

## บทที่สี่

### คำสั่งในภาษา C

คำสั่งในภาษา C สามารถแยกให้ชัดเจนได้เป็น 2 ลักษณะ คือ

1. คำสั่งที่เป็นรูปคำสั่งจริง ๆ เช่น คำสั่งกำหนดค่า, คำสั่งควบคุม เป็นต้น
2. คำสั่งที่เป็นรูปของฟังก์ชัน เช่น คำสั่งรับข้อมูล, คำสั่งแสดงผลข้อมูล เป็นต้น

ดังนั้น ในโปรแกรมภาษา c โปรแกรมหนึ่ง ๆ อาจประกอบด้วยคำสั่งที่เป็นรูปคำสั่งจริงและคำสั่งที่เป็นรูปของฟังก์ชันอยู่ด้วยกันก็ได้เพื่อความเข้าใจในแต่ละรูปคำสั่งจึงอธิบายต่อไปนี้

#### 4.1 คำสั่งคำนวณ (Assignment Statements)

เป็นรูปคำสั่งจริง ๆ ของภาษา C ซึ่งเอาไว้สำหรับให้คอมพิวเตอร์ทำการคำนวณอย่างหนึ่งอย่างใดให้กับเราหรืออาจใช้ในการกำหนดค่าเริ่มต้นให้กับตัวแปร หรือย้ายค่าตัวแปรจากที่หนึ่งไปยังอีกที่หนึ่งก็ได้

รูปแบบ

<Variable> = <expression>;

<Variable> หมายถึง ชื่อตัวแปรที่ทำหน้าที่ในการเก็บค่า

= หมายถึง เป็นเครื่องหมายที่แสดงว่าให้นำค่าที่อยู่ทางขวาไปเก็บไว้ในตัวแปรทางซ้าย

<expression> หมายถึง ตัวแปร, ค่าคงที่หรือนิพจน์ ในกรณีที่ <expression> เป็นค่าคงที่ จะหมายถึงการกำหนดค่าให้กับตัวแปร ดังนี้

##### ตัวอย่างที่ 1

```
main ()
{
    int i;
    i = 10;
```

หมายความว่า มีการประกาศให้ตัวแปร i เป็นชนิดจำนวนเต็ม และให้เอาค่า 10 ไปเก็บไว้ในตัวแปร i

##### ตัวอย่างที่ 2

```
main ()
{
    int a, b, c, d;
    a = b = c = d = 33;
```

หมายความว่า มีการประกาศตัวแปร a, b, c, d เป็นชนิดจำนวนเต็มและทำให้ตัวแปรทุกตัวมีค่า 33 เก็บอยู่

### ตัวอย่างที่ 3

```
main ()
{
    char ch;
    ch = 'A';
```

หมายความว่า มีการประกาศให้ตัวแปร ch เป็นชนิดตัวอักษรซึ่งทำให้ตัวแปร ch สามารถเก็บค่าคงที่ที่เป็นตัวอักษรไว้ได้ 1 ตัวอักษร แล้วจึงนำค่า A ไปเก็บในตัวแปร ch

### ตัวอย่างที่ 4

```
main ()
{
    float f;
    f = 45.5;
```

หมายความว่า ได้มีการประกาศให้ตัวแปร f เป็นตัวแปรชนิดทศนิยมแล้วนำค่า 45.5 ไปเก็บในตัวแปร f

### ตัวอย่างที่ 5

```
main ()
{
    int r;
    float t;
    r = 36.75;
    t = 45;
```

หมายความว่า ได้มีการประกาศให้ตัวแปร r เป็นตัวแปรชนิดจำนวนเต็มและตัวแปร t เป็นตัวแปรชนิดทศนิยมดังนั้นเมื่อเอาค่า 36.75 ไปเก็บในตัวแปร r ตัวแปร r จะเก็บค่า 36 และตัวแปร t จะเก็บค่า 45.000000 ซึ่งเป็นค่าทศนิยมตามที่ประกาศชนิดไว้

ตัวอย่างที่ 6 แสดงการกำหนดค่าข้อมูลที่เป็นข้อความ (string) ไปเก็บในตัวแปรชุด n ชนิด 1 มิติ ขนาด 9 สมาชิก (element)

```
main ()
{
    char n [9] = "computer"
```

