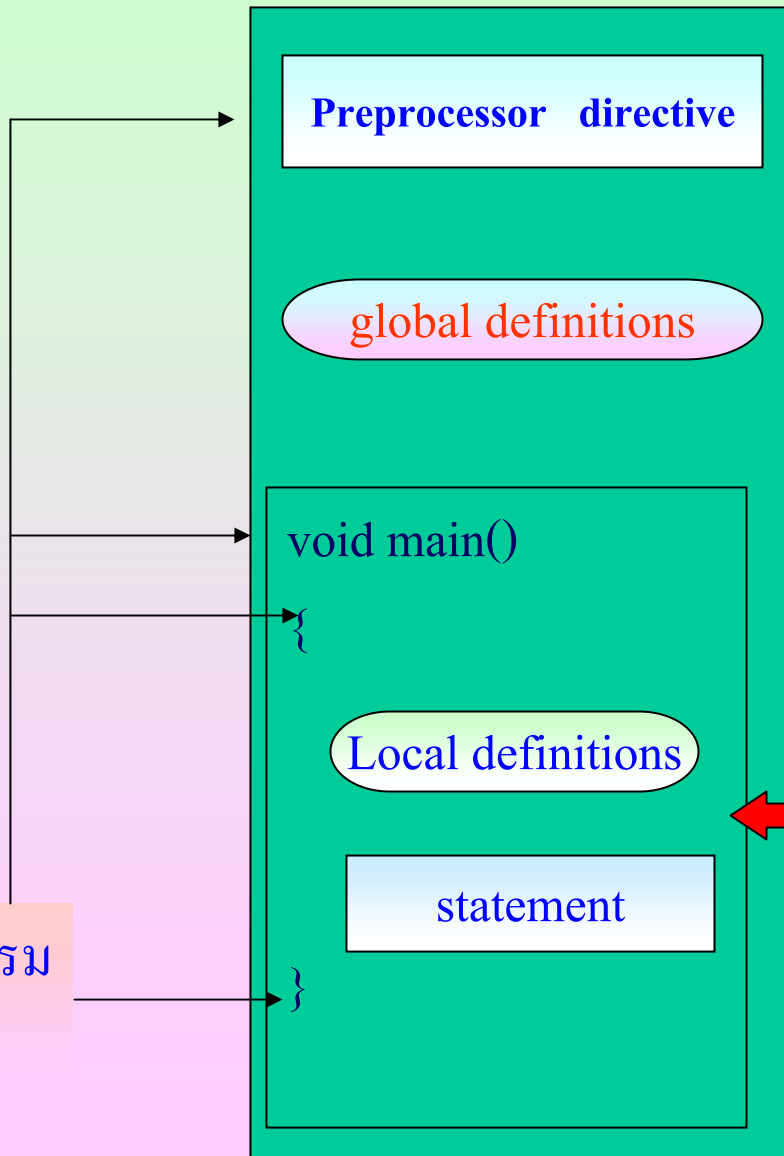


พื้นฐานภาษาซี

1. เริ่มต้นเขียนภาษาซี
2. ประเภทของข้อมูล
3. ตัวดำเนินการ
4. ฟังก์ชันรับและแสดงผลทางจอภาพ

โครงสร้างโปรแกรม



ใช้เรียกไฟล์ที่ต้องการก่อนคอมไพล์
โปรแกรมที่พบได้แก่ #include, #define
เป็นต้น

ใช้ประกาศตัวแปรหรือฟังก์ชัน
ที่สามารถเรียกใช้ได้ทุกส่วน

ตัวแปรและคำสั่งต่าง ๆ ที่ใช้ใน
โปรแกรม

ทุกโปรแกรม
ต้องมี

โปรแกรมแรก

```
#include <stdio.h>

void main()
{
    printf("Easy C \n");
    printf("Computer\n");
}
```

