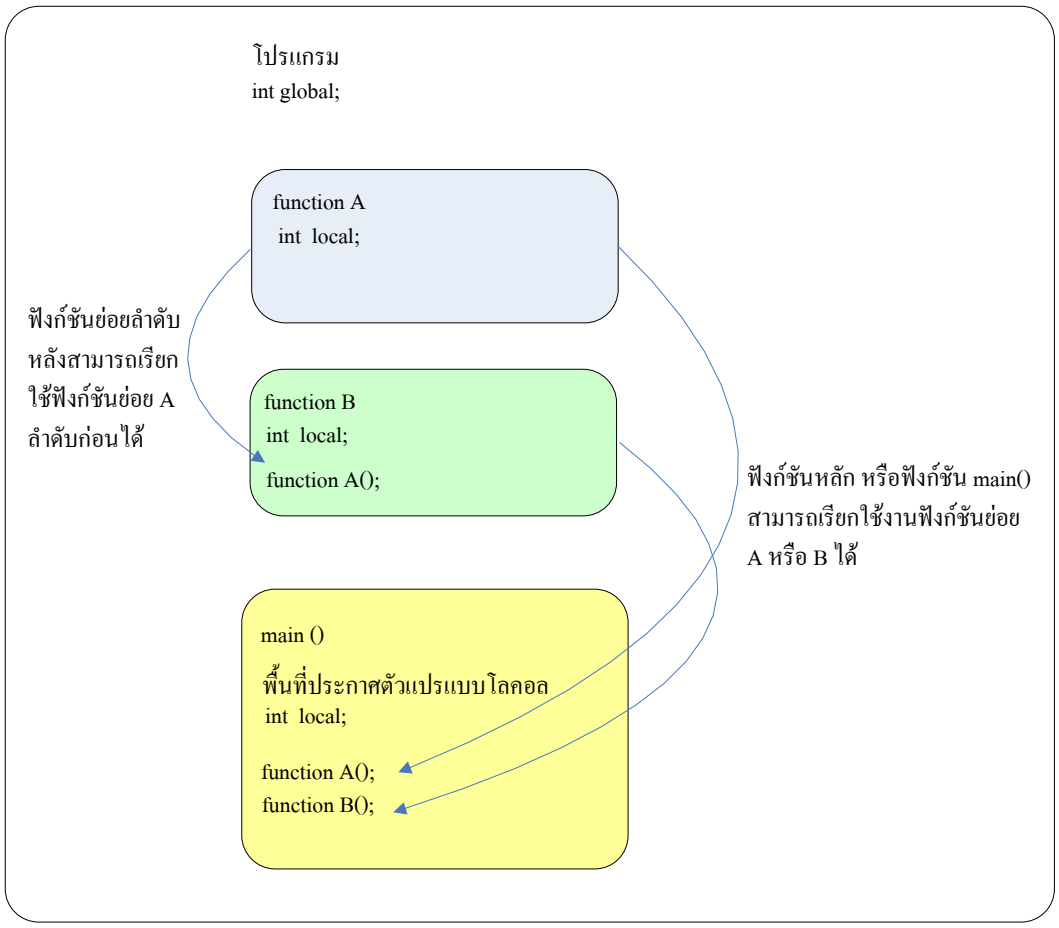
หน่วยเรียนที่ 5

ฟังก์ชัน

ฟังก์ชัน (Function) คือกลุ่มคำสั่งหรือโปรแกรมย่อยที่ทำหน้าที่อย่างใดอย่างหนึ่ง ซึ่งได้ถูกพัฒนาโดยการรวบรวมขั้นตอนการทำงานของคำสั่งต่างๆไว้ด้วยกันจนได้ผลลัพธ์ตามที่ต้องการ ฟังก์ชันในภาษีซีแบ่งออกเป็น 2 ประเภทคือ ฟังก์ชันของคอมไพเลอร์ภาษาซี ซึ่งเป็นฟังก์ชันที่ผู้พัฒนาคอมไพเลอร์สร้างขึ้นมา และฟังก์ชันที่เราสร้างขึ้นเอง หากเป็นฟังก์ชันของคอมไพเลอร์จะอยู่ในไลบราลี ในไฟล์นามสกุล .h ที่นักพัฒนาโปรแกรมจะต้องใช้คำสั่ง #include เพื่อนำไฟล์ดังกล่าวมาใช้ในโปรแกรมหลัก โดยจะต้องทราบชื่อและการทำงานของฟังก์ชันนั้นๆ จึงจะสามารถเรียกใช้ฟังก์ชันได้อย่างถูกต้อง