หมายความว่า ในหน่วยความจำจะเอาค่า computer ไปเก็บไว้ในตัวแปรชุด n ดังนี้

n[0]	n[1]	n[2]	n[3]	n[4]	n[5]	n[6]	n[7]	n[8]
C	O	M	P	U	T	E	R	\0

เครื่องหมาย \0 จะเป็นเครื่องหมายที่เครื่องกำหนดเพื่อให้เกิดการสิ้นสุดข้อความ ตัวอย่างที่ 7 แสดงการกำหนดค่าข้อมูลที่เป็นข้อความในตัวแปรชุดทีละ 1 ตัวอักษร

```
main ()
{
    Char n[9] = {'c', 'o', 'm', 'p', 'u', 't', 'e', 'r', '\0'}
```

หมายความว่า มีการประกาศให้ตัวแปร n เป็นตัวแปรชุดที่มีขนาด 9 สมาชิก แล้วนำค่า computer เข้าไปเก็บในตัวแปรชุด แต่จะเก็บทีละ 1 ตัวอักษร ซึ่งถ้าทำอย่างนี้จะต้องมีการใส่เครื่องหมาย \0 เข้าไปด้วยเพื่อเป็นการบอกการสิ้นสุดข้อความ

ตัวอย่างที่ 8 แสดงการเก็บข้อมูลที่เป็นตัวเลขจำนวนเต็มในตัวแปรชุด

```
main ()
{
    int a[5] = {1, 2, 3, 4, 5};
```

หมายความว่า มีการประกาศว่าตัวแปร a เป็นตัวแปรชุดจำนวนเต็มที่มีขนาด 5 สมาชิก และมีการนำค่า 1, 2, 3, 4, 5 ไปเก็บไว้ในหน่วยความจำ ดังนี้

a[0]    a[1]    a[2]    a[3]    a[4]

1	2	3	4	5
---	---	---	---	---

ในกรณีที่ <expressions> ในรูปแบบเป็นตัวแปร จะเป็นการย้ายค่าจากตัวแปรหนึ่งไปยังตัวแปรอีกตัวแปรหนึ่ง

ตัวอย่างที่ 9

```
main ()
{
    int a, b;
    a = 5;
    b = a;
```

หมายความว่า มีการกำหนดให้ a, b เป็นตัวแปรชนิดจำนวนเต็มแล้วให้เอาค่า 5 ไปเก็บไว้ในตัวแปร a จากนั้นย้ายค่าจาก a ไปเก็บไว้ใน b มีค่าเป็น 5 เหมือนกับค่าของ a ด้วย

ตัวอย่างที่ 10 แสดงการคำนวณโดยนำผลที่ได้จากการคำนวณไปเก็บไว้ในตัวแปร v

```
main ()
{
    int a=5;
    int b=3;
```

```
int v;
```

```
v = a * b / 2 - 10
```

นอกจากนี้ในรูปคำสั่งคำนวณยังใช้แสดงการเพิ่มค่าและการลดค่าภายในตัวแปรตัวเดิมได้ เช่น

```
i = i + 1;
```

หมายความว่า มีการเพิ่มค่าในตัวแปร i จากเดิมขึ้นอีก 1 ซึ่งสามารถเขียนใหม่ได้อีกรูปหนึ่ง ดังนี้

```
i ++;
```

หรือ

```
++ i;
```

ในกรณีที่มีการเพิ่มค่าในตัวแปรมากกว่า 1 เช่น

```
i = i + 5;
```

หมายความว่า มีการเพิ่มค่าในตัวแปร i ขึ้นอีก 5 สามารถเขียนอีกรูปหนึ่งได้ดังนี้

```
i += 5;
```

นอกจากนั้น การเขียนในรูปแบบใหม่นี้ยังใช้ได้กับเครื่องหมายการคำนวณอย่างอื่น ๆ ได้อีกด้วย ดังตัวอย่างต่อไปนี้

คำสั่งคำนวณ	รูปที่นิยมใช้ในภาษา C
$x = x + 5$	$x += 5$
$x = x - 5$	$x -= 5$
$x = x * 5$	$x *= 5$
$x = x / 5$	$x /= 5$
$x = x \% 5$	$x \% = 5$

