

ฟังก์ชัน

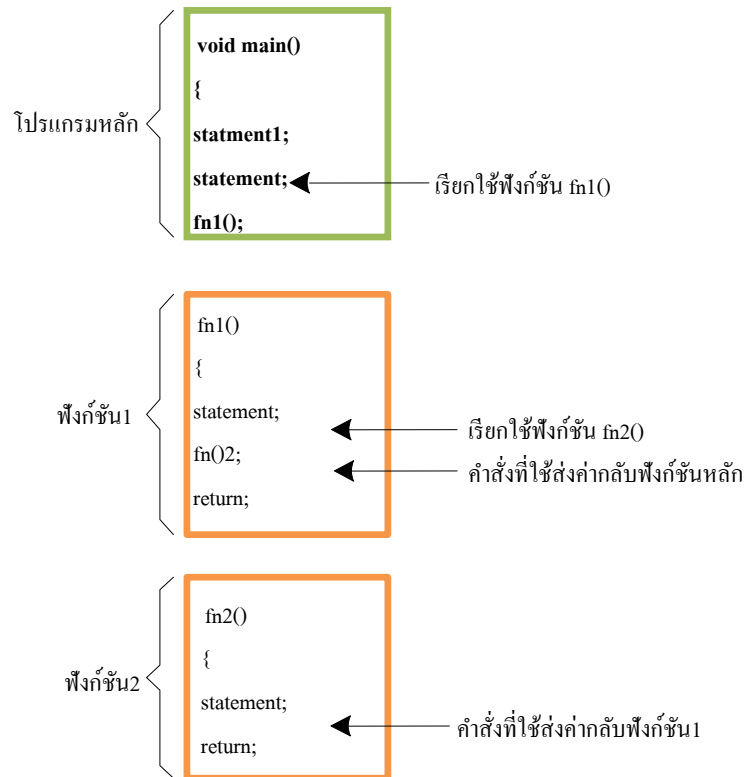
ฟังก์ชัน โดยความหมายทั่วไปหมายถึงหน้าที่ที่ได้รับผิดชอบให้ทำงานให้สำเร็จ และรายงานผลการดำเนินงาน (ศรีธัช อินทโกสุม, 2539) ส่วนฟังก์ชันของโปรแกรมหมายถึงโปรแกรมย่อย (subprograms) ที่เขียนมาให้นำไปใช้ประโยชน์และนำไปงานในการแก้ปัญหาทางสำเร็จสมบูรณ์ภายในตัวเอง ถ้าเทียบกับโปรแกรมภาษาอื่นมีลักษณะเหมือนกับโปรแกรมย่อยหรือสับรูทีน (subroutine) ด้วยเนื่องจากภาษาซี มีการออกแบบโปรแกรมให้อำนวยความสะดวกให้กับผู้ใช้หรือผู้เขียนโปรแกรมให้สามารถทำงานต่าง ๆ ได้อย่างรวดเร็ว และประหยัดเนื้อที่ในหน่วยความจำ ภาษาซีมีฟังก์ชันเตรียมไว้ให้ผู้ใช้เรียกใช้ได้มากมาย (Steven และ Lutfar, 2006) (Brian W. K., Online)

ฟังก์ชันในภาษาซีสามารถแบ่งออกเป็น 2 ประเภท คือ

1. ฟังก์ชันที่ผู้ใช้โปรแกรมกำหนดขึ้น (user defined function)
2. ฟังก์ชันไลบรารี (library function)

ฟังก์ชันที่ผู้ใช้โปรแกรมกำหนดขึ้น (User defined function)

การเขียนฟังก์ชันที่ผู้ใช้โปรแกรมกำหนดขึ้น สามารถเขียนฟังก์ชันโดยฟังก์ชันที่เขียนเป็นส่วนหนึ่งของโปรแกรมหลัก (main()) และทำการบันทึกเป็นไฟล์เดียวกัน กับการเขียนฟังก์ชันมาใช้งาน โดยเก็บฟังก์ชันไว้เป็นชื่อไฟล์อื่น ซึ่งฟังก์ชันที่ผู้ใช้เขียนขึ้นมาใช้งานจะมีลักษณะเหมือนกับโปรแกรมย่อยทั่วไป ลักษณะเดียวกับภาษาอื่น คือฟังก์ชันที่เขียนขึ้นสามารถทำการส่งค่า หรือรับค่า รูปที่ 8.1



รูปที่ 8.1 แสดงการทำงานของ การเรียกฟังก์ชันและการส่งค่ากลับ

รูปแบบ

type FnName();

type

ชนิดของฟังก์ชันที่ส่งกลับค่ากลับไปยังฟังก์ชันที่เรียกใช้งาน สามารถกำหนดชนิดของฟังก์ชันได้ถ้าไม่กำหนดค่าที่ส่งกลับ จะเป็นชนิดจำนวนเต็ม(integer)

FnName() ชื่อฟังก์ชัน()

ตัวอย่างเช่น

void display();

หมายถึง การประกาศฟังก์ชันชื่อ display()
เป็นฟังก์ชันชนิดที่ไม่มีการส่งค่ากลับ(void)

float calculate();

หมายถึง การประกาศฟังก์ชันชื่อ calculate()
เป็นฟังก์ชันชนิดที่มีการส่งค่ากลับเป็นชนิดทศนิยม (float)

ตัวอย่าง โปรแกรมที่ 8.1 เป็น โปรแกรมการเรียกใช้ฟังก์ชันย่อยชื่อ display()

```
#include<stdio.h>
#include<conio.h>
void display();                      //ประกาศฟังก์ชันชื่อ display() ไม่มีการส่งค่ากลับ
(void)
void main()
{
clrscr();
printf("Display Function Main\n");
```

