

บทที่ 3 โครงสร้างของโปรแกรมภาษาซี (ต่อ)

การรับข้อมูลและแสดงผลข้อมูล

ภาษาซีได้เตรียมฟังก์ชันมาตรฐานที่ใช้ในการรับและแสดงผลข้อมูลให้กับผู้พัฒนาโปรแกรมหลายฟังก์ชัน แต่ในที่นี้จะกล่าวถึงเฉพาะฟังก์ชันในการรับข้อมูล คือ ฟังก์ชัน `scanf( )` และฟังก์ชันในการแสดงผลข้อมูล คือ ฟังก์ชัน `printf( )` ในส่วนของการใช้งานพื้นฐาน ซึ่งก่อนจะใช้งานฟังก์ชันดังกล่าวที่ส่วนหัวของโปรแกรมจะต้องมีคำสั่ง

```
#include <stdio.h>
```

3.10 การรับข้อมูล

ฟังก์ชัน ที่ใช้ในการรับข้อมูลมีรูปแบบของการใช้งานคือ

```
scanf(รูปแบบ, argument1, argument2, .....);
```

ในการรับข้อมูลผู้เขียนโปรแกรมจะต้องกำหนดรูปแบบของข้อมูลที่ต้องการรับซึ่งสอดคล้องกับชนิดของข้อมูลที่ต้องการรับเข้า โดยที่ผู้ใช้ต้องส่งตำแหน่ง (หรือ แอดเดรสในหน่วยความจำ – Address) ของตัวแปรที่ต้องการรับเข้าไปยังฟังก์ชัน โดยระบุในตำแหน่งของอาร์กิวเมนต์

ตัวอย่างเช่น ต้องการรับข้อมูลเดือนและปีเป็นจำนวนเต็ม จะต้องใช้คำสั่ง

```
int month, year;  
  
scanf("%d %d", &month, &year);
```

จากตัวอย่างรูปแบบ คือ “%d %d” บอกให้รู้ว่าต้องรับข้อมูล 2 ตัว โดยที่ข้อมูลนั้นแยกจากกันด้วยช่องว่าง %d ตัวแรกแทนข้อมูลตัวแรกที่ป้อนเข้าสู่ระบบ ซึ่งจะนำไปเก็บที่ตัวแปร month ส่วน %d ตัวที่ 2 แทนข้อมูลตัวที่ 2 ที่ป้อนเข้าสู่ระบบ ซึ่งจะถูกลำไปเก็บที่ตัวแปร year ในที่นี้จะระบุตำแหน่งของตัวแปรในหน่วยความจำด้วยเครื่องหมาย & (Address Operator)

รูปแบบของการรับข้อมูลในภาษาซี จะขึ้นต้นด้วยสัญลักษณ์ % อยู่ในเครื่องหมาย “ ” รูปแบบการรับข้อมูลแสดงได้ดังตารางด้านล่าง

ตาราง แสดงรูปแบบของการรับข้อมูล

รูปแบบ	คำอธิบายการใช้งาน
d, i	รับข้อมูลจำนวนเต็มและแปลงข้อมูลเป็น int
ld	รับข้อมูลจำนวนเต็มและแปลงข้อมูลเป็น long
u	รับข้อมูลเป็นจำนวนเต็มบวกและแปลงข้อมูลเป็น unsigned
o	รับข้อมูลเป็นจำนวนเต็มบวกของเลขฐาน 8 และแปลงข้อมูลเป็น unsigned

รูปแบบ	คำอธิบายการใช้งาน
x	รับข้อมูลเป็นจำนวนเต็มบวกของเลขฐาน 16 และแปลงข้อมูลเป็น unsigned
c	รับข้อมูลเป็นอักขระ 1 ตัว
s	ข้อมูลเป็นสตริง
f	ข้อมูลทศนิยมและแปลงข้อมูลเป็น float
lf	ข้อมูลทศนิยมและแปลงข้อมูลเป็น double

การจัดรูปแบบการแสดงผล

กรณีของการใช้รูปแบบในฟังก์ชัน printf() จะแตกต่างกับฟังก์ชัน scanf() ตรงที่ฟังก์ชัน printf() ใช้ในการแสดงผลออกทางหน้าจอ เพราะฉะนั้นจึงมีการกำหนดค่าเพื่อให้สามารถจัดผลลัพธ์ให้แสดงในรูปแบบที่ต้องการได้ โดยปกติแล้วข้อมูลทุกตัวที่แสดงจะแสดงผลจากทางด้านขวามือ เช่น

```
printf("The sum of %5d and %5d is %5s\n", a, b, a+b);
```

ผลลัพธ์ที่ได้คือ The sum of 7 and 2 is 9.