ตัวอย่างที่ 11

```
main ()
```

```
{
```

```
int y, x;
```

```
y = 5;
```

```
x = ++ y;
```

หมายความว่า จะมีการเพิ่มค่า y ขึ้นอีก 1 คือเป็น 6 ก่อนแล้วจึงนำมาเก็บไว้ในตัวแปร x

ตัวอย่างที่ 12

```
main ()
```

```
{
```

```
int y, x;
```

y = 5;

x = y ++;

หมายความว่า จะมีการนำค่า y ซึ่งเป็น 5 ไปเก็บในตัวแปร x ก่อนแล้วจึงจะเพิ่มค่า y ขึ้นอีก 1 เป็น 6

#### 4.2 ฟังก์ชันในการรับข้อมูลและแสดงผลข้อมูล (Input / Output Function)

ในภาษา C การรับข้อมูลและการแสดงผลข้อมูลจะใช้การเรียกฟังก์ชันมาทำงาน ซึ่งฟังก์ชันเหล่านี้จะเป็นฟังก์ชันมาตรฐานที่มีในภาษา C ของทุกบริษัท ซึ่งสามารถแบ่งเป็นกลุ่มใหญ่ ๆ ได้ดังนี้

1. ฟังก์ชันมาตรฐาน ได้แก่ printf(), scanf() เป็นต้น
2. ฟังก์ชันที่เก็บไว้ใน I/O library ได้แก่ฟังก์ชัน getchar(), getch(), putchar() เป็นต้น  
เวลาใช้ฟังก์ชันเหล่านี้จะต้องบอกตัวแปรชุดคำสั่งภาษา C (C compiler) รับรู้ก่อนโดยใช้คำสั่ง #include เพิ่มข้อมูลชื่อ stdio.h เข้ามาก่อนโดยเอาไว้ในส่วนหัว (header) ของโปรแกรม

##### 4.2.1 ฟังก์ชัน printf()

เป็นฟังก์ชันในการแสดงผลข้อมูลที่อยู่ในตัวแปร ค่าคงที่ นิพจน์ออกมาทางจอภาพ

รูปแบบ

```
printf (Control, argument list);
```

argument list เป็นค่าตัวแปรค่าคงที่ หรือนิพจน์ที่ต้องการนำมาแสดงผลถ้ามีมากกว่า 1 ค่าจะแยกกัน โดยใช้เครื่องหมายคอมม่า (,)

control จะต้องเขียนอยู่ภายใต้เครื่องหมาย " " ซึ่งสามารถเขียนได้ 2 ลักษณะ

1. เป็นข้อความที่เป็นคำอธิบายที่ต้องการให้แสดงผลออกมา ส่วนมากจะใช้เพื่อแต่งรูปแบบของรายงานให้สวยงาม เช่น  
printf (" sum of x=");
2. เป็นรหัสรูปแบบ (Format Code) ที่ใช้ในการแสดงผลซึ่งทุกรหัสรูปแบบจะต้องอยู่ตามหลังเครื่องหมาย % รหัสรูปแบบ ข้อมูลที่นิยมใช้ได้แก่

รหัสรูปแบบ	ความหมาย
%c	ใช้กับตัวแปรที่เก็บค่าที่เป็นตัวอักษรเพียงตัวเดียว
%s	ใช้กับตัวแปรที่เก็บค่าที่เป็นข้อความที่เก็บในตัวแปรชุด
%d	ใช้กับตัวแปรที่เก็บค่าที่เป็นเลขจำนวนเต็ม
%u	ใช้กับตัวแปรที่เก็บค่าที่เป็นเลขจำนวนเต็มบวก
%f	ใช้กับตัวแปรที่เก็บค่าที่เป็นเลขทศนิยม
%e	ใช้กับตัวแปรที่เก็บค่าที่เป็นเลขทศนิยมในรูปแบบ e ยกกำลัง
%x	ใช้กับตัวแปรที่เก็บค่าที่เป็นค่าเลขฐานสิบหก

%o	ใช้กับตัวแปรที่เก็บค่าที่เป็นค่าเลขฐานแปด
%p	ใช้กับตัวแปรที่เก็บค่าที่เป็นตัวชี้ตำแหน่ง (pointer)