ต่อจากด้านบน

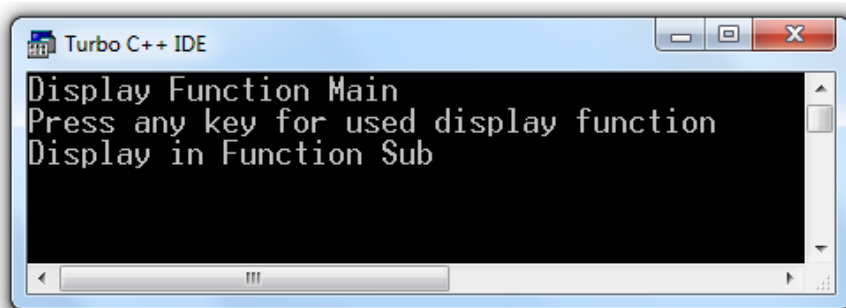
```

printf("Press any key for used display function\n");
getch();           //รอรับการกดคีย์ใด ๆ ของผู้ใช้โปรแกรม
display();         //การเรียกใช้ฟังก์ชัน display()
}                 //จบฟังก์ชันหลัก

void display()     //ตัวฟังก์ชันย่อยชื่อ display
{
    printf("Display in Function Sub");
    getch();
}                 //จบฟังก์ชันย่อย

```

ผลการทำงานโปรแกรม



```

Turbo C++ IDE
Display Function Main
Press any key for used display function
Display in Function Sub

```

การส่งผ่านตัวแปรในฟังก์ชันต่างๆ ค่าข้อมูลที่เก็บไว้ในตัวแปรสามารถส่งค่าที่ถูกระบุหรือค่าที่เก็บผลการคำนวณในแต่ละฟังก์ชันไปยังฟังก์ชันต่างๆ ได้ 2 ลักษณะคือ ตัวแปรแบบโกลบอล(global) และตัวแปรแบบโลคอล(local) ตัวแปรแบบโกลบอล เป็นตัวแปรที่มีการประกาศใช้ก่อนฟังก์ชัน main() ซึ่งค่าเริ่มต้นของตัวแปรจะมีค่าเท่ากับศูนย์อัตโนมัติ ตัวแปรแบบโกลบอลนี้ทุกฟังก์ชันสามารถใช้ร่วมกันได้โดยไม่ต้องประกาศตัวแปรในฟังก์ชันที่เรียกใช้อีก ส่วนตัวแปรแบบโลคอล เป็นตัวแปรที่มีการประกาศหลังฟังก์ชัน main() ซึ่งตัวแปรโลคอลจะสามารถใช้ได้เฉพาะฟังก์ชันที่มีการประกาศตัวแปรเท่านั้น ฟังก์ชันที่ไม่ได้ประกาศจะไม่สามารถงานใช้ได้

การส่งค่าผ่านฟังก์ชัน

การส่งค่าผ่านฟังก์ชัน หมายถึง การส่งค่าจากฟังก์ชันหนึ่งไปยังอีกฟังก์ชันหนึ่งซึ่งจะต้องกำหนดตัวแปรหรือฟังก์ชันไว้ในการรับค่า การส่งค่าอาจส่งได้ค่าเดียวหรือหลายค่าก็ได้ การส่งค่าผ่านฟังก์ชันจะช่วยทำให้การเขียนโปรแกรมง่ายขึ้น สั้นลงและง่ายในการแก้ไขโปรแกรม ซึ่งสามารถศึกษาวิธีการเขียนโปรแกรมการประยุกต์ใช้งานได้ในบทต่อไป

ตัวอย่าง โปรแกรมที่ 8.2 เป็นโปรแกรมสูตรคูณแบบส่งค่าผ่านฟังก์ชัน

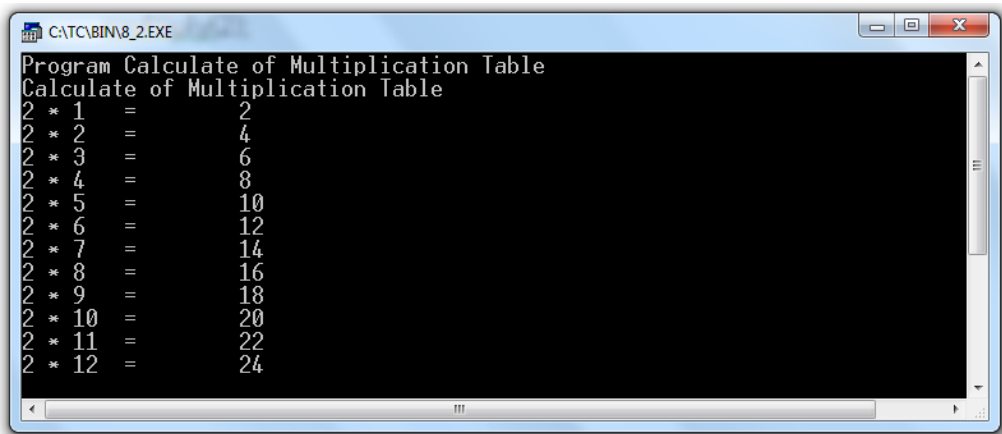
```
#include<stdio.h>
#include<conio.h>
int multi(int n);
void main()
{
    clrscr();
    printf("Program Calculate of Multiplication Table\n");
    multiply(2);
    multiply(12);
    getch();
}
* Sub Program Calculate of Multiplication Table */
int multi(int n)
{
    int i;
    for(i=1;i<13;i++);
        printf("Calculate of Multiplication Table\n");
        printf("%.2d \t*\t %.2d \t=\t %.2d\n",n,i,n*i);
    getch();
}
```

