```

โดย type ได้แก่ชนิดข้อมูลที่ภาษาซีรองรับ เช่น int หรือ float หรือ char หรือ double หรือ long เป็นต้น ส่วน var\_name คือชื่อของตัวแปร โดยชื่อตัวแปรนั้นมีข้อกำหนดในการตั้งชื่อดังนี้

- เริ่มต้นชื่อด้วยตัวอักษร หรือ เครื่องหมาย underscore ( \_ )

- ตัวอักษรตัวที่สองเป็นต้นไปใช้ตัวเลข 0-9 หรือ ตัวอักษร หรือ เครื่องหมาย underscore ได้
- ห้ามใช้เครื่องหมายวรรคตอนอื่นใดเป็นส่วนหนึ่งของชื่อ
- ห้ามใช้ reserved word ของภาษาซี (เช่น for, static, while เป็นต้น) เป็นชื่อตัวแปร

## ข. คำสั่งรับค่าจากแป้นพิมพ์

คำสั่งรับค่าจากแป้นพิมพ์ในภาษาซีที่สามารถรับค่าได้ทุกชนิดข้อมูลที่เป็นที่นิยมใช้คือ คำสั่ง scanf โดยคำสั่ง scanf มีรูปแบบการใช้งานดังนี้

```
scanf("format string", address-list);
```

โดยส่วนของ format string จะเป็นส่วนที่ใช้กำหนดชนิดของข้อมูลที่ใช้จะป้อนเข้ามาทางแป้นพิมพ์ (ตัวอย่างเช่น %s สำหรับการรับข้อความ หรือ %d สำหรับการรับตัวเลขจำนวนเต็ม เป็นต้น) ส่วน address-list จะเป็นรายการตัวแปรที่จะใช้ในการเก็บข้อมูลที่รับจากแป้นพิมพ์ โดยแต่ละตัวแปรในรายการของ address-list จะต้องมีเครื่องหมาย & นำหน้า ยกเว้นตัวแปรชนิดข้อความ

## ค. การดำเนินการทางคณิตศาสตร์

การดำเนินการทางคณิตศาสตร์พื้นฐานในภาษาซีจะกระทำผ่านเครื่องหมาย +, -, \* (คูณ), / (หาร), % (โมดูลัส - หารเอาเศษ) นอกจากนั้นยังมีตัวดำเนินการพิเศษ ++ (เพิ่มค่าขึ้น 1), -- (ลดค่าลง 1), ตัวดำเนินการลรูป (เช่น +=, -=, \*= เป็นต้น)

นอกจากนั้นในการดำเนินการทางคณิตศาสตร์จะต้องคำนึงถึงชนิดของผลลัพธ์ที่จะเกิดขึ้นด้วย เช่น ตัวเลขจำนวนเต็ม กับ ตัวเลขจำนวนเต็ม จะได้ผลลัพธ์เป็นเลขจำนวนเต็มเสมอไม่ว่าจะดำเนินการด้วยตัวดำเนินการใดก็ตาม เป็นต้น

## ตอนที่ 1 ทำความเข้าใจกับการประกาศและใช้งานตัวแปร การใช้ scanf และการคำนวณ

### 1.1 การประกาศและใช้งานตัวแปร

- 1) ใช้โปรแกรมต่อไปนี้ตอบคำถามด้านล่าง

```
#include<stdio.h>

int main()
{
    int x;                //01
    printf("%d", x);

    return 0;
}
```

- ก) โปรแกรมนี้ Compile ผ่านหรือไม่ \_\_\_\_\_
- ข) เมื่อสั่ง Run จะได้ผลลัพธ์คือ \_\_\_\_\_
- ค) หากแก้ไขบรรทัด //01 เป็น float x; เมื่อ Compile & Run แล้วผลลัพธ์ที่ได้เหมือนเดิมหรือไม่ \_\_\_\_\_
- ง) หากแก้ไขบรรทัด //01 เป็น char x; เมื่อ Compile & Run แล้วผลลัพธ์ที่ได้เหมือนเดิมหรือไม่ \_\_\_\_\_
- จ) แก้ไขบรรทัด //01 เป็น int x=10; แล้ว Compile & Run ผลลัพธ์ที่ได้เหมือนเดิมหรือไม่ \_\_\_\_\_ ถ้าคำตอบคือไม่ ผลลัพธ์ที่ได้บนหน้าจอคือ \_\_\_\_\_
- ฉ) เหตุใดก่อนการแก้ไขจึงให้ผลลัพธ์เช่นนั้น \_\_\_\_\_
- ช) เหตุใดหลังการแก้ไขจึงให้ผลลัพธ์เช่นนั้น \_\_\_\_\_

2) ใช้โปรแกรมต่อไปนี้ตอบคำถามด้านล่าง