ตัวอย่างที่ 13 แสดงการใช้คำสั่ง printf ที่มีข้อความเป็นคำอธิบายอยู่ในรูปคำสั่ง

```
main ()
{
    int sum = 5;
    printf ("The sum is % d", sum);
}
```

ผลลัพธ์

The sum is 5

ตัวอย่างที่ 14 แสดงการใช้รหัสรูปแบบ %d และ %u

```
main ()
{
    int d = 12; int u = - 12;
    printf ("d = %d u %d", d, u);
    printf ("d = %u u =%u", d, u);
}
```

ผลลัพธ์

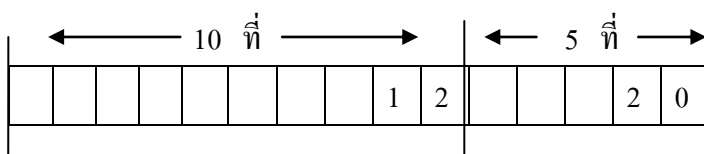
d = 12 u = -12 d = 12 u = 65524

หมายเหตุ ค่า u = 65524 เป็นค่า complement ของ -12 เพราะใช้รหัสรูปแบบ u ซึ่งเป็นรูปแบบ ที่ให้พิมพ์ค่าบวกออกมา

ตัวอย่างที่ 15 แสดงการกำหนดขอบเขตของการพิมพ์ ในแต่ละรหัสรูปแบบ

```
main ()
{
    int a = 12;
    int b = 20;
    printf ("%10d %5d", a, b);
}
```

ผลลัพธ์



ตัวอย่างที่ 16 แสดงในการใช้รหัสรูปแบบ %f และ %e

```
main ()
{
    float e = 11.55;
    printf ("%f %e", e, e);
}
```

ผลลัพธ์

```
11.549999      1.155000e + 01
```

การใช้รหัสรูปแบบ %f และ %e สามารถกำหนดจำนวนตัวเลขหลังจุดทศนิยมได้โดยกำหนดเป็นเลขจำนวนเต็มหลังจุดทศนิยมนางไว้หน้า f หรือ e ตามที่ต้องการ

ตัวอย่างที่ 17 แสดงการใช้รหัสรูปแบบ %f และ %e โดยให้ทศนิยม 2 ตำแหน่ง

```
main ()
{
    float e = 11.55;
    printf ("%0.2f      %0.2e", e, e);
}
```

ผลลัพธ์

```
11.55      1.2e + 01
```

หมายเหตุ %0.2f หมายความว่า ต้องการตัวเลขหลังจุดทศนิยม 2 ตำแหน่ง

%0.2e หมายความว่า ต้องการตัวเลขหลังจุดทศนิยมของเลขที่อยู่หน้า e เท่ากับจำนวนตัวเลขที่กำหนดลบหนึ่ง

ตัวอย่างที่ 18

```
main ()
{
    float a = 88.8, b = 88.8666;
    printf ("%0.2f      %0.4e", a, b);
    printf ("%0.0f      %0.3e", a, b);
}
```

ผลลัพธ์

```
88.80      8.887 e + 01  89      8.9 e + 01
```

ตัวอย่างที่ 19 แสดงการใช้รหัสรูปแบบ %O และ %x

```

main ()
{
    int i = 32;
    printf ("%O      %x", i, i);
}

```

ผลลัพธ์

```

40      20

```

หมายเหตุ ค่า 40 เป็นค่าในฐานแปดของค่า i ซึ่งมีค่าเป็น 32 ในฐานสิบ

ค่า 20 เป็นค่าในฐานสิบหกของค่า i ซึ่งมีค่าเป็น 32 ในฐานสิบ

ระยะห่าง %O กับ %x จะมีผลทำให้ผลลัพธ์มีระยะห่างเท่ากับระยะห่างของ %O กับ %x

ด้วย และถ้า %O กับ %x ติดกันจะมีผลให้ผลลัพธ์ติดกันด้วย

ตัวอย่างที่ 20 แสดงการใช้รหัสรูปแบบ %c และ %s

```

main ()
{
    char ch = 'A';
    char str[] = "computer"
    printf ("%c", ch);
    printf ("%d", ch);
    printf ("%s", str);
    printf ("%3s", str);
}

