ใบความรู้เรื่อง ข้อมูลและตัวดำเนินการ

การตั้งชื่อตัวแปร และ ฟังก์ชัน

ตัวแปร (Variable) หมายถึง ชื่อที่ผู้เขียนโปรแกรมตั้งขึ้นเพื่อใช้เป็นข้อมูลหรือใช้เก็บข้อมูล การกำหนดตัวแปร จะต้องให้สอดคล้องกับชนิดของข้อมูล(data type) เสมอ เพื่อให้ระบบจองเนื้อที่ในหน่วยความจำให้สอดคล้องกับตัวแปรชนิดนั้น ๆ ซึ่งการตั้งตัวแปรขึ้นมาใช้งานก็คือ การประกาศชนิดของตัวแปร นั้นเอง

หลักการในการตั้งชื่อตัวแปรและฟังก์ชันมีดังต่อไปนี้

1. สามารถใช้ตัวอักษร A ถึง Z รวมทั้ง ตังเลข 0 ถึง 9 และ ขีดล่าง (_) ด้วย โดยมีข้อแม้ว่าต้องไม่ใช้ตัวเลขนำ หน้าชื่อตัวแปร เช่น 2noy เป็นตัวแปรที่ไม่ถูกต้อง แต่ noy2 ใช้ได้

2. ชื่อตัวแปรมีความยาวได้ตั้งแต่ 1ตัวอักษร ถึง 32 ตัวอักษร แต่ไม่ตายตัวเนื่องจากความยาวของชื่อตัวแปรจะขึ้นอยู่กับคอมไพเลอร์และระบบเครื่องคอมพิวเตอร์

3. ชื่อตัวแปรแม้เป็นคำเดียวกันแต่หากเขียนเป็นตัวเล็กปนตัวใหญ่หรือตัวใหญ่ปนตัวเล็กที่สลับตำแหน่งกันระบบจะถือว่าเป็นคนละตัวกัน

4. ชื่อตัวแปรและฟังก์ชัน ไม่อาจใช้คำสงวน (Reserved Words) ต่อไปนี้

auto	extern	sizeof
break	float	static
case	for	struct
char	goto	switch
const	if	typedef
continue	int	union
default	long	unsigned
do	register	void
double	return	volatile
else	short	while
enum	signed	

ชนิดของข้อมูล (Data Types)

ตารางแสดงชนิดของข้อมูลและขนาดที่ใช้เก็บ

ชนิดของข้อมูล	ความหมาย	ช่วงข้อมูลที่เก็บได้ (Data Range)	ขนาดหน่วยความจำ (Size)
char	อักษรหรืออักขระ	-128 ถึง 127	1 bytes
Int	จำนวนเต็ม	-32,768 ถึง 32,767	2 bytes
float	จำนวนจริง(เลขทศนิยม)	3.4E-38 ถึง 3.4E+38	4 bytes
double	ใช้กำหนดเพื่อขยายขนาดของตัวแปร ซึ่งจะทำให้ขนาดของเนื้อที่ของตัวแปรนั้น ๆ เพิ่มเป็น 2 เท่าจากเดิม		

นอกจากนี้ยังมีชนิดของข้อมูลที่เป็นแบบ short ซึ่งเป็นแบบ 16 บิต และแบบ long ซึ่งเป็นแบบ 32 บิต โดยใช้ระบุขนาดให้กับตัวแปรเลขจำนวนเต็ม(int) โดย long จะใช้ขยายเนื้อที่เก็บข้อมูลใหญ่ขึ้นเป็น 2 เท่า จึงเหมาะกับงานที่ใช้เก็บตัวเลขขนาดใหญ่ นอกจากนี้ยังมีคำ unsigned เป็นคำที่ใช้ระบุให้กับตัวแปรเลขจำนวนเต็ม(int) เก็บได้เฉพาะข้อมูลที่มีค่ามากกว่า หรือเท่ากับ 0(ไร้เครื่องหมายลบ)เท่านั้น ใช้ขยาย char หรือข้อมูลประเภทจำนวนเต็ม

