

เล่มที่ 3 ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูลและฟังก์ชันทางคณิตศาสตร์

คำชี้แจงสำหรับนักเรียน

เอกสารประกอบการเรียน การเขียนโปรแกรมคอมพิวเตอร์ด้วยภาษาซี กลุ่มสาระการเรียนรู้วิทยาศาสตร์และเทคโนโลยี สำหรับนักเรียนชั้นมัธยมศึกษาปีที่ 6 เล่มที่ 3 เรื่อง ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูลและฟังก์ชันทางคณิตศาสตร์ เป็นสื่อใช้ประกอบการจัดกิจกรรมการเรียนรู้ นักเรียนควรปฏิบัติ ดังนี้

บทบาทของนักเรียน

1. เอกสารประกอบการเรียนเล่มนี้ สร้างขึ้นเพื่อให้ นักเรียนได้ศึกษาด้วยตนเองและใช้ฝึกทักษะการเขียนโปรแกรมคอมพิวเตอร์ด้วยภาษาซี นักเรียนต้องมีวินัย มีความรับผิดชอบและมีความซื่อสัตย์ต่อตนเอง
2. นักเรียนต้องศึกษาจุดประสงค์การเรียนรู้ก่อน เพื่อให้ทราบว่าหลังจากศึกษาเนื้อหาในหน่วยการเรียนรู้แล้ว นักเรียนจะต้องมีความสามารถทำอะไรได้บ้าง
3. นักเรียนทำแบบทดสอบก่อนเรียน เพื่อทดสอบความรู้พื้นฐานของตนเอง
4. นักเรียนศึกษาเนื้อหาไปตามลำดับและปฏิบัติตามกิจกรรมตามคำแนะนำที่ละขั้นตอน พร้อมฝึกเขียนโปรแกรมตามตัวอย่าง
5. เมื่อศึกษาเนื้อหาเสร็จสิ้นแล้ว ให้นักเรียนทำแบบทดสอบหลังเรียน
6. หากมีข้อสงสัยใดๆ ให้ปรึกษาครูผู้สอน

เล่มที่ 3 ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูลและฟังก์ชันทางคณิตศาสตร์

หน่วยการเรียนรู้ที่ 3

เรื่อง ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูลและฟังก์ชันทางคณิตศาสตร์ จำนวน 3 ชั่วโมง

หัวข้อเรื่อง

1. ฟังก์ชันการแสดงผลข้อมูล
 - ฟังก์ชัน `printf()`
 - ฟังก์ชัน `putchar()`
 - ฟังก์ชัน `puts()`
2. ฟังก์ชันการรับข้อมูล
 - ฟังก์ชัน `scanf()`
 - ฟังก์ชัน `getchar()` และ `getch()`
 - ฟังก์ชัน `gets()`
3. ฟังก์ชันจัดการจอภาพ
4. ฟังก์ชันทางคณิตศาสตร์

สาระสำคัญ

1. ในภาษาซี ได้เตรียมฟังก์ชันเพื่อการแสดงผลข้อมูลอยู่หลายฟังก์ชันด้วยกัน ซึ่งสามารถนำมาเลือกใช้งานได้ตามความเหมาะสม เช่น `printf()` เป็นฟังก์ชันที่ใช้สำหรับแสดงผลข้อมูล ไม่ว่าจะเป็นข้อมูลที่เป็นตัวอักษร ข้อความ หรือตัวแปร `putchar()` เป็นฟังก์ชันที่แสดงผลตัวอักษร หรืออักขระ และฟังก์ชัน `puts()` ใช้สำหรับแสดงผลข้อความ

2. ฟังก์ชันการรับข้อมูล ในภาษาซีมีฟังก์ชัน `scanf()` เป็นฟังก์ชันที่ใช้สำหรับรับค่าข้อมูลจากทางแป้นพิมพ์ เพื่อนำค่ามาจัดเก็บไว้ในตัวแปร โดยจะต้องกำหนดรหัสรูปแบบข้อมูลให้สัมพันธ์กับชนิดข้อมูลของตัวแปรที่ต้องการจัดเก็บ มีฟังก์ชัน `getchar()` และ `getch()` สำหรับรับค่าตัวอักษรหรืออักขระ 1 ตัว และมีฟังก์ชัน `gets()` สำหรับรับค่าข้อความจากทางแป้นพิมพ์

3. ฟังก์ชันจัดการจอภาพ ในภาษาซีที่นิยมนำมาใช้งาน ได้แก่ ฟังก์ชัน `system("cls");` เป็นฟังก์ชันที่ใช้สำหรับการล้างหน้าจอเพื่อให้หน้าจอกลับมาเริ่มต้นแสดงผลใหม่โดยที่ไม่ต้องสั่งรันโปรแกรมซ้ำและฟังก์ชันกำหนดตำแหน่งเคอร์เซอร์บนหน้าจอเพื่อทำให้ผู้พัฒนาโปรแกรมสามารถกำหนดการแสดงผลตามที่ต้องการได้สะดวกขึ้น โดยการเขียนฟังก์ชันใช้งาน `gotoxy()` ;

4. ฟังก์ชันทางคณิตศาสตร์ ในภาษาซี เป็นฟังก์ชันที่ถูกประกาศใช้งานอยู่ในเฮดเดอร์ไฟล์ `<math.h>` ได้แก่ `pow()`, `sqrt()`, `sine()`, `cos()`, `tan()`, `ceil()` และ `floor()` อำนวยความสะดวกในการใช้งาน หากต้องการคำนวณหาเลขยกกำลัง การหาค่ารากที่สอง หรือค่าตรีโกณมิติ เป็นต้น

เล่มที่ 3 ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูลและฟังก์ชันทางคณิตศาสตร์