คำสั่ง include ให้นำไฟล์ stdio.h มาร่วมด้วย เพื่อเรียกใช้ฟังก์ชันต่าง ๆ หรืออาจเรียกไฟล์ .h อื่น ๆ อีกก็ได้ โดยไฟล์ที่เรียกจะต้องอยู่ใน “ ” หรือ < >

ฟังก์ชันในภาษาซี จบด้วยเครื่องหมาย ;
ในที่นี้จะให้แสดงข้อความทางจอภาพ และขึ้นบรรทัดใหม่

ผลการรันโปรแกรม

Easy C

Computer

void ที่อยู่หน้า main จะบอกว่าเมื่อทำตั้งแต่ { ถึง } ไม่ต้องคืนค่า หรือไม่ต้องทำอะไรต่อไปอีก

ไฟล์ .h ที่ตามหลัง include

ไฟล์นามสกุล .h เป็น Header file หรือไฟล์ส่วนหัวที่รวบรวมคำสั่งต่าง ๆ ของภาษาซีเอาไว้ เพื่อให้ผู้เขียนสามารถใช้ฟังก์ชันได้ มีอยู่หลายไฟล์เช่น

- **stdio.h** เก็บฟังก์ชันที่ใช้งานทั่วไป เช่น printf , scanf โปรแกรมส่วนมากจะใช้ไฟล์นี้
- **conio.h** เก็บฟังก์ชันควบคุมการแสดงผลต่าง ๆ
- **string.h** เก็บฟังก์ชันที่ใช้ในการประมวลผลกับข้อความ
- **math.h** เก็บฟังก์ชันที่ใช้ในการคำนวณทางคณิตศาสตร์ เช่น sin,cos,log

ตัวอย่างโปรแกรม

```
#include "stdio.h"
```

```
int feet, inches;
```

```
void main()
```

```
{
```

```
    feet = 6;
```

```
    inches = feet * 12;
```

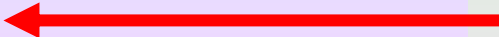
```
    printf("Height in inches is %d",inches);
```

```
}
```

ประกาศตัวแปรสองตัวสำหรับ
เก็บเลขจำนวนเต็ม



กำหนดค่า 6 ให้กับตัวแปร feet ด้วย ;



พิมพ์ค่าออกทางจอเป็นเลขฐานสิบ



นำค่าในตัวแปร feet มาคูณกับ 12 แล้วเก็บในตัวแปร inches



ตัวแปร (variables)

ตัวแปร เป็นชื่อของหน่วยความจำที่ตำแหน่งต่าง ๆ ที่ผู้เขียน โปรแกรม กำหนด มีไว้สำหรับเก็บข้อมูลต่าง ๆ ระหว่างการทำโปรแกรม แบ่งตามประเภทได้ดังนี้

- ตัวแปรเก็บข้อมูลชนิดจำนวนเต็ม
- ตัวแปรเก็บข้อมูลชนิดจำนวนจริง
- ตัวแปรเก็บข้อมูลชนิดตัวอักษร
- ตัวแปรเก็บข้อมูลชนิดสายอักขระ (สตริง)

การประกาศตัวแปร

Data type variables_name

↑ ↑
ประเภทของข้อมูล ชื่อตัวแปร Ex int x;

ประกาศตัวแปรชื่อ x เก็บจำนวนเต็ม

กฎการตั้งชื่อตัวแปร

- ประกอบด้วยตัว a ถึง z , 0 ถึง 9 และ _ เท่านั้น
- ตัวอักษรตัวแรกต้องเป็น a ถึง z และ _
- ห้ามใช้ชื่อเฉพาะ
- ตัวพิมพ์ให้ตัวพิมพ์เล็กระบบจะมองว่ามีความหมายต่างกัน
- ความยาวตัวแปรไม่เกิน 31 ตัวอักษร

ชนิดข้อมูล (data type)