```
#include<stdio.h>

int main()
{
    int x;
    10 = x;                                //01
    printf("%d", x);

    return 0;
}
```

ก) เมื่อ Compile & Run แล้วได้ผลลัพธ์คือ \_\_\_\_\_

ข) แก้ไขบรรทัด //01 เป็น x=10; เมื่อ Compile & Run แล้วได้ผลลัพธ์คือ \_\_\_\_\_

ค) สรุปการกำหนดค่าให้กับตัวแปร \_\_\_\_\_

3) ใช้โปรแกรมต่อไปนี้ตอบคำถามด้านล่าง

```
#include<stdio.h>

int main()
{
    int x=10, y=20, z=0;
    x + y = z;                            //01
    printf("%d\n", z);

    return 0;
}
```

ก) เมื่อ Compile & Run แล้วได้ผลลัพธ์คือ \_\_\_\_\_

ข) แก้ไขบรรทัด //01 เป็น z=x+y; เมื่อ Compile & Run แล้วผลลัพธ์ที่ได้คือ \_\_\_\_\_

4) จากข้อ 2) และ 3) สรุปการใช้เครื่องหมายเท่ากับ (=) \_\_\_\_\_

5) ใช้โปรแกรมต่อไปนี้ตอบคำถามด้านล่าง

```
#include<stdio.h>

int main()
{
    float x=3.14159;
    printf("%d\n", x);

    return ;
}
```

ก) เมื่อ Compile & Run แล้วผลลัพธ์ที่ได้คือ \_\_\_\_\_

ข) เหตุใดจึงเป็นเช่นนั้น \_\_\_\_\_

ค) การใช้ printf เพื่อแสดงผลค่าตัวแปรนั้นมีหลักการเหมือนกันกับการแสดงผลข้อมูลชนิดต่าง ๆ (ตามที่ได้เรียนมาในบทที่ 2) หรือไม่ อย่างไร \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

## 1.2 การใช้งานคำสั่ง scanf

1) ใช้โปรแกรมต่อไปนี้ตอบคำถามด้านล่าง

```
#include<stdio.h>

int main()
{
    int x;                                     //01
    scanf("%d", x);                           //02
    printf("x = %d\n", x);

    return 0;
}
```

ก) เมื่อ Compile & Run ได้ผลลัพธ์คือ \_\_\_\_\_

ข) หากแก้ไขบรรทัด //01 เป็น int x=0; เมื่อ Compile & Run แล้วป้อน 10 ทางแป้นพิมพ์ ได้ผลลัพธ์คือ \_\_\_\_\_

ค) ข้อความแสดงความผิดพลาดจากข้อ ก) และ ข้อ ข) เหมือนกันหรือไม่ \_\_\_\_\_  
อย่างไร \_\_\_\_\_

ง) หากแก้ไขบรรทัด //01 กลับไปเป็น `int x;` และแก้ไขบรรทัด //02 เป็น `scanf("%d", &x);` เมื่อ Compile & Run แล้วป้อน 10 ทางแป้นพิมพ์ ได้ผลลัพธ์บนหน้าจอคือ \_\_\_\_\_

จ) หากแก้ไขบรรทัด //01 เป็น `int x=0;` เมื่อ Compile & Run แล้วป้อน 10 ทางแป้นพิมพ์ ได้ผลลัพธ์คือ \_\_\_\_\_

ฉ) การกำหนดค่าเริ่มต้นให้กับตัวแปร x มีผลต่อการทำงานของ scanf หรือไม่ \_\_\_\_\_

ช) จาก ก) – จ) สรุปได้ว่า ปัจจัยที่มีผลต่อการทำให้ scanf เกิดข้อผิดพลาดคือ \_\_\_\_\_

2) ใช้โปรแกรมต่อไปนี้ตอบคำถามด้านล่าง