จากโปรแกรม 8_2 มีการประกาศฟังก์ชัน `int multi(int n)` เป็นฟังก์ชันที่มีการส่งค่าตัวแปรชนิดเลขจำนวนเต็มขนาด 16 บิต (`int n`) โดยตัวแปร `n` เป็นใช้ในการรับค่า

ในการเรียกใช้ฟังก์ชัน `multi(2)` ที่อยู่ในส่วนของตัวโปรแกรมหลัก `void main()` กำหนดค่าที่ต้องการส่งผ่านตัวแปร `n` มีค่าเท่ากับ 2 ที่อยู่ในวงเล็บจะถูกส่งให้กับฟังก์ชัน `int multi(int n)` ใช้ในการคำนวณ

ตัวฟังก์ชัน `int multi(int n)` จะแสดงผลการคำนวณสูตรคูณมาตรา 2 ด้วยเพราะมีการส่งค่า `n=2` กลับไปยังฟังก์ชันหลัก `void main()` ดังผลการทำงานโปรแกรมด้านล่าง

ผลการทำงานโปรแกรม



```

CATC\BIN\8_2.EXE
Program Calculate of Multiplication Table
Calculate of Multiplication Table
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18
2 * 10 = 20
2 * 11 = 22
2 * 12 = 24

```

ตัวอย่าง โปรแกรมที่ 8.3 โปรแกรมส่งค่าผ่านฟังก์ชันแบบหลายค่า

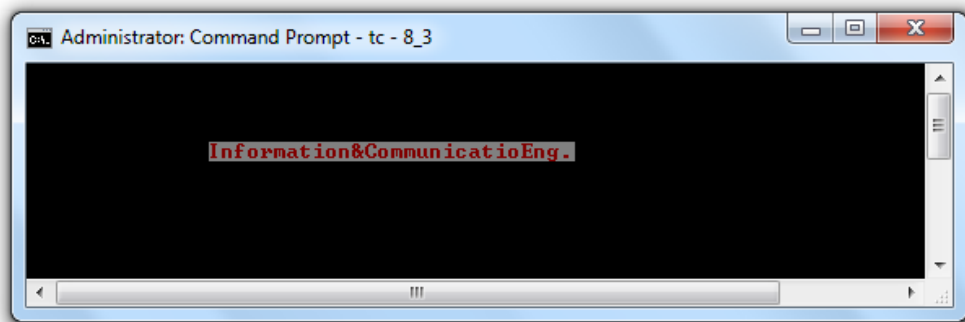
```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<dos.h>
char show(char col, char row);
void main()
{
    clrscr();
    show(15,5);
    getch();
}
char show(char c, char r)
{
    do {
        gotoxy(c, r);
        textcolor(random(16)|128);
        printf("Information&CommunicatioEng.");
        delay(5000);
    } while(!kbhit());
    return 0;
}
```


จากโปรแกรม 8.3 มีการประกาศฟังก์ชัน `char show(char col, char row)` โดยมีตัวแปร `col` และ `row` เป็นตัวแปรชนิด `char` (character) มีขนาด 8 บิต เพื่อใช้ในการรับค่า

ในฟังก์ชัน `void main()` เรียกใช้ฟังก์ชัน `show(col, row)` และทำการส่งค่า 15 และ 5 ให้กับฟังก์ชัน `char show(char col, char row)` ในฟังก์ชัน `char show(char c, char r)` จะรับค่า 15 มาเก็บไว้ที่ `c` และ 5 มาเก็บไว้ที่ `r`

ใน ฟังก์ชัน `char show(char c, char r)` แสดงข้อความ `Information & CommunicationEng.` ตามตำแหน่งกำหนดให้กับหน้าจอกอมพิวเตอร์ด้วยค่าในตัวแปรของ `c` และ `r` โดยที่ข้อความ จะกะพริบและเปลี่ยนสีแบบสลับไปเรื่อย ๆ จนกว่าจะมีการกดแป้นพิมพ์ใด ๆ จึงออกจากโปรแกรม

ผลการทำงานโปรแกรม



ฟังก์ชันไลบรารี (library function)

โปรแกรมในภาษาซีได้ใช้ฟังก์ชัน (function) เป็นหลักในการเขียนโปรแกรม ตัวโปรแกรมหลัก(main()) ก็เป็นฟังก์ชัน ภาษาซีเป็นภาษาสร้างขึ้นมาเพื่ออำนวยความสะดวกแก่ผู้เขียนโปรแกรมให้สามารถทำงานด้วยความรวดเร็วและประหยัดเนื้อที่ในหน่วยความจำ โดยภาษาซีได้เตรียมฟังก์ชันไว้ให้ผู้เขียนโปรแกรมได้เรียกใช้ตามลักษณะงานที่ต้องการได้อย่างครบถ้วน และภาษาซียังเปิดโอกาสให้ผู้เขียนโปรแกรมสามารถเขียนฟังก์ชันขึ้นมาใช้เองได้ ดังนั้นฟังก์ชันที่ภาษาซีได้จัดเตรียมไว้ให้ เป็นฟังก์ชันพื้นฐานที่จำเป็นต้องเรียกใช้เสมอ เช่น ฟังก์ชัน printf(), scanf(), getchar(), getch() putchar() และ putch() เป็นต้น ซึ่งต่างก็เป็นโปรแกรมย่อยของระบบภาษาซี สำหรับฟังก์ชัน printf() และ scanf() เป็นฟังก์ชันมาตรฐานที่สามารถใช้ได้กับซีคอมไพเลอร์ของผู้ผลิตทุกบริษัท ฟังก์ชัน ในภาษาซี ทำหน้าที่ให้การเขียนโปรแกรมเป็นแบบโครงสร้างสามารถอ่านโปรแกรมทำความเข้าใจได้ง่าย สะดวกต่อตรวจหาจุดผิดพลาด และง่ายต่อการปรับปรุงโปรแกรม

