

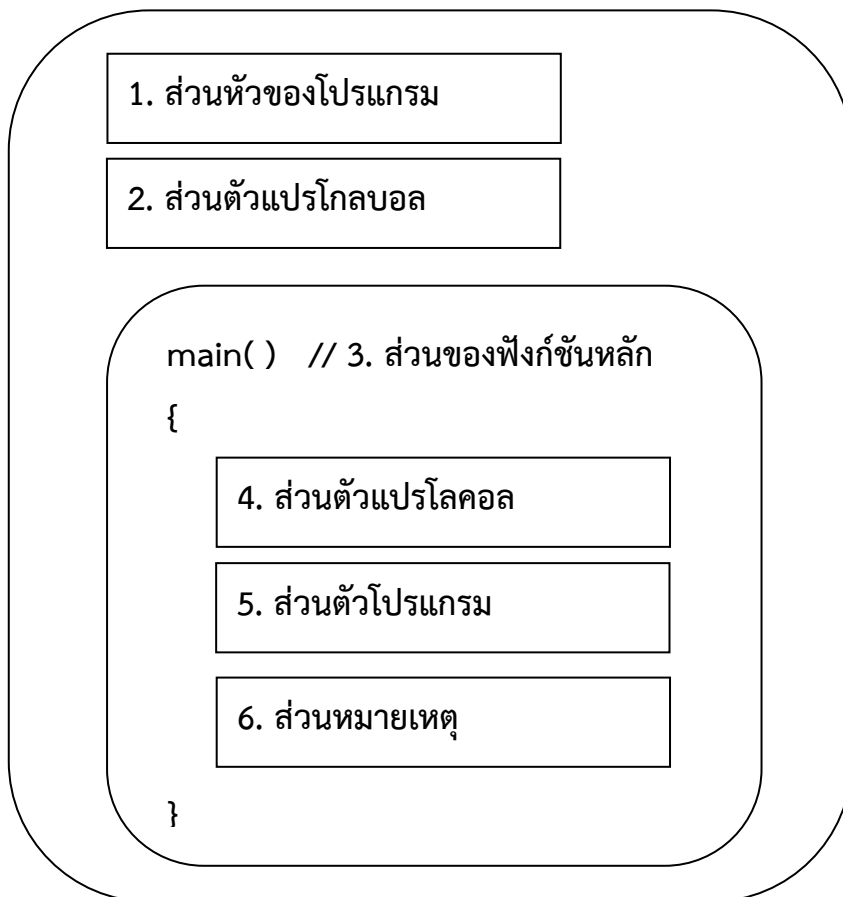
หน่วยที่ 5 โครงสร้างภาษาซีเบื้องต้น

การเขียนโปรแกรมคอมพิวเตอร์นั้น ผู้เขียนโปรแกรมจะต้องทราบหลักการและรูปแบบของการเขียนโปรแกรมภาษานั้น ๆ ภาษาซีก็เช่นเดียวกันรูปแบบของภาษานี้ประกอบไปด้วยส่วนต่าง ๆ 6 ส่วน โดยผู้พัฒนาโปรแกรมจะต้องทราบว่าแต่ละส่วนนั้นมีรูปแบบการเขียนอย่างไร ซึ่งหัวข้อที่จะเรียนในหน่วยนี้ มีดังนี้

1. โครงสร้างของโปรแกรมภาษาซี
2. คำสั่งคำนวณ
3. คำสั่งรับและแสดงผลข้อมูล

1. โครงสร้างของโปรแกรมภาษาซี

การเขียนโปรแกรมแต่ละภาษานั้นโครงสร้างของภาษาจะแตกต่างกัน ลักษณะโครงสร้างของภาษาซี แบ่งออกได้เป็น 6 ส่วนดังต่อไปนี้



ภาพที่ 5.1 โครงสร้างของโปรแกรมภาษาซี

โครงสร้างของโปรแกรมประกอบไปด้วยหลายส่วน แต่ในการเขียนโปรแกรมนั้นไม่จำเป็นต้องเขียนหมดทุกส่วน ส่วนใดไม่ใช้ก็สามารถตัดทิ้งได้ แต่ทุกโปรแกรมต้องมีส่วนหัวและส่วนฟังก์ชันหลัก รายละเอียดของส่วนต่าง ๆ เป็นดังนี้

1. ส่วนหัวของโปรแกรม (Header Part)

ส่วนหัวของโปรแกรม หรืออาจเรียกอีกชื่อหนึ่งว่า **Preprocessor Directives** จะใช้สำหรับเรียกไฟล์ที่โปรแกรมต้องการในการทำงาน และกำหนดค่าต่าง ๆ โดยคอมไพเลอร์ (compiler) จะกระทำการใด ๆ ตามคำสั่งก่อนที่จะคอมไพล์หรือแปลผลโปรแกรม ซึ่งจะต้องขึ้นต้นด้วยเครื่องหมาย ไดรექทีฟ (Directive) **#** แล้วตามด้วยชื่อโปรแกรมหรือชื่อตัวแปรที่ต้องการกำหนดค่า

ส่วนหัวของโปรแกรม เป็นส่วนที่เก็บไลบรารีมาตรฐานของภาษาซี ซึ่งจะถูกดึงเข้ามารวมกับโปรแกรมในขณะที่กำลังทำการคอมไพล์ ส่วนหัวของโปรแกรมจะมีส่วนขยายเป็น .h หรือ .c และเป็นส่วนที่จำเป็นต้องมีอย่างน้อย 1 เฮดเดอร์ไฟล์ก็คือ stdio.h โดยจะเป็นที่เก็บไลบรารีมาตรฐานที่จัดการเกี่ยวกับอินพุตและเอาต์พุต ดังนี้

#include เป็นการแจ้งให้คอมไพเลอร์อ่านไฟล์อื่นเข้ามาคอมไพล์ร่วมด้วย รูปแบบการใช้งานจะทำได้โดยการเขียน #include แล้วตามด้วยชื่อไฟล์ มีรูปแบบคำสั่ง 2 แบบ ดังนี้

รูปแบบ

#include <File_Name>

หรือ

#include "File_Name"

ความหมาย

File_Name คือ ชื่อไฟล์ที่ต้องการเรียกใช้

ตัวอย่าง เช่น

#include <stdio.h> หมายความว่า ให้อ่านไฟล์ stdio.h เข้ามาด้วย

#include "Pro1.c" หมายความว่า ให้อ่านไฟล์ Pro1.c เข้ามาด้วย

ตัวอย่าง ไฟล์นามสกุล .h ที่ตามหลัง #include เป็น Header file หรือไฟล์ส่วนหัวที่รวบรวมคำสั่งต่าง ๆ ของภาษาซีเอาไว้ เพื่อให้ผู้เขียนโปรแกรมสามารถใช้ฟังก์ชันได้ มีอยู่หลายไฟล์ เช่น

stdio.h เก็บฟังก์ชันที่ใช้งานทั่วไป เช่น printf , scanf โปรแกรมส่วนมากจะใช้ไฟล์นี้

conio.h เก็บฟังก์ชันควบคุมการแสดงผลต่าง ๆ

string.h เก็บฟังก์ชันที่ใช้ในการประมวลผลกับข้อความ

math.h เก็บฟังก์ชันที่ใช้ในการคำนวณทางคณิตศาสตร์ เช่น sin, cos, log, pow, sqrt ฯลฯ

graphics.h เก็บฟังก์ชันที่ใช้ในกราฟิกโหมด

การกำหนดชื่อไฟล์หลัง #include นั้น อาจใช้เครื่องหมาย < > คร่อมชื่อไฟล์ก็ได้ ซึ่งจะเป็นการอ่านไฟล์จากไดเรกทอรีที่กำหนดไว้ก่อน แต่ถ้าใช้ “ ” เป็นการอ่านไฟล์จากไดเรกทอรีปัจจุบันที่กำลังติดต่อกำลังอยู่ และไฟล์ที่จะ include เข้ามานี้จะต้องไม่มีฟังก์ชัน main() โดยมากแล้วจะประกอบด้วยโปรแกรมน้อย ค่าคงที่ และข้อกำหนดต่าง ๆ

#define การประกาศตัวแทนค่าคงที่ หรือการกำหนดค่านิพจน์ต่าง ๆ มีรูปแบบดังนี้

รูปแบบ

#define Name Value

ความหมาย

Name คือ ชื่อค่าคงที่ นิยมเขียนเป็นตัวอักษรพิมพ์ใหญ่ เพื่อให้เกิดความแตกต่างระหว่างชื่อตัวแปรกับชื่อค่าคงที่

Value คือ ค่าคงที่ หรือนิพจน์ต่าง ๆ

ตัวอย่าง เช่น

```
#define GPA 3.81          #define CH 'a'
#define PI 3.14           #define AD 2564-543
#define NAME "SORALAK"    #define AREA(x) PI*x*x
```

2. ส่วนตัวแปรแบบโกลบอล (Global Declarations หรือ Global Variable)

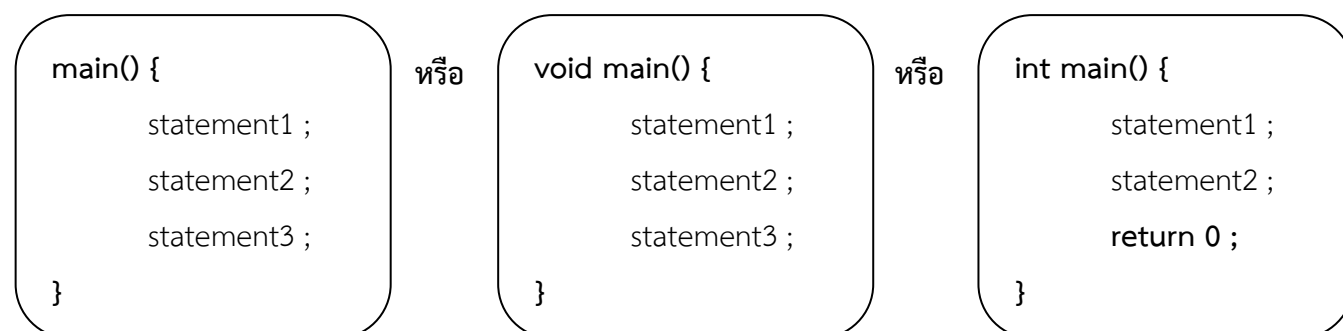
ตัวแปรแบบโกลบอล (Global variables) เป็นส่วนที่ใช้ประกาศตัวแปรหรือค่าต่าง ๆ ที่ให้สามารถเรียกใช้งานได้ทั้งโปรแกรม โดยฟังก์ชันทุก ๆ ตัวสามารถเรียกใช้งานได้ ซึ่งถือว่าเป็นตัวแปรสาธารณะ และในส่วนนี้ไม่จำเป็นต้องมีก็ได้

3. ส่วนฟังก์ชันหลัก (main() function)

ฟังก์ชัน (functions) เป็นส่วนที่ประกอบไปด้วยคำสั่งต่าง ๆ ที่จะทำให้โปรแกรมทำงาน โดยนำคำสั่งต่าง ๆ มาเรียงต่อกัน และคำสั่งในแต่ละบรรทัดจะจบท้ายด้วยเครื่องหมายเซมิโคลอน (;) ซึ่งในภาษาซีจะบังคับให้ต้องมีฟังก์ชันอย่างน้อย

1 ฟังก์ชัน นั่นคือ ฟังก์ชัน main() และในโปรแกรม 1 โปรแกรมสามารถมีฟังก์ชันได้มากกว่า 1 ฟังก์ชัน โดยขอบเขตของการเริ่มต้นการทำงานของฟังก์ชัน จะเริ่มจากเครื่องหมายปีกกาเปิด "{" และสิ้นสุดการทำงานด้วยเครื่องหมายปีกกาปิด "}"

ฟังก์ชัน main() สามารถเขียนตามรูปแบบได้ ดังนี้



4. ส่วนตัวแปรแบบโลคอล (Local Declarations หรือ Local Variable)

ตัวแปรแบบโลคอล (Local variables) เป็นส่วนที่ใช้สำหรับประกาศตัวแปรหรือค่าต่าง ๆ ที่จะใช้ได้เฉพาะในฟังก์ชันนั้น ๆ ฟังก์ชันอื่นจะไม่สามารถเข้าถึงหรือเรียกใช้งานได้ โดยจะทำการประกาศตัวแปรก่อนการใช้งานเสมอ และจะต้องประกาศไว้ในส่วนนี้เท่านั้น

5. ส่วนตัวโปรแกรม

ตัวแปรโปรแกรม (Statements) เป็นส่วนที่ประกอบไปด้วยคำสั่งต่าง ๆ เพื่อให้โปรแกรมทำงานตามที่ต้องการ และเมื่อจบคำสั่งจะใช้เครื่องหมายเซมิโคลอน (Semi-colon) ";" เพื่อเป็นการบอกให้รู้ว่าจบหนึ่งคำสั่ง ทำให้ภาษาซีไม่สับสนกับการขึ้นบรรทัดใหม่ เพราะสามารถพิมพ์คำสั่งหลายคำสั่งในบรรทัดเดียวกันได้ โดยมีเครื่องหมายเซมิโคลอน (Semi-colon) ";" เป็นตัวจบคำสั่ง โดยปกติแล้วคำสั่งของภาษาซีจะเขียนด้วยตัวอักษรภาษาอังกฤษพิมพ์เล็ก เนื่องจากภาษาซีจะแยกความแตกต่างของตัวอักษรพิมพ์เล็กและพิมพ์ใหญ่ (case sensitive) นั่นเอง

6. ส่วนหมายเหตุ (Comments or Remark)

การอธิบาย (Comments) หรือหมายเหตุ (Remark) ในโปรแกรม เป็นส่วนที่ใช้สำหรับแสดงข้อความเพื่อเขียนอธิบายสิ่งที่ต้องการในโปรแกรม โดยคอมไพเลอร์จะไม่นำไปแปลผลในส่วนนี้ หมายเหตุของภาษามี 2 รูปแบบ ได้แก่

6.1 แบบทัดเดียว

เขียนข้อความอธิบายหลังเครื่องหมาย //..... ใ้กับการ Comments หลังคำสั่งเป็นส่วนใหญ่

6.2 แบบหลายบรรทัด

เขียนข้อความอธิบายในเครื่องหมาย /*.....*/ มักใ้กับ Comment หลาย ๆ บรรทัด

ตัวอย่าง เช่น

```
#include <stdio.h>
int main() {
    //Single line Welcome user comment
    printf("Hello world"); //This comment
}
```

หรือ

```
#include <stdio.h>
int main() {
    /* Multi-line Welcome user comment
    written to demonstrate comments
    in C/C++ */
    printf("Hello world");
}
```

ตัวอย่างโปรแกรมภาษาซี

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

```
int a, b, c;
```

```
int sum(int x, int y);
```

```
int main()
```

```
{
```

```
    int i = 5, j = 3;
```

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

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

```
    c = sum(a, b);
```

```
    printf("\n%d + %d = %d", a, b, c);
```

```
    printf("\n%d + %d = %d", i, j, i+j);
```

```
}
```

```
int sum (int x, int y)
```

```
{
```

```
    return (x + y);
```

```
}
```

//ส่วนหัวของโปรแกรม หรือ คำสั่งตัวประมวลผลก่อน Preprocessor Directives

//ส่วนประกาศตัวแปรแบบโกลบอล

//คำสั่งแบบฟังก์ชัน

//ส่วนฟังก์ชันหลัก main()

//เริ่มต้นฟังก์ชัน main()