5.1 การสร้างฟังก์ชัน

การสร้างหรือเขียนฟังก์ชันจะมีประโยชน์มากในการเขียนโปรแกรมที่มีความซับซ้อนขึ้น ทำให้การพัฒนาโปรแกรมทำได้ง่ายขึ้น ขนาดของโปรแกรมสั้นลง ทำให้การประมวลผลเร็วขึ้น การเขียนโปรแกรมมีลักษณะเป็นลำดับขั้นตอนทำให้แก้ไขโปรแกรมได้ง่ายขึ้น ซึ่งเมื่อได้สร้างฟังก์ชันย่อยขึ้นมาแล้ว ก็สามารถเรียกใช้ฟังก์ชันย่อยนั้น ณ ตำแหน่งของฟังก์ชันหลัก หรือฟังก์ชันย่อยอื่นๆ ทั้งนี้ฟังก์ชันย่อยลำดับหลังจะสามารถเรียกใช้ฟังก์ชันลำดับก่อนได้ โดยลักษณะโครงสร้างของฟังก์ชันย่อยเป็นดังภาพที่ 5.1



ภาพที่ 5.1 โครงสร้างของฟังก์ชันย่อยที่สร้างขึ้นเอง

5.1.1 ฟังก์ชันชนิดไม่ผ่านค่าข้อมูล

ฟังก์ชันชนิดไม่ผ่านค่าข้อมูลเป็นฟังก์ชันที่มีการทำงานตามคำสั่งต่างๆ โดยที่ไม่มีการรับข้อมูลหรือส่งค่าข้อมูลออกจากฟังก์ชัน หากต้องการให้โปรแกรมทำงานตามลักษณะของฟังก์ชันที่สร้างขึ้นมานั้น ก็เพียงเรียกชื่อฟังก์ชันนั้น ณ ตำแหน่งใดๆในโปรแกรมหลัก ก็สามารถใช้งานได้ทันที การสร้างฟังก์ชันชนิดไม่ผ่านค่าข้อมูลมีลักษณะดังนี้

```
void ชื่อฟังก์ชัน( )
{
  คำสั่งต่างๆ
}
```

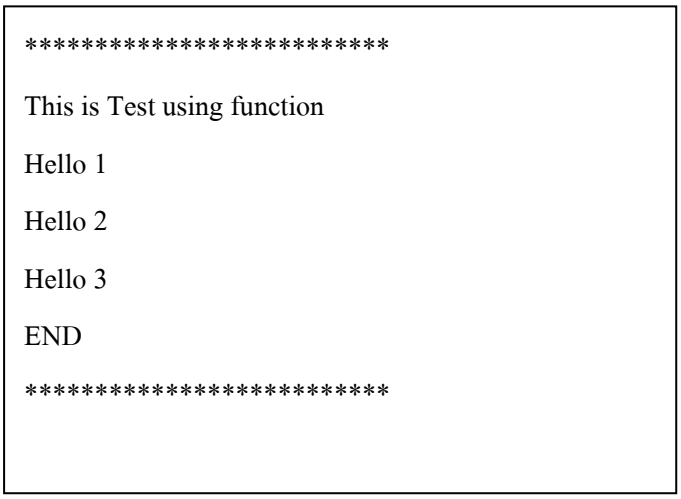
ตัวอย่างที่ 5.1 การสร้างฟังก์ชันที่ไม่ผ่านค่าข้อมูล

```
#include <stdio.h>
void hello()                //              ฟังก์ชันย่อยที่สร้างขึ้น
{
   for(i=1;i<=5;i++)       // กำหนดให้ทำงานวนซ้ำ 5 ครั้ง
    printf("Hello %d\n",i);              // ปริ้นข้อความ Hello
}

void line()
{
    printf("*************************\n");
}

main()                      //  ฟังก์ชันหลัก
{
  line();                   // เรียกใช้ฟังก์ชัน line()
  printf(" This is Test using function\n");
  hello();    //  เรียกใช้ฟังก์ชันที่สร้างขึ้น
  printf("END");
  line();                   // เรียกใช้ฟังก์ชัน line()
}
```

เมื่อรันโปรแกรม ฟังก์ชันหลัก main() จะมีการเรียกใช้งานฟังก์ชันย่อยที่สร้างขึ้นซึ่งจะมีการเรียกใช้ฟังก์ชันย่อยชื่อ line() เพื่อวาดเส้น และฟังก์ชัน hello() ซึ่งภายในฟังก์ชัน hello() จะมีการวนซ้ำแสดงข้อความ Hello จำนวน 10 รอบ เมื่อครบจำนวนก็จะเรียกใช้ฟังก์ชัน line() เพื่อวาดเส้นอีกครั้ง โดยจะได้ผลลัพธ์การทำงานดังภาพที่ 5.2