ฟังก์ชันที่อยู่ในไลบรารี เปรียบเสมือนเป็นฟังก์ชันสำเร็จรูปที่ผู้ผลิตภาษาซี สร้างไว้ให้ใช้ตามมาตรฐาน ไลบรารีฟังก์ชันจะเก็บอยู่ในไฟล์ที่มีนามสกุล .h หรือที่เรียกว่าไฟล์ header โดยจัดแบ่งตามประเภทการใช้งาน เช่น ไลบรารีฟังก์ชันที่ต้องเรียกใช้เสมอ stdio.h กับ conio.h ไลบรารีฟังก์ชันเกี่ยวกับการคำนวณทางคณิตศาสตร์จะเก็บอยู่ในไฟล์ math.h ส่วนไลบรารีฟังก์ชันเกี่ยวกับการจัดการข้อความจะถูกเก็บใน string.h เป็นต้น

การเรียกใช้ไลบรารีฟังก์ชันใด มีรูปแบบการเรียกใช้ด้วยคำสั่ง `#include<ไฟล์ header>` ของฟังก์ชันนั้นไว้ที่ส่วนต้นหรือส่วนหัวของโปรแกรมก่อนฟังก์ชันหลัก (`main()`) เสมอ

ฟังก์ชันมาตรฐาน

ฟังก์ชันที่เรียกใช้อยู่เสมอในการเขียน โปรแกรม อยู่ในไลบรารี `stdio.h` มีดังตาราง 8.1

ตาราง 8.1 แสดงฟังก์ชันในไลบรารี `stdio.h`

ชื่อฟังก์ชัน	ความหมาย
<code>printf()</code>	พิมพ์ค่าข้อมูลไปยังหน้าจอคอมพิวเตอร์
<code>putchar()</code>	แสดงผลทางจอภาพครั้งละ 1 อักขระ
<code>getchar()</code>	ป้อนตัวอักษรผ่านทางแป้นพิมพ์โดยจะรับตัวอักษรเพียง 1 ตัวเท่านั้น และแสดงตัวอักษรบนจอภาพ
<code>gets()</code>	ใช้สำหรับรับข้อมูลชนิดสตริงหรือข้อความทางแป้นพิมพ์
<code>getch();</code>	รอรับข้อมูลจากแป้นพิมพ์ 1 ตัวอักษร

ฟังก์ชัน `printf()`

เป็นฟังก์ชันใช้ พิมพ์ค่าข้อมูลไปยังหน้าจอคอมพิวเตอร์

รูปแบบคำสั่ง

`printf("control string", variable list);`

`control string` อาจจะเป็นตัวอักษร ข้อความหรือตัวกำหนดชนิดข้อมูล (specified) ซึ่งใช้กำหนดชนิดข้อมูลที่จะพิมพ์ ตัวกำหนดชนิดข้อมูลมีดังนี้

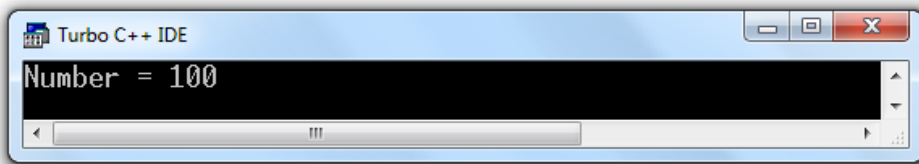
%c	แทนตัวอักษร
%d	แทนเลขจำนวนเต็ม
%e	แทนเลขในรูปเอกซ์โพเนนเชียล (exponential form)
%f	แทนเลขทศนิยม
%o	แทนเลขฐานแปด
%s	แทนสตริงก์
%u	แทนเลขจำนวนเต็มไม่ติดเครื่องหมาย
%x	แทนเลขฐานสิบหก
%p	แทนข้อมูลแบบพอยน์เตอร์ (pointer)
%%	แทนเครื่องหมาย %

ตัวอย่าง โปรแกรมที่ 8.4

```
#include <stdio.h>
#include <conio.h>
void main(void)
{
    int n;
    n=100;
    printf("Number = %d",n);
    getch();
}
```

จากโปรแกรม 8.4 ประกาศตัวแปร n เป็นชนิดเลขจำนวนเต็ม กำหนดค่า n มีค่าเท่ากับ 100 เรียกใช้ฟังก์ชันมาตรฐาน `printf()` โดยในส่วนของ control string เป็นข้อความ `Number =` และมีตัวกำหนดชนิดข้อมูล `%d` ทำหน้าที่ในการแสดงค่าข้อมูลที่ตัวแปร n เก็บอยู่ให้แสดงหน้าจอภาพเป็นเลขจำนวนเต็ม ตามการประกาศชนิดข้อมูล ส่วนฟังก์ชัน `getch()` เป็นการรอรับข้อมูลจากแป้นพิมพ์ 1 ตัวอักษร เพื่อเป็นการให้โปรแกรมแสดงผลยังคงแสดงอยู่ จนกว่าผู้ใช้โปรแกรมต้องการออกจากโปรแกรม สามารถกระทำได้โดยการกดแป้นพิมพ์ใดๆ

ผลการทำงาน โปรแกรม



ตัวอย่าง โปรแกรมที่ 8.5

```
#include <stdio.h>      #include<conio.h>
void main(void)
{
    clrscr();
    char c= 'A';    int x=10;
    printf("Value is = %c\n",c);
    printf("Value is = %d\n",c);
    printf("Value is = %x\n",x);
    getch();      }
```