```
#include<stdio.h>

int main()
{
    int x;
    scanf("x: %d", &x);                //01
    printf("x = %d\n\n", x);

    return 0;
}
```

ก) เมื่อ Compile & Run แล้วป้อน 10 ได้ผลลัพธ์บนหน้าจอคือ \_\_\_\_\_

ข) Run อีกครั้ง แล้วป้อน x: 10 ได้ผลลัพธ์บนหน้าจอคือ \_\_\_\_\_

ค) แก้ไขบรรทัด //01 เป็น `scanf("input number %d", &x);` เมื่อ Compile & Run ป้อน 10 ได้ผลลัพธ์บนหน้าจอคือ \_\_\_\_\_

ง) Run อีกครั้ง แล้วป้อน input number 10 ได้ผลลัพธ์บนหน้าจอคือ \_\_\_\_\_

จ) แก้ไขบรรทัด //01 เป็น scanf ("%d\n", &x) ; เมื่อ Compile & Run ป้อน 10  
ได้ผลลัพธ์บนหน้าจอคือ \_\_\_\_\_

ฉ) จาก ก) – ข) สรุปได้ว่าการใส่ตัวอักษรใด ๆ นอกเหนือไปจาก %d ในพารามิเตอร์  
แรกของ scanf ส่งผลให้ \_\_\_\_\_  
\_\_\_\_\_

3) ใช้โปรแกรมต่อไปนี้ตอบคำถามด้านล่าง

```
#include<stdio.h>

int main()
{
    int x;                                //01
    scanf("%d", &x);                      //02
    printf("x = %d\n", x);                //03

    return 0;
}
```

ก) Compile & Run แล้วป้อน 3.14159 ได้ผลลัพธ์บนหน้าจอคือ \_\_\_\_\_

ข) Run อีกครั้งแล้วป้อน 3.94159 ได้ผลลัพธ์บนหน้าจอคือ \_\_\_\_\_

ค) จากข้อ ก) และ ข) สรุปได้ว่า หากสั่งให้ scanf รับข้อมูลเป็นชนิดเลขจำนวนเต็ม (%d)  
เมื่อข้อมูลที่ป้อนเข้ามาเป็นเลขทศนิยม scanf สามารถรับค่าที่ป้อนเข้ามาได้หรือไม่  
และค่าที่รับเข้ามจะปัดทศนิยมให้หรือไม่ \_\_\_\_\_  
\_\_\_\_\_

ง) Run อีกครั้งแล้วป้อน a ได้ผลลัพธ์บนหน้าจอคือ \_\_\_\_\_

จ) Run อีกครั้งแล้วป้อน Test ได้ผลลัพธ์บนหน้าจอคือ \_\_\_\_\_

ฉ) แก้ไขบรรทัด //01 เป็น float x; แก้ไขบรรทัด //02 เป็น scanf ("%f",  
&x) ; เมื่อ Compile & Run แล้วป้อน 15 ได้ผลลัพธ์บนหน้าจอคือ \_\_\_\_\_

ข) แก้ไขบรรทัด //02 เป็น `printf("x = %f\n", x);` เมื่อ Compile & Run

แล้วป้อน 15 ได้ผลลัพธ์บนหน้าจอคือ \_\_\_\_\_

4) ใช้โปรแกรมต่อไปนี้ตอบคำถามด้านล่าง

```
#include<stdio.h>

int main()
{
    float x;
    scanf("%f", &x);                //01
    printf("x = %f\n", x);          //02

    return 0; }
```

ก) Compile & Run ป้อน 3.14159 ได้ผลลัพธ์บนหน้าจอคือ \_\_\_\_\_

ข) Run อีกครั้งป้อน 19 ได้ผลลัพธ์บนหน้าจอคือ \_\_\_\_\_

ค) Run อีกครั้งป้อน a ได้ผลลัพธ์บนหน้าจอคือ \_\_\_\_\_

ง) Run อีกครั้ง ป้อน Test ได้ผลลัพธ์บนหน้าจอคือ \_\_\_\_\_

จ) แก้ไขบรรทัด //01 เป็น `scanf ("%d", &x);` เมื่อ Compile & Run ป้อน

3.14159 ได้ผลลัพธ์บนหน้าจอคือ \_\_\_\_\_

ฉ) แก้ไขบรรทัด //02 เป็น `printf("x = %d\n", x);` เมื่อ Compile & Run

ป้อน 3.14159 ได้ผลลัพธ์บนหน้าจอคือ \_\_\_\_\_

ช) สามารถใช้รูปแบบ `%m.nf` (เช่น `%.2f`) กับคำสั่ง `scanf` เหมือนเช่นที่ใช้กับ `printf` ได้

หรือไม่ \_\_\_\_\_

5) จากข้อ 3) – 4) สามารถสรุปได้ว่า \_\_\_\_\_

---

---

---



6) ใช้โปรแกรมต่อไปนี้ตอบคำถามด้านล่าง