ชนิดข้อมูล	การประกาศ	ขนาด (Byte)	ช่วงของข้อมูล
ตัวอักษร	char	1	-128 ถึง 127
เลขจำนวนเต็ม	int	2	-32,768 ถึง 32,767
long integer	long	4	สองพันล้าน
จำนวนเต็ม ไม่คิดเครื่องหมาย	unsigned int	2	0 ถึง 65,535
เลขทศนิยม	float	4	1.2×10^{-38} - 3.4×10^{38}

ดูเพิ่มเติมในหนังสือ

การประกาศตัวแปร

จบด้วยเครื่องหมายอัฒภาค (;)

ชนิดข้อมูล ชื่อตัวแปร;

ถ้าหากมีตัวแปรหลายตัวให้ใช้เครื่องหมายจุลภาคคั่น

ชนิดข้อมูล ชื่อตัวแปร, ชื่อตัวแปร,....;

ตัวอย่าง

int count;	ประกาศตัวแปรชื่อ count สำหรับเก็บเลขจำนวนเต็ม
int m, n;	ประกาศตัวแปรชื่อ m และ n สำหรับเก็บเลขจำนวนเต็ม
int no = 10	ประกาศตัวแปรชื่อ no สำหรับเก็บจำนวนเต็มและให้เท่ากับ 10
float percent, total;	ประกาศตัวแปรชื่อ percent และ total สำหรับเก็บเลขทศนิยม

การกำหนดค่าให้กับตัวแปรตัวเลข

- การให้ค่ากับข้อมูลการกำหนดค่าพร้อมกับประกาศตัวแปร หรือประกาศแล้วจึงกำหนดก็ได้ ตัวอย่างเช่น
 - `int no = 10;           /* ประกาศตัวแปร no เป็นเลขจำนวนเต็มและให้เท่ากับ 10*/`
 - `int no;                   /* ประกาศตัวแปร no เป็นจำนวนเต็ม */`
`no = 10;                /* ให้ no เท่ากับ 10 */`
 - `int num                 /* ประกาศตัวแปร num เป็นจำนวนเต็ม */`
`num = 0x10;           /* ให้ num เท่ากับ 10 ฐานสิบหก */`

เลขฐานสิบหกให้เริ่มต้นด้วย 0x

ตัวแปรชนิดอักขระ

```
char a, b;
```

← ประกาศตัวแปร a และ b เป็นตัวอักขระ

```
a = 'T';
```

```
b = 'A';
```

← กำหนดค่าตัวอักขระให้กับตัวแปรจะต้องอยู่ในเครื่องหมาย ' '

ตัวอย่าง

```
int a, b;      /* ประกาศตัวแปร a และ b สำหรับเก็บเลขจำนวนเต็ม */
```

```
char c, d;     /* ประกาศตัวแปร c และ d เก็บตัวอักขระ */
```

```
a = 3;
```

```
b = a;         /* เอาค่าในตัวแปร a ใส่ในตัวแปร b จะทำให้ b เท่ากับ 3 */
```

```
c = 'a';       /* ให้ค่าในตัวแปร c มีค่าเป็นตัวอักขระตัว a */
```

```
b = a + 4;     /* ให้ค่าในตัวแปร b เท่ากับ 7 */
```

```
b = 'a' + a;   /* ค่าในตัวแปร b จะเท่ากับรหัส ASCII ของ a บวกกับ 3 */
```

ตัวแปรชนิดข้อความ

ข้อความก็คือตัวอักษรหลาย ๆ ตัวมาต่อเรียงกันเรียกว่าสตริง (string) เมื่อมีการประกาศตัวแปรประเภทนี้ คอมพิวเตอร์จะนำตัวอักษรแต่ละตัวมาเก็บในหน่วยความจำแบบ char เรียงกันไปทีละตัว