ตัวอย่าง

ประเภท (Type)	ช่วงข้อมูลที่เก็บได้ (Data Range)	ขนาดหน่วยความจำ (Size)
Int	-32,768 to 32,767	2Bytes
Unsingned int	0 to 65,535	4 Byte
long int	-2,147,483,648 to 2,147,483,647	4 bytes
Float	3.4E-38 to 3.4E+38	4byte
Double	1.7E-308 to 1.7E+308	8 byte
Long double	3.4E-4932to +1.1E4932	10 byte

การประกาศตัวแปร

ตัวแปรในภาษาซี จะต้องมีการประกาศ ชนิดของตัวแปรให้สอดคล้องกับข้อมูลที่เก็บอยู่ในตัวแปรนั้นและจะต้องมีการประกาศชนิดของตัวแปรเหล่านี้ไว้ก่อนที่จะใช้ตัวแปรนั้น ๆ ซึ่งในภาษาซี สามารถทำการประกาศชนิดของตัวแปร ได้โดยใช้ฟังก์ชัน type

รูปแบบ type variable-list;

variable-list ชื่อของตัวแปรที่ต้องการประกาศชนิดถ้ามีมากกว่า 1 ตัว จะแยกกันด้วยเครื่องหมายคอมม่า (,)

type ชนิดของตัวแปร ซึ่งจะสามารถประกาศชนิดได้ดังนี้

int เป็นตัวแปรที่เก็บค่าเป็นเลขจำนวนเต็ม Integer

float เป็นตัวแปรที่เก็บค่าที่เป็นเลขทศนิยม

short เป็นตัวแปรที่เก็บค่าที่เป็นเลขจำนวนเต็มที่มีค่าน้อยกว่าค่าของตัวแปร ที่ประกาศไว้เป็น ชนิด int

long เป็นตัวแปรที่เก็บค่าที่เป็นเลขจำนวนเต็มที่มีจำนวนบิทมากเป็น 2 เท่าของเดิม

double เป็นตัวแปรที่เก็บค่าที่เป็นทศนิยมที่มีจำนวนบิทมากเป็น 2 เท่าของเดิม

unsigned เป็นตัวแปรที่เก็บค่าจำนวนเต็มที่มีค่าเป็นบวกเท่านั้นเอาไว้

char เป็นตัวแปรที่เก็บค่าเป็นตัวอักษร

ตัวอย่าง int data1,data2;

เป็นการประกาศตัวแปรชื่อ data1และ data2 ให้เก็บข้อมูลที่เป็นจำนวนเต็ม

แต่ถ้าประกาศตัวแปรชื่อ data1 เก็บเลขจำนวนเต็มและ ตัวแปรชื่อ data2 เก็บเลขจำนวนจริงสามารถทำได้ดังนี้ Int data1; float data2;

ตัวแปรชุด (Array Variable)ในการนำค่าไปเก็บในหน่วยความจำหลาย ๆ ค่า จะต้องตั้งชื่อตัวแปรหลายตัวซึ่งทำให้มีปัญหากับการตั้งชื่อดังนั้นการตั้งชื่อตัวแปรสามารถทำได้อีกแบบหนึ่ง คือตั้งชื่อตัวแปรเพียงชื่อเดียวแล้วใช้ตัวเลขกำกับว่าเป็นตัวแปรตัวที่เท่าไร ในชุดนั้นลักษณะการตั้งชื่อตัวแปรแบบนี้เรียกว่าตัวแปรชุดและตัวเลขที่บอกตำแหน่งนั้นเรียกว่า ดรรชนี ที่นิยมใช้ได้แก่

ตัวแปรชุดชนิด 1 มิติ (One Dimension) หมายถึงตัวแปรชุดที่มีตัวเลขแสดงตำแหน่งเพียงตัวเดียวซึ่งสามารถประกาศชนิดของตัวแปรชุดได้ดังนี้