จากโปรแกรม 8.5 ได้เรียกใช้ฟังก์ชันที่น่าสนใจ clrscr() เป็นฟังก์ชันที่อยู่ในไลบรารี conio.h ทำหน้าที่ ล้างข้อมูลหน้าจอภาพที่แสดงก่อนหน้าที่โปรแกรม 8.5 จะทำการแสดงการประกาศตัวแปร c เป็นชนิดข้อมูลตัวอักษร เก็บตัวอักษร A ส่วนตัวแปร x เป็นตัวแปรชนิดตัวเลขจำนวนเต็ม เก็บค่า 10 ซึ่งเป็นเลขฐานสิบ

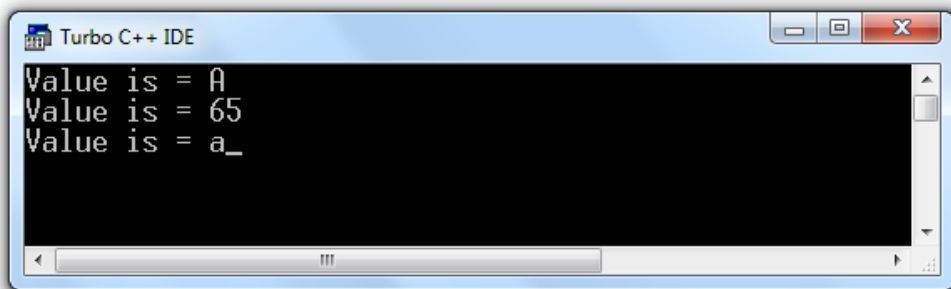
ในการแสดงผลการทำงานของโปรแกรมทำการเรียกฟังก์ชัน printf() โดยที่มีการกำหนด control string 3 รูปแบบ

printf("Value is = %c\n",c); เป็นการให้แสดงค่าตัวแปร c ในรูปตัวอักษร 1 ตัวคือ A

printf("Value is = %d\n",c); เป็นการให้แสดงค่าตัวแปร c เป็นเลขจำนวนเต็มเลขฐาน 10 1 ตัวคือ 65 (ค่า A ตามตาราง ASCII มีค่าเท่ากับ 65)

printf("Value is = %x\n",x); เป็นการให้แสดงค่าตัวแปร x ในรูปเลขฐานสิบหกของ ซึ่ง 10 เลขฐานสิบมีค่าเท่ากับเลขฐานสิบหกคือ a

ผลการทำงานโปรแกรม



```
Turbo C++ IDE
Value is = A
Value is = 65
Value is = a_
```

ฟังก์ชันเกี่ยวกับตัวอักษร

ฟังก์ชันเกี่ยวกับตัวอักษรในการเขียนโปรแกรม อยู่ในไลบรารี ctype.h ตาราง 8.2

ตาราง 8.2 แสดงฟังก์ชันในไลบรารี ctype.h

ชื่อฟังก์ชัน	ความหมาย
isalnum(x)	ฟังก์ชันนี้จะให้ค่า ไม่เท่ากับศูนย์ ถ้า x มีค่าเป็นตัวอักษรหรือตัวเลข และจะให้ค่าเป็นศูนย์ ถ้า x ไม่ได้มีค่าเป็นตัวอักษรหรือตัวเลข
isalpha(x)	ฟังก์ชันนี้จะให้ค่า ไม่เท่ากับศูนย์ ถ้า x มีค่าเป็นตัวอักษร และจะให้ค่าเป็นศูนย์ถ้า x ไม่ได้เป็นตัวอักษรหรือเครื่องหมายต่าง ๆ
isctrl(x)	ฟังก์ชันนี้จะให้ค่า ไม่เท่ากับศูนย์ ถ้า x มีค่ารหัสแอสกี ตั้งแต่ 0 ถึง 1F (เลขฐาน 16) หรือมีค่ารหัสแอสกี 7F (DEL หรือ Delete) รหัสแอสกี ดังกล่าวจัดเป็นตัวอักษรที่ไม่สามารถพิมพ์ได้ (unprintable character) และจะให้ค่าเป็นศูนย์ถ้า x ไม่มีค่าในรหัสแอสกีดังกล่าว
isdigit(x)	ฟังก์ชันนี้จะให้ค่า ไม่เท่ากับศูนย์ถ้า x มีค่าเป็นตัวเลข 0-9 และจะให้ค่าเป็นศูนย์ถ้า x ไม่มีค่าเป็นตัวเลข
isgraph(x)	ฟังก์ชันนี้จะให้ค่า ไม่เท่ากับศูนย์ ถ้า x มีค่าเป็นตัวอักษรที่สามารถพิมพ์ได้ (printable character) ยกเว้นช่องว่าง (space) ตัวอักษรที่สามารถพิมพ์ได้นี้จะมีการรหัสแอสกี ระหว่าง 21-7E กับค่าเลขฐานสิบ 33-126 และจะให้ค่าเป็นศูนย์ถ้า x ไม่มีค่าเป็นรหัสแอสกีดังกล่าว
islower(x)	ฟังก์ชันนี้จะให้ค่า ไม่เท่ากับศูนย์ ถ้า x มีค่าเป็นตัวอักษรซึ่งเป็นอักษรตัวพิมพ์เล็ก และจะให้ค่าเป็นศูนย์ ถ้า x มีค่าเป็นตัวอักษรตัวพิมพ์ใหญ่ เครื่องหมายต่าง ๆ