//ส่วนประกาศตัวแปรแบบโลคอล

//คำสั่งรับค่าตัวแปร

//คำสั่งรับค่าตัวแปร

//เรียกฟังก์ชัน sum()

//คำสั่งแสดงผล

//คำสั่งแสดงผล

//จบฟังก์ชัน main()

//ฟังก์ชัน sum()

//เริ่มต้นฟังก์ชัน sum()

//คำสั่งรวมค่าและส่งค่ากลับ

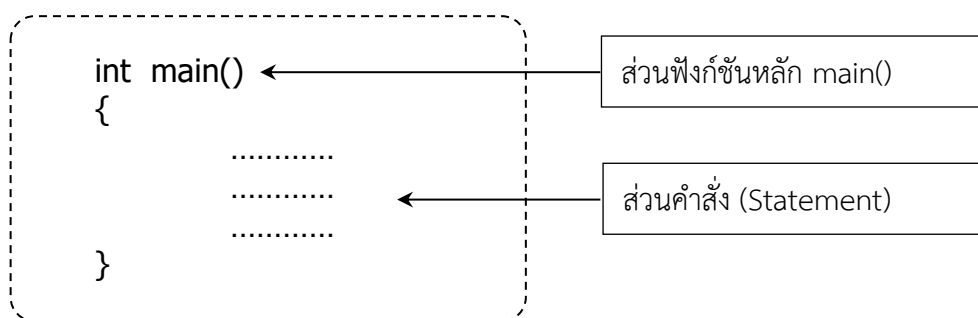
//จบฟังก์ชัน sum()

ตัวอย่างโปรแกรมภาษาซี

```
#include <stdio.h>
int i = 1;           // i เป็นตัวแปรแบบโกลบอล
int test(int j) {
    int x = 3;       // j, x เป็นตัวแปรแบบโลคอลของฟังก์ชัน test
    return x + j;
}
int main() {
    int y;           // y เป็นตัวแปรแบบโลคอลของฟังก์ชัน main
    printf ("Global variable i is %d\n", i);
    y = test(i);
    y = y * x;        // บรรทัดนี้ error เพราะไม่สามารถเรียกใช้ตัวแปร x ได้
    printf ("Local variable y is %d\n", y);
}
```

โปรแกรมภาษาซีเบื้องต้น

ตัวอย่างการเขียนโปรแกรมภาษาซีอย่างง่าย

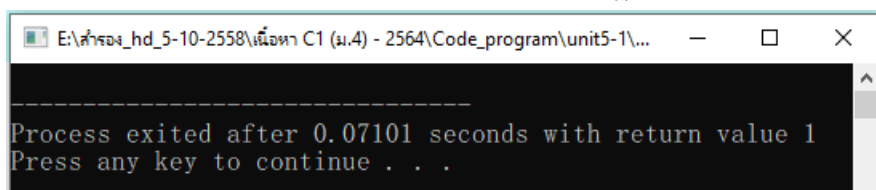


โปรแกรมภาษาซีที่สั้นที่สุด ที่สามารถคอมไพล์และรันได้ โดยคอมไพเลอร์ตามมาตรฐานของ ANSI C

บรรทัดที่	รหัสต้นฉบับ (Source Code)	อธิบายคำสั่ง
/*1*/	int main() {	/* ประกาศ main เป็นฟังก์ชันหลัก ชนิดจำนวนเต็ม และเริ่มต้น block ของฟังก์ชัน main (เริ่มต้นโปรแกรม) */
/*2*/	}	/* สิ้นสุด block ของฟังก์ชัน main (สิ้นสุดโปรแกรม) */

ผลการรันโปรแกรม (Output)

ไม่มีคำสั่งในฟังก์ชัน main จึงไม่ปรากฏผลลัพธ์ใด ๆ บนจอภาพ

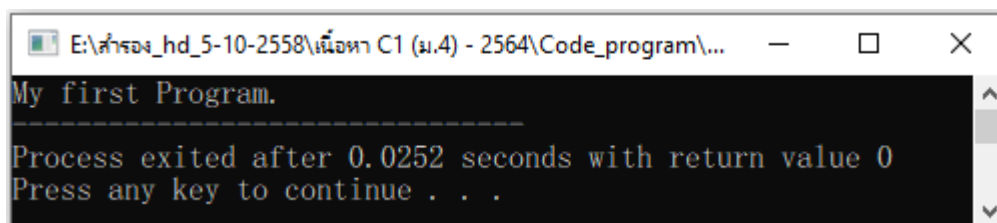


หมายเหตุ แม้ว่าจะไม่มีความหมายใด ๆ ในฟังก์ชัน main แต่ก็ยังต้องมีเครื่องหมาย { และ } เพื่อบอกขอบเขตของฟังก์ชัน main

โปรแกรมการแสดงผลพร้อมเบื้องต้น (โปรแกรมแรกของฉัน)

บรรทัดที่	รหัสต้นฉบับ (Source Code)	อธิบายคำสั่ง
/*1*/	#include <stdio.h>	/* เป็นคำสั่งของตัวประมวลผลก่อนซี จะมีผลให้เพิ่ม stdio.h ซึ่งเป็น เพิ่มส่วนหัวที่ใช้เก็บรวบรวมคำสั่งเกี่ยวกับการรับและแสดงผลข้อมูล ถูกอ่านเข้ามาเพื่อประมวลผลร่วมกับโปรแกรมนี้นี้ */
/*2*/	int main() {	/* ประกาศ main เป็นฟังก์ชันหลัก ชนิดจำนวนเต็ม และเริ่มต้น block ของฟังก์ชัน main (เริ่มต้นโปรแกรม) */
/*3*/	printf("My First Program");	/* แสดงข้อความ "My First Program" ทางจอภาพ */
/*4*/	return 0;	/* คืนค่า 0 ให้ฟังก์ชัน main โปรแกรมทำงานถูกต้องสมบูรณ์ */
/*5*/	}	/* สิ้นสุด block ของฟังก์ชัน main (สิ้นสุดโปรแกรม) */

ผลการรันโปรแกรม (Output)



```

E:\โครงงาน_5-10-2558\เนื้อหา C1 (ม.4) - 2564\Code_program\...
My first Program.
-----
Process exited after 0.0252 seconds with return value 0
Press any key to continue . . .

```

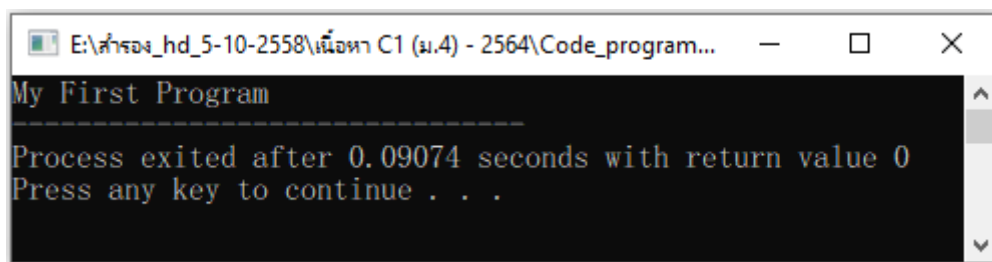
การเขียนอธิบายหรือหมายเหตุ (Comments or Remark)

ในภาษาซี นิยมเขียนการอธิบาย (Comments) หรือหมายเหตุ (Remark) อธิบายการทำงานในส่วนต่าง ๆ ของโปรแกรม เพื่อให้เข้าใจและอ่านโปรแกรมง่ายขึ้น โดยสิ่งที่เขียนอธิบายจะไม่มีผลต่อการประมวลผลของโปรแกรม

บรรทัดที่	รหัสต้นฉบับ (Source Code)	อธิบายคำสั่ง
/*1*/	/* Program : ex2-3.c	
/*2*/	Print text to screen */	
/*3*/	#include <stdio.h>	/* เป็นคำสั่งของตัวประมวลผลก่อนซี จะมีผลให้เพิ่ม stdio.h ซึ่งเป็น เพิ่มส่วนหัวที่ใช้เก็บรวบรวมคำสั่งเกี่ยวกับการรับและแสดงผลข้อมูล ถูกอ่านเข้ามาเพื่อประมวลผลร่วมกับโปรแกรมนี้นี้ */
/*4*/	int main() { // Start main	/* ประกาศ main เป็นฟังก์ชันหลัก ชนิดจำนวนเต็ม และเริ่มต้น block ของฟังก์ชัน main (เริ่มต้นโปรแกรม) */
/*5*/	printf("My First Program");	/* แสดงข้อความ "My First Program" ทางจอภาพ */
/*6*/	return 0;	/* คืนค่า 0 ให้ฟังก์ชัน main โปรแกรมทำงานถูกต้องสมบูรณ์ */
/*7*/	} // End main	/* สิ้นสุด block ของฟังก์ชัน main (สิ้นสุดโปรแกรม) */

ผลการรันโปรแกรม (Output)

ผลการรันโปรแกรม จะให้ผลเหมือนโปรแกรมตัวอย่างด้านบน



```

E:\source_hd_5-10-2558\เนื้อหา C1 (ม.4) - 2564\Code_program...
My First Program
-----
Process exited after 0.09074 seconds with return value 0
Press any key to continue . . .
  
```

การเขียนอธิบายหรือหมายเหตุ สามารถเลือกใช้ได้ตามความสะดวก ในภาษาซีมี 2 รูปแบบ ได้แก่

1. การเขียนแบบทัดเดียว จะเขียนข้อความอธิบายหลังเครื่องหมาย `//.....` ใช้กับการ Comments หลังคำสั่งเป็นส่วนใหญ่
2. การเขียนแบบหลายบรรทัด จะเขียนข้อความอธิบายในเครื่องหมาย `/*.....*/` มักใช้กับ Comment หลาย ๆ บรรทัด

2. คำสั่งคำนวณ หรือ ตัวดำเนินการกำหนดค่า

คำสั่งในภาษาซี สามารถแยกให้เห็นชัดเจนได้ เป็น 2 ลักษณะ คือ

- คำสั่งที่เป็นรูปแบบคำสั่งจริง ๆ เช่น คำสั่งกำหนดค่า คำสั่งควบคุม เป็นต้น
- คำสั่งที่เป็นรูปแบบของฟังก์ชัน เช่น คำสั่งรับข้อมูล คำสั่งแสดงผลข้อมูล เป็นต้น

ในโปรแกรมภาษาซี โปรแกรมหนึ่ง ๆ อาจประกอบด้วย คำสั่งที่เป็นรูปแบบคำสั่งจริง ๆ และคำสั่งที่เป็นรูปแบบของฟังก์ชันอยู่ด้วยกันก็ได้

2.1 คำสั่งคำนวณ (Assignment Statements) เป็นรูปแบบคำสั่งจริง ๆ ของภาษาซี เอาไว้สำหรับสั่งให้คอมพิวเตอร์คำนวณ หรืออาจใช้ในการกำหนดค่าเริ่มต้นให้กับตัวแปร หรือย้ายค่าตัวแปรจากที่หนึ่งไปยังอีกที่หนึ่งก็ได้

รูปแบบ

ตัวแปร = ตัวแปร ;

หรือ

ตัวแปร = ค่าคงที่ ;

หรือ

ตัวแปร = นิพจน์ ;

ความหมาย

เครื่องหมาย = (เท่ากับ) มีความหมายแตกต่างจากคณิตศาสตร์

ในภาษาซี หมายถึง การกำหนดค่า (Assignment) ฝั่งขวา ให้กับ ตัวแปรฝั่งซ้าย โดย

ตัวแปร คือ การนำค่าตัวแปรฝั่งขวาไปเก็บในตัวแปรฝั่งซ้าย

ค่าคงที่ คือ การนำค่าคงที่ไปเก็บในตัวแปรฝั่งซ้าย

นิพจน์ คือ การนำค่าผลลัพธ์การประมวลผลจากนิพจน์ ไปเก็บในตัวแปรฝั่งซ้าย

ตัวอย่าง เช่น

ตัวแปร	ค่าคงที่	นิพจน์
i = j ; x = y ; a = b ;	age = 10 ; j = 3.55 ; ch = 'A' ; name ="Pong";	area = 1.0 / 2.0 * b * h ; speed = distance / time ; total = 60 * hours + minutes ; pi = 22.0 / 7.0 ;

ตัวอย่างที่ 1 การประกาศตัวแปรจำนวนเต็ม และการกำหนดค่าให้ตัวแปร

บรรทัดที่	รหัสต้นฉบับ (Source Code)	อธิบายคำสั่ง
/*1*/	int main() {	/* ประกาศ main เป็นฟังก์ชันหลัก ชนิดจำนวนเต็ม และเริ่มต้น block ของฟังก์ชัน main (เริ่มต้นโปรแกรม) */
/*2*/	int i;	/* ประกาศตัวแปร i เป็นชนิดจำนวนเต็ม */
/*3*/	i = 20;	/* กำหนดค่า 20 ให้กับตัวแปร i */
/*4*/	return 0;	/* คืนค่า 0 ให้ฟังก์ชัน main โปรแกรมทำงานถูกต้องสมบูรณ์ */
/*5*/	}	/* สิ้นสุด block ของฟังก์ชัน main (สิ้นสุดโปรแกรม) */

ตัวอย่างที่ 2 การประกาศตัวแปรหลายตัวพร้อมการกำหนดค่าเริ่มต้น และการกำหนดค่าให้กับตัวแปรโดยใช้นิพจน์

บรรทัดที่	รหัสต้นฉบับ (Source Code)	อธิบายคำสั่ง
/*1*/	int main() {	/* ประกาศ main เป็นฟังก์ชันหลัก ชนิดจำนวนเต็ม และเริ่มต้น block ของฟังก์ชัน main (เริ่มต้นโปรแกรม) */
/*2*/	int x = 5 , y = 7;	/* ประกาศตัวแปร x และ y เป็นชนิดจำนวนเต็ม พร้อมกำหนดค่าเริ่มต้นให้กับตัวแปร x และ y เป็น 5 และ 7 ตามลำดับ */
/*3*/	float a;	/* ประกาศตัวแปร a เป็นจำนวนจริง */
/*4*/	a = 0.5 * x * y;	/* ประมวลผลนิพจน์ $0.5 * x * y$ จะได้ $0.5 * 5 * 7$ เป็น 17.500000 แล้วนำค่าที่ได้ไปเก็บไว้ในตัวแปร a */
/*5*/	return 0;	/* คืนค่า 0 ให้ฟังก์ชัน main โปรแกรมทำงานถูกต้องสมบูรณ์ */
/*6*/	}	/* สิ้นสุด block ของฟังก์ชัน main (สิ้นสุดโปรแกรม) */