รูปแบบ type array - name[n];

array-name ชื่อตัวแปรที่ต้องการประกาศชนิดว่าเป็นตัวแปรชุด

n ตัวเลขจำนวนเต็มที่ใช้แสดงขนาดของตัวแปรชุดนั้น โดยเริ่ม จากศูนย์เสมอถ้าไม่มีการบอกขนาด เครื่องจะเตรียมที่ใน หน่วยความจำไว้เท่ากับจำนวนของขนาดข้อมูลจริง

ตัวอย่าง char name[80];

หมายความว่า name เป็นตัวแปรชุดที่มีชื่อตัวแปรตั้งแต่ name[0],name[1],name[2],.....name[79] โดยทุกชื่อตัวแปรจะเก็บข้อมูลเป็นตัวอักษรความยาว 1 ตัวอักษร

ตัวดำเนินการ(Operater)

ในภาษา C มีการใช้เครื่องหมายดำเนินการ ซึ่งแบ่งออกได้เป็น 3 ชนิดใหญ่ ๆ คือ

1. **เครื่องหมายคณิตศาสตร์** (Arithmetic Operators)

ได้แก่ เครื่องหมายที่ใช้ในการคำนวณบวก ลบ คูณ หาร ค่าต่าง ๆ เครื่องหมายที่ใช้มีดังนี้

เครื่องหมาย	ความหมาย	ตัวอย่าง
+	การบวก	A + B
-	การลบ	A - B
*	การคูณ	A * B
/	การหาร	A / B
%	หารเอาแต่เศษไว้ (Modulus) 5 % 3 = 1 เศษ 2 จะเก็บแต่เศษ 2 เอาไว้	
- -	การลดค่าครั้งละ 1 เสมอ A - - จะเหมือนกับ A = A - 1	
+ +	การเพิ่มค่าขึ้นครั้งละ 1 เสมอ A + + จะเหมือนกับ A = A + 1	

2. **เครื่องหมายเปรียบเทียบ** (Relational and Logical Operaters)

หมายถึง เครื่องหมายที่ใช้ในการเปรียบเทียบและตัดสินใจ ซึ่งผลของการเปรียบเทียบจะได้เป็น 2 กรณี คือ จริง (true) จะให้ค่าเป็น 1 เท็จ (false) จะให้ค่าเป็น 0

เครื่องหมายที่ใช้มีดังนี้

เครื่องหมาย	ความหมาย	ตัวอย่าง
>	มากกว่า	A > B (A มากกว่า B)
>=	มากกว่าหรือเท่ากับ	A >= B (A มากกว่าหรือเท่ากับ B)
<	น้อยกว่า	A < B (A น้อยกว่า B)
<=	น้อยกว่าหรือเท่ากับ	A <= B (A น้อยกว่าหรือเท่ากับ B)
==	เท่ากับ	A == B (A เท่ากับ B)
!=	ไม่เท่ากับ	A != B (A ไม่เท่ากับ B)

3. **เครื่องหมายตรรก** (Logical Operaters)

หมายถึง เครื่องหมายที่ใช้ในการเปรียบเทียบและตัดสินใจโดยเอาเงื่อนไขตั้งแต่ 2 เงื่อนไข มาเปรียบเทียบ กัน ผลที่ได้จากการเปรียบเทียบจะได้ผลเป็น 2 กรณี เช่นเดียวกับเครื่องหมายเปรียบเทียบ เครื่องหมายตรรกที่ใช้มี ดังนี้

&& (AND) หมายถึง การนำเงื่อนไข 2 เงื่อนไขมาเปรียบเทียบกัน แล้วจะได้ผลของการเปรียบเทียบดังตาราง

P	**Q**	**P && Q**
0	0	0
0	1	0
1	0	0
1	1	1

|| (OR) หมายถึง การนำเงื่อนไข 2 เงื่อนไขมาเปรียบเทียบกันแล้วจะได้ผลของการเปรียบเทียบดังตาราง

P	**Q**	**P || Q**
0	0	0
0	1	1
1	0	1
1	1	1

! (NOT) หมายถึงการนำเงื่อนไขมาเปรียบเทียบแล้วจะได้ผลของการเปรียบเทียบ ดังตาราง

P	**!P**
1	0
0	1

นิพจน์ (Expression)