ชื่อฟังก์ชัน	ความหมาย
isupper(x)	ฟังก์ชันนี้จะให้ค่า ไม่เท่ากับศูนย์ ถ้า x มีค่าเป็นตัวอักษรซึ่งเป็นอักษรตัวพิมพ์ใหญ่และจะให้ค่าเป็นศูนย์ ถ้า x มีค่าเป็นตัวอักษรตัวพิมพ์เล็ก เครื่องหมายต่าง ๆ
isprint(x)	ฟังก์ชันนี้จะให้ค่า ไม่เท่ากับศูนย์ ถ้า x มีค่าเป็นตัวอักษรที่สามารถพิมพ์ได้รวมทั้งช่องว่าง ตัวอักษรที่สามารถพิมพ์ได้จะมีรหัสแอสกีระหว่าง 20-7E และจะให้ค่าเป็นศูนย์ ถ้า x ไม่ได้มีค่ารหัสแอสกีดังกล่าว
ispunct(x)	ฟังก์ชันนี้จะให้ค่า ไม่เท่ากับศูนย์ ถ้า x มีค่าเป็นเครื่องหมายวรรคตอน (punctuation character) ซึ่งในที่นี้ได้แก่เครื่องหมายต่าง ๆ ยกเว้น ตัวเลข ตัวอักษร (A-Z) และช่องว่าง และจะให้ค่าเป็นศูนย์ ถ้า x ไม่ได้มีค่าเครื่องหมายต่าง ๆ
isspace(x)	ฟังก์ชันนี้จะให้ค่า ไม่เท่ากับศูนย์ ถ้า x มีค่าเป็น space, tab, vertical tab, form feed, carriage return หรือ newline character ค่าต่าง ๆ เหล่านี้เรียกว่า whitespace และจะให้ค่าเป็นศูนย์ ถ้า x ไม่มีค่าดังกล่าว
isxdigit(x)	ฟังก์ชันนี้จะให้ค่า ไม่เท่ากับศูนย์ ถ้า x มีค่าเป็นตัวเลขในระบบเลขฐานสิบหก ซึ่งได้แก่เลข 0-9, A-F และ a-f และจะให้ค่าเป็นศูนย์ ถ้า x ไม่ได้มีค่าดังกล่าว
tolower(x)	ฟังก์ชันนี้จะทำการเปลี่ยนตัวอักษรซึ่งเก็บอยู่ในตัวแปร x เป็นตัวอักษรตัวพิมพ์เล็ก
toupper(x)	ฟังก์ชันนี้จะทำการเปลี่ยนตัวอักษรซึ่งเก็บอยู่ในตัวแปร x เป็นตัวอักษรตัวพิมพ์ใหญ่

ฟังก์ชันเกี่ยวกับสตริง

ฟังก์ชันเกี่ยวกับสตริงในการเขียน โปรแกรม อยู่ในไลบรารี string.h ดังตาราง 8.3 ดังนี้

ตาราง 8.3 แสดงฟังก์ชันในไลบรารี string.h

ชื่อฟังก์ชัน	ความหมาย
strcmp (str1,str2)	ฟังก์ชันนี้จะนำค่ารหัสแอสกีของสตริงทั้งสองตัวคือ str1 กับ str2 มาทำการเปรียบเทียบข้อมูลสตริงและจะให้ค่าของฟังก์ชันดังนี้คือน้อยกว่าศูนย์ ถ้าค่าใน str1 มีอันดับน้อยกว่า str2 เท่ากับศูนย์ ถ้าค่าใน str1 มีอันดับเท่ากับ str2 มากกว่าศูนย์ถ้าค่าใน str1 มีอันดับมากกว่า str2 โดยการกำหนดค่าใน str1 และ str2 อาจจะเป็นสตริงที่อยู่ในเครื่องหมาย " " หรืออยู่ในรูปอาร์เรย์
strcpy (str1,str2)	ฟังก์ชันนี้ทำการสำเนาข้อมูลโดยจะนำค่าของสตริง str2 ไปเก็บไว้ในสตริง str1 ซึ่ง str2 อาจเป็นค่าคงที่สตริงที่อยู่ในเครื่องหมาย " "หรืออยู่ในรูปอาร์เรย์
strlen(str)	ฟังก์ชันนี้จะทำการให้ค่าตัวเลขจำนวนเต็มบอกขนาดความยาวของสตริงซึ่งเก็บอยู่ในสตริง str โดย str อาจจะเป็นค่าคงที่สตริงซึ่งอยู่ในเครื่องหมาย " "หรืออยู่ในรูปอาร์เรย์
strcat(str1,str2)	ฟังก์ชันนี้ทำการจัดเชื่อมค่าในสตริง str1 กับ str2 โดยผลลัพธ์จะถูกเก็บในสตริง str1 ทั้งนี้ str1 จะต้องมีความยาวพอที่จะเก็บผลลัพธ์ที่ทำการนำผลรวมของสตริง str1 กับ str2 โดย str2 อาจจะเป็นค่าคงที่สตริงซึ่งอยู่ในเครื่องหมาย " " หรืออยู่ในรูปอาร์เรย์

จากหัวข้อ 2.3 และ 2.4 เป็นการนำตัวอักษรและสตริงมาทำการ ตรวจสอบค่าของข้อมูลที่อยู่ในตัวแปรแต่ละชนิด กับการทำการเปรียบเทียบค่าของสตริงรวมทั้งสามารถทำการสำเนาหรือนำค่าสตริงมาทำการเชื่อมต่อกัน ตามลำดับ การเขียนโปรแกรมนำหัวข้อดังกล่าวมาใช้งานสามารถทำได้ทั้งสองลักษณะคือ ค่าที่อยู่ในรูปของสตริง กับค่าที่อยู่ในรูปของอาร์เรย์ ต่างๆ