```

ผลลัพธ์

```

A      65      computer      com

```

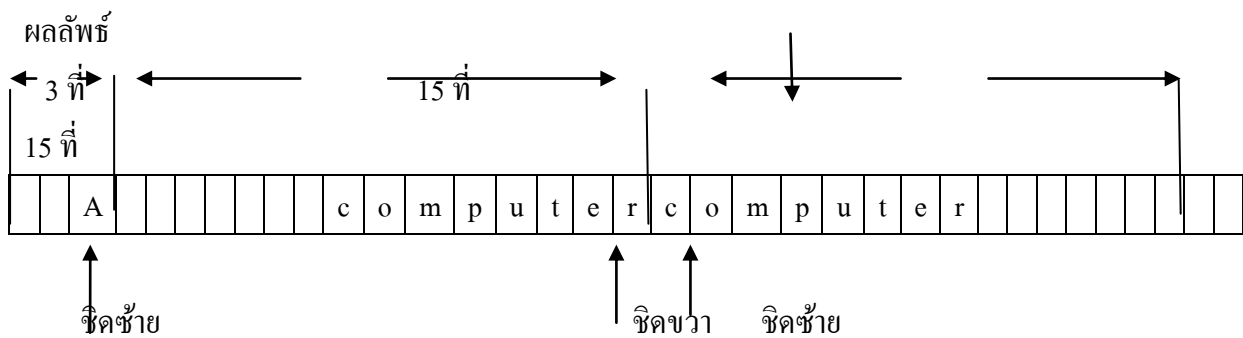
ตัวอย่างที่ 21

```

main ()
{
    char ch = 'A';
    char str [] = "computer";
    printf ("% 3c", ch);
    printf ("%15.8 s", str);
    printf ("% -15.8 s", str);
}

```





ผลลัพธ์ จากตัวอย่างที่ 20 และตัวอย่างที่ 21 สำหรับ format %c และ %s ตัวเลขที่อยู่หลัง % จะแสดงขนาดของพื้นที่ที่จะแสดงผล เช่น

%3c หมายถึงเตรียมไว้ 3 ที่ สำหรับแสดงผลของตัวแปรชนิดอักขระ (character) ที่สั่งพิมพ์

%15.8s หมายถึงเตรียมไว้ที่ 3 ที่ สำหรับแสดงผลของตัวแปรชนิดข้อความ str แต่จะนำค่ามาแสดงผลเพียง 8 ตัวอักษรเท่านั้น

%-15.8s เครื่องหมาย - จะเป็นตัวบอกให้เครื่องรู้ว่าจะให้พิมพ์ข้อความชิดขอบซ้ายของเขตข้อมูล

การใช้รหัสควบคุมการแสดงผล (Carriage Control)

ในภาษา C จะใช้

\n หมายถึง การขึ้นบรรทัดใหม่

\t หมายถึง การเว้นระยะครั้งละ 6 ตัวอักษรแนวนอน

\r หมายถึง ให้ตัวชี้ตำแหน่ง (cursor) กลับไปอยู่ที่ต้นบรรทัด

\f หมายถึง ให้เว้นการแสดงผลไปหนึ่งหน้า

\b หมายถึง ให้ตัวชี้ตำแหน่งถอยหลังกลับไปทีละ 1 ตัวอักษรพร้อมกับลบตัวอักษรไปด้วย  
รหัสควบคุมการแสดงผลเหล่านี้จะเขียนอยู่ในของ control ในรูปคำสั่ง printf

ตัวอย่างที่ 22

```
main ()
{
    char ch = 'A';
    char str [ ] = "computer"
    printf ("\n %c \n \n %s", ch, str);
}
```

ผลลัพธ์

```
} เว้น 1 บรรทัด
```

```
A
```

```
} เว้น 1 บรรทัด
```

```
computer
```

ตัวอย่างที่ 23

```
main ()
{
    char ch = 'A';
    char str [ ] = "computer";
    printf ("%c \t \t CHARACTER \b %s ", ch, str);
}
```

ผลลัพธ์

```
A          CHARACTER computer
```

```
12 ตัวอักษร
```

4.2.2 ฟังก์ชัน scanf()