ตัวอย่าง

ตัวแปรชื่อ name มีความยาวไม่เกิน 15 ตัวอักษร

`char name[15] = "COMPUTER";` `/* เก็บข้อความลงตัวแปร name */`

`char phone[15] = "0-2737-3000";`

`char age[3] = "20";`

ตัวแปรชื่อ age มีความยาวไม่เกิน 3 ตัวอักษร

การประกาศค่าคงที่

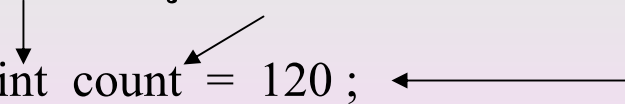
ค่าคงที่เป็นค่าที่คงเดิมตลอดทั้งโปรแกรม แต่ตัวแปรจะมีค่าเปลี่ยนแปลงได้ระหว่างทำโปรแกรม

การประกาศค่าคงที่

1. ใช้คำสั่ง const ตามรูปแบบดังนี้

const ชนิดข้อมูล ชื่อตัวแปร = ค่าที่เก็บในตัวแปร;

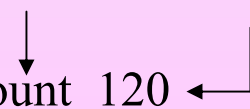
Ex: const int count = 120 ;



2. ใช้ตัวประมวลผลก่อน ตามรูปแบบดังนี้

#define ชื่อค่าคงที่ ค่าคงที่ที่ต้องการเก็บ

Ex: #define count 120 /* กำหนดให้ count เป็นค่าคงที่แบบ int มีค่าเป็น 120 */



นิพจน์ (Expression)

หมายถึงสิ่งที่ประมวลผลแล้วให้ค่าผลลัพธ์ออกมา มีความยากง่ายแตกต่างกันออกไป

นิพจน์ที่เป็นสัญลักษณ์

```
#define VAT 7  
  
#define PI 3.14159  
  
const int a = 35;
```

นิพจน์ตัวแปร

```
int count;  
  
float amount;  
  
char ch;
```

นิพจน์ที่เชื่อมโยงด้วยตัวดำเนินการ

```
3+5;  
  
score1 * 2 + score2 * 5 + score3 * 7;
```

ข้อความกำหนดค่า (Assignment statement)

เป็นการกำหนดค่าให้กับตัวแปร มีรูปแบบดังนี้

ตัวแปร = นิพจน์; ← จบด้วยเครื่องหมาย ;
↑
ใช้เครื่องหมายเท่ากับ

ตัวอย่าง

```
con = 10.5;
```

```
result = 25 * 6;
```

```
point = score1 * 2 + score2 * 5 + score3 * 8;
```

```
a = b = 0;      /*ให้ตัวแปร a เท่ากับตัวแปร b โดยมีค่าเป็นศูนย์ */
```

ตัวอย่าง

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

```
void main()
```

```
{
```

```
    int sum, total;
```

```
    sum = total = 0;
```

```
    printf(" sum = %d \n",sum);
```

```
    printf(" total = %d \n",total);
```

```
}
```

คำสั่งนี้จะได้เรียนต่อไป

```
sum = 0
```

```
total = 0
```


ตัวดำเนินการทางคณิตศาสตร์

การคำนวณ	ตัวดำเนินการ	ตัวอย่าง	การทำงาน
บวก	+	$c = a + b;$	นำ a บวก b ผลลัพธ์เก็บใน c
ลบ	-	$c = a - b;$	นำ a ลบ b แล้วเก็บใน c
คูณ	*	$c = a * b;$	นำ a คูณ b แล้วเก็บใน c
หาร	/	$c = a / b;$	การหารต้องระวังจะขึ้น กับประเภทของข้อมูลด้วย
มอดุลัส	%	$c = a \% b;$	เป็นการหารที่เก็บเศษไว้ใน c

ตัวดำเนินการเอกภาค (unary operator)

เป็นการใช้ตัวดำเนินการกับตัวแปรตัวเดียว มีการใช้สองแบบคือ

1. ตัวดำเนินการเอกภาคเติมหลัง (postfix mode)