ภาพที่ 5.2 ผลลัพธ์บนจอภาพของโปรแกรมตัวอย่างที่ 5.1

เมื่อสร้างฟังก์ชันย่อยแล้วสามารถเรียกใช้ฟังก์ชันย่อยกี่ครั้งก็ได้ หรือเรียกตำแหน่งใดก็ได้ ยกเว้นฟังก์ชันย่อยลำดับที่ถูกสร้างขึ้นก่อนจะไม่สามารถเรียกใช้ฟังก์ชันที่ถูกสร้างขึ้นมาลำดับหลังได้

5.1.2 ฟังก์ชันชนิดผ่านค่าข้อมูล

เป็นฟังก์ชันที่ผ่านค่าข้อมูลทางตัวแปร เพื่อนำค่าข้อมูลนั้นมาใช้ในฟังก์ชันย่อยที่สร้างขึ้น โดยฟังก์ชันชนิดรับค่าข้อมูลจะต้องประกาศชนิด และ ตัวแปรที่ต้องการนำเข้าข้อมูลไว้ภายในวงเล็บหลังชื่อฟังก์ชัน และขณะเรียกใช้ฟังก์ชัน จะต้องกำหนดตัวเลขในวงเล็บหลังชื่อฟังก์ชัน เพื่อส่งข้อมูลดังกล่าวไปใช้ประมวลผลภายในฟังก์ชัน การสร้างฟังก์ชันชนิดผ่านค่าข้อมูลมีลักษณะดังนี้

```
void ชื่อฟังก์ชัน(ชนิดตัวแปรรับค่าข้อมูล  ชื่อตัวแปร )
{
  คำสั่งต่างๆ
}
```

ตัวอย่างที่ 5.2 การสร้างฟังก์ชันชนิดผ่านค่าข้อมูล

```
#include <stdio.h>
void hello(int k)            //ฟังก์ชันย่อยที่สร้างขึ้นซึ่งจะรับค่าข้อมูลผ่านตัวแปร k
{
  for(i=1;i<=k;i++)         // ตัวแปร k กำหนดจำนวนการวนซ้ำ
  printf("Hello %d\n",i);
}

main()        // ฟังก์ชันหลัก
{
  printf(" This is Test using function\n");
  hello(4);      //เรียกใช้ฟังก์ชันที่สร้างขึ้นและกำหนดค่าให้ตัวแปร k
  printf("END");
}
```

เมื่อรันโปรแกรม ฟังก์ชันหลัก main() จะมีการเรียกใช้งานฟังก์ชันย่อยชนิดผ่านค่าข้อมูลที่สร้างขึ้น ซึ่งจะต้องกำหนดข้อมูลให้แก่ฟังก์ชันย่อย เช่น hello(4); โดย 4 คือค่าที่จะส่งให้กับตัวแปร k เพื่อนำไปใช้ฟังก์ชันย่อย ซึ่งเป็นจำนวนรอบของการวนซ้ำแสดงข้อความ Hello โดยผลลัพธ์ของโปรแกรมเป็นดังภารพที่ 5.3

```
This is Test using function
Hello 1
Hello 2
Hello 3
Hello 4
END
```

ภาพที่ 5.3 ผลลัพธ์บนจอภาพของโปรแกรมตัวอย่างที่ 5.2

5.1.3 ฟังก์ชันชนิดรับและคืนค่าข้อมูล

เป็นฟังก์ชันที่มีการรับและส่งคืนค่าข้อมูลออกมาจากฟังก์ชันหลังจากได้ทำงานในฟังก์ชันเสร็จสิ้นแล้ว โดยอาจจะมีการรับค่าข้อมูลเข้าไปใช้งานและเมื่อทำงานเสร็จจะมีการส่งค่าคืนหรือผลลัพธ์ออกมาด้วยคำสั่ง return() เพื่อนำผลลัพธ์นั้นไปใช้งานต่อไป การสร้างฟังชันชนิดรับและคืนค่าข้อมูลมีลักษณะดังนี้

```
ชนิดข้อมูล ชื่อฟังก์ชัน(ชนิดตัวแปร ตัวแปร )
 {
   คำสั่งต่างๆ
   return(ข้อมูล)
 }
```