หมายถึง การนำตัวแปร ค่าคงที่ มาสัมพันธ์กันโดยใช้เครื่องหมายอย่างหนึ่งอย่างใดเป็นตัวเชื่อม แบ่งออกเป็น

นิพจน์คณิตศาสตร์ (Arithmetic Expression)

หมายถึง การนำตัวแปร ค่าคงที่ มาสัมพันธ์กันโดยใช้เครื่องหมายคณิตศาสตร์เป็นตัวเชื่อม ผลที่ได้จากนิพจน์แบบนี้จะเป็นตัวเลข เช่น

3x + 5y	=	3 * x + 5 * y
x2 - y2	=	x * x - y * y

นิพจน์ตรรก (Logical Expression)

หมายถึง การนำตัวแปร ค่าคงที่ หรือนิพจน์ มาสัมพันธ์กัน โดยใช้เครื่องหมายเปรียบเทียบและเครื่องหมายตรรกเป็นตัวเชื่อมผลลัพธ์ที่ได้จะเป็นจริง หรือเท็จ คือจะให้ค่าเป็น 1 หรือ 0 ออกมาเป็นผลลัพธ์ ซึ่งสามารถนำไปคำนวณต่อได้ เช่น

ถ้า i มีค่าเป็น 3 j มีค่าเป็น 5 a มีค่าเป็น 3 ถ้าเขียนนิพจน์ดังนี้

i = = j ผลลัพธ์ ไม่จริง (0)

i = = a ผลลัพธ์ จริง (1)

i > j * 5 ผลลัพธ์ ไม่จริง (0)

i + 3 > j - 2 && a * 2 > 10 ผลลัพธ์ ไม่จริง (0)

กฎเกณฑ์ในการเขียนนิพจน์

1. ห้ามเขียนตัวแปร 2 ตัวติดกันโดยไม่มีเครื่องหมาย

เช่น ab ในภาษา C ต้องเขียน a * b จะเขียนเป็น ab ไม่ได้ เพราะจะถือเป็น ชื่อตัวแปรตัวเดียวชื่อ ab ไม่ ใช่ค่า a คูณ b

2. ถ้าเขียนนิพจน์โดยมีค่าของตัวแปรหรือค่าคงที่ต่างชนิดกันในนิพจน์เดียวกัน ภาษา C จะเปลี่ยนข้อมูลชนิดที่มีขนาดเล็กให้เป็นชนิดของข้อมูลที่ใหญ่ขึ้น เช่น

char short กับ int จะเปลี่ยนเป็น int

float กับ double จะเปลี่ยนเป็น double

int กับ long จะเปลี่ยนเป็น long

int กับ double จะเปลี่ยนเป็น double

int กับ unsigned จะเปลี่ยนเป็น unsigned

long กับ double จะเปลี่ยนเป็น double

อะไร กับ long จะเปลี่ยนเป็น long

อะไร กับ long double จะเปลี่ยนเป็น long double

ขั้นตอนการทำงานของนิพจน์

นิพจน์ในภาษา C จะทำงานจากซ้ายไปขวาตามลำดับการทำงานของเครื่องหมายต่อไปนี้ โดยเรียงจากการทำงานอันดับที่ 1 ลงไปถึงอันดับสุดท้าย คือ

ถ้ามีวงเล็บจะทำในวงเล็บก่อนและทำจากซ้ายไปขวาตามลำดับเครื่องหมาย

!, ++, - - (ทำจากขวาไปซ้าย)

*, /, %

+, -

<, <=, >, >=

= =, ! =

&&

||

การกำหนดค่า

เป็นรูปฟังก์ชันจริง ๆ ของภาษา C สำหรับให้คอมพิวเตอร์ทำการคำนวณอย่างหนึ่งอย่างใด หรืออาจใช้ในการกำหนดค่าเริ่มต้นให้กับตัวแปร หรือย้ายค่าตัวแปรจากที่หนึ่งไปยังอีกที่หนึ่งก็ได้

