ข้อมูลและตัวแปรในภาษาซี

ชนิดของข้อมูลในภาษาซี

ชนิดของข้อมูล	ขนาด (bits)	ขอบเขต	ข้อมูลที่เก็บ
char	8	-128 ถึง 127	ข้อมูลชนิดอักขระ ใช้เนื้อที่ 1 byte
int	16	-32,768 ถึง 32,767	ข้อมูลชนิดจำนวนเต็ม ใช้เนื้อที่ 2 byte
short	8	-128 ถึง 127	ข้อมูลชนิดจำนวนเต็มแบบสั้น ใช้เนื้อที่ 1 byte
long	32	-2,147,483,648 ถึง 2,147,483,649	ข้อมูลชนิดจำนวนเต็มแบบยาว ใช้เนื้อที่ 4 byte
float	32	3.4*10e(-38) ถึง 3.4*10e(38)	ข้อมูลชนิดเลขทศนิยม ใช้เนื้อที่ 4 byte
double	64	3.4*10e(-308) ถึง 3.4*10e(308)	ข้อมูลชนิดเลขทศนิยม ใช้เนื้อที่ 8 byte
long double	128	3.4*10e(-4032) ถึง 1.1*10e(4032)	ข้อมูลชนิดเลขทศนิยม ใช้เนื้อที่ 16 byte

ข้อมูลชนิดตัวเลข (จำนวนเต็ม)

ชนิด	ขนาด (bits)	ขอบเขต	ข้อมูลที่เก็บ
int	16	-32,768 ถึง 32,767	ข้อมูลชนิดจำนวนเต็ม
unsigned int	16	0 ถึง 65,535	ข้อมูลชนิดจำนวนเต็ม ไม่คิดเครื่องหมาย
short	8	-128 ถึง 127	ข้อมูลชนิดจำนวนเต็มแบบสั้น
unsigned short	8	0 ถึง 255	ข้อมูลชนิดจำนวนเต็มแบบสั้น ไม่คิดเครื่องหมาย
long	32	-2,147,483,648 ถึง 2,147,483,649	ข้อมูลชนิดจำนวนเต็มแบบยาว
unsigned long	32	0 ถึง 4,294,967,296	ข้อมูลชนิดจำนวนเต็มแบบยาว ไม่คิดเครื่องหมาย

ข้อมูลชนิดตัวเลข (ทศนิยม)

ชนิด	ขนาด (bits)	ขอบเขต	ข้อมูลที่เก็บ
float	32	-3.4*10e(-38) ถึง 3.4*10e(38)	ข้อมูลชนิดเลขทศนิยม
double	64	3.4*10e(-308) ถึง 3.4*10e(308)	ข้อมูลชนิดเลขทศนิยม
long double	128	3.4*10e(-4032) ถึง 1.1*10e(4032)	ข้อมูลชนิดเลขทศนิยม

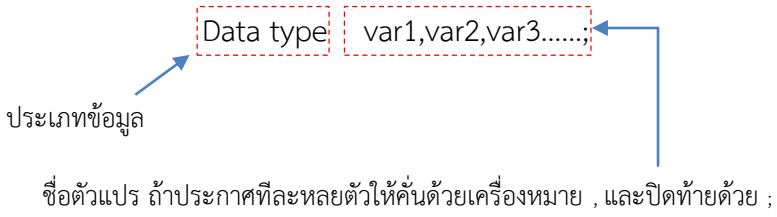
ข้อมูลชนิดอักขระหรือตัวอักษร

ชนิด	ขนาด (bits)	ขอบเขต	ข้อมูลที่เก็บ
char	8	-128 ถึง 127	ข้อมูลชนิดอักขระ
unsigned char	8	0 ถึง 255	ข้อมูลชนิดอักขระ ไม่คิดเครื่องหมาย

การประกาศตัวแปร (variable declaration)

การประกาศตัวแปร คือการจองเนื้อที่ในหน่วยความจำสำหรับเก็บค่าบางอย่างพร้อมทั้งกำหนดชื่อเรียกแทนหน่วยความจำในตำแหน่งนั้นเพื่อให้ผู้เขียนโปรแกรมมีความสะดวกในการเข้าถึงค่าที่เก็บอยู่ในหน่วยความจำดังกล่าว