EX: `a++;`

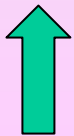
2. ตัวดำเนินการเอกภาคเติมหน้า (prefix mode)

EX: `++a;`

ตัวอย่างการใช้งาน

การคำนวณ	ตัวดำเนินการ	ตัวอย่าง	การทำงาน
เพิ่มค่าที่ละหนึ่ง	++	$x++$	$x = x + 1$
		$++x$	$x = x + 1$
ลดค่าที่ละหนึ่ง	--	$x--$	$x = x - 1$
		$--x$	$x = x - 1$

$y = ++x;$



เพิ่มค่า x ขึ้นหนึ่งแล้วใส่ให้ตัวแปร y

$y = x++;$



นำค่า x ให้ตัวแปร y ก่อน จากนั้นเพิ่มค่า x ขึ้นหนึ่ง

ตัวดำเนินการประกอบ

เป็นการใช้ตัวดำเนินการหนึ่งตัวร่วมกับเครื่องหมายเท่ากับ

ตัวดำเนินการประกอบ

ตัวอย่าง

การทำงาน

`+=`

`x += 5`

`x = x + 5`

`-=`

`x -= 5`

`x = x - 5`

`*=`

`x *= y`

`x = x * y`

`/=`

`x /= y`

`x = x / y`

`%=`

`x %= 5`

`x = x % 5`

`+=`

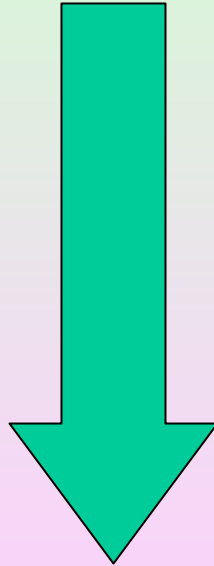
`x += y/8`

`x = x + y / 8`

ลำดับในการดำเนินการ

ถ้าหากมีการใช้ตัวดำเนินการประกอบและตัวดำเนินการเอกภาคหลายตัว จะมีลำดับทำงานดังนี้

1. ()
2. ++ --
3. * / %
4. + -
5. += *= /= -= %=



ตัวอย่างการใช้งาน

คำสั่ง	x	y
<code>int x = 10, y = 20;</code>	10	20
<code>++x;</code>	11	20
<code>y = - -x;</code>	10	10
<code>x = x-- +y;</code>	19	10
<code>y = x- ++x;</code>	20	0

การแปลงชนิดข้อมูล (type cast)

เป็นการเปลี่ยนแปลงข้อมูลชนิดหนึ่งให้เป็นอีกชนิดหนึ่ง

ตัวแปร = (ชนิดของข้อมูล) นิพจน์;

ข้อมูลใหม่ที่ต้องการให้กับตัวแปร

ตัวอย่าง

```
#include "stdio.h"
```

```
main()
```

```
{
```

```
    int x;
```

```
    x = 5.6 + 3.5;
```

/* จะได้ x = 9 */

```
    x = (int) 5.6 + (int) 3.5;
```

/* แปลง 5.6 ให้เป็น 5 แปลง 3.5 ให้เป็น 3 */

```
}
```

จะได้ x = 8

ฟังก์ชันรับข้อมูลและแสดงผล

ฟังก์ชันรับข้อมูล (Input Function)

- `scanf ( )` รับข้อมูลจากคีย์บอร์ดมาเก็บในตัวแปรที่กำหนด
- `getchar( )` รับข้อมูลอักขระตัวเดียว และต้องกด Enter ทุกครั้ง
- `getch( )` รับข้อมูลอักขระตัวเดียว ไม่ต้องกด Enter และไม่แสดงผล
- `getche( )` รับข้อมูลอักขระตัวเดียวเหมือน `getch( )` แต่จะแสดงผลด้วย
- `gets( )` รับข้อความ (string) มาเก็บในตัวแปรที่กำหนด

scanf()

scanf(control string ; argument list)