รูปแบบ <Variable> = <expression>;

<Variable> หมายถึง ตัวแปรที่ทำหน้าที่ในการเก็บค่า

= หมายถึง เครื่องหมายที่แสดงว่าให้นำค่าที่อยู่ทางขวา ไปเก็บไว้ในตัวแปรทางซ้าย

<expression> หมายถึง ตัวแปร, ค่าคงที่หรือนิพจน์

ในกรณีที่ เป็นค่าคงที่จะ หมายถึง การกำหนดค่าให้กับตัวแปร

ภาษา C ยอมให้มีการกำหนดค่า หรือ เปลี่ยนแปลง type หนึ่งไปเป็นอีก type หนึ่งได้เมื่อจำเป็น

ตัวอย่างที่ 1

#include <stdio.h>	ขอใช้ไฟล์ stdio.h จากไฟล์ไลบรารี #include เป็นแหล่งรวมสารสนเทศ
main()	ฟังก์ชัน main()เป็นโปรแกรมหลัก ไม่มีพารามิเตอร์จึงไม่มีการรับส่งข้อมูล
{	เริ่มต้นบล็อกของฟังก์ชัน main()
printf(“Hello TU\n”);	ฟังก์ชัน printf()จาก stdio.h ทำการพิมพ์ข้อความที่อยู่ในเครื่องหมาย “ ”ออกจอภาพ \n ทำให้ฟังก์ชัน printf ข้ามไปเริ่มต้นที่บรรทัดใหม่
return(0);	การตรวจสอบฟังก์ชัน main()ซึ่งจะต้องให้ค่าฟังก์ชันเป็นเลขจำนวนเต็ม ปกติโปรแกรมจะส่งกลับค่า 0 แต่ถ้าส่งกลับค่าเป็น 1 ถือว่าโปรแกรมทำงานผิดปกติ
}	สิ้นสุดบล็อกของฟังก์ชัน main()

ตัวอย่างที่ 2

#include <stdio.h>	ขอใช้ไฟล์ stdio.h จากไฟล์ไลบรารี #include เป็นแหล่งรวมสารสนเทศ
void main()	ไม่มีการรับข้อมูลและส่งข้อมูลไปยังฟังก์ชันใด ๆ .
{	เริ่มต้นบล็อกของฟังก์ชัน main()
printf(“Hello TU\n”);	ฟังก์ชัน printf()จาก stdio.h ทำการพิมพ์ข้อความที่อยู่ในเครื่องหมาย “ ”ออกจอภาพ \n ทำให้ฟังก์ชัน printf ข้ามไปเริ่มต้นที่บรรทัดใหม่
return(0);	ไม่จำเป็นต้องใช้คำสั่ง return(0) เมื่อมี void ข้าง main()
}	สิ้นสุดบล็อกของฟังก์ชัน main()

ตัวอย่างที่ 3

#include <stdio.h>	ขอใช้ไฟล์ stdio.h จากไฟล์ไลบรารี #include เป็นแหล่งรวมสารสนเทศ
main()	ฟังก์ชัน main()เป็นโปรแกรมหลัก ไม่มีพารามิเตอร์จึงไม่มีการรับส่งข้อมูล
{	เริ่มต้นบล็อกของฟังก์ชัน main()
Int x_value = 19;	กำหนดค่า 19 ให้ตัวแปร x_value ซึ่งเป็นตัวแปรชนิดจำนวนเต็ม
printf(“x = %d\n ”,x_value);	ฟังก์ชัน printf()จาก stdio.h ทำการพิมพ์ข้อความ x=19 ออกจอภาพ แล้วขึ้นบรรทัดใหม่
return(0);	ส่งค่า 0 กลับจริง ถือว่าโปรแกรมทำงานถูกต้อง
}	สิ้นสุดบล็อกของฟังก์ชัน main()