จุดประสงค์การเรียนรู้

หลังจากศึกษาเนื้อหาในหน่วยการเรียนรู้นี้แล้วนักเรียนสามารถ

1. อธิบายลักษณะการทำงานของฟังก์ชันแสดงผลได้
2. อธิบายลักษณะการทำงานของฟังก์ชันรับข้อมูลได้
3. เลือกใช้งานฟังก์ชันแสดงผลและรับข้อมูลได้เหมาะสม
4. บอกคุณสมบัติและการใช้งานฟังก์ชันเพื่อการจัดการกับจอภาพได้
5. บอกคุณสมบัติและการใช้งานฟังก์ชันทางคณิตศาสตร์ได้
6. เขียนโปรแกรมการรับข้อมูลและแสดงผลข้อมูลด้วยภาษาซีได้

เล่มที่ 3 ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูลและฟังก์ชันทางคณิตศาสตร์

หัวเรื่องที่ 3.1 ฟังก์ชันการแสดงผลข้อมูล

- เวลาเรียน 30 นาที -

คำสั่งในภาษาซี ล้วนอยู่ในรูปแบบของฟังก์ชันทั้งสิ้น ซึ่งอาจเป็นฟังก์ชันมาตรฐานที่ภาษาซีได้จัดเตรียมมาให้แล้ว (ประกาศใช้งานอยู่ในเฮดเดอร์ไฟล์ต่าง ๆ) หรือฟังก์ชันที่เราสามารถเขียนขึ้นเพื่อใช้งานเอง

ฟังก์ชันการแสดงผล (Output) ปกติก็คือ การแสดงออกทางจอภาพ ในภาษาซีใช้ในรูปแบบของฟังก์ชันเช่นเดียวกับงานอื่น ๆ โดยฟังก์ชันมาตรฐานที่มีใน library ฟังก์ชันที่ใช้แสดงผลทางจอภาพ เช่น printf(), putchar(), puts() ฟังก์ชันเหล่านี้ประกาศใช้งานอยู่ในเฮดเดอร์ไฟล์ ชื่อ stdio.h

1. ฟังก์ชัน printf()

การแสดงผลออกทางหน้าจอ สามารถทำได้หลายวิธีด้วยกัน แต่วิธีที่นิยมใช้กันมากที่สุดก็คือ การเรียกใช้ฟังก์ชัน printf() ซึ่งเป็นฟังก์ชันมาตรฐานที่ใช้ในการแสดงผลข้อมูลทุกชนิดออกทางจอภาพ ไม่ว่าจะเป็นจำนวนเต็ม ทศนิยม อักขระ ข้อความ หรือค่าตัวแปร ซึ่งมีรูปแบบการเขียนดังนี้

รูปแบบการเรียกใช้ฟังก์ชัน printf() แสดงดังต่อไปนี้

```
printf(" format ",variable);
```

☞ format : ข้อมูลที่ต้องการแสดงผลออกทางหน้าจอ โดยข้อมูลนี้ต้องเขียนไว้ในเครื่องหมาย " " ข้อมูลที่สามารถแสดงผลได้มีอยู่ 2 ประเภท คือ ข้อความธรรมดาและค่าที่เก็บไว้ในตัวแปร ซึ่งถ้าเป็นค่าที่เก็บไว้ในตัวแปรต้องใส่รหัสควบคุมรูปแบบให้ตรงกับชนิดของข้อมูลที่เก็บไว้ในตัวแปรนั้นด้วย

☞ variable คือ ตัวแปร หรือนิพจน์ที่ต้องการนำค่าไปแสดงผล

การเรียกใช้งานฟังก์ชัน printf() เพื่อแสดงข้อความออกทางหน้าจอ แสดงดังต่อไปนี้

เล่มที่ 3 ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูลและฟังก์ชันทางคณิตศาสตร์

ตัวอย่าง

```
#include<stdio.h>
main()
{
    printf("Hello, nice to meet you.");
    printf("C Programming is easy.");
    printf(" Nakhonsawan Thailand.");
}
```

เมื่อสั่งรันโปรแกรมข้อความจะแสดงออกทางหน้าจอ ดังต่อไปนี้

Hello, nice to meet you. C Programming is easy. Nakhonsawan Thailand.

รหัสรูปแบบที่ใช้กับฟังก์ชัน printf() จะเริ่มด้วยเครื่องหมาย % ดังตารางต่อไปนี้

รหัสรูปแบบการแสดงผล	ความหมาย / การนำไปใช้
%c	แสดงผลอักขระ 1 ตัว (char)
%d	แสดงผลเลขจำนวนเต็ม (int)
%ld	แสดงผลเลขจำนวนเต็มแบบยาว (long int)
%e	แสดงผลเลขจำนวนจริง แบบเอ็กซ์โปเนนต์
%f	แสดงผลเลขจำนวนจริง หรือทศนิยม (float)
%o	แสดงผลออกมาในรูปแบบของเลขฐานแปด
%s	แสดงผลข้อความ (string)
%u	แสดงผลตัวเลขจำนวนเต็มบวก
%x	แสดงผลออกมาในรูปแบบของเลขฐานสิบหก

เล่มที่ 3 ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูลและฟังก์ชันทางคณิตศาสตร์

รูปแบบการเขียนฟังก์ชัน printf () พร้อมกำหนดรหัสรูปแบบ แสดงดังต่อไปนี้

```
printf(“%d %f”,variable1, variable2);
```

โดย %d %f : รหัสรูปแบบการแสดงผลของตัวแปร

variable1, variable2 : ตัวแปรที่ต้องการให้แสดงค่าออกทางจอภาพ

นอกจากนี้เรายังสามารถผนวกรหัสควบคุม (Escape Sequence) เข้าไปในการจัดรูปแบบการแสดงผลให้ดูเป็นระเบียบมากขึ้นได้ อย่างเช่น ขึ้นบรรทัดใหม่หลังแสดงข้อความ หรือเว้นระยะแท็บระหว่างข้อความ โดยใช้อักขระควบคุมการแสดงผลร่วมกับคำสั่ง printf () ซึ่งรหัสควบคุมเหล่านี้จะเขียนด้วยเครื่องหมาย \ (Back Slash) และตามด้วยรหัสควบคุม ซึ่งประกอบด้วย