- control string เป็นรหัสรูปแบบข้อมูล (format code)
- argument list ตัวแปรที่ใช้เก็บข้อมูล ใช้เครื่องหมาย & นำหน้าชื่อ แต่ถ้ารับข้อมูลเป็นสตริงไม่ต้องมีเครื่องหมายนี้

Format code

รูปแบบ	ความหมาย
%c	ใช้แสดงอักขระตัวเดียว
%d	ใช้แสดงจำนวนเต็มเลขฐานสิบ
%e	ใช้แสดงข้อมูลตัวเลขจุดทศนิยม
%f , %lf	ใช้แสดงข้อมูลแบบ flot
%g	
%h	ใช้แสดงข้อมูล short integer
%l	
%o	ใช้แสดงข้อมูลจำนวนเต็มเลขฐานแปด
%u	ใช้แสดงข้อมูล unsigned int
%x	ใช้แสดงข้อมูลเลขฐานสิบหก
%s	ใช้แสดงสตริง

getchar()

```
getchar ( );
```

```
หรือ char_var = getchar( );
```

char_var เป็นตัวแปรชนิด char ที่ใช้เก็บข้อมูลตัวอักขระที่ป้อนผ่าน
คีย์บอร์ด

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

```
void main( )
```

```
{
```

```
    char c;
```

```
    clrscr();
```

```
    printf("Enter a single character :");
```

```
    c = getchar( );
```

```
    printf("You type a character is .... %c\n",c);
```

```
}
```

getch()

getch ();

หรือ char_var = getch();

char_var เป็นตัวแปรชนิด char ที่ใช้เก็บข้อมูลตัวอักขระที่ป้อนผ่านคีย์บอร์ด

ฟังก์ชันนี้ใช้รับอักขระตัวเดียวจากคีย์บอร์ด ไม่ต้องกด Enter ข้อมูลที่รับเข้าไปจะไม่แสดงผล

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

```
void main( )
```

```
{
```

```
    char d;
```

```
    clrscr();
```

```
    printf("Enter a single character :");
```

```
    d = getch( );
```

```
    printf("You type a character is .... %c\n",d);
```

```
}
```

getche()

getche ();

หรือ char_var = getche();

char_var เป็นตัวแปรชนิด char ที่ใช้เก็บข้อมูลตัวอักขระที่ป้อนผ่านคีย์บอร์ด

ฟังก์ชันนี้ใช้รับอักขระตัวเดียวจากคีย์บอร์ด ไม่ต้องกด Enter ข้อมูลที่รับเข้าไปจะแสดงผลออกมาด้วย


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

```
void main( )
```

```
{
```

```
    char e;
```

```
    clrscr();
```

```
    printf("Enter a single character :");
```

```
    e = getche( );
```

```
    printf("You type a character is .... %c\n",e);
```

```
}
```

gets()

```
gets(string_var);
```

string_var เป็นตัวแปรสตริงที่ใช้เก็บข้อความ (string constant)

ฟังก์ชันนี้ใช้รับข้อมูลจากคีย์บอร์ดมาเก็บในตัวแปรสตริง

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

```
void main( )
```

```
{
```

```
    char name[30];
```

```
    int age
```

```
    printf("Enter your name :");
```

```
    gets(name);
```

```
    printf("Enter your age :");
```

```
    scanf("%d",&age);
```

```
    printf("Your name is ...%s\n",name)
```

```
    printf("Your are %d years old. \n",age);
```

```
}
```

สรุปการใช้ input function

- รับข้อมูล string ควรใช้ฟังก์ชัน gets() หรือ scanf()
- รับข้อมูลตัวเลขควรใช้ฟังก์ชัน scanf()
- รับตัวเลขหนึ่งตัวหรือตัวอักษรหนึ่งตัว ไม่แสดงทางจอภาพ ใช้ getch()
แต่ถ้าต้องการให้แสดงทางจอภาพใช้ getche()
- การใช้ scanf() รับข้อมูลติดกันมากกว่า 2 ครั้งอาจเกิดข้อผิดพลาดได้