รูปแบบการประกาศตัวแปร



ตัวอย่าง การประกาศตัวแปร a เป็นประเภท integer

```
int a ;
```

การกำหนดค่าคงที่ให้กับตัวแปร

สามารถทำได้ตั้งแต่เริ่มการประกาศตัวแปร หรือจะกำหนดค่าในภายหลังก็ได้ โดยการใช้เครื่องหมาย “=” แต่ในกรณีของตัวแปรประเภท char ค่าที่กำหนดให้ต้องมีเครื่องหมาย ‘’ ด้วย เช่น ‘a’ ‘b’

รูปแบบการกำหนดค่าตัวแปร

1.กำหนดค่าตั้งแต่เริ่มการประกาศตัวแปร

```
int a =10;      ||      char a = ‘m’;
```

2.กำหนดค่าภายหลังการประกาศตัวแปร

```
int y ;         ||      char a;
y =10;          ||      a = ‘m’
```

การแสดงผลข้อมูล printf()

คำสั่ง printf( ) เป็นคำสั่งที่ใช้แสดงผลข้อมูลหรือตัวแปรที่เราต้องการ เป็นคำสั่งที่ใช้ได้หลายรูปแบบ โดยในวงเล็บจะเป็นการใส่ข้อมูลที่เราต้องการโดยผู้ใช้จะต้องใส่ข้อมูลไว้ในเครื่องหมาย “ ” ส่วนการใช้คำสั่ง printf( ) ในการแสดงผลตัวแปรนั้น เราจะใช้ควบคู่ไปกับ Conversion Specifiers

การรับข้อมูลทางคีย์บอร์ด scanf()

คำสั่ง scanf( ) เป็นฟังก์ชั่นรับข้อมูลทางคีย์บอร์ด ทั้งตัวเลข อักขระ และข้อความ โดยจะนำข้อมูลที่ได้ไปเก็บไว้ในพื้นที่ๆ ผู้ใช้กำหนด ซึ่งผู้ใช้จะต้องควบคุมให้ชนิดข้อมูลที่นำเข้าและชนิดข้อมูลของตัวแปรจะใช้นั้นให้สัมพันธ์กัน ไม่เช่นนั้นโปรแกรมอาจจะดำเนินการผิดพลาดได้

การใช้ Conversion Specifiers ควบคุมรูปแบบการแสดงผล

ตัวควบคุมรูปแบบการแสดงผล (The printf Conversion Specifers)

Conversion Specifiers	ข้อมูลที่ใช้
%c	อักขระ
%s	ข้อความ(สตริงหรือสายอักขระ)
%d	เลขจำนวนเต็ม
%f	เลขทศนิยม
%g หรือ %G	เลขในทศนยิมแบบอัตโนมัติ
%u	เลขจำนวนเต็มแบบไม่ติดเครื่องหมาย
%o	เลขฐาน 8
%x หรือ %X	เลขฐาน 16
%%	แสดงอักขระ %

การใช้อักขระหลีก (Escape Characters)

คือ อักขระที่ทำหน้าที่ควบคุมตำแหน่ง Cursor หรือควบคุมการแสดงผลของอักขระพิเศษที่ไม่สามารถแสดงผลแสดงผลตามปกติ อักขระหลีกจะประกอบด้วย \ กับอักขระที่ต้องการ เช่น \ n ซึ่งอักขระหลีกแต่ละตัวจะถือว่าเป็นอักษรเพียง 1 ตัว

ตารางอักขระหลีก (Escape Characters)

อักขระหลีก	ความหมาย
\n	New line
\t	Tab
\v	Vertical Tab
\b	Back Space
\r	Return
\f	Form Feed
\\	Back Slash
\”	Quotation marks
\’	Apostrophe
\0	null

ตัวอย่างการใช้อักขระหลีก

```
printf(“Hello!\nHow are you?\n”);
printf(“My name is Han.\t\And You ?\n”);
```

จากโค้ดตัวอย่างข้างต้นจะได้ผลลัพธ์คือ

```
Hello!
How are you?
My name is Han.        And you?
```