รหัสควบคุม	ความหมาย
\a	ส่งเสียงบี๊ป 1 ครั้ง
\b	ถอยหลังหนึ่งตัวอักษร (backspace)
\f	ขึ้นหน้าใหม่ (form feed)
\n	ขึ้นบรรทัดใหม่ (new line)
\r	ย้ายเคอร์เซอร์กลับไปต้นบรรทัด
\t	แท็บแนวนอน (horizontal tab) (เท่ากับ 6 ตัวอักษร)
\v	แท็บแนวตั้ง (vertical tab)
\'	แสดงเครื่องหมาย '
\"	แสดงเครื่องหมาย "
\\	แสดงเครื่องหมาย \
\0 (ศูนย์)	ค่าว่าง (null)

รูปแบบการควบคุมการแสดงผล แสดงดังต่อไปนี้

```
printf(“number = %d \n”,variable);
```

โดย \n : รหัสควบคุมการแสดงผล ซึ่งจะต้องเขียนไว้ภายในเครื่องหมาย “ ”

ร่วมกับข้อความและรหัสรูปแบบการแสดงผลของตัวแปร

นอกจากนี้การแสดงผลตัวเลขทศนิยม ตามปกติถ้าไม่ได้กำหนดค่าใด ๆ เพิ่มเติม เมื่อใช้รหัสควบคุมรูปแบบ %f โปรแกรมจะแสดงตัวเลขหลังจุดทศนิยมทั้งหมด 6 ตำแหน่ง ดังตัวอย่างโปรแกรม ต่อไปนี้

เล่มที่ 3 ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูลและฟังก์ชันทางคณิตศาสตร์

```
#include<stdio.h>
main( )
{
    printf(“%f\n”,45.90);
    printf(“%f\n”,-12.556);
    printf(“%f\n”,1.67);
    printf(“%f\n”,-320.5);
}
```

เมื่อสั่งรันโปรแกรมจะปรากฏผลการทำงาน ดังต่อไปนี้

```
45.900000
-12.556000
1.670000
-320.500000
```

ในกรณีที่เราต้องการให้แสดงตัวเลขหลังจุดทศนิยม 2 หรือ 3 ตำแหน่ง ก็สามารถกำหนดค่าเพิ่มไปกับรหัสรูปแบบได้ ตามรูปแบบต่อไปนี้

```
%nf
```

โดย n : ใส่หลังเครื่องหมายจุด เป็นตัวเลขของจำนวนอักขระ
ในข้อความที่จะให้แสดงผล

เล่มที่ 3 ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูลและฟังก์ชันทางคณิตศาสตร์

```
#include<stdio.h>
main()
{
    printf("%.2f\n",45.90);
    printf("%.3f\n",-12.556);
    printf("%.1f\n",1.67);
    printf("%.3f\n",-320.5);
}
```

เมื่อสั่งรันโปรแกรมจะปรากฏผลการทำงาน ดังต่อไปนี้

```
45.90
-12.556
1.7
-320.500
```

2. ฟังก์ชัน putchar ()

ในการแสดงผลตัวอักษร หรืออักขระ (char) ออกทางหน้าจอ นอกจากใช้ฟังก์ชัน printf () พร้อมกับกำหนดรหัสรูปแบบ %c แล้ว เราสามารถเรียกใช้คำสั่ง สำหรับแสดงตัวอักษร หรืออักขระ โดยเฉพาะได้อีกด้วย โดยคำสั่งนั้นคือ คำสั่ง putchar () ซึ่งมีรูปแบบการเรียกใช้คำสั่ง ดังต่อไปนี้

```
putchar(ch);
```

โดย ch : ตัวอักษร หรืออักขระเขียนอยู่ภายในเครื่องหมาย ‘ ’
หรือตัวแปรชนิด char

เล่มที่ 3 ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูลและฟังก์ชันทางคณิตศาสตร์

ตัวอย่าง การใช้คำสั่ง putchar() ในการแสดงผลอักขระออกทางหน้าจอ แสดงได้ดังนี้

```
#include<stdio.h>
char letter = 'T'
char vol = 'y'
main()
{
    putchar(letter);    // ใช้คำสั่ง putchar() แสดงอักขระในตัวแปร letter
    putchar('o');      // ใช้คำสั่ง putchar() แสดงอักขระ o ออกทางหน้าจอ
    putchar(vol);      // ใช้คำสั่ง putchar() แสดงอักขระในตัวแปร vol
}
```

เมื่อสั่งรันโปรแกรมจะปรากฏผลการทำงาน ดังต่อไปนี้

Toy

3. ฟังก์ชัน puts()

นอกจากการแสดงผลอักขระด้วยฟังก์ชัน putchar() แล้ว ในภาษาซียังมีคำสั่งสำหรับแสดงผลข้อความออกทางหน้าจอโดยเฉพาะอีกด้วย นั่นก็คือ คำสั่ง puts() ซึ่งมีรูปแบบการเรียกใช้คำสั่ง ดังนี้

```
puts(str);
```

โดย str : ข้อความซึ่งเขียนอยู่ในเครื่องหมาย “ ” หรือตัวแปร
ที่เก็บข้อมูลชนิดข้อความ

เล่มที่ 3 ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูลและฟังก์ชันทางคณิตศาสตร์