ฟังก์ชันแสดงข้อมูล (Output Function)

- `printf ( )` แสดงผลลัพธ์ออกทางจอภาพตามรูปแบบที่กำหนด
- `putchar( )` แสดงตัวอักขระตัวเดียวออกทางจอภาพ
- `puts( )` แสดงข้อความสตริงออกทางจอภาพ

printf()

```
printf (“ control “,value);
```

control : เป็นส่วนควบคุมการแสดงผล ประกอบด้วยข้อความ
รหัสควบคุม(เช่น %d %f) และอักขระควบคุมการแสดงผล(\n)

value : ได้แก่ค่าตัวแปร นิพจน์ ที่ต้องการแสดงผล ถ้าหากมีมากกว่า
หนึ่งตัวให้ใช้ comma คั่นระหว่างแต่ละตัว

รหัสรูปแบบ (Format Code)

- %d แสดงเลขจำนวนเต็ม (int,short,unsigned ,....)
- %f แสดงเลขทศนิยม (float,double,...)
- %c แสดงตัวอักขระหนึ่งตัว
- %s แสดงผลเป็นข้อความ
- %p แสดงผลเป็นตัวชี้ตำแหน่ง (pointer)

รูปแบบอื่น ๆ ดูเพิ่มเติมในกลุ่มมือ

ตัวอย่าง

```
int x1 = 43, x2 = 0x77 , x3 = 0573;
```

```
float y1 = -764.512 , y2 = 1.25e02;
```

```
char z = 'A';
```

```
char name[10] = "COMPUTER ";
```

```
main()
```

```
{
```

```
    printf("%d ",x1);
```

```
    printf("%x %o ",x2,x3);
```

```
    printf("%f %e ",y1,y2);
```

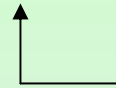
```
    printf("%c %s ",z,name);
```

```
}
```


การกำหนดรูปแบบทศนิยม

รูปแบบ

%.nf



จำนวนจุดทศนิยมที่ต้องการแสดงผล

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

```
float x1 = 17.765 ,x2 = 1.578e+03;
```

```
main()
```

```
{
```

```
    printf("%.4f ",x1);
```

```
    printf("%.1f ",x1);
```

```
    printf("%.2e ",x2);
```

```
    printf("%.1e ",x2);
```

```
}
```

การกำหนดพื้นที่แสดงผล

รูปแบบ %m.nf

m : จำนวนพื้นที่แสดงผล

n : จำนวนอักขระที่ต้องการนำมาแสดงผล

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

```
char message[ ] = "C Language";
```

```
main()
```

```
{
```

```
    printf("%12.6s",message);
```

```
}
```

รหัสควบคุมการแสดงผล

- \n ขึ้นบรรทัดใหม่
- \t เว้นช่องว่างเป็นระยะ 1 tab หรือหกตัวอักษร
- \r ให้เคอร์เซอร์ขึ้นบรรทัดใหม่
- \f เว้นช่องว่างเป็นระยะหนึ่งหน้าจอ

ตัวอย่างการแสดงผล

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

```
#define PUBLISH “Infopress”
```

```
int x1 = 14, x2 = 5;
```

```
main()
```

```
{
```

```
    printf(“This is the book from %s\n”,PUBLISH);
```

```
    printf(“Sum of %d + %d = \t%d\n”,x1 , x2 , x1+x2);
```

```
}
```

putchar() แสดงอักขระตัวเดียว

```
putchar(char);
```

↑ เป็นตัวแปรชนิด char หรือเขียนภายในเครื่องหมาย ‘ ‘

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

```
char first = 'A'
```

```
{
```

```
    clrscr();
```

```
    putchar(first);
```

```
    putchar('T');
```

```
}
```