เป็นฟังก์ชันที่ใช้ในการรับข้อมูลจากแป้นพิมพ์ (key board) เข้ามาเก็บไว้ในตัวแปร

รูปแบบ

```
scanf (Control, argument list);
```

argument list เป็นตัวแปรที่จะรับค่าจากแป้นพิมพ์มาเก็บไว้ถ้ามีมากกว่าหนึ่งค่า จะต้องแยกกันด้วยเครื่องหมายคอมม่า (,) และตัวแปรทุกตัวที่ใช้จะต้องนำหน้าด้วยเครื่องหมาย & ทุกตัว ยกเว้นตัวแปรชุด

control เป็นรหัสรูปแบบที่ใช้ในการรับข้อมูลซึ่งจะต้องเขียนอยู่ภายใต้เครื่องหมาย " " และจะมีรูปแบบเหมือนกับรหัสรูปแบบที่ใช้ในคำสั่ง printf

ตัวอย่างที่ 24

```
main ()
{
    int a;
    scanf ("%d" , &a);
```

หมายความว่า เมื่อเครื่องทำคำสั่ง scanf เครื่องจะรอรับค่าที่เป็นจำนวนเต็ม 1 ค่าที่จะป้อนเข้ามาทางแป้นพิมพ์มาเก็บไว้ในตัวแปร a

ตัวอย่างที่ 25

```
main ()
{
    int a, b;
    scanf("%d %d", &a, &b);
```

ลักษณะข้อมูลที่ป้อน

123 34

หมายความว่า เครื่องจะรับค่าที่เป็นจำนวนเต็ม 2 ค่าจากแป้นพิมพ์มาเก็บไว้ในตัวแปร a และ b โดย a มีค่า 123 b มีค่า 34 ตามลำดับ สังเกต format จะต้องมีการเว้นระยะระหว่าง format ให้สอดคล้องกับค่าของข้อมูลที่ได้รับเข้ามา

ตัวอย่างที่ 26

```
main ()
{
    int a, b;
    scanf(" %d, %d", &a, &b);
```

ลักษณะข้อมูลที่ป้อน

125, 45

หมายความว่า เครื่องจะรับค่าที่เป็นจำนวนเต็ม 2 ค่าจากแป้นพิมพ์มาเก็บไว้ในตัวแปร a และ b โดย a มีค่า 125 b มีค่า 45 ค่าของ a, b ที่ป้อนเข้าไปจะแยกกันด้วยเครื่องหมายคอมม่าใน format ที่ใช้ก็จะแยกกันด้วยเครื่องหมายคอมม่า ตามลักษณะข้อมูลที่ป้อนเข้าไป

ในกรณีที่ตัวแปรจะเก็บค่าตัวเลขจะไม่สามารถใช้ format ที่เหมือนกัน 2 format ติดกันได้ เพราะเครื่องจะไม่สามารถแยกความยาวของแต่ละค่าออกจากกันได้ แต่ถ้า format ต่างชนิดกัน หรือเป็น format ของตัวอักษร หรือตัวแปรชุดจะสามารถเขียนติดกันได้เพราะเครื่องจะรู้ขนาดอยู่แล้ว

ตัวอย่างที่ 27

```
main ()
{
    int a;
    char b;
    scanf(" %d %c", &a, &b);
```

ลักษณะข้อมูลที่ป้อน

789g

หมายความว่า เครื่องจะค่าที่เป็นเลขจำนวนเต็ม 789 เข้าไปเก็บในตัวแปร a และตัวแปร b จะมีค่าตัวอักษรเก็บอยู่อย่างถูกต้อง

ตัวอย่างที่ 28

```
main ()
{
    char a, b, c;
    scanf ("%c %c %c", %a, %b, %c);
```

ลักษณะข้อมูลที่ป้อน

IBM

หมายความว่า เครื่องจะเก็บค่า I มาเก็บไว้ในตัวแปร a B มาเก็บไว้ในตัวแปร b และ M จะเก็บค่าไว้ในตัวแปร C ตามลำดับ