หมายเหตุ เมื่อจบโปรแกรม ค่าที่เก็บในตัวแปร a คือ 17.500000

ตัวอย่างที่ 3 การใช้คำสั่งคำนวณเพิ่มค่าและลดค่า ครั้งที่ 1

บรรทัดที่	รหัสต้นฉบับ (Source Code)	อธิบายคำสั่ง
/*1*/	int main() {	/* ประกาศ main เป็นฟังก์ชันหลัก ชนิดจำนวนเต็ม และเริ่มต้น block ของฟังก์ชัน main (เริ่มต้นโปรแกรม) */
/*2*/	int i = 5, j = 0, k = 7 ;	/* ประกาศตัวแปร i , j และ k เป็นชนิดจำนวนเต็ม พร้อมกำหนดค่า เริ่มต้นให้กับตัวแปร i , j และ k เป็น 5 , 0 และ 7 ตามลำดับ */
/*3*/	int x = 4, y = 2, z = 8 ;	/* ประกาศตัวแปร x , y และ z เป็นชนิดจำนวนเต็ม พร้อมกำหนดค่า เริ่มต้นให้กับตัวแปร x , y และ z เป็น 4 , 2 และ 9 ตามลำดับ */
/*4*/	i = i + 1 ;	/* ประมวลผล i + 1 ก่อน จะได้ 5 + 1 เป็น 6 แล้วจึงนำ 6 ไปเก็บใน i */
/*5*/	j++ ;	/* เป็นคำสั่งเหมือน j = j + 1 ; จึงประมวลผล j + 1 ก่อน จะได้ 0 + 1 เป็น 1 แล้วจึงนำ 1 ไปเก็บใน j */
/*6*/	k += 1 ;	/* เป็นคำสั่งเหมือน k = k + 1 ; จึงประมวลผล k + 1 ก่อน จะได้ 7 + 1 เป็น 8 แล้วจึงนำ 8 ไปเก็บใน k */
/*7*/	x = x - 1 ;	/* ประมวลผล x - 1 ก่อน จะได้ 4 - 1 เป็น 3 แล้วจึงนำ 3 ไปเก็บใน x */
/*8*/	y-- ;	/* เป็นคำสั่งเหมือน y = y - 1 ; จึงประมวลผล y - 1 ก่อน จะได้ 2 - 1 เป็น 1 แล้วจึงนำ 1 ไปเก็บใน y */
/*9*/	z -= 1 ;	/* เป็นคำสั่งเหมือน z = z - 1 ; จึงประมวลผล z - 1 ก่อน จะได้ 8 - 1 เป็น 7 แล้วจึงนำ 7 ไปเก็บใน z */
/*10*/	return 0;	/* คืนค่า 0 ให้ฟังก์ชัน main โปรแกรมทำงานถูกต้องสมบูรณ์ */
/*11*/	}	/* สิ้นสุด block ของฟังก์ชัน main (สิ้นสุดโปรแกรม) */

หมายเหตุ เมื่อจบโปรแกรม

ค่าที่เก็บในตัวแปร i , j และ k คือ 6 , 1 และ 8 ตามลำดับ

ค่าที่เก็บในตัวแปร x , y และ z คือ 3 , 1 และ 7 ตามลำดับ

ตัวอย่างที่ 4 การใช้คำสั่งคำนวณเพิ่มค่าและลดค่า ครึ่งละใด ๆ

บรรทัดที่	รหัสต้นฉบับ (Source Code)	อธิบายคำสั่ง
/*1*/	int main() {	/* ประกาศ main เป็นฟังก์ชันหลัก ชนิดจำนวนเต็ม และเริ่มต้น block ของฟังก์ชัน main (เริ่มต้นโปรแกรม) */
/*2*/	int i = 5 , j = 0 , k = 7 , a = 12 ;	/* ประกาศตัวแปร i j k และ a เป็นชนิดจำนวนเต็ม พร้อมกำหนดค่า เริ่มต้นให้กับตัวแปร i j k และ a เป็น 5 0 7 และ 12 ตามลำดับ */
/*3*/	int x = 4 , y = 2 , z = 8 , c = 7 ;	/* ประกาศตัวแปร x , y , z และ c เป็นชนิดจำนวนเต็ม พร้อม กำหนดค่าเริ่มต้นให้กับตัวแปร x , y , z และ c เป็น 4 , 2 , 8 และ 7 ตามลำดับ */
/*4*/	i = i + 5 ;	/* ประมวลผล i + 5 ก่อน จะได้ 5 + 5 เป็น 10 แล้วจึงนำ 10 ไปเก็บใน i */
/*5*/	j += 6 ;	/* เป็นคำสั่งเหมือน j = j + 6 ; จึงประมวลผล j + 6 ก่อน จะได้ 0 + 6 เป็น 6 แล้วจึงนำ 6 ไปเก็บใน j */
/*6*/	k += a ;	/* เป็นคำสั่งเหมือน k = k + a ; จึงประมวลผล k + a ก่อน จะได้ 7 + 12 เป็น 19 แล้วจึงนำ 19 ไปเก็บใน k */
/*7*/	x = x - 4 ;	/* ประมวลผล x - 4 ก่อน จะได้ 4 - 4 เป็น 0 แล้วจึงนำ 0 ไปเก็บใน x */
/*8*/	y -= 2 ;	/* เป็นคำสั่งเหมือน y = y - 2 ; จึงประมวลผล y - 2 ก่อน จะได้ 2 - 2 เป็น 0 แล้วจึงนำ 0 ไปเก็บใน y */
/*9*/	z -= c ;	/* เป็นคำสั่งเหมือน z = z - c ; จึงประมวลผล z - c ก่อน จะได้ 8 - 7 เป็น 1 แล้วจึงนำ 1 ไปเก็บใน z */
/*10*/	return 0;	/* คืนค่า 0 ให้ฟังก์ชัน main โปรแกรมทำงานถูกต้องสมบูรณ์ */
/*11*/	}	/* สิ้นสุด block ของฟังก์ชัน main (สิ้นสุดโปรแกรม) */

หมายเหตุ เมื่อจบโปรแกรม

ค่าที่เก็บในตัวแปร i , j , k และ a คือ 10 , 6 , 19 และ 12 ตามลำดับ

ค่าที่เก็บในตัวแปร x , y , z และ c คือ 0 , 0 , 1 และ 7 ตามลำดับ

ตัวอย่างที่ 5 การใช้คำสั่งคำนวณ คูณ หาร และหารเอาเศษ (Modulus)

บรรทัดที่	รหัสต้นฉบับ (Source Code)	อธิบายคำสั่ง
/*1*/	<code>int main() {</code>	/* ประกาศ main เป็นฟังก์ชันหลัก ชนิดจำนวนเต็ม และเริ่มต้น block ของฟังก์ชัน main (เริ่มต้นโปรแกรม) */
/*2*/	<code>int i = 5 , j = 4 , k = 7 , a = 3 ;</code>	/* ประกาศตัวแปร i , j , k และ a เป็นชนิดจำนวนเต็ม พร้อมกำหนดค่าเริ่มต้นให้กับตัวแปร i , j , k และ a เป็น 5 , 4 , 7 และ 3 ตามลำดับ */
/*3*/	<code>i = i * 3 ;</code>	/* ประมวลผล $i \times 3$ ก่อน จะได้ 5×3 เป็น 15 แล้วจึงนำ 15 ไปเก็บใน i */
/*4*/	<code>j /= 2 ;</code>	/* เป็นคำสั่งเหมือน $j = j / 2$; จึงประมวลผล $j / 2$ ก่อน จะได้ $4 / 2$ เป็น 2 แล้วจึงนำ 2 ไปเก็บใน j */
/*5*/	<code>k %= a ;</code>	/* เป็นคำสั่งเหมือน $k = k \% a$; จึงประมวลผล k modulus a ก่อน จะได้ 7 modulus 3 เป็น 1 แล้วจึงนำ 1 ไปเก็บใน k */
/*6*/	<code>return 0;</code>	/* คืนค่า 0 ให้ฟังก์ชัน main โปรแกรมทำงานถูกต้องสมบูรณ์ */
/*7*/	<code>}</code>	/* สิ้นสุด block ของฟังก์ชัน main (สิ้นสุดโปรแกรม) */

หมายเหตุ เมื่อจบโปรแกรม

ค่าที่เก็บในตัวแปร i , j , k และ a คือ 15 , 2 , 1 และ 3 ตามลำดับ

ใบงานหน่วยที่ 5.1

1. จงเขียนคำสั่งอื่น ที่ใช้แทนคำสั่งที่กำหนดให้ โดยการประมวลผลยังคงเหมือนเดิม

```
int main() {
    int i = 5 , j = 3 , k = 20 ;
    i++ ;           ให้ผลเหมือนกับคำสั่ง i = i + 1 ;
    i = i + 2 ;     ให้ผลเหมือนกับคำสั่ง i += 2 ;
    i = i * 2 ;     ให้ผลเหมือนกับคำสั่ง i *= 2 ;
    1.1) j++ ;      ให้ผลเหมือนกับคำสั่ง.....
    1.2) j = j + 3 ; ให้ผลเหมือนกับคำสั่ง.....
    1.3) j = j * 4 ; ให้ผลเหมือนกับคำสั่ง.....
    1.4) k-- ;      ให้ผลเหมือนกับคำสั่ง.....
    1.5) k = k - 3 ; ให้ผลเหมือนกับคำสั่ง.....
    1.6) k = k / 2 ; ให้ผลเหมือนกับคำสั่ง.....
    1.7) k = k % 2 ; ให้ผลเหมือนกับคำสั่ง.....
    return 0 ;
}
```

2. จงประมวลผลโปรแกรมที่กำหนดให้ในกระดาษ แล้วให้บอกค่าที่เก็บในตัวแปรแต่ละบรรทัด

```
int main() {
    int i = 5 , j = 3 , k = 20 ;
    i++ ;           i เก็บค่า 6
    i = i + 2 ;     i เก็บค่า 8
    i = i * 2 ;     i เก็บค่า 16
    2.1) j++ ;      j เก็บค่า.....
    2.2) j = j + 3 ; j เก็บค่า.....
    2.3) j = j * 4 ; j เก็บค่า.....
    2.4) k-- ;      k เก็บค่า.....
    2.5) k = k - 3 ; k เก็บค่า.....
    2.6) k = k / 2 ; k เก็บค่า.....
    2.7) k = k % 2 ; k เก็บค่า.....
    return 0 ;
}
```

หน่วยที่ 5 โครงสร้างภาษาซีเบื้องต้น

สาระการเรียนรู้

1. โครงสร้างของโปรแกรมภาษาซี
2. คำสั่งคำนวณ
3. คำสั่งรับและแสดงผลข้อมูล

3. คำสั่งรับและแสดงผลข้อมูล

3.1 คำสั่งแสดงผลข้อมูล

คำสั่ง printf()

คำสั่ง printf() มีชื่อเต็มว่า print format เป็นคำสั่งที่ใช้ในการแสดงผลข้อความหรือข้อมูลที่อยู่ในตัวแปรหรือค่าคงที่หรือนิพจน์ ออกมาแสดงผลทางจอภาพ ซึ่งคำสั่ง printf() นี้ถูกเก็บไว้ใน I/O Library เวลาจะใช้คำสั่งต้องบอกให้ตัวแปลชุดคำสั่งภาษาซีรับรู้ก่อน โดยใช้คำสั่งตัวประมวลผลก่อนซึ่ง #include <stdio.h> เข้ามาก่อน โดยเอาไว้ส่วนหัวของโปรแกรม

รูปแบบ

```
printf (format-control-string , argument-list) ;
```

ความหมาย

format-control-string คือ สตริงควบคุมรูปแบบ จะต้องเขียนอยู่ภายใต้เครื่องหมาย " " (Double Quote) ภายในเครื่องหมาย " " จะประกอบด้วย ข้อความ รหัสควบคุมการแสดงผล และรหัสรูปแบบข้อมูล รายละเอียดดังนี้

1. ข้อความที่เป็นคำอธิบาย ที่ต้องการให้แสดงผลออกมาทางจอภาพ ส่วนมากจะใช้เพื่อแต่งรูปแบบของรายการให้อ่านเข้าใจและสวยงาม เช่น `printf("Hello world");`

2. รหัสควบคุมการแสดงผล (Escape character) ที่ใช้ในการควบคุมการแสดงผล จะต้องอยู่หลังเครื่องหมาย \ (Backslash) เช่น `printf("Hello \nworld");`

3. รหัสรูปแบบข้อมูล (String format) ที่ใช้ในการกำหนดการแสดงผลของข้อมูล จะต้องอยู่ตามหลังเครื่องหมาย % จำนวนรหัสรูปแบบข้อมูล ต้องสอดคล้องกับจำนวนค่าข้อมูลใน argument-list เช่น `printf("%d + %d = %d", 5, 3, 5+3);`

argument-list คือ ตัวแปร หรือค่าคงที่ หรือนิพจน์ ที่ต้องการนำไปแสดงผล ถ้ามีมากกว่า 1 ค่า จะต้องแยกกัน โดยใช้เครื่องหมายคอมม่า (,) คั่น ต้องสอดคล้องกับจำนวนรหัสรูปแบบข้อมูล เช่น

```
printf("x = %d", x);           // x = ?
printf("i + j = %d", i+j);     // i + j = ?
printf("%d + %d = %d", 5, 3, 5+3); // 5 + 3 = 8
printf("Area of Triangle = %f", 0.5*b*h); // Area of Triangle = ?
printf("Area of Circle = %f", 22.0/7.0*r*r); // Area of Circle = ?
```

ตารางแสดง รหัสควบคุมการแสดงผล

รหัสควบคุมการแสดงผล (escape character)	ใช้สำหรับ
<code>\n</code>	ขึ้นบรรทัดใหม่ 1 บรรทัด
<code>\t</code>	tab เว้นวรรค 1 tab (8 อักขระ)
<code>\a</code>	ส่งเสียงบี๊บเตือน 1 ครั้ง
<code>\'</code>	แสดงเครื่องหมาย '
<code>\"</code>	แสดงเครื่องหมาย "
<code>\\</code>	แสดงเครื่องหมาย \

ตารางแสดง รหัสรูปแบบข้อมูล

รหัสรูปแบบข้อมูล	ใช้สำหรับข้อมูลหรือตัวแปรแบบ
<code>%c</code>	char และ unsigned char
<code>%f</code>	float และ double
<code>%d</code>	int และ unsigned int
<code>%s</code>	char[] หรือ สายอักขระ
<code>%o</code>	ข้อมูลแบบตัวเลขฐานแปด
<code>%x , %X</code>	ข้อมูลแบบตัวเลขฐานสิบหก
<code>%e , %E</code>	ข้อมูลในรูป e ยกกำลัง

ตัวอย่างที่ 6 การใช้คำสั่งแสดงข้อความ "My First Program" ทางจอภาพ

บรรทัดที่	รหัสต้นฉบับ (Source Code)	อธิบายคำสั่ง
<code>/*1*/</code>	<code>#include <stdio.h></code>	<code>/*</code> เป็นคำสั่งของตัวประมวลผลก่อนซี จะมีผลให้แฟ้ม <code>stdio.h</code> ซึ่งเป็นแฟ้มส่วนหัวที่ใช้เก็บรวบรวมคำสั่งเกี่ยวกับการรับและแสดงผลข้อมูล ถูกอ่านเข้ามาเพื่อประมวลผลร่วมกับโปรแกรมนี้ <code>*/</code>
<code>/*2*/</code>	<code>int main() {</code>	<code>/*</code> ประกาศ <code>main</code> เป็นฟังก์ชันหลัก ชนิดจำนวนเต็ม และเริ่มต้น block ของฟังก์ชัน <code>main</code> (เริ่มต้นโปรแกรม) <code>*/</code>
<code>/*3*/</code>	<code>printf("My First Program");</code>	<code>/*</code> แสดงข้อความ "My First Program" ทางจอภาพ <code>*/</code>
<code>/*4*/</code>	<code>return 0;</code>	<code>/*</code> คืนค่า 0 ให้ฟังก์ชัน <code>main</code> โปรแกรมทำงานถูกต้องสมบูรณ์ <code>*/</code>
<code>/*5*/</code>	<code>}</code>	<code>/*</code> สิ้นสุด block ของฟังก์ชัน <code>main</code> (สิ้นสุดโปรแกรม) <code>*/</code>