การรับข้อมูลทางคีย์บอร์ด

คำสั่ง scanf( ) เป็นฟังก์ชั่นรับข้อมูลทางคีย์บอร์ดแบบอรรถประโยชน์ คือสามารถรับ ข้อมูลทางคีย์บอร์ด ให้กับตัวแปรทุกชนิด โดยการนำข้อมูลที่รับมาทางคีย์บอร์ดไปเก็บไว้ในหน่วยความจำที่กำหนดให้ผ่านตำแหน่งที่อยู่ของหน่วยความจำนั้น ซึ่งโดยปกติหน่วยความจำนี้ก็ คือตัวแปรในโปรแกรมนั่นเอง การใช้ scanf() จะต้องคำนึงถึงชนิดของข้อมูลของตัวแปรที่ใช้ โดย scanf() จะนำข้อมูลที่ได้ทางคีย์บอร์ดมาแปลงให้เป็นชนิดที่เหมาะสมกับตัวแปรที่กำหนดตามรหัสแปลง(conversion specifiers) ที่ผู้เขียนโปรแกรมระบุไว้ ถ้ารหัสควบคุมไม่สอดคล้องกับตัวแปร หรือ ข้อมูลนำเข้าไม่สอดคล้องกับตัวแปรจะทำให้เกิดการผิดพลาดของการประมวลผลได้

รูปแบบของ scanf ( ) มีดังนี้

scanf (“ข้อความกำหนดรูปแบบ”, ชุดตำแหน่งที่อยู่ของตัวแปรที่ใช้เก็บข้อมูล);

เช่น

scanf(“%d”, &x) ;

จากรูปแบบจะเห็นว่าพารามิเตอร์ที่ส่งให้กับ scanf() มี 2 ส่วน คือ

1. ส่วนข้อความกำหนดรูปแบบข้อมูล (Format String หรือ control string) ซึ่งเป็น พารามิเตอร์ชุดแรกจะอยู่ในเครื่องหมายคำพูด “ ” ประกอบด้วยชุดของเครื่องหมายควบคุม (conversion specifier) (เครื่องหมาย% ตามด้วยอักขระกำหนดชนิดข้อมูล) เช่น %d, %f เป็นต้น และอาจประกอบด้วย modifies บางตัวเช่น h, l เป็นต้น

2. ส่วนของตำแหน่งที่อยู่ (Address) ของหน่วยความจำที่ใช้เก็บข้อมูล โดยปกติเมื่อ กำหนดตัวแปรที่ใช้เก็บข้อมูล จะใช้ตัวดำเนินการอ่านตำแหน่งที่อยู่ (&: Address-of Operator) นำหน้าตัวแปรที่ต้องการใช้เพื่อส่งตำแหน่งที่อยู่ไปให้ scanf() แล้วจึงสามารถนำข้อมูลไปเก็บใน ตัวแปรที่ถูกชี้ตำแหน่งเหล่านี้
เช่น &x , &y

การแก้ไขปัญหาที่ scanf( ) ไม่ยอมรับค่า

ปัญหาที่ scanf( ) ไม่ยอมรับค่า เกิดจากการอ่านค่าของ scanf( ) ในรูปแบบฟอแมทของพวกตัวเลข จะอ่านเฉพาะตัว string ที่เกี่ยวกับตัวเลขเท่านั้น (0 1 2 3 4 5 6 7 8 9 . + -) อาจจะมีตัวอื่นอีกนะครับ แต่ในส่วนของตัว '\n' นั้นถือว่าไม่ใช่ string เกี่ยวกับตัวเลข scanf( ) ในฟอแมทของพวกตัวเลข จึงไม่รับ '\n' และค้างอยู่ใน Keyboard Buffer ทำให้ถูกส่งไปยัง scanf( ) ตัวต่อไป

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

main()
{
      char b,c;
      printf("first : ");
      scanf("%c",&b);
      printf("\n");

      printf("second : ");
      scanf("%c",&c);
      printf("\n");

      printf("%c%c",b,c);
      getch();

}
```

ตัวอย่างโปรแกรมนี้ scanf( ) จะข้ามการรับค่าข้อมูลครั้งที่ 2 ไป เนื่องจากยังมีอักขระ \n ค้างอยู่ใน Keyboard Buffer ทำให้ถูกส่งไปยัง scanf( ) ตัวต่อไป

เราสามารถแก้ไขได้โดยการใช้ฟังก์ชัน getchar() ไว้หลัง scanf( ) เพื่อให้ \n ไม่ถูกส่งให้ scanf( ) ตัวต่อไป แต่จะส่งให้ getchar( ) เอง

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

main()
{
      char b,c;
      printf("first : ");
      scanf("%c",&b);
      getchar();
      printf("\n");

      printf("second : ");
      scanf("%c",&c);
      getchar();
      printf("\n");

      printf("%c%c",b,c);
      getch();

}
```

เมื่อนำ getchar( ) มาแทรกหลังจาก scanf( ) แล้ว อักขระ \n จะถูกส่งไปยัง getchar( ) แทน ซึ่งจะทำให้ scanf( ) ตัวต่อไปทำงานได้ปกติ