ตัวอย่างที่ 29 เป็นการแสดงการรับค่าความยาวเป็นฟุตมาจากแป้นพิมพ์แล้วเปลี่ยนค่านั้นให้เป็นเมตร

```
/*sample program feet to meters*/
main () {
    int feet;
    float meters;
    printf ("Enter number of feet :");
    scanf ("%d", & feet);
    meters = feet * 0.3048;
    printf ("%d feet is %.4f meters", feet,meters);
}
```

ผลลัพธ์

```
Enter number of feet : 10
10 feet is 3.0480 maters
```

ตัวอย่างที่ 30 แสดงการรับค่าที่เป็นข้อความจากแป้นพิมพ์มาเก็บไว้ในตัวแปร a ที่กำหนดให้

```
main ()
{
    Char a[9]
    scanf ("%s", a);
```

ลักษณะข้อมูลที่ป้อน

COMPUTER

ผลจะได้ข้อความ COMPUTER ถูกนำไปเก็บไว้ในตัวแปรชุด a ดังนี้

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]
C	O	M	P	U	T	E	R	\0

ในกรณีที่ถ้ามีการป้อนข้อมูลเข้าไปไม่ตรงกับชนิดของตัวแปรเครื่องจะเลิกรับข้อมูลทันที ตัวอย่างที่ 31 แสดงการรับข้อมูลที่ประกอบด้วย เลขประจำตัว, ชื่อ, เงินเดือน, อายุ จากเป็นพิมพ์ แล้วนำมาแสดงผลทางจอภาพ

```
main ()
{
    char id[12], name[30];
    float salary;
    printf age;
    printf ("ID NUMBER :");
    scanf ("%s", id);
    printf ("SALARY : ");
    scanf ("%f",& salary);
    printf ("AGE:");
    scanf ("%d", & age);
    printf ("\n \n \n ID. NO.= %s \t \t NAME = \n", id, name);
    printf ("SALARY = %.2f \t \t AGE = %d", salary, age);
}
```

ผลลัพธ์

```
ID NUMBER : 123
NAME : SUDA
SALARY : 4525.56
AGE : 30
ID.NO. = 123 NAME = SUDA
SALARY = 4525.56 AGE = 30
```

#### 4.2.3 ฟังก์ชัน getchar ()

เป็นฟังก์ชันที่ใช้รับข้อมูลเข้ามาทางแป้นพิมพ์ทีละ 1 ตัวอักษร โดยต้องกด enter ทุกครั้งเมื่อสิ้นสุดข้อมูล และข้อมูลที่ป้อนจะปรากฏให้เห็นบนหน้าจอด้วย

รูปแบบ

```
getchar ();
```

ตัวอย่างที่ 32

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

```
main ()
```

```
{
```

```
    char ch;
```

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

หมายความว่า เครื่องจะรับข้อมูลจากแป้นพิมพ์เข้ามา 1 ตัวอักษรเก็บไว้ในตัวแปร ch แล้วกดปุ่ม enter เมื่อป้อนข้อมูลเสร็จ

#### 4.2.4 ฟังก์ชัน getch ()

เป็นฟังก์ชันที่ใช้รับข้อมูลที่เป็นตัวอักษร 1 ตัวอักษร เข้ามาทางแป้นพิมพ์โดยเมื่อป้อนข้อมูลเสร็จไม่ต้องกดปุ่ม enter และตัวอักษรที่ป้อนเข้ามาจะไม่ปรากฏบนจอภาพ

รูปแบบ

```
getch ();
```

ตัวอย่างที่ 33

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

```
main ()
```

```
{
```

```
    char x.;
```

```
    x = getch ();
```

หมายความว่า เครื่องจะรอรับข้อมูลจากแป้นพิมพ์เข้ามา 1 ตัวอักษร เพื่อนำมาเก็บไว้ในตัวแปร x

#### 4.2.5 ฟังก์ชัน gets ()

เป็นฟังก์ชันที่ใช้รับข้อมูลที่เป็นข้อความจากแป้นพิมพ์เข้ามาเก็บไว้ในตัวแปรชุด

รูปแบบ

```
gets ( n );
```

n เป็นตัวแปรชุดที่ต้องการรับค่าที่เป็นข้อความจากแป้นพิมพ์เข้ามาเก็บไว้โดยตัวสุดท้ายจะเก็บ \0 เอาไว้เพื่อแสดงการสิ้นสุดของข้อความที่รับเข้ามาเมื่อผู้ใช้กดปุ่ม enter