ตัวอย่าง การใช้งานคำสั่ง puts() ในการแสดงข้อความออกทางหน้าจอ แสดงดังต่อไปนี้

```
#include<stdio.h>
char name[10] = "Tawee";
main()
{
puts(name);           // ใช้คำสั่ง puts() แสดงข้อความในตัวแปร name
puts("Navamin");     // ใช้คำสั่ง puts() แสดงข้อความ Navamin
puts("Nakhonsawan"); // ใช้คำสั่ง puts() แสดงข้อความ Nakhonsawan
}
```

คำสั่ง puts() เมื่อแสดงข้อความแล้ว จะขึ้นบรรทัดใหม่ให้โดยอัตโนมัติ ซึ่งผลการรันโปรแกรม แสดงดังต่อไปนี้

```
Tawee
Navamin
Nakhonsawan
```

สรุป ในภาษาซี ได้เตรียมฟังก์ชันเพื่อการแสดงผลข้อมูลอยู่หลายฟังก์ชันด้วยกัน ซึ่งสามารถนำมาเลือกใช้งานได้ตามความเหมาะสม เช่น printf() เป็นฟังก์ชันที่ใช้สำหรับแสดงผลข้อมูล ไม่ว่าจะเป็นข้อมูลที่เป็นตัวอักขระ ข้อความ หรือตัวแปร putchar() เป็นฟังก์ชันที่แสดงผลตัวอักขระหรืออักขระและฟังก์ชัน puts() ใช้สำหรับแสดงผลข้อความ

เล่มที่ 3 ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูลและฟังก์ชันทางคณิตศาสตร์

กิจกรรมที่ 3.1 ฟังก์ชันการแสดงผลข้อมูล (คะแนนเต็ม 5 คะแนน)

<ใช้เวลาทำ 10 นาที>

คำชี้แจง

ให้นักเรียนตอบคำถามต่อไปนี้

1. จงเขียนคำสั่งเพื่อแสดงค่าของตัวแปรเหล่านี้ออกทางหน้าจอ (5 คะแนน)

โดยกำหนดตัวแปรให้ดังนี้

```
int Num1 = 2560;
float Num2 = 12.745;
char answer = 'Y';
char city[10] = "Thailand";
```

- 1.1 เขียนคำสั่งเพื่อแสดงค่าของตัวแปร Num1 ออกทางหน้าจอ

.....

- 1.2 เขียนคำสั่งเพื่อแสดงค่าของตัวแปร answer แล้วขึ้นบรรทัดใหม่ จากนั้นจึงแสดงค่าของตัวแปร city ออกทางหน้าจอ

.....

- 1.3 เขียนคำสั่งเพื่อแสดงค่าของตัวแปร Num2 ออกทางหน้าจอ โดยให้แสดงตัวเลขหลังจุดทศนิยมแค่ 2 ตำแหน่ง พร้อมทั้งขึ้นบรรทัดใหม่

- 1.4 เขียนคำสั่งเพื่อแสดงค่าผลบวกของตัวแปร Num1 และ Num2 ออกทางหน้าจอ

- 1.5 เขียนคำสั่งเพื่อแสดงข้อความในตัวแปร city ออกทางหน้าจอ

แนวตอบกิจกรรมที่ 3.1

printf () เป็นฟังก์ชันที่ใช้สำหรับแสดงผลข้อมูล ไม่ว่าจะเป็นข้อมูลที่เป็นตัวอักขระข้อความ หรือตัวแปร

putchar () เป็นฟังก์ชันที่แสดงผลตัวอักขระ หรืออักขระ

puts () เป็นฟังก์ชันใช้สำหรับแสดงผลข้อความ

เล่มที่ 3 ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูลและฟังก์ชันทางคณิตศาสตร์

หัวเรื่องที่ 3.2 ฟังก์ชันการรับข้อมูล

- เวลาเรียน 40 นาที -

การทำงานของโปรแกรมโดยส่วนใหญ่มักจะเป็นการเชื่อมโยงกับผู้ใช้แบบ 2 ทิศทาง นั่นก็คือ ในส่วนการแสดงผลการทำงานออกทางหน้าจอและมีการรับข้อมูลจากผู้ใช้เข้ามาทางคีย์บอร์ด เพื่อนำมารวมในการประมวลผลของโปรแกรม ซึ่งในส่วนการรับข้อมูลจากผู้ใช้ ภาษาซีได้กำหนดคำสั่งและฟังก์ชันมาตรฐานเอาไว้ให้เรียกใช้แล้ว เช่นเดียวกับส่วนของการแสดงผล รายละเอียดของคำสั่งและฟังก์ชันเหล่านี้ ได้แก่

1. ฟังก์ชัน scanf()

ในภาษาซี การรับข้อมูลจากคีย์บอร์ดสามารถทำได้โดยการเรียกใช้ฟังก์ชัน scanf() ซึ่งเป็นฟังก์ชันมาตรฐานสำหรับรับข้อมูลจากคีย์บอร์ด โดยสามารถรับข้อมูลได้ทุกประเภท ไม่ว่าจะเป็นจำนวนเต็ม ทศนิยม อักขระ หรือข้อความ

รูปแบบการเรียกใช้ฟังก์ชัน scanf() คล้ายกับการเรียกใช้ฟังก์ชัน printf() ดังแสดงต่อไปนี้

```
scanf("format", &variable);
```

☞ format : ข้อความและรหัสรูปแบบ เพื่อกำหนดรูปแบบและชนิดของข้อมูลที่จะรับเข้ามาจากคีย์บอร์ด โดยรหัสรูปแบบจะใช้ชุดเดียวกับรหัสรูปแบบของฟังก์ชัน printf()