ตัวอย่างที่ 4

#include <stdio.h>	ขอใช้ไฟล์ stdio.h จากไฟล์ไลบรารี #include เป็นแหล่งรวมสารสนเทศ
main()	ฟังก์ชัน main()เป็นโปรแกรมหลัก ไม่มีพารามิเตอร์

{	เริ่มต้นบล็อกของฟังก์ชัน main()
Int x,y,sum;	กำหนดให้ตัวแปร x,y,sum เป็นตัวแปรชนิดจำนวนเต็ม
x=20;	กำหนดค่า 20 ให้ตัวแปร x
y=70;	กำหนดค่า 70 ให้ตัวแปร y
Sum=x+y	ผลรวมของค่าที่เก็บในตัวแปร x และ y นำไปไว้ในตัวแปร sum
printf("sum = %d\n ",sum);	ฟังก์ชัน printf()จาก stdio.h ทำการพิมพ์ข้อความ sum=90 ออกจอภาพ แล้วขึ้นบรรทัดใหม่
return(0);	ส่งค่า 0 กลับจริง ถือว่าโปรแกรมทำงานถูกต้อง
}	สิ้นสุดบล็อกของฟังก์ชัน main()

ตัวอย่างที่ 5

#include <stdio.h>	ขอใช้ไฟล์ stdio.h จากไฟล์ไลบรารี #include เป็นแหล่งรวมสารสนเทศ
main()	ฟังก์ชัน main()เป็นโปรแกรมหลัก ไม่มีพารามิเตอร์ จึงไม่มีการรับส่งข้อมูล
{	เริ่มต้นบล็อกของฟังก์ชัน main()
Int x,y;	กำหนดให้ตัวแปร x,y เป็นตัวแปรชนิดจำนวนเต็ม
x=10;	กำหนดค่า 10 ให้ตัวแปร x
y=20;	กำหนดค่า 20 ให้ตัวแปร y
--x	จะมีค่าเหมือน x = x-1 ดังนั้น x จะเก็บค่า 9
++y	จะมีค่าเหมือน y = y+1 ดังนั้น y จะเก็บค่า 21
printf("x = %d\n ",x);	ฟังก์ชัน printf()จาก stdio.h ทำการพิมพ์ข้อความ x=9 ออกจอภาพ แล้วขึ้นบรรทัดใหม่
printf("y = %d\n ",x);	ฟังก์ชัน printf()จาก stdio.h ทำการพิมพ์ข้อความ y=21ออกจอภาพ แล้วขึ้นบรรทัดใหม่
return(0);	ส่งค่า 0 กลับจริง ถือว่าโปรแกรมทำงานถูกต้อง
}	สิ้นสุดบล็อกของฟังก์ชัน main()

ตัวอย่างที่ 6

#include <stdio.h>	ขอใช้ไฟล์ stdio.h จากไฟล์ไลบรารี #include เป็นแหล่งรวมสารสนเทศ
main()	ฟังก์ชัน main()เป็นโปรแกรมหลัก ไม่มีพารามิเตอร์จึงไม่มีการรับส่งข้อมูล
{	เริ่มต้นบล็อกของฟังก์ชัน main()
char any_char;	any_char เป็นตัวแปรชนิดอักขระ
printf("Please type a character : ");	ฟังก์ชัน printf()จาก stdio.h ทำการพิมพ์ข้อความ Please type a character : ออกจอภาพ
scanf("%c",&any_char);	ฟังก์ชัน scanf()จะทำการอ่านคีย์ใด ๆ ที่ผู้ใช้พิมพ์ที่คีย์บอร์ด %c เป็นรหัสควบคุมรูปแบบการพิมพ์ใช้ & ผ่านแอดเดรสของ any_char ไปยังฟังก์ชัน scanf()
printf("Thank you,your character are %c " any_char",);	พิมพ์ข้อความ Thank you,your character are ที่จอภาพพร้อมทั้งค่าอักขระต่าง ๆ ที่ any_char เก็บไว้
return(0);	ส่งค่า 0 กลับจริง ถือว่าโปรแกรมทำงานถูกต้อง
}	สิ้นสุดบล็อกของฟังก์ชัน main()