ฟังก์ชันการคำนวณทางคณิตศาสตร์

ฟังก์ชันการคำนวณทางคณิตศาสตร์ อยู่ในไลบรารี `math.h` ตาราง 8.4

ตาราง 8.4 แสดงฟังก์ชันในไลบรารี `math.h` ซึ่งมีดังนี้

ชื่อฟังก์ชัน	ความหมาย
<code>abort()</code>	ฟังก์ชันทำการยกเลิกการทำงานของโปรแกรมขณะนั้น โดยจะมีข้อความ "Abnormal program termination" แสดงหน้าจอ
<code>abs(x)</code>	ฟังก์ชันนี้ทำหน้าที่ในการกำหนดค่า x ให้เป็นค่าสัมบูรณ์ (หมายถึงค่าไม่ติดลบ) ซึ่งเป็นตัวแปรชนิดเลขจำนวนเต็ม (<code>int</code>)
<code>atof(str)</code>	ฟังก์ชันนี้ทำการในการเปลี่ยนสตริงในอาร์เรย์สตริง <code>str</code> ให้เป็นข้อมูลชนิดตัวเลขจำนวนจริง (<code>float</code>) มีความละเอียด 2 เท่า ซึ่งสามารถนำไปใช้งานทางคณิตศาสตร์กับตัวเลขจำนวนเต็มหรือทศนิยมส่วนค่าที่เป็นสตริง ตัวอักษรหรือสัญลักษณ์ เครื่องหมายต่าง ๆ จะไม่มีผลต่อการเปลี่ยนแปลงข้อมูล

ชื่อฟังก์ชัน	ความหมาย
atoi(str)	ฟังก์ชันนี้ทำการเปลี่ยนสตริงในอาร์เรย์สตริง str เป็นข้อมูลเลขจำนวนเต็ม(int) ซึ่งไม่สามารถนำไปคำนวณทางคณิตศาสตร์ได้ สตริงดังกล่าวสามารถนำมาใช้กับข้อมูลตัวอักษร สัญลักษณ์ หรือเครื่องหมายต่างๆ จะมีผลในการเปลี่ยนชนิดข้อมูล
atol(str)	ฟังก์ชันนี้ใช้ทำการเปลี่ยนสตริงในอาร์เรย์สตริง str เป็นข้อมูลชนิดตัวเลขจำนวนเต็มยาว(long) ซึ่งสามารถนำไปคำนวณทางคณิตศาสตร์ได้ สตริงดังกล่าวนอกจากจะมีตัวเลขแล้วอาจจะตามด้วยตัวอักษร สัญลักษณ์ หรือเครื่องหมายต่าง ๆ จะไม่มีผลต่อการเปลี่ยนชนิดข้อมูล
exit(status)	ฟังก์ชันนี้ทำการยกเลิกการทำงานของโปรแกรมในขณะนั้น ซึ่งไม่เหมือนกับการจบโปรแกรมตามปกติ status โดยปกติจะมีค่าเป็นศูนย์ แต่ถ้าเป็นการจบโปรแกรมไม่ปกติ ค่า status จะใช้แทนด้วยข้อผิดพลาดต่าง ๆ

สรุป

การเขียนโปรแกรม จะมีการเรียกใช้กลุ่มคำสั่งที่ซ้ำๆ กันอย่างสม่ำเสมอ ดังนั้น การกำหนดให้ชุดคำสั่งที่ซ้ำๆ มากำหนดเป็นโปรแกรมย่อย และทำการตั้งให้เป็นฟังก์ชัน เป็นการอำนวยความสะดวกในการเรียกชุดคำสั่งได้ โดยภาษาซีได้สร้างชุดคำสั่งที่เป็นประโยชน์อำนวยความสะดวกในการใช้งาน ด้วยการเรียกกลุ่มของฟังก์ชันในไลบรารี และถ้าผู้เขียนโปรแกรมต้องการสร้างฟังก์ชันขึ้นมาใหม่ สามารถทำได้ด้วย 3 ขั้นตอนดังนี้ ขั้นตอนแรก ต้องประกาศชื่อฟังก์ชันก่อน ต่อมาสร้างตัวฟังก์ชัน และขั้นตอนสุดท้าย การเรียกใช้ฟังก์ชัน ซึ่งการเรียกใช้ฟังก์ชันที่สร้างขึ้นมาใช้เองสามารถเรียกใช้ได้ทันที แต่มีความแตกต่างกับการเรียกใช้งาน จากไลบรารี ต้องทำการลิงค์เพิ่มข้อมูลด้วยคำสั่ง include ตามด้วยชื่อของไลบรารี หลังจากนั้นจึงสามารถเรียกใช้ฟังก์ชันได้

แบบฝึกหัดท้ายบทที่ 8

ตอนที่ 1 จงเติมคำหรือข้อความในช่องว่างต่อไปนี้ให้ถูกต้อง

1. จงเขียน โค้ดการคำนวณสมการ $z = 5^2 + 4x + 321$
.....
2. จากสมการ $z = 5^2 + 4x + 321$ กำหนดค่า x มีค่าเท่ากับ 25 จงเขียนใช้ฟังก์ชันในการแก้สมการ
.....
3. จงเขียน โค้ดทำการแปลงเลขฐานสิบให้เป็นเลขฐาน 16
.....
4. เมื่อฟังก์ชันรับค่าข้อมูล จากผู้งาน โปรแกรม 20 ค่าเลขฐานสิบ ทำการแปลงเลขฐานสิบให้เป็นเลขฐาน 16
.....
5. จงเขียน โค้ดเพื่อตรวจสอบว่าเลขที่รับเข้าทางแป้นพิมพ์เป็น จำนวนเฉพาะหรือไม่ (จำนวนเฉพาะ หมายถึง เลขจำนวนเต็มบวกที่มีค่ามากกว่า 1 นอกจาก 1 และตัวมันเองแล้วไม่มีเลขจำนวนใดหารลงตัว)
.....
6. เขียนในรูปของฟังก์ชันตรวจสอบว่าเลขที่รับเข้าทางแป้นพิมพ์เป็น จำนวนเฉพาะหรือไม่
.....
7. จงเขียนการรับค่าเลขเป็นตัวแปรชนิดจำนวนเต็มบวก เช่น 1234
.....