☞ &variable : ตัวแปรที่จะใช้เก็บข้อมูลที่รับเข้ามาจากคีย์บอร์ด โดยชนิดของตัวแปรจะต้องตรงกับรหัสรูปแบบที่กำหนดไว้ นอกจากนี้หน้าชื่อของตัวแปรจะต้องนำหน้าด้วยเครื่องหมาย & ยกเว้นตัวแปรชนิด string เท่านั้นที่ไม่ต้องนำหน้าด้วยเครื่องหมาย &

การใช้ฟังก์ชัน scanf() เพื่อรับข้อมูลจากคีย์บอร์ดเข้ามาเก็บไว้ในตัวแปร แสดงดังต่อไปนี้

เล่มที่ 3 ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูลและฟังก์ชันทางคณิตศาสตร์

ตัวอย่าง

```
#include<stdio.h>
int high;
main()
{
    printf("Enter high of you (cm) :"); // ให้กรอกค่าความสูง
    scanf("%d",&high); // รับค่าความสูงเป็นจำนวนเต็ม (%d)
                               มาเก็บไว้ในตัวแปร high
    printf("You is high = %d cm.",speed); // แสดงค่าความสูงที่ได้รับเข้ามา
}
```

เมื่อสั่งรันโปรแกรมจะปรากฏข้อความ Enter high of you (cm) : พร้อมเคอร์เซอร์กะพริบรอให้กรอกข้อมูล ดังรูปต่อไปนี้

```
Enter high of you (cm) : _
```

ทดลองกรอกตัวเลขจำนวนเต็มแทนค่าความสูงลงไปแล้วกดปุ่ม <Enter> เพื่อให้โปรแกรมกลับไปทำงานต่อ จากนั้นโปรแกรมจะแสดงข้อความพร้อมทั้งค่าความสูงที่ป้อนเข้าไปขึ้นมาให้บนหน้าจอ

```
Enter high of you (cm) : 170
You is high = 170 cm.
```


เล่มที่ 3 ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูลและฟังก์ชันทางคณิตศาสตร์

ตัวอย่าง การใช้คำสั่ง scanf ()

```
#include<stdio.h>
main()
{
int age;
printf("How old are you ? ");      // แสดงข้อความถามว่าคุณอายุเท่าไร
scanf("%d", &age);                // รับเลขจำนวนเต็มของอายุมาเก็บไว้ใน age
printf("You are %d years old" ,age // แสดงข้อความพร้อมทั้งอายุออกทางหน้าจอ
}
```

เมื่อสั่งรันโปรแกรมจะปรากฏข้อความ How old are you ? ขึ้นมาให้เรากรอกอายุ

```
How old are you ? _
```

เมื่อกรอกอายุพร้อมทั้งกดปุ่ม <Enter> แล้ว โปรแกรมจะแสดงข้อความพร้อมทั้งอายุที่กรอกขึ้นมาให้บนหน้าจอ

```
How old are you ? 40
You are 40 years old
```

ในกรณีที่เราต้องการให้กรอกข้อมูลเข้ามามากกว่าหนึ่งค่าในครั้งเดียวกัน ก็สามารถทำได้ โดยกำหนดรูปแบบการป้อนข้อมูลเหล่านั้นในฟังก์ชัน scanf () ดังแสดงในตัวอย่างต่อไปนี้

เล่มที่ 3 ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูลและฟังก์ชันทางคณิตศาสตร์

```
scanf("%d %f %d", &variable1, &variable2, &variable3);
```

%d %f %d : ให้กรอกข้อมูลเข้ามา 3 ค่า ตัวแรกเป็นจำนวนเต็ม (%d) ตัวที่สองเป็นจำนวนทศนิยม (%f) ตัวสุดท้ายเป็นจำนวนเต็ม (%d) โดยเวลากรอกข้อมูลต้องเว้นช่องว่างระหว่างข้อมูลแต่ละตัวตามรูปแบบที่กำหนดไว้
%d %f %d

&variable1, &variable2, &variable3 :

ตัวแปรที่จะใช้เก็บข้อมูลที่รับเข้ามาจากคีย์บอร์ดทั้ง 3 ค่าตามลำดับ

```
scanf ("%d /%f/ %d", &variable1, &variable2, &variable3);
```

%d /%f/ %d : ให้กรอกข้อมูลเข้ามา 3 ค่า ตัวแรกเป็นจำนวนเต็ม (%d) ตัวที่สองเป็นจำนวนทศนิยม (%f) และตัวสุดท้ายเป็นจำนวนเต็ม (%d) โดยกรอกข้อมูลต้องใส่เครื่องหมาย / คั่นระหว่างข้อมูลแต่ละตัวตามรูปแบบที่กำหนดไว้ %d/ %f/ %d

&variable1, &variable2, &variable3 :

ตัวแปรที่จะใช้เก็บข้อมูลที่รับเข้ามาจากคีย์บอร์ดทั้ง 3 ค่าตามลำดับ

2. ฟังก์ชัน getchar() และ getch()

ภาษาซีมีฟังก์ชันพิเศษสำหรับรับข้อมูลประเภทอักขระ หรือตัวอักษรจากคีย์บอร์ด โดยเฉพาะนั่นคือ ฟังก์ชัน getchar() และ getch() ทั้ง 2 ฟังก์ชันนี้ใช้รับข้อมูลประเภทอักขระ (char) โดยมีรูปแบบการเรียกใช้คำสั่ง ดังแสดงต่อไปนี้

```
ch = getchar();
```

```
ch = getch();
```

ch : ตัวแปรชนิด char เพื่อใช้เก็บค่าของอักขระ หรือตัวอักษรที่รับเข้ามา

เล่มที่ 3 ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูลและฟังก์ชันทางคณิตศาสตร์