หมายเหตุ คำสั่ง `printf` เป็นคำสั่งแสดงผลข้อมูลที่รวมอยู่ในแฟ้ม `stdio.h` ใช้สั่งให้แสดงผลของข้อความที่อยู่ในเครื่องหมาย " " ออกทางจอภาพ

ผลการรันโปรแกรม (Output)

My First Program

ตัวอย่างที่ 7 การใช้คำสั่งแสดงข้อความ 2 บรรทัดด้านล่าง โดยใช้คำสั่ง printf เพียงครั้งเดียว

Name : Peereaphong
Surname : Meepuangphol

บรรทัดที่	รหัสต้นฉบับ (Source Code)	อธิบายคำสั่ง
/*1*/	#include <stdio.h>	/* เป็นคำสั่งของตัวประมวลผลก่อนซี จะมีผลให้แฟ้ม stdio.h ซึ่งเป็นแฟ้มส่วนหัวที่ใช้เก็บรวบรวมคำสั่งเกี่ยวกับการรับและแสดงผลข้อมูล ถูกอ่านเข้ามาเพื่อประมวลผลร่วมกับโปรแกรมนี้ */
/*2*/	int main() {	/* ประกาศ main เป็นฟังก์ชันหลัก ชนิดจำนวนเต็ม และเริ่มต้น block ของฟังก์ชัน main (เริ่มต้นโปรแกรม) */
/*3*/	printf("\tName : Peeraphong \tSurname : Meepuangphol");	/* เว้นช่องว่าง 1 tab แล้วแสดงข้อความ "Name : Peeraphong" แล้วขึ้นบรรทัดใหม่ 1 บรรทัด แล้วเว้นช่องว่าง 1 tab แล้วแสดงข้อความ "Surname : Meepuangphol" ทางจอภาพ */
/*4*/	return 0;	/* คืนค่า 0 ให้ฟังก์ชัน main โปรแกรมทำงานถูกต้องสมบูรณ์ */
/*5*/	}	/* สิ้นสุด block ของฟังก์ชัน main (สิ้นสุดโปรแกรม) */

หมายเหตุ คำสั่ง printf จะไม่แสดง \t และ \n ออกมา แต่จะเว้นช่องว่าง 1 tab และขึ้นบรรทัดใหม่ 1 บรรทัดแทน

ผลการรันโปรแกรม (Output)

Name : Peereaphong
Surname : Meepuangphol

ตัวอย่างที่ 8 การใช้คำสั่งแสดงข้อความ "The sum is " และจำนวน 15 ที่เก็บในตัวแปร sum โดยใช้คำสั่ง printf เพียงครั้งเดียว

บรรทัดที่	รหัสต้นฉบับ (Source Code)	อธิบายคำสั่ง
/*1*/	#include <stdio.h>	/* เป็นคำสั่งของตัวประมวลผลก่อนซี จะมีผลให้แฟ้ม stdio.h ซึ่งเป็นแฟ้มส่วนหัวที่ใช้เก็บรวบรวมคำสั่งเกี่ยวกับการรับและแสดงผลข้อมูล ถูกอ่านเข้ามาเพื่อประมวลผลร่วมกับโปรแกรมนี้ */
/*2*/	int main() {	/* ประกาศ main เป็นฟังก์ชันหลัก ชนิดจำนวนเต็ม และเริ่มต้น block ของฟังก์ชัน main (เริ่มต้นโปรแกรม) */
/*3*/	int sum = 15;	/* ประกาศตัวแปร sum เป็นชนิดจำนวนเต็ม พร้อมกำหนดค่าเริ่มต้นให้กับตัวแปร sum เป็น 15 */
/*4*/	printf("The sum is %d" , sum);	/* แสดงข้อความ "The sum is %d" ทางจอภาพ โดยแสดงค่าของ sum ใน %d แทน */
/*5*/	return 0;	/* คืนค่า 0 ให้ฟังก์ชัน main โปรแกรมทำงานถูกต้องสมบูรณ์ */
/*6*/	}	/* สิ้นสุด block ของฟังก์ชัน main (สิ้นสุดโปรแกรม) */

หมายเหตุ คำสั่ง printf จะไม่แสดง %d ออกมา แต่จะแสดงค่าของข้อมูลออกมาแทน

ผลการรันโปรแกรม (Output)

The sum is 15

ตัวอย่างที่ 9 การใช้คำสั่งแสดงข้อความและการประมวลผลนิพจน์ทางคณิตศาสตร์

บรรทัดที่	รหัสต้นฉบับ (Source Code)	อธิบายคำสั่ง
/*1*/	#include <stdio.h>	/* เป็นคำสั่งของตัวประมวลผลก่อนซี จะมีผลให้เพิ่ม stdio.h ซึ่งเป็น เพิ่มส่วนหัวที่ใช้เก็บรวบรวมคำสั่งเกี่ยวกับการรับและแสดงผลข้อมูล ถูกอ่านเข้ามาเพื่อประมวลผลร่วมกับโปรแกรมนี้ */
/*2*/	int main() {	/* ประกาศ main เป็นฟังก์ชันหลัก ชนิดจำนวนเต็ม และเริ่มต้น block ของฟังก์ชัน main (เริ่มต้นโปรแกรม) */
/*3*/	int x = 8 , y = 4;	/* ประกาศตัวแปร x และ y เป็นชนิดจำนวนเต็ม พร้อมกำหนดค่า เริ่มต้นให้กับตัวแปร x และ y เป็น 8 และ 4 ตามลำดับ */
/*4*/	printf("%d + %d = %d" , x , y , x+y);	/* แสดงข้อความ "%d + %d = %d" ทางจอภาพ โดยแสดงค่าของ x ใน %d ตัวแรก แสดงค่า y ใน %d ตัวที่ 2 และแสดงค่าผลบวก x + y ใน %d ตัวที่ 3 แทน */
/*5*/	return 0;	/* คืนค่า 0 ให้ฟังก์ชัน main โปรแกรมทำงานถูกต้องสมบูรณ์ */
/*6*/	}	/* สิ้นสุด block ของฟังก์ชัน main (สิ้นสุดโปรแกรม) */

หมายเหตุ การแสดงผลนิพจน์ ในคำสั่ง printf จะประมวลผลนิพจน์นั้นก่อน แล้วจึงนำผลลัพธ์ไปแสดงในรหัส
รูปแบบข้อมูลที่กำหนด

ผลการรันโปรแกรม (Output)

8 + 4 = 12

Example 10 : C Output

```
#include <stdio.h>
int main()
{
    printf("C Programming");
    return 0;
}
```

Output

C Programming

Example 10-1 : C Output and escape character

```
#include <stdio.h>
int main()
{
    printf("Nangrong\n");
    printf("Buriram");
    return 0;
}
```

```
printf("Nangrong");
printf("\nBuriram");
return 0;
```

```
printf("Nangrong");
printf("\n");
printf("\Buriram");
return 0;
```

```
printf("Nangrong\nBuriram");
```

Output

Nangrong
Buriram

Example 11 : Integer Output

```
#include <stdio.h>
int main()
{
    int testInteger = 5;
    printf("Number = %d", testInteger);
    return 0;
}
```

Output

Number = 5

Example 11-2 : Integer Output

```
#include <stdio.h>
int main()
{
    int x = 7;
    printf("%d", x);
    printf("\n%d", x+9);
    return 0;
}
```

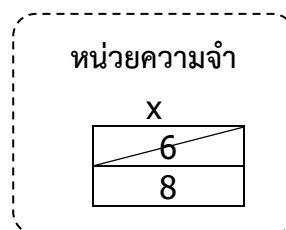
printf("%d\n%d ", x, x+9);

Output

7
16

Example 11-3 : Integer Output

```
#include <stdio.h>
int main()
{
    int x;
    x = 6;
    x = 8;
    printf("x = %d", x);
    return 0;
}
```

**Output**

x = 8

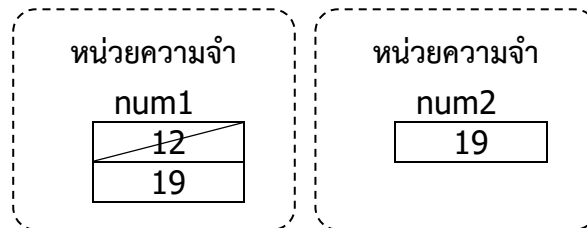
ในการประกาศตัวแปร เมื่อโปรแกรมถูกรันหรือมีการทำงานใด ๆ ค่าในหน่วยความจำหรือตัวแปรอาจมีการเปลี่ยนแปลงได้ โดยค่าเก่าจะหายไปและจะถูกแทนด้วยค่าใหม่ในหน่วยความจำ

Example 11-4 : Integer Output

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

```
int main()
```

```
{
    int num1, num2;
    num1 = 12;
    num2 = 19;
    num1 = num2;
    printf("num1 = %d", num1);
    printf("\nnum2 = %d", num2);
    return 0;
}
```



```
printf("num1 = %d\nnum2 = %d ", num1, num2);
```

Output

```
num1 = 19
```

```
num2 = 19
```

Example 12 : float and double Output

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

```
int main()
```

```
{
    float number1 = 13.5;
    double number2 = 12.4;
    printf("number1 = %f\n", number1);
    printf("number2 = %lf", number2);
    return 0;
}
```

```
printf("number1 = %f\nnumber2 = %lf", number1, number2);
```

Output

```
number1 = 13.500000
```

```
number2 = 12.400000
```

* %f, %lf จะแสดงผลตัวเลขจำนวนจริงที่มีทศนิยม 6 หลัก

Example 13 : Characters Output

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

```
int main()
```

```
{
    char chr = 'a';
    printf("character = %c", chr);
    return 0;
}
```

```
int chr = 97;
printf("character = %c", chr);
```

```
printf("character = %c", 97);
```

Output

```
character = a
```

Example 14 : String Output

```
#include <stdio.h>
int main()
{
    char chr[] = "Nangrong School";
    printf("string = %s", chr);
    return 0;
}
```

Output

string = Nangrong School

Example 15 : Octal and Hexadecimal

```
#include <stdio.h>
int main()
{
    int i = 27;
    printf("Octal = 0%o\n", i);
    printf("Hexadecimal = 0x%x\n", i);
    printf("Hexadecimal = 0x%X ", i);
    return 0;
}
```

Output

Octal = 033
Hexadecimal = 0x1b
Hexadecimal = 0x1B

Example 16 : Multi

```
printf("%s %d %f %c", "John", 14, -8.25, 'A');
```

Output

John 14 -8.250000 A

Example 17 : Float Output Decimals

```
printf("%f ", 8.25 );
```

Output

8.250000

การจัดรูปแบบการแสดงผล

ความกว้างพื้นที่การแสดงผล

การกำหนดความกว้าง (Width) ให้ใส่ตัวเลขจำนวนเต็มหลัง %

- หากกำหนดขนาดมากกว่าค่าที่ต้องการให้แสดงผล ค่าจะถูกแสดงผลชิดขอบด้านขวา พื้นที่ที่เหลือจะเป็นช่องว่างทางด้านซ้าย

- หากกำหนดขนาดน้อยกว่าค่าที่ต้องการให้แสดงผล ค่าจะถูกแสดงผลเหมือนไม่มีการกำหนดค่าความกว้างให้ จะแสดงชิดด้านซ้ายเท่ากับค่านั้น

Example 18

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

```
int main()
```

```
{
```

```
    printf("Width = 8 with integer.\n");
```

```
    printf("%8d\n", 12);
```

```
    printf("%8d\n", 1234);
```

```
    printf("%8d\n", 123456);
```

```
    printf("%8d\n", 123456789);
```

```
    return 0;
```

```
}
```

Output

```
Width = 8 with integer.
```

```
*****12
```

```
****1234
```

```
**123456
```

```
123456789
```

						1	2	
				1	2	3	4	
		1	2	3	4	5	6	
1	2	3	4	5	6	7	8	9

* แทนช่องว่าง 1 ช่อง

จำนวนหลักทศนิยม

การแสดงผลค่าส่ง printf() โดยปกติถ้าใช้ %f %e หรือ %E จะแสดงผลค่าทศนิยม 6 ตำแหน่ง

Example 19 : Float & Exponential Output

```
printf("%f %e %E", 300.12, 300.12, 300.12 );
```

Output

```
300.120000 3.001200e+02 3.001200E+02
```

การกำหนดจำนวนหลักเลขทศนิยม (precision) หลัง % ให้ใส่ . แล้วตามด้วยตัวเลขจำนวนเต็ม ที่ต้องการแสดงผลจำนวนทศนิยมตามความต้องการ

หากใช้กำหนดกับชนิดข้อมูลเลขจำนวนเต็ม ที่มีจำนวนตัวเลขน้อยกว่าจำนวนหลักทศนิยมจะมีการเพิ่ม 0 ให้ไว้ด้านหน้าตัวเลขจนครบตามจำนวนหลัก

หากนำไปใช้กำหนดกับข้อความตัวอักษร เป็นจำนวนตัวอักษรมากที่สุด ที่จะนำออกมาแสดง ส่วนตัวอักษรที่เกินจะถูกตัดทิ้งไป

Example 20

```
#include <stdio.h>
int main()
{
    printf("Floating normal = %f, 3 precision = %.3f\n", 123.4567890, 123.4567890);
    printf("Science 2 precision = %.2e\n", 123.4567890);
    printf("Integer 8 precision = %.8d\n", 12345);
    printf("String 10 precision = %.10s\n", "C Programming");
    return 0;
}
```

Output

```
Floating normal = 123.456789, 3 precision = 123.457
Science 2 precision = 1.23e+02
Integer 8 precision = 00012345
String 10 precision = C Programm
```

```
printf("%f %%.3f %%.2f %%.1f", 4.5678, 4.5678, 4.5678, 4.5678 );
```

Output

```
4.567800 4.568 4.57 4.6
```

```
printf("%8.2f", 4.5678 );
```

Output

***4.57 //แสดงเลข 4.57 มีความกว้างเท่ากับ 8 ตัวอักษรรวมทศนิยมด้วย ดังนั้นผลลัพธ์ที่ได้จะแสดงช่องว่างด้านหน้า 4 ตัวอักษร (* แทนช่องว่าง 1 ช่อง)

				4	.	5	7
--	--	--	--	---	---	---	---

```
printf("%12.3e", 12.345);
```

Output

***1.235e+01 // (* แทนช่องว่าง 1 ช่อง)

			1	.	2	3	5	e	+	0	1
--	--	--	---	---	---	---	---	---	---	---	---

```
printf("%8s", "Hello" );
```

Output

***Hello //แสดงข้อความ Hello ที่มีความกว้างเท่ากับ 8 ตัวอักษร
 ดังนั้นผลลัพธ์ที่ได้จะแสดงช่องว่างด้านหน้า 3 ตัวอักษร (* แทนช่องว่าง 1 ช่อง)

			H	e	l	l	o
--	--	--	---	---	---	---	---

* ในกรณีที่ต้องการเปลี่ยนการแสดงผลจากชิดขอบด้านขวาให้ชิดขอบด้านซ้ายแทน สามารถทำได้โดยการใส่เครื่องหมายลบ (-) หน้าตัวเลขจำนวนเต็มหลัง % เช่น

Example 21

```
#include <stdio.h>
int main()
{
    printf("%8d\n", 234);
    printf("%08d\n", 234);
    printf("%-8d\n", 234);
    return 0;
}
```

Output

*****234 // * แทนช่องว่าง 1 ช่อง
 00000234
 234***** // * แทนช่องว่าง 1 ช่อง

					2	3	4
0	0	0	0	0	2	3	4
2	3	4					

```
printf("%-8d", 234 );
printf("%-8d", 567 );
```