```
#include<stdio.h>

int main()
{
    int x, y, z;
    scanf("%d%d%d", &x, &y, &z);          //01
    printf("x = %d, y = %d, z = %d\n", x, y, z);

    return 0;
}
```

ก) Compile & Run แล้วป้อน 10 20 30 (10 เว้นวรรค 20 เว้นวรรค 30 แล้วกดปุ่ม enter)

ได้ผลลัพธ์บนหน้าจอคือ \_\_\_\_\_

ข) Run อีกครั้งแล้วป้อน 10 <enter> 20 <enter> 30 (<enter> หมายถึงให้กดปุ่ม enter

บนแป้นพิมพ์) ได้ผลลัพธ์บนหน้าจอคือ \_\_\_\_\_

ค) แก้ไขบรรทัด //01 เป็น scanf ("%d %d %d", &x, &y, &z); เมื่อ

Compile & Run แล้วป้อน 10 20 30 (10 เว้นวรรค 20 เว้นวรรค 30 แล้วกดปุ่ม enter)

ได้ผลลัพธ์บนหน้าจอคือ \_\_\_\_\_

ง) Run อีกครั้งแล้วป้อน 10 <enter> 20 <enter> 30 (<enter> หมายถึงให้กดปุ่ม enter

บนแป้นพิมพ์) ได้ผลลัพธ์บนหน้าจอคือ \_\_\_\_\_

จ) แก้ไขบรรทัด //01 เป็น scanf ("%d-%d-%d", &x, &y, &z); เมื่อ

Compile & Run แล้วป้อน 10 20 30 (10 เว้นวรรค 20 เว้นวรรค 30 แล้วกดปุ่ม enter)

ได้ผลลัพธ์บนหน้าจอคือ \_\_\_\_\_

ฉ) Run อีกครั้งแล้วป้อน 10 <enter> 20 <enter> 30 (<enter> หมายถึงให้กดปุ่ม enter

บนแป้นพิมพ์) ได้ผลลัพธ์บนหน้าจอคือ \_\_\_\_\_

ช) Run อีกครั้งแล้วป้อน 10-20-30 ได้ผลลัพธ์บนหน้าจอคือ \_\_\_\_\_

ซ) Run อีกครั้งแล้วป้อน 10,20,30 ได้ผลลัพธ์บนหน้าจอคือ \_\_\_\_\_

7) ใช้โปรแกรมต่อไปนี้ตอบคำถามด้านล่าง

```
#include<stdio.h>

int main()
{
    char str[20];
    scanf("%s", str);          //01
    printf("str: %s\n", str);