ถึงแม้ว่าทั้งฟังก์ชัน `getchar()` และ `getch()` จะใช้สำหรับรับข้อมูลชนิดอักขระเหมือนกัน แต่ทั้ง 2 ฟังก์ชันนี้ มีความแตกต่างกันอยู่ โดยฟังก์ชัน `getchar()` เมื่อป้อนอักขระเข้ามาแล้วจะต้องกดปุ่ม `<Enter>` โปรแกรมจึงจะกลับไปทำงานต่อได้ และตัวอักษรที่เรากรอกจะแสดงขึ้นมาให้เห็นบนหน้าจอด้วย ตัวอย่างการเรียกใช้ฟังก์ชัน `getchar()` รับตัวอักษรเข้ามาจากคีย์บอร์ดแสดงดังต่อไปนี้

```
#include<stdio.h>
main()
{
char ans;
printf("Enter you answer (Y/N) : ");
ans = getchar(); // เรียกใช้คำสั่ง getchar() รับอักขระเข้ามาเก็บไว้ในตัวแปร ans
printf("Thank you\n"); //แสดงข้อความขอบคุณออกทางหน้าจอ
}
```

เมื่อรันโปรแกรมจะปรากฏข้อความขึ้นมาให้ป้อนอักขระ ดังรูปต่อไปนี้

```
Enter you answer (Y/N) :
```

เมื่อเราป้อนอักขระลงไป โปรแกรมก็จะกลับไปทำงานต่อ โดยแสดงข้อความ `Thank you` ขึ้นมา แต่จะสังเกตได้ว่าการใช้คำสั่ง `getchar()` จะแสดงอักขระที่เราป้อนเข้าไปให้เห็นบนหน้าจอ

```
Enter you answer (Y/N) : Y
Thank you
```

เล่มที่ 3 ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูลและฟังก์ชันทางคณิตศาสตร์

ในส่วนของการใช้คำสั่ง `getch()` หลังจากป้อนอักขระเข้ามา 1 ตัวแล้ว เราไม่ต้องกดปุ่ม `<Enter>` โปรแกรมจะกลับไปทำงานต่อทันที และอักขระที่เราป้อนเข้าไปในโปรแกรมก็จะไม่แสดงขึ้นมาให้เห็นบนหน้าจอด้วย โดยในการใช้คำสั่ง `getch()` เราต้องเขียนพรีโพรเซสเซอร์โคเรคทีฟ `#include<conio.h>` ไว้ที่ส่วนหัวของโปรแกรมด้วย ดังแสดงในตัวอย่างต่อไปนี้

```
#include <stdio.h>
#include <conio.h> // นำไฟล์ conio.h มารวมกับโปรแกรม เพื่อใช้ฟังก์ชัน getch()
main()
{
    char ans;
    printf("Enter you answer (Y/N) : ");
    ans = getch(); // เรียกใช้ฟังก์ชัน getch() รับอักขระเข้ามาเก็บไว้ในตัวแปร ans
    printf("Thank you\n");
}
```

เมื่อสั่งรันโปรแกรมจะปรากฏข้อความขึ้นมาให้ป้อนอักขระ เมื่อป้อนอักขระเข้าไปแล้ว โปรแกรมก็จะกลับไปทำงานต่อทันทีโดยไม่กดปุ่ม `<Enter>` และจะไม่แสดงอักขระที่ป้อนขึ้นมาให้เห็นบนหน้าจอ

```
Enter you answer (Y/N) : Thank you
```

3. ฟังก์ชัน `gets()`

นอกจากคำสั่งพิเศษในการรับข้อมูลชนิดอักขระ หรือตัวอักษรแล้ว การรับข้อความจากคีย์บอร์ดก็มีคำสั่งพิเศษให้เรียกใช้เช่นกัน นั่นก็คือ คำสั่ง `gets()` ซึ่งมีรูปแบบการเรียกใช้ดังนี้

```
gets(str);
```

str : ตัวแปรสำหรับเก็บข้อความที่รับเข้ามาจากคีย์บอร์ด

เล่มที่ 3 ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูลและฟังก์ชันทางคณิตศาสตร์

ตัวอย่าง การเขียนโปรแกรมที่มีการเรียกใช้คำสั่ง gets() รับข้อความเข้ามาจากคีย์บอร์ด

```
#include<stdio.h>
main()
{
char province[30]; // สร้างตัวแปรชนิด string สำหรับเก็บข้อความขนาดสูงสุด 29 ตัวอักษร
printf("Enter province name : "); // แสดงข้อความให้กรอกชื่อจังหวัด
gets(province); // ใช้คำสั่ง gets รับข้อความเข้ามาเก็บไว้ในตัวแปร province
printf("You stay in %s\n",province); // แสดงข้อความใน province ทางหน้าจอ
}
```

เมื่อสั่งรันโปรแกรมจะปรากฏข้อความขึ้นมาให้ป้อนอักขระ

Enter province name : _

ให้ทดลองพิมพ์ข้อความ แล้วกดปุ่ม <Enter> เพื่อให้โปรแกรมกลับไปทำงานในคำสั่งต่อไป

Enter province name : NakhonSawan

โปรแกรมจะแสดงข้อความที่เราป้อนขึ้นมาบนหน้าจอ ดังรูปต่อไปนี้

Enter province name : NakhonSawan

You stay in NakhonSawan

สรุป ฟังก์ชันการรับข้อมูล ในภาษาซีมีฟังก์ชัน scanf () เป็นฟังก์ชันที่ใช้สำหรับรับค่าข้อมูลจากทางแป้นพิมพ์ เพื่อนำค่ามาจัดเก็บไว้ในตัวแปร โดยจะต้องกำหนดรหัสรูปแบบข้อมูลให้สัมพันธ์กับชนิดข้อมูลของตัวแปรที่ต้องการจัดเก็บ มีฟังก์ชัน getchar () และ getch () สำหรับรับค่าตัวอักษรหรืออักขระ 1 ตัว และมีฟังก์ชัน gets () สำหรับรับค่าข้อความจากทางแป้นพิมพ์