Output

234*****567***** (* แทนช่องว่าง 1 ช่อง)

2	3	4					5	6	7				
---	---	---	--	--	--	--	---	---	---	--	--	--	--

```
printf("%-8s", "Hello" );
```

Output

Hello*** //แสดงข้อความ Hello ชิดด้านซ้าย มีความกว้างเท่ากับ 8 ตัวอักษร ดังนั้นผลลัพธ์ที่ได้จะแสดงช่องว่าง
 ด้านหลัง 3 ตัวอักษร (* แทนช่องว่าง 1 ช่อง)

H	e	l	l	o			
---	---	---	---	---	--	--	--

```
printf("%-8c", 'z');
```

Output

z***** //แสดงตัวอักษร z ซิดด้านซ้าย มีความกว้างเท่ากับ 8 ตัวอักษร ดังนั้นผลลัพธ์ที่ได้จะแสดงช่องว่างด้านหลัง 7 ตัวอักษร (* แทนช่องว่าง 1 ช่อง)

z							
---	--	--	--	--	--	--	--

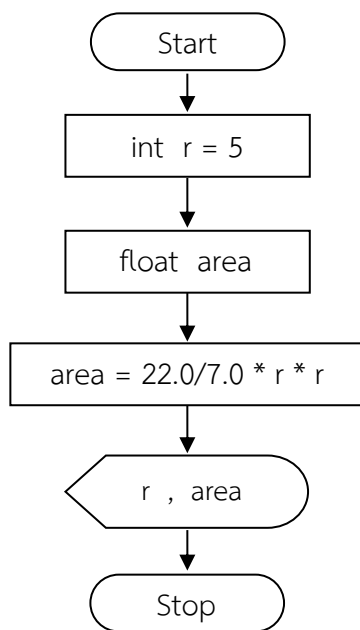
```
printf("%-12f", 11.234);
```

Output

11.234000*** //แสดงตัวอักษร 11.234000 ซิดด้านซ้าย มีความกว้างเท่ากับ 128 ตัวอักษร ดังนั้นผลลัพธ์ที่ได้จะแสดงช่องว่างด้านหลัง 3 ตัวอักษร (* แทนช่องว่าง 1 ช่อง)

1	1	.	2	3	4	0	0	0			
---	---	---	---	---	---	---	---	---	--	--	--

ตัวอย่างที่ 22 จงเขียนโปรแกรมคำนวณหาพื้นที่รูปวงกลม รัศมีเท่ากับ 5
จากโจทย์ เขียนผังงาน แสดงการแก้ปัญหา ได้ดังนี้



```
#include <stdio.h>
int main() {
    int r = 5;
    float area;
    area = 22.0 / 7.0 * r * r;
    printf("radius : %3d , Area of Circle : %7.2f" , r, area);
    return 0;
}
```

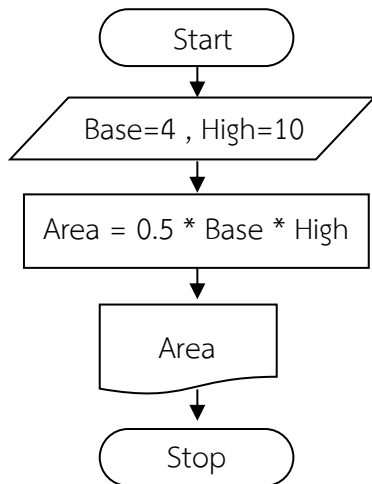
หมายเหตุ %3d หมายถึง แสดงผลจำนวนเต็ม ได้ 3 ตำแหน่ง

%7.2f หมายถึง แสดงผลทั้งหมด 7 ตำแหน่ง แบ่งเป็น แสดงจำนวนเต็มได้ 4 ตำแหน่ง แสดงทศนิยมได้ 2 ตำแหน่ง และแสดงจุดทศนิยม 1 ช่อง

ผลการรันโปรแกรม (Output)

radius : 5 ,Area of Circle : 78.57

ตัวอย่างที่ 23 จงเขียนโปรแกรมคำนวณหาพื้นที่รูปสามเหลี่ยม ที่มีความยาวฐาน 4 หน่วย สูง 10 หน่วย

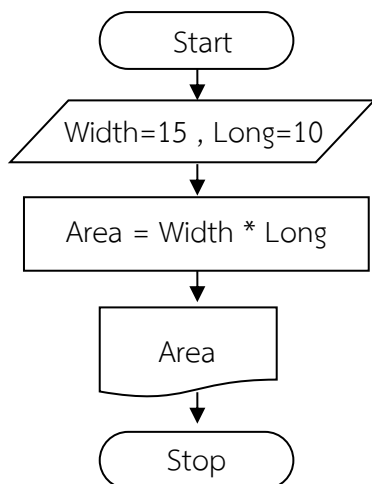


```
#include <stdio.h>
int main() {
    int Base = 4 , High = 10 ;
    float Area;
    Area = 0.5 * Base * High;
    printf("Area = %.2f" , Area );
    return 0;
}
```

ผลการรันโปรแกรม

Area = 20.00

ตัวอย่างที่ 24 จงเขียนโปรแกรมคำนวณหาพื้นที่รูปสี่เหลี่ยม ที่มีความกว้าง 15 หน่วย ยาว 10 หน่วย

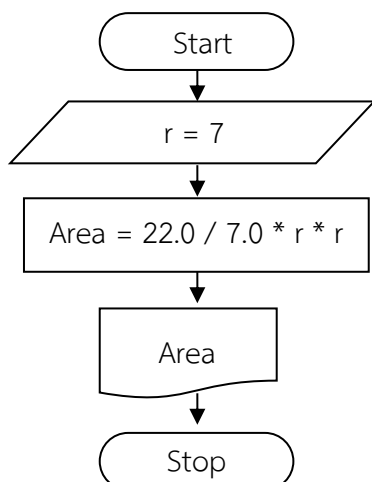


```
#include <stdio.h>
int main() {
    int Width=15 , Long=10, Area ;
    Area = Width * Long;
    printf("Area = %d" , Area );
    return 0;
}
```

ผลการรันโปรแกรม

Area = 150

ตัวอย่างที่ 25 จงเขียนโปรแกรมคำนวณหาพื้นที่รูปวงกลม ที่มีความยาวรัศมี 7 หน่วย

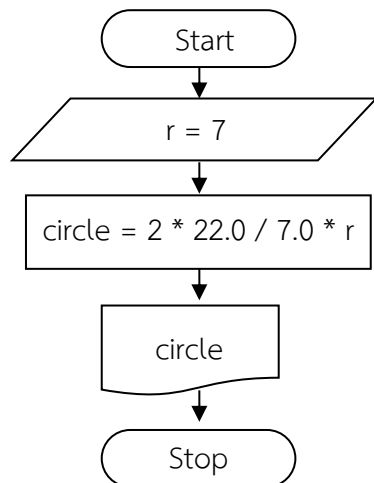


```
#include <stdio.h>
int main() {
    float r = 7, Area ;
    Area = 22.0/7.0 * r * r ;
    printf("Area = %.2f" , Area );
    return 0;
}
```

ผลการรันโปรแกรม

Area = 154.00

ตัวอย่างที่ 26 จงเขียนโปรแกรมคำนวณหาความยาวของเส้นรอบรูปวงกลม ที่มีความยาวรัศมี 7 หน่วย

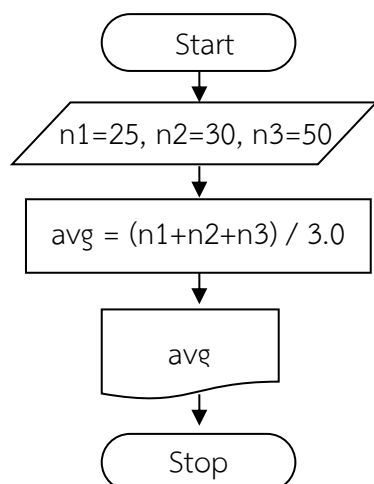


```
#include <stdio.h>
int main() {
    float r = 7, circle ;
    circle = 2*22.0/7.0*r ;
    printf("circle = %.2f" , circle );
    return 0;
}
```

ผลการรันโปรแกรม

circle = 44.00

ตัวอย่างที่ 27 จงเขียนโปรแกรมคำนวณหาค่าเฉลี่ยของตัวเลขจำนวนเต็ม 3 จำนวนดังนี้ 25 30 50



```
#include <stdio.h>
int main() {
    int n1=25, n2=30, n3=50 ;
    float avg ;
    avg = (n1+n2+n3) / 3.0 ;
    printf("Average = %f" , avg );
    return 0;
}
```

ผลการรันโปรแกรม

Average = 35.000000

ฝึกทำโจทย์ จงเขียนโปรแกรมรับค่าตัวเลขจำนวนนาฬิกาที่เป็นตัวเลขจำนวนเต็ม 1 จำนวนที่มากกว่าศูนย์ แล้วคำนวณเปลี่ยนตัวเลขค่านาฬิกาที่รับเข้ามา ให้แสดงผลเป็นจำนวนวัน จำนวนชั่วโมง และจำนวนนาฬิกา ตัวอย่าง เช่น กรอกตัวเลข 500 นาฬิกา ให้แสดงผล 0 วัน 8 ชั่วโมง 20 นาที , กรอกตัวเลข 4000 นาฬิกา ให้แสดงผล 2 วัน 18 ชั่วโมง 40 นาที

Input	Output		
4000	2	18	40
500	0	8	20

3.2 คำสั่งรับค่าข้อมูล

คำสั่ง scanf()

คำสั่ง scanf() เป็นคำสั่งที่ใช้ในการรับข้อมูลจากแป้นพิมพ์ (Keyboard) เข้ามาเก็บไว้ในตัวแปร

รูปแบบ

scanf (format-control-string , argument-list);

ความหมาย

format-control-string จะต้องเขียนอยู่ภายใต้เครื่องหมาย " " (Double Quote) ภายในเครื่องหมายประกอบด้วย **รหัสรูปแบบข้อมูล** (รหัสเหมือนกับที่ใช้ในคำสั่ง printf) หากต้องการรับข้อมูลมากกว่าหนึ่งตัวให้แยกกันด้วยเว้นวรรค (space bar) หรือจะกำหนดเป็นรูปแบบอื่นก็ได้ แต่ตอนรับค่าต้องป้อนรูปแบบให้ตรงกัน โดยจำนวนรหัสรูปแบบข้อมูลต้องสอดคล้องกับจำนวนตัวแปรใน argument-list

argument-list ตัวแปรสำหรับไว้เก็บค่าข้อมูลที่รับจากแป้นพิมพ์ ถ้ามีตัวแปรมากกว่าหนึ่งตัว ให้แยกกันด้วยเครื่องหมายคอมม่า (,) และต้องนำหน้าตัวแปรด้วยเครื่องหมาย & ทุกตัว ยกเว้นตัวแปรชุด/ตัวแปรสตริงที่ใช้เก็บข้อความ

ตัวอย่างที่ 28 การรับค่าจำนวนเต็ม 1 จำนวน จากแป้นพิมพ์

บรรทัดที่	รหัสต้นฉบับ (Source Code)	อธิบายคำสั่ง
/*1*/	#include <stdio.h>	/* เป็นคำสั่งของตัวประมวลผลก่อนซี จะมีผลให้เพิ่ม stdio.h ซึ่งเป็นเพิ่มส่วนหัวที่ใช้เก็บรวบรวมคำสั่งเกี่ยวกับการรับและแสดงผลข้อมูล ถูกอ่านเข้ามาเพื่อประมวลผลร่วมกับโปรแกรมนี้ */
/*2*/	int main() {	/* ประกาศ main เป็นฟังก์ชันหลัก ชนิดจำนวนเต็ม และเริ่มต้น block ของฟังก์ชัน main (เริ่มต้นโปรแกรม) */
/*3*/	int x;	/* ประกาศตัวแปร x เป็นชนิดจำนวนเต็ม */
/*4*/	scanf("%d" , &x);	/* รับค่าเลขจำนวนเต็ม จากแป้นพิมพ์ ไปเก็บไว้ในตัวแปร x เมื่อป้อนข้อมูลเสร็จแล้ว ต้องกดแป้น Enter เพื่อยืนยันข้อมูลและทำคำสั่งถัดไป */
/*5*/	return 0;	/* คืนค่า 0 ให้ฟังก์ชัน main โปรแกรมทำงานถูกต้องสมบูรณ์ */
/*6*/	}	/* สิ้นสุด block ของฟังก์ชัน main (สิ้นสุดโปรแกรม) */

หมายเหตุ เมื่อคำสั่ง scanf ทำงาน เครื่องจะหยุดรอรับค่า (เคอร์เซอร์กระพริบ)

เมื่อป้อนข้อมูลเสร็จแล้ว ต้องกดแป้น Enter เพื่อยืนยันข้อมูลและทำคำสั่งถัดไป

ผลการรันโปรแกรม (Output)

—

10←

(← คือ กดปุ่ม Enter เพื่อยืนยันข้อมูล นำไปเก็บไว้ในตัวแปรที่กำหนด)

ตัวอย่างที่ 29 การรับค่าจำนวนเต็ม 2 จำนวน จากแป้นพิมพ์

บรรทัดที่	รหัสต้นฉบับ (Source Code)	อธิบายคำสั่ง
/*1*/	<code>#include <stdio.h></code>	/* เป็นคำสั่งของตัวประมวลผลก่อนซี จะมีผลให้เพิ่ม stdio.h ซึ่งเป็น เพิ่มส่วนหัวที่ใช้เก็บรวบรวมคำสั่งเกี่ยวกับการรับและแสดงผลข้อมูล ถูกอ่านเข้ามาเพื่อประมวลผลร่วมกับโปรแกรมนี้ */
/*2*/	<code>int main() {</code>	/* ประกาศ main เป็นฟังก์ชันหลัก ชนิดจำนวนเต็ม และเริ่มต้น block ของฟังก์ชัน main (เริ่มต้นโปรแกรม) */
/*3*/	<code>int x , y;</code>	/* ประกาศตัวแปร x และ y เป็นชนิดจำนวนเต็ม */
/*4*/	<code>scanf("%d %d" , &x , &y);</code>	/* รับค่าเลขจำนวนเต็ม 2 จำนวน จากแป้นพิมพ์ ไปเก็บไว้ในตัวแปร x และ y ตามลำดับ คำนับจำนวนสองจำนวนด้วยเว้นวรรค เมื่อป้อน ข้อมูลเสร็จแล้ว ต้องกดแป้น Enter เพื่อยืนยันข้อมูลและทำคำสั่ง ถัดไป */
/*5*/	<code>printf("x = %d y = %d" , x , y);</code>	/* แสดงข้อความ "x = %d y = %d" โดยแสดงค่า x ใน %d ตัวแรก และแสดงค่า y ใน %d ตัวที่ 2 */
/*6*/	<code>return 0;</code>	/* คืนค่า 0 ให้ฟังก์ชัน main โปรแกรมทำงานถูกต้องสมบูรณ์ */
/*7*/	<code>}</code>	/* สิ้นสุด block ของฟังก์ชัน main (สิ้นสุดโปรแกรม) */

หมายเหตุ คำสั่ง `scanf("%d %d" , &x , &y);` เป็นคำสั่งรับค่าจำนวนเต็ม 2 จำนวน จากแป้นพิมพ์ ไปเก็บไว้ในตัวแปร x และ y ตามลำดับ โดยป้อนค่าที่ 1 แล้วเว้นวรรค (space bar) แล้วจึงป้อนค่าที่ 2 เมื่อป้อนข้อมูลเสร็จแล้ว ต้องกดแป้น Enter เพื่อยืนยันข้อมูลและทำคำสั่งถัดไป