จะเกิดการเว้นว่าง 5 ช่อง และแสดงผลจากทางด้านขวามือของช่องว่างนั้น

ตัวอย่างอื่นได้แก่

```
float x=43.34, y=2.231;
```

```
printf("Minus %f with %f, answer is %f", x, y, x-y);
```

ผลลัพธ์ที่ได้คือ

Minus 43.340000 with 2.231000, answer is 41.109000

แต่หากกำหนดว่า

```
printf("Minus %4.2f with %4.2f, answer is %4.2f");
```

ผลลัพธ์ที่ได้คือ

Minus 43.34 with 2.23, answer is 41.11

การจัดรูปแบบเมื่อใช้สัญลักษณ์ – ประกอบในรูปแบบ จะทำให้เกิดการบังคับให้การแสดงผลชิดทางด้านซ้ายมือ แสดงดังตัวอย่างที่ด้านล่าง (การจัดรูปแบบการแสดงผล)

```
#include <stdio.h>
```

```
#include <conio.h>
```

```

void main( ) {
    clrscr( );
    printf("\n%s", "Computer"); /* [Computer] */
    printf("\n%2s", "Computer"); /* [Computer] */
    printf("\n%.3s", "Computer"); /* [Com] */
    printf("\n%10s", "Computer"); /* [ Computer] */
    printf("\n%-10s", "Computer"); /* [Computer ] */
    printf("\n%-10.3s", "Computer"); /* [Com ] */
    printf("\n");
    printf("\n[%x]", 15); /* Hexa [f] */
    printf("\n[%o]", 15); /* Octal [17] */
    printf("\n");
    printf("\n[%d]", 100); /* [100] */
    printf("\n%.2d", 100); /* [100] */
    printf("\n%10d", 100); /* [ 100] */
    printf("\n%-10d", 100); /* [100 ] */
    printf("\n%-10.2d", 100); /* [100 ] */
    printf("\n");
    printf("\n[%f]", 32.5762); /* [32.576200] */
    printf("\n%.2f", 32.5762); /* [32.58] */
    printf("\n%10.2f", 32.5762); /* [ 32.58] */
    printf("\n%-10.2f", 32.5762); /* [32.58 ] */
    getch( );
}

```

ในตัวนี้มีการใช้ฟังก์ชันเพิ่มเติม เพื่อช่วยให้เห็นผลลัพธ์ของการแสดงผลที่จอภาพ ในกรณีนี้ผู้เขียนโปรแกรมใช้คอมไพเลอร์ Turbo C คือ

- clrscr() ใช้เพื่อให้ลบข้อความต่าง ๆ ที่ค้างบนจอภาพก่อนที่จะแสดงผลโปรแกรม
- getch() ใช้เพื่อรอรับการกดแป้นพิมพ์ใด ๆ หลังจากแสดงผลแล้ว เพื่อให้เกิดการค้างรอให้ผู้ใช้งานป้อนพิมพ์ก่อนจะไปทำงานอื่นต่อไป

ทั้ง 2 ฟังก์ชันอยู่ในฟังก์ชันมาตรฐานของคอมไพเลอร์ Turbo C ต้องเรียกใช้ include ไฟล์ชื่อ conio.h

ตัวอย่างที่ แสดงตัวอย่างการรับข้อมูลเพื่อนำมาแสดงผล

```
#include <stdio.h>

void main( ) {

    char name[100];

    printf("What is your name ?\n");

    scanf("%s", name);

    printf("Very glad to know you, ");

    printf("%s.",name);

}
```

ผลลัพธ์ของการทำงาน

What is your name ?

Willy

Very glad to know you, Willy.

หมายเหตุ ตัวเอียงคือข้อความที่ผู้ใช้ป้อนเข้าสู่ระบบ

ข้อควรระวัง

- การป้อนข้อมูลที่เป็นสตริงไม่จำเป็นต้องใส่เครื่องหมาย & นำหน้าตัวแปรสตริง
- หากชื่อที่ป้อนเข้าสู่โปรแกรมมีช่องว่าง เช่นผู้ใช้ป้อนชื่อและนามสกุล โดยมีช่องระหว่างชื่อและนามสกุล โปรแกรมจะรับได้เฉพาะข้อมูลชื่อเท่านั้น โปรแกรมไม่สามารถรับข้อมูล ที่มีช่องว่างเข้ามาเก็บในตัวแปรได้