เล่มที่ 3 ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูลและฟังก์ชันทางคณิตศาสตร์

กิจกรรมที่ 3.2 ฟังก์ชันการรับข้อมูล (คะแนนเต็ม 5 คะแนน)

<ใช้เวลาทำ 20 นาที>

คำชี้แจง

ให้นักเรียนตอบคำถามต่อไปนี้

1. จงเขียนคำสั่งเพื่อรับข้อมูลจากคีย์บอร์ดจากเงื่อนไขต่อไปนี้ (2 คะแนน)

1.1 รับข้อมูลชนิดจำนวนเต็มเข้ามา 1 จำนวน โดยให้เก็บไว้ที่ตัวแปร a

1.2 รับข้อมูลชนิดจำนวนเต็มและทศนิยมอย่างละ 1 จำนวน โดยใช้เครื่องหมาย , (Comma) เป็นตัวแยกในการป้อนข้อมูล ข้อมูลที่รับเข้ามาให้เก็บไว้ที่ตัวแปร u และ v ตามลำดับ

1.3 รับอักขระจากคีย์บอร์ดเข้ามาเก็บไว้ที่ตัวแปร Q

1.4 รับข้อความจากคีย์บอร์ดเข้ามาเก็บไว้ในตัวแปร document

2. จงเขียนโปรแกรมเพื่อรับค่ารัศมีของวงกลม (radius) ผ่านทางคีย์บอร์ด จากนั้นให้คำนวณหาพื้นที่ของวงกลม (area) และขนาดเส้นรอบวง (circumference) (3 คะแนน)

แนวตอบกิจกรรมที่ 3.2

- ฟังก์ชัน scanf() เป็นฟังก์ชันที่ใช้สำหรับรับค่าข้อมูลจากทางแป้นพิมพ์
ฟังก์ชัน getchar() และ getch() สำหรับรับค่าตัวอักษรหรืออักขระ 1 ตัว
ฟังก์ชัน gets() สำหรับรับค่าข้อความจากทางแป้นพิมพ์
- พื้นที่ของวงกลม (area) = $PI * radius * radius$
เส้นรอบวง (circumference) = $2 * PI * radius$
 $PI = 3.14$

เล่มที่ 3 ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูลและฟังก์ชันทางคณิตศาสตร์

หัวเรื่องที่ 3.3 ฟังก์ชันจัดการจอภาพ

- เวลาเรียน 10 นาที -

สำหรับฟังก์ชันเพื่อการจัดการจอภาพ ในที่นี้จะขอกล่าวถึงฟังก์ชันที่ใช้สำหรับการล้างหน้าจอและฟังก์ชันที่ใช้กำหนดตำแหน่งบนจอภาพ ได้แก่

1. ฟังก์ชันสำหรับการล้างหน้าจอ

เป็นฟังก์ชันที่ใช้สำหรับล้างจอภาพ โดยจะล้างข้อความเดิมบนจอภาพออกทั้งหมด
รูปแบบ

```
system("cls");
```

ตัวอย่าง

```
#include<stdio.h>
main()
{
    printf("This is message #1");
    getch();
    system("cls");
    printf("This is message #2");
}
```

เมื่อสั่งรันโปรแกรมจะปรากฏข้อความขึ้นมาว่า

```
This is message #1
```

จากนั้นกดปุ่ม <Enter> เพื่อให้โปรแกรมกลับไปทำงานในคำสั่งต่อไป ก็จะปรากฏข้อความขึ้นมาว่า

```
This is message #2
```

โดยที่ข้อความแรกจะหายไปจากหน้าจอ

เล่มที่ 3 ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูลและฟังก์ชันทางคณิตศาสตร์

2. ฟังก์ชันกำหนดตำแหน่งบนจอภาพ

เป็นการกำหนดตำแหน่งเคอร์เซอร์บนหน้าจอแบบเท็กซ์โหมด โดยใช้การเขียนโปรแกรมด้วยรูปแบบ

```
gotoxy(column, row);
```

โดยที่ column คือ สดมภ์
row คือ แถว

ตัวอย่าง

```
#include<stdio.h>
#include<conio.h>
#include<windows.h>
void gotoxy(short x, short y);
main()
{ gotoxy(40,15);
  printf("Hi i am,here");
  return 0; }
void gotoxy(short x, short y)
{ COORD pos ={x,y};
  SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), pos); }
```

เมื่อสั่งรันโปรแกรมจะปรากฏข้อความขึ้นมาว่า

Hi i am,here บนหน้าจอที่ตำแหน่งคอลัมน์ที่ 40 และแถวที่ 15

สรุป ฟังก์ชันจัดการจอภาพ ในภาษาซีที่นิยมนำมาใช้งาน ได้แก่ ฟังก์ชัน system("cls"); เป็นฟังก์ชันที่ใช้สำหรับการล้างหน้าจอเพื่อให้หน้าจอกลับมาเริ่มต้นแสดงผลใหม่โดยที่ไม่ต้องสั่งรันโปรแกรมซ้ำและฟังก์ชันกำหนดตำแหน่งเคอร์เซอร์บนหน้าจอเพื่อทำให้ผู้พัฒนาโปรแกรมสามารถกำหนดการแสดงผลตามที่ต้องการได้สะดวกขึ้น โดยการเขียนฟังก์ชันใช้งาน gotoxy();

เล่มที่ 3 ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูลและฟังก์ชันทางคณิตศาสตร์

กิจกรรมที่ 3.3 ฟังก์ชันจัดการจอภาพ (คะแนนเต็ม 5 คะแนน)