ผลการรันโปรแกรม (Output)

7 9↵

x = 7 y = 9

(↵ คือ กดปุ่ม Enter)

ตัวอย่างที่ 30 การจัดรูปแบบอื่น ๆ ในการรับค่าจำนวนเต็ม 2 จำนวน จากแป้นพิมพ์

บรรทัดที่	รหัสต้นฉบับ (Source Code)	อธิบายคำสั่ง
/*1*/	#include <stdio.h>	/* เป็นคำสั่งของตัวประมวลผลก่อนซี จะมีผลให้เพิ่ม stdio.h ซึ่งเป็น เพิ่มส่วนหัวที่ใช้เก็บรวบรวมคำสั่งเกี่ยวกับการรับและแสดงผลข้อมูล ถูกอ่านเข้ามาเพื่อประมวลผลร่วมกับโปรแกรมนี้ */
/*2*/	int main() {	/* ประกาศ main เป็นฟังก์ชันหลัก ชนิดจำนวนเต็ม และเริ่มต้น block ของฟังก์ชัน main (เริ่มต้นโปรแกรม) */
/*3*/	int x , y;	/* ประกาศตัวแปร x และ y เป็นชนิดจำนวนเต็ม */
/*4*/	scanf("%d, %d" , &x , &y);	/* รับค่าเลขจำนวนเต็ม 2 จำนวน จากแป้นพิมพ์ ไปเก็บไว้ในตัวแปร x และ y ตามลำดับ คำนวณสองจำนวนด้วยเครื่องหมายคอมมา (,) ตามที่กำหนดไว้ในรูปแบบคำสั่ง เมื่อป้อนข้อมูลเสร็จแล้ว ต้องกด แป้น Enter เพื่อยืนยันข้อมูลและทำคำสั่งถัดไป */
/*5*/	printf("x = %d y = %d" , x , y);	/* แสดงข้อความ "x = %d y = %d" โดยแสดงค่า x ใน %d ตัวแรก และแสดงค่า y ใน %d ตัวที่ 2 */
/*6*/	return 0;	/* คืนค่า 0 ให้ฟังก์ชัน main โปรแกรมทำงานถูกต้องสมบูรณ์ */
/*7*/	}	/* สิ้นสุด block ของฟังก์ชัน main (สิ้นสุดโปรแกรม) */

ผลการรันโปรแกรม (Output)

7,9↵

x = 7 y = 9

(↵ คือ กดปุ่ม Enter)

ตัวอย่างที่ 31 การรับค่า 3 ค่า ต่างชนิดกัน จากแป้นพิมพ์ วิธีที่ 1

บรรทัดที่	รหัสต้นฉบับ (Source Code)	อธิบายคำสั่ง
/*1*/	#include <stdio.h>	/* เป็นคำสั่งของตัวประมวลผลก่อนซี จะมีผลให้เพิ่ม stdio.h ซึ่งเป็น เพิ่มส่วนหัวที่ใช้เก็บรวบรวมคำสั่งเกี่ยวกับการรับและแสดงผลข้อมูล ถูกอ่านเข้ามาเพื่อประมวลผลร่วมกับโปรแกรมนี้ */
/*2*/	int main() {	/* ประกาศ main เป็นฟังก์ชันหลัก ชนิดจำนวนเต็ม และเริ่มต้น block ของฟังก์ชัน main (เริ่มต้นโปรแกรม) */
/*3*/	char x;	/* ประกาศตัวแปร x เป็นชนิดอักขระ */
/*4*/	int y;	/* ประกาศตัวแปร y เป็นชนิดจำนวนเต็ม */
/*5*/	float z;	/* ประกาศตัวแปร z เป็นชนิดจำนวนจริง */
/*6*/	scanf("%c" , &x);	/* รับค่าอักขระ จากแป้นพิมพ์ ไปเก็บไว้ในตัวแปร x เมื่อป้อนข้อมูล เสร็จแล้ว ต้องกดแป้น Enter เพื่อยืนยันข้อมูลและทำคำสั่งถัดไป */
/*7*/	scanf("%d" , &y);	/* รับค่าเลขจำนวนเต็ม จากแป้นพิมพ์ ไปเก็บไว้ในตัวแปร y เมื่อป้อน ข้อมูลเสร็จแล้ว ต้องกดแป้น Enter เพื่อยืนยันข้อมูลและทำคำสั่ง ถัดไป */
/*8*/	scanf("%f" , &z);	/* รับค่าเลขจำนวนจริง จากแป้นพิมพ์ ไปเก็บไว้ในตัวแปร z เมื่อป้อน ข้อมูลเสร็จแล้ว ต้องกดแป้น Enter เพื่อยืนยันข้อมูลและทำคำสั่ง ถัดไป */
/*9*/	printf("x = %c y = %d z = %f" , x , y , z);	/* แสดงข้อความ "x = %c y = %d z = %f" โดยแสดงค่า x ใน %c แสดงค่า y ใน %d และแสดงค่า z ใน %f */
/*10*/	return 0;	/* คืนค่า 0 ให้ฟังก์ชัน main โปรแกรมทำงานถูกต้องสมบูรณ์ */
/*11*/	}	/* สิ้นสุด block ของฟังก์ชัน main (สิ้นสุดโปรแกรม) */

ผลการรันโปรแกรม (Output)

A↵

45↵

12.45↵

x = A y = 45 z = 12.450000

(↵ คือ กดปุ่ม Enter)

ตัวอย่างที่ 32 การรับค่า 3 ค่า ต่างชนิดกัน จากแป้นพิมพ์ วิธีที่ 2

บรรทัดที่	รหัสต้นฉบับ (Source Code)	อธิบายคำสั่ง
/*1*/	#include <stdio.h>	/* เป็นคำสั่งของตัวประมวลผลก่อนซี จะมีผลให้เพิ่ม stdio.h ซึ่งเป็น เพิ่มส่วนหัวที่ใช้เก็บรวบรวมคำสั่งเกี่ยวกับการรับและแสดงผลข้อมูล ถูกอ่านเข้ามาเพื่อประมวลผลร่วมกับโปรแกรมนี้ */
/*2*/	int main() {	/* ประกาศ main เป็นฟังก์ชันหลัก ชนิดจำนวนเต็ม และเริ่มต้น block ของฟังก์ชัน main (เริ่มต้นโปรแกรม) */
/*3*/	char x;	/* ประกาศตัวแปร x เป็นชนิดอักขระ */
/*4*/	int y;	/* ประกาศตัวแปร y เป็นชนิดจำนวนเต็ม */
/*5*/	float z;	/* ประกาศตัวแปร z เป็นชนิดจำนวนจริง */
/*6*/	scanf("%c %d %f", &x , &y , &z);	/* รับค่า อักขระ จำนวนเต็ม และจำนวนจริง จากแป้นพิมพ์ ไปเก็บไว้ใน ตัวแปร x y และ z ตามลำดับ แต่ละค่าคั่นด้วยเว้นวรรค (space bar) เมื่อป้อนข้อมูลเสร็จทั้ง 3 ค่าแล้ว ต้องกดแป้น Enter เพื่อยืนยัน ข้อมูลและทำคำสั่งถัดไป */
/*7*/	printf("x = %c y = %d z = %f", x , y , z);	/* แสดงข้อความ "x = %c y = %d z = %f" โดยแสดงค่า x ใน %c แสดงค่า y ใน %d และแสดงค่า z ใน %f */
/*8*/	return 0;	/* คืนค่า 0 ให้ฟังก์ชัน main โปรแกรมทำงานถูกต้องสมบูรณ์ */
/*9*/	}	/* สิ้นสุด block ของฟังก์ชัน main (สิ้นสุดโปรแกรม) */

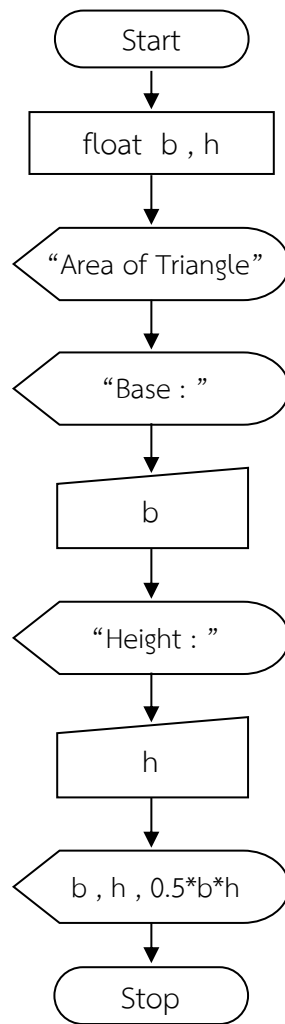
ผลการรันโปรแกรม (Output)

A 45 12.45↵

x = A y = 45 z = 12.450000

(↵ คือ กดปุ่ม Enter)

ตัวอย่างที่ 33 จงเขียนโปรแกรมคำนวณหาพื้นที่รูปสามเหลี่ยม
จากโจทย์ เขียนผังงาน แสดงการแก้ปัญหา ได้ดังนี้



จากผังงาน เขียนโปรแกรมได้ดังนี้

บรรทัดที่	รหัสต้นฉบับ (Source Code)	อธิบายคำสั่ง
/*1*/	#include <stdio.h>	/* เป็นคำสั่งของตัวประมวลผลก่อนซี จะมีผลให้เพิ่ม stdio.h ซึ่งเป็น เพิ่มส่วนหัวที่ใช้เก็บรวบรวมคำสั่งเกี่ยวกับการรับและแสดงผลข้อมูล ถูกอ่านเข้ามาเพื่อประมวลผลร่วมกับโปรแกรมนี้ */
/*2*/	int main() {	/* ประกาศ main เป็นฟังก์ชันหลัก ชนิดจำนวนเต็ม และเริ่มต้น block ของฟังก์ชัน main (เริ่มต้นโปรแกรม) */
/*3*/	float b , h;	/* ประกาศตัวแปร b และ h เป็นชนิดจำนวนจริง */
/*4*/	printf("Area of Triangle\n");	/* แสดงข้อความ "Area of Triangle" แล้วขึ้นบรรทัดใหม่ */
/*5*/	printf("Base : ");	/* แสดงข้อความ "Base : " เพื่อแจ้งให้ผู้ใช้โปรแกรมทราบว่า ต้อง ป้อนข้อมูลอะไร */
/*6*/	scanf("%f" , &b);	/* รับค่าจำนวนจริง จากแป้นพิมพ์ ไปเก็บไว้ในตัวแปร b เมื่อป้อน ข้อมูลเสร็จแล้ว ต้องกดแป้น Enter เพื่อยืนยันข้อมูลและทำคำสั่ง ถัดไป */
/*7*/	printf("Height : ");	/* แสดงข้อความ "Height : " เพื่อแจ้งให้ผู้ใช้โปรแกรมทราบว่า ต้อง ป้อนข้อมูลอะไร */
/*8*/	scanf("%f" , &h);	/* รับค่าจำนวนจริง จากแป้นพิมพ์ ไปเก็บไว้ในตัวแปร h เมื่อป้อน ข้อมูลเสร็จแล้ว ต้องกดแป้น Enter เพื่อยืนยันข้อมูลและทำคำสั่ง ถัดไป */
/*9*/	printf("Triangle : Base = %.2f Height = %.2f Area = %.2f", b,h,0.5*b*h);	/* แสดงข้อความ "Triangle :Base =%.2f Height =%.2f Area =%.2f" โดยแสดงค่า b ใน %.2f ตัวแรก แสดงค่า h ใน %.2f ตัวที่ 2 และแสดงค่า 0.5*b*h ใน %.2f ตัวที่ 3 (%.2f คือ แสดงผลเป็นจุด ทศนิยม 2 ตำแหน่ง) */
/*10/	return 0;	/* คืนค่า 0 ให้ฟังก์ชัน main โปรแกรมทำงานถูกต้องสมบูรณ์ */
/*11*/	}	/* สิ้นสุด block ของฟังก์ชัน main (สิ้นสุดโปรแกรม) */

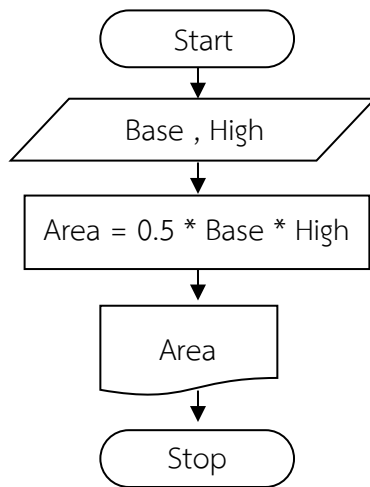
หมายเหตุ แสดงการใช้คำสั่ง printf กับ scanf คู่กัน โดยใช้คำสั่ง printf เพื่อแสดงข้อความ ให้ผู้ใช้โปรแกรมทราบว่า ถึงขั้นตอนไหนและต้องทำอะไร (เป็นการสื่อสารกับผู้ใช้โปรแกรม) แล้วจึงใช้คำสั่ง scanf ให้ผู้ใช้ป้อนข้อมูล

ผลการรันโปรแกรม (Output)

```
Area of Triangle
Base : 4↵
Height : 5↵
Triangle : Base = 4.00 Height = 5.00 Area = 10.00
```

(↵ คือ กดปุ่ม Enter)

ตัวอย่างที่ 34 จงเขียนโปรแกรมรับค่าจำนวนเต็มความยาวฐานและความสูงของรูปสามเหลี่ยม คำนวณหาพื้นที่รูปสามเหลี่ยม และแสดงผลพื้นที่รูปสามเหลี่ยมทางจอภาพ

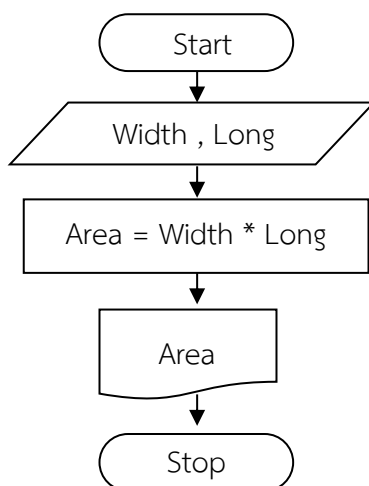


```
#include <stdio.h>
int main() {
    int Base , High ;
    float Area;
    scanf ("%d" , &Base);
    scanf ("%d" , &High);
    Area = 0.5 * Base * High;
    printf("Area = %.2f" , Area);
    return 0;
}
```

ผลการรันโปรแกรม

```
4↵
10↵
Area = 20.00
```

ตัวอย่างที่ 35 จงเขียนโปรแกรมรับค่าจำนวนเต็มความกว้างและความยาวของรูปสี่เหลี่ยม คำนวณหาพื้นที่รูปสี่เหลี่ยม และแสดงผลพื้นที่รูปสี่เหลี่ยมออกทางจอภาพ

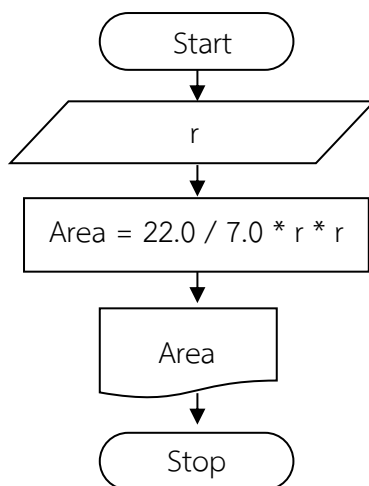


```
#include <stdio.h>
int main() {
    int Width, Long, Area ;
    scanf ("%d" , &Width);
    scanf ("%d" , &Long);
    Area = Width * Long;
    printf("Area = %d" , Area );
    return 0;
}
```