ตัวอย่าง

การกำหนดค่า และการแสดงผลลัพธ์ในภาษา C

```
/* What will be printed ? */
#include
void main(void)
{
        int i = 67;
        double d = 123456.7;
        float f = 1.234;
        char c;
        unsigned u;
        long l = 123456L;
                c = i;
                i = l;
```

```
                    l = d;
                    d = u;
                    u = -9;
                    printf("i = %d, d = %f, f = %f\n"  , i, d, f);
                    printf("c = %c, u = %u, l = %ld", c, u, l);
        }
```

ผลลัพธ์

i = -7616m d = 32750.000000, f = 1.234000

c = C, u = 65527, l = 123456

คำอธิบาย

6 บรรทัดแรก เป็นส่วนของการประกาศ พิจารณาประโยคกำหนดค่า สามารถกำหนดค่า data type หนึ่งแก่ data type ต่างชนิดกันได้ เช่น c = i; เป็นการกำหนดค่า int ให้กับchar ซึ่ง c จะเก็บค่า decimal int. ของ i ซึ่งเป็น ASCII code ของ char c การกำหนดค่าหลายค่าภาษา C ยอมให้ผู้เขียนโปรแกรมสามารถกำหนดค่าได้มากกว่า 1 ค่า (Multiple Assignments) ภายใน 1 ประโยคฟังก์ชัน

รูปแบบ var1 = var2 = expression;

ตัวอย่าง

```
n1 = n2 = 10;
a = b = c = 5;
x = y = x = w;
```

คำอธิบาย

ในประโยคที่มีการกำหนดค่าหลายค่านั้น คอมไพเลอร์จะทำการกำหนดค่าให้จากขวามือไปซ้ายมือ เช่น มีการกำหนดค่า 10 ให้แก่ n2 แล้วจึงกำหนดค่า n2 ให้แก่ n1 ซึ่งก็คือ 10 นั่นเอง นอกจากนี้ ภาษา C ยังพิเศษกว่าภาษาอื่น ๆ โดยมี Special Saaignment Operators เพื่อให้สามารถทำงานได้เร็วขึ้น เช่น

n + = 4; มีความหมายเช่นเดียวกับ n = n + 4;

n - = 4; มีความหมายเช่นเดียวกับ n = n - 4;

n * = 4; มีความหมายเช่นเดียวกับ n = n * 4;

รหัสควบคุมการแสดงผล

'\n'	newline	การขึ้นบรรทัดใหม่
'\\'	backslash (\ )	
'\t'	horizontal lab	เว้นระยะ 6 ตัวอักษรในแนวนอน

‘\v’	vertical tab	
‘\"’	quotation mark (")	
‘\’'	apostrophe (')	
'\r'	carriage return	ให้ตัวชี้ตำแหน่ง(cursor)กลับไปอยู่ที่ต้นบรรทัด
'\ddd'	bit pattern in octal digits	
'\f'	form feed	ให้เว้นการแสดงผลไปหนึ่งหน้าจอ
'\0'	ASCII char NULL	
'\b'	backspace	
'\xdd'	bit pattern in hex. degits	
‘\a’	bell	
‘\?’	question mark (?)	

ตัวกำหนดชนิดข้อมูลของภาษาซี

ตัวกำหนดชนิดข้อมูล	ความหมาย
%c	แสดงข้อมูลในรูปตัวอักษร
%d	แสดงข้อมูลในรูปตัวเลขจำนวนเต็ม
%e	แสดงข้อมูลในรูปเอกซ์โพเนนเซียล
%f	แสดงข้อมูลในรูปตัวเลขทศนิยม
%o	แสดงข้อมูลในรูปของตัวเลขฐานแปด
%x	แสดงข้อมูลในรูปของตัวเลขฐานสิบหก
%u	แสดงข้อมูลในรูปของจำนวนเต็มบวก
%s	แสดงข้อมูลในรูปของข้อความ(string)
%p	แสดงข้อมูลในรูปของที่อยู่ของตัวแปร(address)
%%	แสดงเครื่องหมาย %