8. นำค่าที่รับเลขเป็นตัวแปรชนิดจำนวนเต็มบวก มาทำการสลับตำแหน่ง เช่น
`converse(2255) = 5522`

.....

9. เขียนอธิบายแนวคิดการนำฟังก์ชันการสลับตำแหน่ง

.....

10. เขียนอธิบายการทำงานฟังก์ชันการสลับตำแหน่ง

.....

ตอนที่ 2 จงทำเครื่องหมายกากบาท (x) ทับหน้าข้อที่ถูกต้องที่สุด

- ข้อใดเป็นการประกาศฟังก์ชัน (declare function)

ก. <code>func()</code> ;	ข. <code>char func()</code> ;
ค. <code>func() float</code> ;	ง. <code>func(12)</code> ;
- ข้อใดเป็นการเรียกฟังก์ชันมาใช้งานพร้อมส่งค่าให้กับฟังก์ชัน

ก. <code>func(){...}</code>	ข. <code>char func()</code> ;
ค. <code>func() float</code> ;	ง. <code>func(12)</code> ;
- ข้อใดเป็นตัวฟังก์ชันที่ต้องการสร้างโค้ดมาใช้งาน

ก. <code>func(){...}</code>	ข. <code>char func()</code> ;
ค. <code>func() float</code> ;	ง. <code>func(12)</code> ;
- ข้อใดเป็นการส่งค่าให้กับฟังก์ชันนำค่าที่เป็นเลขจำนวนเต็ม 2 จำนวน ไปคำนวณ

ก. <code>func(2)</code>	ข. <code>char func(int a = 2)</code> ;
ค. <code>func(10, 12)</code> ;	ง. <code>func(12)</code> ;
- ฟังก์ชันใดไม่ได้อยู่ในไลบรารีของภาษาซี

ก. <code>get()</code>	ข. <code>getchar()</code>
ค. <code>printf()</code>	ง. <code>scanf()</code>

ตอนที่ 3 จงทำการวิเคราะห์คำถามและทำการเขียนอธิบายคำตอบตามที่ผู้อ่านเข้าใจโดยยึดความถูกต้องของเนื้อหาประกอบการบรรยาย

1. จากโค้ดด้านล่างจงเขียนทำแก้ไขข้อผิดพลาดให้ถูกต้อง

```
#include <stdio.h>

void fn1 ( int )
{
    int x;
    printf("X is %d", a);
}

void main()
{
    int x = 10;
    fn1 ( x );
}
```

2. จากโค้ดด้านล่างจงเขียนทำแก้ไขข้อผิดพลาดให้ถูกต้อง

```
#include <stdio.h>

int main( )
{
    float x;
    printf("Enter number :");
    scanf("%d", &x);
}
```

```

void main()
{
    float v;
    v = main();
    printf("Value v is %d", v);
}

```

3. จากโค้ดด้านล่างจงเขียนทำแก้ไขข้อผิดพลาดให้ถูกต้อง

```

#include <stdio.h>
fnY( int , int )
void fnZ()
void main()
{
    int x=10;
    fnY(x);
    x = fnZ(x+2);
}

fnA()
{
    fnY(x);
    printf("\nx = %d", x);
    return(x);
}

```



```

void fnY()
{
    printf("\nx = %d", x);
}

void fnZ( int )
{
    printf("\nx = %d", x);
}

```

4. จากโค้ดด้านล่างจงเขียนผลการทำงานโปรแกรม

```

#include <stdio.h>
int t=10;
void fn1(int t)
{
    int x;
    x = t*t;
    printf("\nx2= %d, T = %d", x, t); }
int fn2()
{
    int x;
    printf("\n x4 = %d, T = %d", x, t);
    x = t*t;
    printf("\n x5 = %d, T = %d", x, t);
    return(x); }

```

```
void main()
{
    int x=50;
    printf("\n x1 = %d, T = %d", x, t);
    fn1(x);
    printf("\n x3 = %d, T = %d", x, t);
    x = fn2();
    printf("\n x6 = %d, T = %d", x, t);
}
```

5. จงเขียนโปรแกรมทำการคำนวณหาผลลัพธ์ของสมการ $x = 2ab + c$ โดยผู้อ่านต้องเขียนฟังก์ชันในการรับข้อมูลเลขจำนวนเต็มจากแป้นพิมพ์ทั้งสามตัวแปร (a, b, และ C) และเขียนผลการทำงานของโปรแกรม ด้วยการทดลองกำหนดค่าตัวแปรทั้ง 3 ตัวให้ถูกต้อง อย่างน้อย 1 ชุด
-

เอกสารอ้างอิง

- ศรัณย์ อินทโกสุม (2539). ทฤษฎีและตัวอย่างโจทย์การเขียนโปรแกรมด้วยภาษาซี
กรุงเทพฯ : แมคกรอฮิล อินเทอร์เน็ต เนชั่นแนล เอ็นเตอร์ไพรส์, ینگค์.
- ชันวา ศรีประโมง (2539). การเขียนโปรแกรมภาษาซีสำหรับวิศวกรรม. พิมพ์ครั้งที่ 4. กรุงเทพฯ : มหาวิทยาลัยเทคโนโลยีมหานคร.
- วิจักขณ์ ศรีดีจะเลิศาจา และคุณผู้ ประเสริฐฐิติพงษ์ ออนไลน์ :
www.satit.su.ac.th/soottin.
- Brian, W. K. Programming in C: A Tutorial Online :
<http://www.lysator.liu.se/c/bwktutor.html>.
- Byron S. Gottfried, “Schaum ’s Theory and problems of programming with c”
McGraw-Hill,Inc., 1990.
- Steven, H. & Lutfar, R. (2006). Art of Programming Contest: C Programming |
Data Structure | Algorithms (ACM supported), 2nd Edition.