<ใช้เวลาทำ 10 นาที>

คำชี้แจง

ให้นักเรียนตอบคำถามต่อไปนี้

1. จงเขียนโปรแกรมแสดงผลชื่อนักเรียนบนหน้าจอที่ตำแหน่งคอลัมน์ที่ 10 และแถวที่ 5 จากนั้นเมื่อกดแป้นพิมพ์ enter ให้ล้างหน้าจอและแสดงผลนามสกุลของนักเรียนบนหน้าจอที่ตำแหน่งคอลัมน์ที่ 30 และแถวที่ 15 (5 คะแนน)

แนวตอบกิจกรรมที่ 3.3

ตัวอย่างโปรแกรมการกำหนดตำแหน่งการแสดงผลบนหน้าจอ

```
#include<stdio.h>
#include<conio.h>
#include<windows.h>
void gotoxy(short x, short y);
main()
{
    gotoxy(40,15);
    printf("Hi i am,here");
    return 0;
}
void gotoxy(short x, short y)
{
    COORD pos ={x,y};
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), pos);
}
```

ฟังก์ชันล้างหน้าจอ คือ system("cls");

เล่มที่ 3 ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูลและฟังก์ชันทางคณิตศาสตร์

หัวเรื่องที่ 3.4 ฟังก์ชันทางคณิตศาสตร์

- เวลาเรียน 10 นาที -

จากเนื้อหาที่ผ่านมา นักเรียนได้เรียนรู้เกี่ยวกับตัวดำเนินการทางคณิตศาสตร์มาบ้างแล้ว ซึ่งประกอบด้วยเครื่องหมาย +, -, *, / และ % สิ่งเหล่านี้เป็นเพียงพื้นฐานเท่านั้น หากต้องการคำนวณหาค่าเลขยกกำลัง หาค่ารากที่สอง ค่าตรีโกณมิติอย่าง เช่น sine, cos, tan ในภาษาซี ตัวดำเนินการทางคณิตศาสตร์ดังกล่าว จะถูกประกาศใช้งานอยู่ในเฮดเดอร์ไฟล์ชื่อ <math.h> โดยเฉพาะเครื่องหมายยกกำลัง ซึ่งในภาษาระดับสูงทั่วไป มักจะใช้เครื่องหมาย ^ หรือ ** แต่ภาษาซีจะใช้ฟังก์ชัน pow(); แทน รายละเอียดของฟังก์ชันทางคณิตศาสตร์ที่ประกาศใช้งานในเฮดเดอร์ไฟล์ ได้แก่

ฟังก์ชัน	ความหมาย
pow (x, y)	ค่าของ x ยกกำลัง y
sqrt (x)	หารากที่สองของ x
fmod (x, y)	หารเพื่อเอาเศษ ระหว่าง x โมดูลัส y
sin (x)	หาค่า sine ของ x
cos (x)	หาค่า cosine ของ x
tan (x)	หาค่า tangent ของ x
exp (x)	ฟังก์ชันเอ็กซ์โปเนนต์
log (x), log10 (x)	ลอการิทึมของ x
fabs (x)	ค่าสัมบูรณ์ (Absolute) ของ x
ceil (x)	ปัดเศษทศนิยมขึ้น
floor (x)	ปัดเศษทศนิยมทิ้ง

เล่มที่ 3 ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูลและฟังก์ชันทางคณิตศาสตร์

ตัวอย่าง

โปรแกรมทดสอบการใช้งานฟังก์ชันเพื่อหาค่ายกกำลังและหารากที่สอง

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
main()
{
    double x,y,z;
    x=64.0;
    y=3.0;
    z=0.5;
    printf("Pow(64.0, 3.0) return : %7.0f\n",pow(x,y));
    printf("Pow(64.0, 0.5) return : %2.0f\n",pow(x,z));
    printf("sqrt(64.0)    return : %2.0f\n",sqrt(x));
}
```

ผลการรันโปรแกรม

Pow(64.0, 3.0) return : 262144

Pow(64.0, 0.5) return : 8

sqrt(64.0) return : 8

สรุป ฟังก์ชันทางคณิตศาสตร์ ในภาษาซี เป็นฟังก์ชันที่ถูกประกาศใช้งานอยู่ในเฮดเดอร์ไฟล์ `<math.h>` ได้แก่ `pow()`, `sqrt()`, `sine()`, `cos()`, `tan()`, `ceil()` และ `floor()` อำนวยความสะดวกในการใช้งาน หากต้องการคำนวณหาเลขยกกำลัง การหารากที่สอง หรือค่าตรีโกณมิติ เป็นต้น

เล่มที่ 3 ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูลและฟังก์ชันทางคณิตศาสตร์

กิจกรรมที่ 3.4 ฟังก์ชันทางคณิตศาสตร์ (คะแนนเต็ม 5 คะแนน)

<ใช้เวลาทำ 10 นาที>

คำชี้แจง

ให้นักเรียนตอบคำถามต่อไปนี้

1. จงเขียนโปรแกรมรับตัวเลขทศนิยม 1 จำนวน ผ่านทางแป้นพิมพ์ จากนั้นให้แสดงผลทางหน้าจอที่เกิดจากการนำตัวเลขที่รับเข้าไปนำไปปัดเศษขึ้นและปัดเศษลง (5 คะแนน)

ตัวอย่างเช่น ป้อนจำนวน 8.5 เข้าไปทางแป้นพิมพ์ ผลลัพธ์ที่แสดงผลบนหน้าจอจะมี 2 ค่า คือ 8.0 (กรณีปัดเศษลง) และ 9.0 (กรณีปัดเศษขึ้น)

แนวตอบกิจกรรมที่ 3.4ฟังก์ชัน $\text{ceil}(x)$ ใช้สำหรับการปัดเศษทศนิยมขึ้นฟังก์ชัน $\text{floor}(x)$ ใช้สำหรับการปัดเศษทศนิยมทิ้งเมื่อ x คือตัวแปรที่ใช้เก็บค่าที่ป้อนเข้าไปผ่านทางแป้นพิมพ์