ผลการรันโปรแกรม

```
10↵
20↵
Area = 200
```

ตัวอย่างที่ 36 จงเขียนโปรแกรมรับค่าจำนวนเต็มที่เป็นความยาวรัศมี คำนวณหาพื้นที่รูปวงกลม และแสดงผลพื้นที่รูปวงกลมออกทางจอภาพ

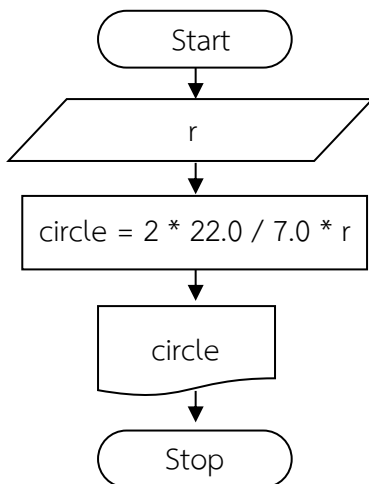


```
#include <stdio.h>
int main() {
    int r ;
    float Area ;
    scanf ("%d" , &r );
    Area = 22.0/7.0 * r * r ;
    printf("Area = %.2f" , Area );
    return 0;
}
```

ผลการรันโปรแกรม

7↵
Area = 154.00

ตัวอย่างที่ 37 จงเขียนโปรแกรมรับค่าจำนวนเต็มที่เป็นความยาวรัศมี คำนวณหาความยาวของเส้นรอบรูปวงกลม และแสดงผลความยาวของเส้นรอบรูปวงกลมออกทางจอภาพ



```
#include <stdio.h>
int main() {
    int r ;
    float circle ;
    scanf ("%d" , &r );
    circle = 2*22.0/7.0*r ;
    printf("circle = %.2f" , circle );
    return 0;
}
```

ผลการรันโปรแกรม

7↵
circle = 44.00

การเขียนโปรแกรมที่ถือว่าเป็นพื้นฐานที่สุด คือ การเขียนโปรแกรมแสดงผลทางหน้าจอและการรับข้อมูลทางแป้นพิมพ์เข้าไปประมวลผล ที่ผ่านมาได้ศึกษาฟังก์ชันการรับและแสดงผลข้อมูลพื้นฐานแล้ว จะพบว่าฟังก์ชัน printf() สามารถใช้แสดงผลข้อมูลที่เป็นอักขระ ข้อความ หรือตัวเลขได้ พร้อมทั้งใส่รหัสควบคุมต่าง ๆ ลงไปด้วยได้ ฟังก์ชัน scanf() สามารถใช้รับข้อมูลทางแป้นพิมพ์ได้ทั้งตัวเลข และตัวอักขระ แต่โดยทั่วไปแล้วฟังก์ชันนี้นิยมใช้รับตัวเลขเท่านั้น การรับข้อมูลแบบอื่น ๆ มักจะใช้ฟังก์ชันอื่นแทน ในการเขียนโปรแกรมภาษาซี ยังมีฟังก์ชันแสดงผลและรับข้อมูลที่น่าสนใจอีกดังนี้

ฟังก์ชันแสดงผล

คำสั่ง putchar()

เป็นคำสั่งที่ใช้สำหรับแสดงตัวอักขระตัวเดียวออกทางหน้าจอ

รูปแบบ

```
putchar(ch);
```

ความหมาย

ch ตัวอักษรหรือตัวอักขระเขียนอยู่ภายในเครื่องหมาย ' ' (Single Quote) หรือตัวแปรชนิด char

ตัวอย่าง เช่น

```
#include <stdio.h>
int main()
{
    char m = 'T';
    char letter = 'z';
    putchar(m);           //แสดงค่าของตัวแปร m   ออกทางจอภาพ ซึ่งก็คือตัวอักษร T
    putchar(letter);      //แสดงค่าของตัวแปร letter ออกทางจอภาพ ซึ่งก็คือตัวอักษร z
    putchar('Q');         //แสดงตัวอักษร Q ทางจอภาพ
    putchar('K');         //แสดงตัวอักษร K ทางจอภาพ
    return 0;
}
```

Output

TzQK

คำสั่ง puts()

เป็นคำสั่งที่ใช้สำหรับแสดงข้อความหรือสายอักขระออกทางจอภาพ

รูปแบบ

```
puts(str);
```

ความหมาย

str ข้อความที่อยู่ภายในเครื่องหมาย " " (Double Quote) หรือตัวแปรที่เก็บข้อมูลชนิดข้อความหรือสายอักขระหรือตัวแปรชุดหรือตัวแปรสตริง

ตัวอย่าง เช่น

```
#include <stdio.h>
int main()
{
    char name[10] = "Nangrong";
    char sch[7] = "School";
    puts(name);           //แสดงข้อความในตัวแปร name ออกทางจอภาพ ซึ่งก็คือคำว่า Nangrong
    puts(sch);            //แสดงข้อความในตัวแปร sch   ออกทางจอภาพ ซึ่งก็คือคำว่า School
    printf("\n");
    puts("Computer");     //แสดงข้อความ Computer ทางจอภาพ
    puts("Room");         //แสดงข้อความ Room ทางจอภาพ
    return 0;
}
```

Output

Nangrong
School

Computer
Room

ฟังก์ชันรับข้อมูล

คำสั่ง getchar()

เป็นคำสั่งที่ใช้สำหรับรับข้อมูลตัวอักขระหนึ่งตัวทางแป้นพิมพ์ เมื่อป้อนข้อมูลแล้วต้องกดปุ่ม Enter

โปรแกรมจึงจะทำงานต่อไปได้

รูปแบบ

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

ความหมาย

ch ตัวแปรชนิด char เพื่อใช้เก็บค่าของอักขระที่รับเข้ามาจากคำสั่ง getchar() และจะแสดงอักขระนั้นออกทางจอภาพ แล้วจึงทำงานต่อไป

ตัวอย่าง เช่น

```
#include <stdio.h>
int main()
{
    char ch;
    printf("Enter your answer (y/n) : "); //แสดงข้อความให้ป้อนตัวอักษร y หรือ n
    ch = getchar();                      //รับตัวอักษรที่ป้อนจากแป้นพิมพ์ มาเก็บไว้ในตัวแปร ch
    printf("Good Bye");                  //แสดงข้อความ Good Bye ทางจอภาพ
    return 0;
}
```

Output

Enter your answer (y/n) : **y** ↵ //ป้อนตัวอักษร y หรือ n แล้วกดปุ่ม Enter จึงทำงานต่อไป
Good Bye

(↵ คือ กดปุ่ม Enter)

เมื่อพิมพ์ตัวอักษร 1 ตัว แล้วกดปุ่ม Enter ตัวอักษรนั้นจะเก็บไว้ในตัวแปร ch และจะแสดงตัวอักษรนั้นทางจอภาพด้วย แล้วจึงทำงานต่อไป

คำสั่ง getch()

เป็นคำสั่งที่ใช้สำหรับรับข้อมูลตัวอักษรหนึ่งตัวทางแป้นพิมพ์ เมื่อป้อนข้อมูลแล้วไม่ต้องกดปุ่ม Enter โปรแกรมจะทำงานต่อไป

รูปแบบ

```
ch = getch();
```

ความหมาย

ch ตัวแปรชนิด char เพื่อใช้เก็บค่าของอักขระที่รับเข้ามาจากคำสั่ง getch() และจะไม่แสดงอักขระนั้นออกทางจอภาพ จึงทำงานต่อไป

ตัวอย่าง เช่น

```
#include <stdio.h>
int main()
{
    char ch;
    printf("Enter your answer (y/n) : "); //แสดงข้อความให้ป้อนตัวอักษร y หรือ n
    ch = getch();                        //รับตัวอักษรที่ป้อนจากแป้นพิมพ์ มาเก็บไว้ในตัวแปร ch
    printf("\nGood Bye");                //แสดงข้อความ Good Bye ทางจอภาพ
    return 0;
}
```

Output

Enter your answer (y/n) : **_** //ป้อนตัวอักษร y หรือ n ไม่ต้องกดปุ่ม Enter จะทำงานต่อไป
Good Bye

เมื่อพิมพ์ตัวอักษร 1 ตัว **ไม่ต้องกดปุ่ม Enter** ตัวอักษรนั้นจะเก็บไว้ในตัวแปร ch และจะไม่แสดงตัวอักษรนั้นทางจอภาพ แล้วจึงทำงานต่อไป

คำสั่ง gets()

เป็นคำสั่งที่ใช้สำหรับรับข้อความทางแป้นพิมพ์ เข้ามาเก็บไว้ในตัวแปร

รูปแบบ

```
gets(str);
```

ความหมาย

str ตัวแปรสำหรับเก็บข้อความที่ป้อนทางแป้นพิมพ์ ซึ่งจะต้องประกาศตัวแปร **str** เป็นตัวแปรชนิดข้อความหรือสายอักขระหรือตัวแปรชุดหรือตัวแปรสตริงไว้ก่อน

ตัวอย่าง เช่น

```
#include <stdio.h>
int main()
{
    char str[50];
    printf("Enter province name : "); //แสดงข้อความให้ป้อนตัวอักษร y หรือ n
    gets(str);                       //รับข้อความที่ป้อนจากแป้นพิมพ์ มาเก็บไว้ในตัวแปร str
    printf("You stay in %s", str);    //แสดงข้อความทางจอภาพ
    return 0;
}
```

Output

```
Enter province name : Buriram↵ //ป้อนข้อความ แล้วกดปุ่ม Enter จะทำงานต่อไป
You stay in Buriram
( ↵ คือ กดปุ่ม Enter)
```

เปรียบเทียบการทำงานระหว่างฟังก์ชัน gets() และ scanf()

```
#include <stdio.h>
int main() {
    char name[50] ;
    printf("What is your name ? \n" );
    gets(name);
    printf("Hi %s", name);
    return 0;
}
```

ผลการรันโปรแกรม

```
What is your name ?
Peeraphong Meepuangphol↵
Hi Peeraphong Meepuangphol
```

```
#include <stdio.h>
int main() {
    char name[50] ;
    printf("What is your name ? \n" );
    scanf("%s", name);
    printf("Hi %s" , name);
    return 0;
}
```

ผลการรันโปรแกรม

```
What is your name ?
Peeraphong Meepuangphol↵
Hi Peeraphong
```

คำสั่ง gets() จะรับข้อความทั้งหมดรวมถึงช่องว่างด้วยจนกระทั่งมีการกดปุ่ม Enter จึงหยุดรับค่า ดังนั้นผลลัพธ์ของโปรแกรมทางด้านซ้ายมือ จึงแสดงทั้งชื่อและนามสกุลออกมาได้ ต่างจากฟังก์ชัน scanf() ด้านขวาจะรับค่าข้อความมาเมื่ออ่านพบช่องว่างจะตัดข้อความหลังช่องว่างออก และฟังก์ชัน scanf() ยังไม่มีการเติม '\0' เพื่อปิดสตริงให้ด้วย

ใบงานหน่วยที่ 5

1. จับคู่รหัสรูปแบบข้อมูลและรหัสควบคุมการแสดงผลกับความหมายทางด้านขวามือ โดยนำตัวอักษรที่อยู่หน้าความหมายทางด้านขวามือ มาเติมในช่องว่างหน้าชื่อทางด้านซ้ายให้สอดคล้องกัน

คำตอบ ข้อ	รหัสในภาษาซี	ความหมาย
..... 1)	\n	ก. char และ unsigned char
..... 2)	\\	ข. float และ double
..... 3)	%f	ค. int และ unsigned int
..... 4)	\"	ง. char[] หรือ สายอักขระ
..... 5)	\t	จ. ขึ้นบรรทัดใหม่ 1 บรรทัด
..... 6)	%d	ฉ. tab เว้นวรรค 1 tab (8 อักขระ)
..... 7)	\a	ช. ส่งเสียงบี๊บเตือน 1 ครั้ง
..... 8)	%c	ซ. แสดงอักขระ ' 1 ตัว
..... 9)	\'	ฅ. แสดงอักขระ " 1 ตัว
..... 10)	%s	ญ. แสดงอักขระ \ 1 ตัว

2. จงเติมรหัสรูปแบบข้อมูลในคำสั่ง printf ให้สอดคล้องกับชนิดตัวแปรที่กำหนดให้

```
int main() {
    int i = 5 , j = 3 ;
    float x = 50.50 , y = 100.99;
    char a = 'A' , b = 'B';
    1) printf("i = %.....\n" , i );
    2) printf("j = %..... x = %.....\n" , j , x );
    3) printf("a = %..... b = %.....\n" , a , b );
    4) printf("i - 3 = %..... x * 2 = %..... b = %..... \n" , i - 3 , x*2 , b );
    return 0;
}
```

3. จงเติมรหัสรูปแบบข้อมูลในคำสั่ง scanf ให้สอดคล้องกับชนิดตัวแปรที่กำหนดให้

```
int main() {
    int i , j ;
    float x , y ;
    char a , b ;
    1) scanf("%....." , &i );
    2) scanf("%..... %....." , &x , &a);
    3) scanf("%..... %....." , &a , &b);
    4) scanf("%..... %..... %....." , &i , &y , &j );
    return 0;
}
```

4. จงเขียนโปรแกรมคำนวณหาพื้นที่สี่เหลี่ยมผืนผ้า โดยกำหนดตัวแปร Width และ Hight เป็นชนิดจำนวนเต็ม แทนด้านของสี่เหลี่ยมผืนผ้า ความกว้างและความยาว ตามลำดับ และตัวแปร Area เป็นชนิดจำนวนเต็ม แทนพื้นที่สี่เหลี่ยมผืนผ้า สูตรในการคำนวณหาพื้นที่สี่เหลี่ยมผืนผ้า คือ ความกว้าง คูณ ความยาว ($\text{Area} = \text{Width} * \text{Hight}$) โดยกำหนดให้รับค่าความกว้าง (Width) และความยาว (Hight) ทางแป้นพิมพ์ แสดงผลพื้นที่ (Area) ทางจอภาพดังนี้

ผลการรันโปรแกรม

Width : 4
Hight : 5
Area of Rectangle : 20

ตัวอย่างแสดงผลลัพธ์ เมื่อป้อนความกว้าง เป็น 4 และความยาว เป็น 5 จะแสดงผลพื้นที่ เป็น 20

รหัสคำสั่ง (Source Code)

```
#include <stdio.h>

int main() {
    ..... Width , Hight , Area;           //ประกาศตัวแปร
    printf("Width : ");                     //แสดงข้อความแจ้งผู้ใช้ให้ป้อนข้อมูล
    .....;                                  //รับข้อมูล Width
    printf("Hight : ");                     //แสดงข้อความแจ้งผู้ใช้ให้ป้อนข้อมูล
    .....;                                  //รับข้อมูล Hight
    .....;                                  //คำนวณพื้นที่สี่เหลี่ยมผืนผ้า (จากสูตร)
    .....;                                  //แสดงผลพื้นที่สี่เหลี่ยมผืนผ้า

    return 0;
}
```

5. จงเขียนโปรแกรมรับค่าตัวเลขทศนิยมสองตำแหน่ง 5 จำนวน แล้วให้แสดงผลทศนิยมที่ตรงกันและผลรวมของจำนวนทั้งหมดดังนี้

Input	Output
7.89	7.89
67.89	67.89
567.89	567.89
4567.89	4567.89
34567.89	34567.89
	39779.45
5.67	5.67
45.67	45.67
345.67	345.67
2345.67	2345.67
12345.67	12345.67
	15088.35

6. จงเขียนโปรแกรมรับค่าปี พ.ศ. แล้วแสดงผลลัพธ์เป็นปี ค.ศ.