    return 0;
}
```

ก) Compile & Run แล้วป้อน Hello ผลลัพธ์ที่ได้บนหน้าจอคือ \_\_\_\_\_

ข) Run อีกครั้ง ป้อน Hello there ผลลัพธ์ที่ได้บนหน้าจอคือ \_\_\_\_\_

ค) แก้ไขบรรทัด //01 เป็น scanf ("%[^\\n]", str); เมื่อ Compile & Run

แล้วป้อน Hello ผลลัพธ์ที่ได้บนหน้าจอคือ \_\_\_\_\_

ง) Run อีกครั้ง ป้อน Hello there ผลลัพธ์ที่ได้บนหน้าจอคือ \_\_\_\_\_

จ) ความแตกต่างระหว่างการใช้ %s กับ %[\\n] คือ \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

### 1.3 การดำเนินการทางคณิตศาสตร์

1) หาผลลัพธ์ของการดำเนินการทางคณิตศาสตร์ต่อไปนี้ด้วยลำดับการดำเนินการแบบภาษาซี

(ให้คิดเอง, สามารถใช้เครื่องคิดเลขช่วยคำนวณได้) พร้อมทั้งให้เขียนใหม่จัดรูปให้ง่าย

โดยที่ผลลัพธ์ไม่เปลี่ยน ดังตัวอย่างในข้อ ก)

ก)  $3 + 5 * 2 * 9 / 3 - 8$

ผลลัพธ์ 25      จัดรูปใหม่       $(3 + (((5 * 2) * 9) / 3)) - 8$

ข)  $3.5 + 9 - 6 * 3 * 7 / 2$

---

---

---

ค)  $8.3 * 2 + 5 * 9 / 2 + 10 * 4$

---

---

---

ง)  $7 / 6 + 2 - 3 * 9 * 5 * 16$

---

---

---

จ)  $7.8 + 9 - 3.6 * 2 / 8 + 5 * 9 - 3.9$

---

---

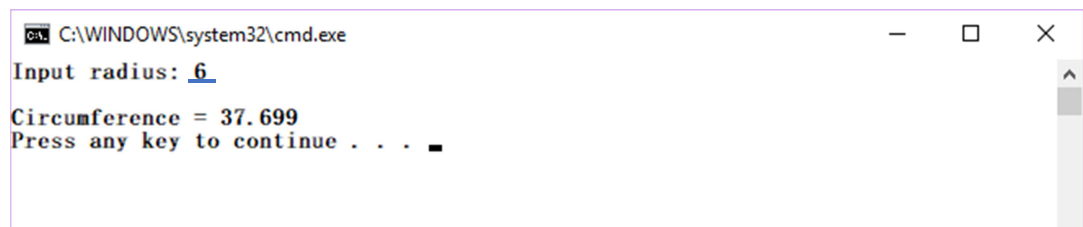
---

2) เขียนโปรแกรมเพื่อตรวจสอบผลลัพธ์ในข้อ 1) ถ้าผิดให้เขียนคำตอบที่ถูกต้องด้วยปากกาแดง

## ตอนที่ 2 การประยุกต์

### 2.1 สามารถเขียนโปรแกรมที่ให้ผลลัพธ์ตามต้องการได้

- 1) โปรแกรมหาความยาวเส้นรอบวงกลม (กำหนดให้ค่า  $\pi = 3.14159$ ) โดยเมื่อโปรแกรมทำงานจะให้ผลลัพธ์ตามรูปต่อไปนี้ (ตัวเลขที่ขีดเส้นใต้ในรูปคือตัวเลขที่ป้อนให้กับโปรแกรมทางแป้นพิมพ์)

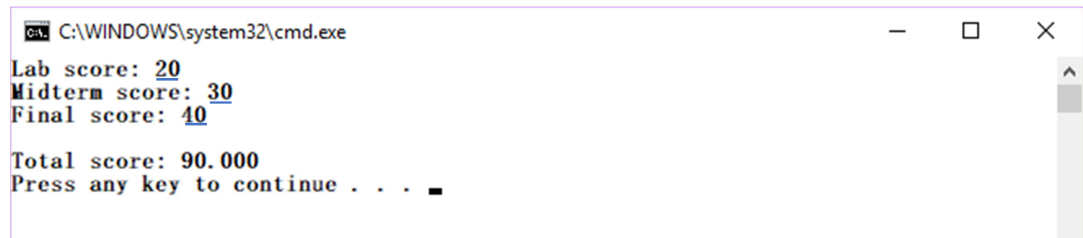


```
cmd: C:\WINDOWS\system32\cmd.exe
Input radius: 6
Circumference = 37.699
Press any key to continue . . .
```

- 2) โปรแกรมหาค่าพลังงานศักย์จากสูตร  $E_p = mgh$  (m มวลหน่วยเป็น kg., g ค่าคงที่ กำหนดให้ มีค่า 9.8, h คือความสูงหน่วยเป็นเมตร) โดยเมื่อโปรแกรมทำงานจะให้ผลลัพธ์ตามรูปต่อไปนี (ตัวเลขที่ขีดเส้นใต้ในรูปคือตัวเลขที่ป้อนให้กับโปรแกรมทางแป้นพิมพ์)

```
C:\WINDOWS\system32\cmd.exe
Input mass (g): 1500
Input height (cm): 50
Potential Energy = 7.35
Press any key to continue . . .
```

- 3) โปรแกรมรวมคะแนนวิชา Programming โดยโปรแกรมจะรับข้อมูล คะแนน Lab, คะแนนสอบกลางภาค และคะแนนสอบปลายภาค โดยเมื่อโปรแกรมทำงานจะให้ผลลัพธ์ตามรูปต่อไปนี (ตัวเลขที่ขีดเส้นใต้ในรูปคือตัวเลขที่ป้อนให้กับโปรแกรมทางแป้นพิมพ์)



```
ca. C:\WINDOWS\system32\cmd.exe
Lab score: 20
Midterm score: 30
Final score: 40

Total score: 90.000
Press any key to continue . . . █
```



```
ca. C:\WINDOWS\system32\cmd.exe
Lab score: 15.5
Midterm score: 24.45
Final score: 19.87

Total score: 59.820
Press any key to continue . . . █
```