ตัวอย่างที่ 34

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

```
main ()
```

```
{
```

```
    char name [10];
```

```
    gets (name);
```

หมายความว่า เครื่องจะจองที่ตัวแปรที่ชื่อ name ไว้ 10 ที่ และรอรับค่าที่เป็นข้อความเข้ามาเก็บไว้ในตัวแปรชุดที่ชื่อ name ได้ยาวไม่เกิน 9 ตัวอักษรเพื่อให้ name [9] เก็บ \0 เอาไว้

#### 4.2.6 ฟังก์ชัน putchar ()

เป็นฟังก์ชันที่ใช้ให้คอมพิวเตอร์แสดงผลบนจอภาพทีละ 1 ตัวอักษร

รูปแบบ

```
putchar ( )
```

ตัวอย่างที่ 35

```
#include <stdio.h>
main ( )
{
    char x;
    x = getch ( );
    printf ("Here is the output \n");
    putchar ( x );
}
```

ลักษณะข้อมูลที่ป้อน

A

ผลลัพธ์

Here is the output

A

#### 1.27. ฟังก์ชัน puts ()

เป็นฟังก์ชันที่ใช้แสดงผลที่เป็นข้อความที่เก็บไว้ในตัวแปรชุดออกมาบนจอภาพ

รูปแบบ

```
puts ( n );
```

n เป็นชื่อตัวแปรชุดที่ต้องการนำค่ามาแสดงผล

ตัวอย่างที่ 36

```
#include <stdio.h>
main ( )
{
    char name [10];
    gets (name);
    puts (name);
}
```

หมายความว่า เครื่องจะนำค่าที่เก็บในตัวแปรชื่อ name มาแสดงผลบนจอภาพ

### แบบฝึกหัด

#### 1. จงหาที่ผิดของโปรแกรมต่อไปนี้

a. char n;

...

printf ("%d", n);

...

b. char alpha;

...

scanf ("%c", alpha);

...

c. main ()

{

int a;

float b;

...

printf ("%d %f", b, a);

d. main ()

{

char str = 'ATOM';

...

printf ("\n %s, str");

...

e. main ()

{

int p, q, r;

char mat [ ] = "test score"

...



```
p = getchar (q);
```

```
...
```

```
printf("answer is %d, %d, %d, %s", p, q, r, mat);
```

2. จงเขียนโปรแกรมกำหนดให้  $a = 33.5$ ;  $b = 42.7$   $c = 3.5$   $d = 12.0$  แล้วคำนวณหาค่า

$$x = \frac{ab}{cd}$$

$$y = \frac{a + b}{c + d}$$

ให้แสดงค่า  $a, b, c, d, x, y$  ทางจอภาพ

3. จงเขียนโปรแกรมคำนวณหาค่ารหัสสินค้า ชื่อสินค้า จำนวนสินค้าที่ขายได้ ราคาสินค้าต่อหน่วยมาทางแป้นพิมพ์แล้วคำนวณหาราคาที่ขายสินค้าได้ทั้งหมดจากการนำเอาจำนวนสินค้าที่ขายคูณด้วยราคาสินค้าต่อหน่วย ให้แสดงผลเป็นรหัสสินค้า ชื่อสินค้า จำนวนสินค้าที่ขายได้ราคาสินค้าต่อหน่วย ราคาสินค้าที่ขายได้ทั้งหมดทางจอภาพ
4. จงเขียนโปรแกรมรับข้อมูลที่เป็นชื่อ นามสกุล และที่อยู่ของลูกค้าเข้ามาแล้วทำการแสดงผลชื่อนามสกุล และที่อยู่ของลูกค้านั้นออกมาทางจอภาพ
5. จงเขียนโปรแกรมรับค่ารหัสสินค้า ราคาสินค้า เข้ามาทางแป้นพิมพ์แล้วคำนวณหาส่วนลดและราคาที่ขายจริง ของสินค้านั้น ถ้าบริษัทคิดส่วนลด 10% ของราคาสินค้า โดยแสดงผลเป็นรหัสสินค้า ราคาสินค้า ส่วนลด ราคาที่ขายจริงทางจอภาพ