Input	Output
2566	2023
2542	1999

7. จงเขียนโปรแกรมรับค่าอักขระ 1 ตัว แล้วนำมาแสดงผลเป็นรหัสแอสกีของอักขระนั้น

Input	Output
Enter character : r	Ascii code of r is 114
Enter character : A	Ascii code of A is 65

8. จงเขียนโปรแกรมรับค่าตัวอักษร 5 ตัว แล้วให้ทำการแปลงตัวอักษรตัวที่ 2 และ 4 เป็นตัวอักษรตัวถัดไป (เช่น a อักษรถัดไปคือ b , t อักษรถัดไปคือ u , y อักษรถัดไปคือ z)

Input	Output
Enter 5 character : Table	Tbbme
Enter 5 character : Horse	Hprte
Enter 5 character : Sheep	Siefp
Enter 5 character : Shark	Siask

9. จงเขียนโปรแกรมคำนวณอายุ โดยรับข้อมูลชื่อ (Name) ปีที่เกิด (Birth year) และปีปัจจุบัน (Current year) ทางแป้นพิมพ์ แล้วแสดงผลดังนี้

Input	Output
Name : Manee Birth of Year : 2551 Current Year : 2566	Manee is 15 years old.
Name : Boss Birth of Year : 2542 Current Year : 2561	Boss is 19 years old.

10. จงเขียนโปรแกรมคำนวณอายุ โดยรับข้อมูลชื่อ-สกุล (Name) ทางแป้นพิมพ์ แล้วแสดงผลดังนี้

Input	Output
What 's your name : Manee Meeta	Hi Manee Meeta , nice to meet you.
What 's your name : Steve Jobs	Hi Steve Jobs , nice to meet you.
What 's your name : Bill Gates	Hi Bill Gates , nice to meet you.

11. จงเขียนโปรแกรมรับค่าจำนวนเต็ม x แล้วคำนวณหาผลลัพธ์จากสมการ $x^2 + 8x + 4$

Input	Output
9	157
15	349
4	52

12. จงเขียนโปรแกรมแก้สมการหาค่า x จากสมการ $Ax^2 + Bx + C = 0$ และแสดงค่า x (รับค่า A, B, C)

Input	Output
1 4 3	-1 -3
1 -4 3	1 3

$$ax^2 + bx + c = 0$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

13. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนนาฬิกา ที่เป็นตัวเลขจำนวนเต็มที่มีมากกว่าศูนย์ แล้วคำนวณเปลี่ยนตัวเลขค่านาฬิกาที่รับเข้ามา ให้แสดงผลเป็นจำนวนวัน จำนวนชั่วโมง และจำนวนนาฬิกา ตัวอย่าง เช่น กรอกตัวเลข 500 นาฬิกา ให้แสดงผล 0 8 20 (0 วัน 8 ชั่วโมง 20 นาฬิกา) , กรอกตัวเลข 4000 นาฬิกา ให้แสดงผล 2 18 40 (2 วัน 18 ชั่วโมง 40 นาฬิกา)

Input	Output
4000	2 18 40
500	0 8 20

14. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนพื้นที่ที่ดินหน่วยเป็นตารางวา ที่เป็นตัวเลขจำนวนจริงที่มากกว่าศูนย์ แล้วคำนวณเปลี่ยนตัวเลขที่รับเข้ามา ให้แสดงผลเป็นจำนวนไร่ จำนวนงาน และจำนวนตารางวา ตัวอย่าง เช่น กรอกเลข 2289 ตารางวา ให้แสดงผล 5 2 89 (5 ไร่ 2 งาน 89 ตรว.) , กรอกเลข 520.5 ตารางวา ให้แสดงผล 1 1 20.5 (1 ไร่ 1 งาน 20.5 ตรว.)

Input	Output
1140	2 3 40.0
520.5	1 1 20.5
2289	5 2 89.0

1 ไร่ เท่ากับ 4 งาน

1 งาน เท่ากับ 100 ตรว.

15. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็ม 1 จำนวน กรณีตู้ ATM ให้นักเรียนเขียนโปรแกรมคำนวณหาว่า เมื่อมีผู้มาถอนเงินตามจำนวนนั้น ให้แจ้งว่าจะได้รับเงินธนบัตรใบละ 1000 บาท จำนวนกี่ใบ ธนบัตรใบละ 500 บาท จำนวนกี่ใบ และธนบัตรใบละ 100 บาท จำนวนกี่ใบ โดยเรียงจากธนบัตรมูลค่าที่มากที่สุดเรียงมาน้อยสุด (1000 , 500 , 100)

ตัวอย่าง เช่น กรอกตัวเลข 800 บาท ให้แสดงผลลัพธ์ 0 1 3 (หมายถึง จะได้รับเงินธนบัตรใบละ 1000 บาท จำนวน 0 ใบ ธนบัตรใบละ 500 บาท จำนวน 1 ใบ และธนบัตรใบละ 100 บาท จำนวน 3 ใบ) หรือ

กรอกตัวเลข 35600 บาท ให้แสดงผลลัพธ์ 35 1 1 (หมายถึง จะได้รับเงินธนบัตรใบละ 1000 บาท จำนวน 35 ใบ ธนบัตรใบละ 500 บาท จำนวน 1 ใบ และธนบัตรใบละ 100 บาท จำนวน 1 ใบ)

Input	Output
35600	35 1 1
800	0 1 3

โจทย์เพิ่มเติม

15.1 จงเขียนโปรแกรมเช็คค่าผู้ใช้งานกรอกจำนวนเงินที่มีเศษเงินน้อยกว่าธนบัตร 100 บาท หรือเป็นยอดเงินที่ไม่ลงท้ายด้วย 00 หรือไม่ ซึ่งผิดเงื่อนไขการให้บริการตู้เอทีเอ็มของธนาคาร และให้แสดงข้อความแจ้งเตือนว่า “You must enter for 1000 or 500 or 100 Baht”

15.2 จงเขียนโปรแกรมเช็คค่าผู้ใช้งานกรอกจำนวนเงินเกิน 25000 บาทหรือไม่ ซึ่งผิดเงื่อนไขการให้บริการตู้เอทีเอ็มของธนาคารต่อการกดเงินหนึ่งครั้ง และให้แสดงข้อความแจ้งเตือนว่า “You must enter between 100-25000 Baht”

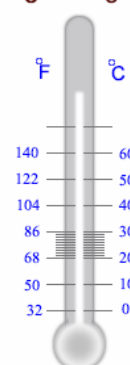
16. จงเขียนโปรแกรมรับค่าอุณหภูมิองศาเซลเซียสแล้วแปลงเป็นองศาฟาเรนไฮต์

Input	Output
45	113.00
37	98.60
40	104.00

17. จงเขียนโปรแกรมรับค่าอุณหภูมิองศาฟาเรนไฮต์แล้วแปลงเป็นองศาเซลเซียส

Input	Output
37.5	3.055556
45.25	7.361111
32.71	0.394444

$$\frac{C}{5} = \frac{F - 32}{9}$$



$$C = (F - 32) / 9$$

$$F = (9C + (32 * 5)) / 5$$

18. จงเขียนโปรแกรมเพื่อรับค่ารัศมีของวงกลม (radius) ผ่านทางแป้นพิมพ์ จากนั้นคำนวณหาพื้นที่ของวงกลม (Area) และความยาวเส้นรอบวงกลม (Circumference) โดยมีสูตรในการคำนวณดังนี้

➤ $\text{Area} = \pi * \text{radius} * \text{radius}$

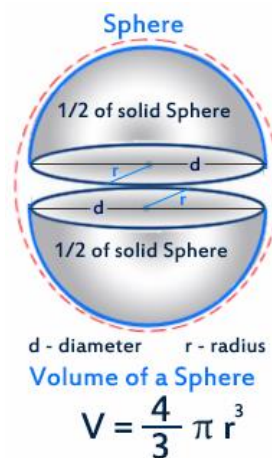
➤ $\text{Circumference} = 2 * \pi * \text{radius}$

โดยกำหนดให้ pi เป็นค่าคงที่มีค่าเท่ากับ 3.14159

Input	Output
3.2	Area of Circle = 32.17 Circumference of Circle = 20.11
12.25	Area of Circle = 471.43 Circumference of Circle = 76.97

19. จงเขียนโปรแกรมเพื่อรับค่ารัศมีของวงกลม (radius) ผ่านทางแป้นพิมพ์ จากนั้นคำนวณหาปริมาตรของวงกลม (Area) โดยกำหนดให้ pi เป็นค่าคงที่มีค่าเท่ากับ 3.14159

Input	Output
2.56	70.276176
5.75	796.327637



20. จงเขียนโปรแกรมรับค่าชั่วโมง (Hour) และนาที (Minute) ผ่านทางแป้นพิมพ์ แล้วคำนวณหาเวลารวมเป็นนาทีทั้งหมด

Input	Output
Hour : 5 Minute : 37	Total 337 minutes.
Hour : 9 Minute : 40	Total 580 minutes.

21. จงเขียนโปรแกรมรับค่านาทีรวม (Minute) ผ่านทางแป้นพิมพ์ แล้วคำนวณหาว่ามีกี่ชั่วโมง (Hour) และกี่นาที (Minute)

Input	Output
546	9 6
4000	66 40

22. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็ม 2 จำนวน โดยการรับค่าให้เว้นวรรค 1 เคาะระหว่างเลข 2 จำนวน แล้วคำนวณหาผลรวม ผลต่าง ผลคูณ ผลหาร ของตัวเลขสองจำนวนนั้น

Input	Output
10 5	10 + 5 = 15 10 - 5 = 5 10 * 5 = 50 10 / 5 = 2.00
24 4	24 + 4 = 28 24 - 4 = 20 24 * 4 = 96 24 / 4 = 6.00

23. จงเขียนโปรแกรมรับค่ามุม 2 มุมของรูปสามเหลี่ยมทางแป้นพิมพ์ แล้วให้คำนวณหามุมที่สามของสามเหลี่ยมนั้น

Input	Output
angles one : 50 angles two : 70	60
angles one : 60 angles two : 60	60

24. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็ม 3 จำนวน แล้วคำนวณหาผลรวม และค่าเฉลี่ยของตัวเลขสามจำนวนนั้น

Input	Output
10 5 35	Sum = 50 Average = 16.666666
21 18 15	Sum = 54 Average = 18.000000

25. จงเขียนโปรแกรมเพื่อรับค่าเส้นผ่านศูนย์กลางของวงกลม (Diameter) ผ่านทางแป้นพิมพ์ จากนั้นคำนวณหาพื้นที่ของวงกลม (Area) และความยาวเส้นรอบวงกลม (Circumference) โดยกำหนดให้ pi มีค่าเท่ากับ 22.0/7.0

Input	Output
14	Area of Circle = 154.00 Circumference of Circle = 44.00
120.25	Area of Circle = 11361.48 Circumference of Circle = 377.93

26. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มแทนระยะทางที่มีหน่วยเป็นไมล์ แล้วคำนวณแปลงค่าที่รับเข้ามาให้มีหน่วยเป็นกิโลเมตร (กำหนดให้ 1 ไมล์ เท่ากับ 1.6 กิโลเมตร)

Input	Output
93	148.80
155	248.00
26	41.60

27. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มแทนจำนวนเงินที่มีหน่วยเป็นดอลลาร์ (USD) แล้วคำนวณอัตราแลกเปลี่ยนจากค่าที่รับเข้ามาให้มีหน่วยเป็นบาท (กำหนดให้ 1 USD เท่ากับ 35.60 BTH)

Input	Output
55	1958.00
172	6123.20
350	12460.00

28. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มฐานสิบ แล้วคำนวณแปลงค่าที่รับเข้ามาให้แสดงเป็นเลขฐานแปด (0...) และเลขฐานสิบหก (0x...)

Input	Output
1247	02337 0x4df
435	0663 0x1b3
99	0143 0x63

29. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มบวก 1 จำนวน ที่มีค่าระหว่าง 100 ถึง 999 จากนั้นให้แสดงผลการกลับค่าตัวเลข (Reversed number) จากหลังไปหน้า และให้แสดงผลการคูณของตัวเลขโดดสามจำนวนนั้น

Input	Output
123	321 6
456	654 120
284	482 64

30. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มบวก 1 จำนวน ที่มีค่าระหว่าง 100 ถึง 999 จากนั้นให้แสดงผลการกลับค่าตัวเลข (Reversed number) จากหลังไปหน้า และให้แสดงผลการคูณจำนวนนั้น กับ 3

Input	Output
123	321 963
456	654 1962
284	482 1446

31. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเงินสะสม และอัตราเงินปันผลเป็นเปอร์เซ็นต์ แล้วให้คำนวณหาจำนวนเงินปันผล และยอดเงินรวมทั้งหมดที่ได้

Input	Output
100 12	12.00 112.00
5000 6.5	325.00 5325.00
850 4.7	39.95 889.95

32. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนชั่วโมงที่เรียนพิเศษ (Hour) และจำนวนเงินที่ชำระค่าเรียนพิเศษ (Pay) แล้วคำนวณหาค่าเรียนพิเศษรวม (Total) และเงินทอน (Change) แสดงผลทางจอภาพ (กำหนดให้ ค่าเรียนชั่วโมงละ 350 บาท)

Input	Output
5 2000	Hour : 5 Total : 1750 Pay : 2000 Change : 250
1 500	Hour : 1 Total : 350 Pay : 500 Change : 150
3 1100	Hour : 3 Total : 1050 Pay : 1100 Change : 50

33. นักเรียนเข้าไปซื้อสินค้าในร้านค้าแห่งหนึ่ง เจ้าของร้านคิดราคารวมสินค้าทั้งหมดที่นักเรียนซื้อ (Buy) และรับเงินที่จากนักเรียนจ่าย (Pay) ถ้าในลิ้นชักของร้านค้า มีธนบัตรใบละ 20 บาท เหรียญ 10 บาท เหรียญ 5 บาท และเหรียญ 1 บาท เท่านั้น ให้นักเรียนเขียนโปรแกรมแสดงผลจำนวนเงินทอน (Change) จำนวนธนบัตรใบละ 20 บาท จำนวนเหรียญเหรียญ 10 บาท จำนวนเหรียญเหรียญ 5 บาท และจำนวนเหรียญเหรียญ 1 บาทที่จะได้รับ

Input	Output
117 200	Change : 83 4 0 0 3
242 500	Change : 258 12 1 1 3
216 250	Change : 34 1 1 0 4

34. จงเขียนโปรแกรมแสดงผลลัพธ์ดังภาพ

```

c C C C C c      c C C c      c C c      C C      c C C c      c C C C c
C C      C C      C C      C C      c C C c      C C      c C c
C C      C C      C C      C C      C C      C C      c C c
C C C      C C      C C      c C C C C c      C C      c C C C c

```

ตัวอย่าง

	c	C	C	C	C	c			c	C	C	c			c	C	c			C	C				c	C	C	c			c	C	C	C	c
	C	C			C	C				C	C				C	C		c	C	C	c					C	C			c	C	c			
	C	C			C	C				C	C				C	C										C	C			c	C	c			
C	C	C			C	C				C	C				c	C	C	C	C	C	c						C	C			c	C	C	C	c