ตัวอย่างที่ 5.3 การสร้างฟังก์ชันรับค่าและคืนค่าข้อมูล

```
#include <stdio.h>
int add_num(int j, int k)        // ฟังก์ชันย่อยที่สร้างขึ้น โดยรับข้อมูล 2 ตัวแปร
{
  int q;
  q = j + k;
  return(q);                     // คืนค่าออกจากฟังก์ชัน
 }
void main()                      // โปรแกรมหลัก
{
  int a, b, c,d;                 // สร้างตัวแปร
  a=3;                           // กำหนดค่าให้ตัวแปร a
  b=5;                           // กำหนดค่าให้ตัวแปร b
  c = add_num(a,b);              // เรียกใช้ฟังก์ชันที่สร้างขึ้นและผลลัพธ์แก่ตัวแปร c
  d = ad_num(6,9);               // เรียกใช้ฟังก์ชันที่สร้างขึ้นและผลลัพธ์แก่ตัวแปร d
  printf(" This is Test using function\n");
  printf("result C = %d\n",c);
  printf("result D = %d\n",d);
```

```
  printf("END");
}
```

เมื่อรันโปรแกรม ฟังก์ชันหลัก main() จะมีการเรียกใช้งานฟังก์ชันย่อยรับข้อมูลและส่งค่าออก ที่สร้างขึ้น โดยมีการกำหนดข้อมูล 6 และ 9 ให้แก่ฟังก์ชันย่อย add_num(6,9) มีผลให้ผลลัพธ์ฟังก์ชันย่อยที่สร้างขึ้นเป็น 15 และเก็บไว้ในตัวแปร d และมีการแสดงผลลัพธ์ดังภาพที่ 5.4

```
This is Test using function
result C = 8
result D = 15
END
```

ภาพที่ 5.4 ผลลัพธ์บนจอภาพของโปรแกรมตัวอย่างที่ 5.3

ตัวอย่างที่ 5.4 การสร้างฟังก์ชันคำนวณผลคูณ

```
#include <stdio.h>
float  multi_num(float x, float y)       // ฟังก์ชันย่อยที่สร้างขึ้น โดยรับข้อมูล 2 ตัวแปร
{
  float z;
  z = x * y;
  return(z);                  // ส่งค่าออกจากฟังก์ชัน
 }
void main()                   // โปรแกรมหลัก
{
  float  a, b,c;              //  สร้างตัวแปร
  printf("This is Multiple Function  Z = X * Y");
  printf("\nPlease Enter X  =");
  scanf("%f",&a);
  printf("\nPlease Enter Y =");
  scanf("%f",&b);
```

```
  c = multi_num(a,b);
  printf("\nResult  Z = %.2f x %.2f  = %.2f",a,b,c);
  printf("END");
}
```

เมื่อรันโปรแกรมจะแสดงข้อความให้เราป้อนค่า X และค่า Y แล้วเก็บค่าไว้ในตัวแปร a และตัวแปร b ตามลำดับ และมีการเรียกใช้ฟังก์ชัน multi_num(a,b); พร้อมกับส่งค่าให้แก่ตัวแปร x และ ตัวแปร y ของฟังก์ชัน multi_num(a,b); ซึ่งเป็นฟังก์ชันการคูณ ผลลัพธ์ของฟังก์ชันจะถูกคืนค่าออกมาด้วยคำสั่ง return(z) ให้แก่ตัวแปร c เพื่อนำมาแสดงผลบนจอภาพ ดังในภาพที่ 5.5

```
This is Multiple Function  Z = X * Y
Please Enter X  = 10
Please Enter Y  = 15
Result  Z = 10.00 * 15.00 = 150.00
END
```

ภาพที่ 5.5 ผลลัพธ์บนจอภาพของโปรแกรมตัวอย่างที่ 5.4

5.2 การประกาศโปรโตไทป์ของฟังก์ชัน

การประกาศโปรโตไทป์ (Prototype) เป็นสิ่งจำเป็นในภาษาซี เนื่องจากภาษาซีเป็นภาษาในลักษณะที่ต้องมีการประกาศฟังก์ชันก่อนจะเรียกใช้ฟังก์ชันนั้นๆ จากตัวอย่างที่ผ่านมาจะเห็นว่า ฟังก์ชัน main() ซึ่งเป็นฟังก์ชันหลักที่เรียกใช้ฟังก์ชันย่อยอื่นๆ จะมีลำดับหลังฟังก์ชันย่อยที่ถูกเรียกใช้ นั่นคือมีการประกาศฟังก์ชันย่อยไว้ก่อนแล้วที่จะมีการเรียกใช้ฟังก์ชันดังกล่าว แต่หากต้องการย้ายฟังก์ชัน main() ขึ้นไปไว้ลำดับแรก หรือ ด้านบนก่อนการสร้างฟังก์ชันย่อย จะต้องมีการประกาศโปรโตไทป์ของฟังก์ชันย่อยที่ต้องการเรียกใช้ก่อนเสมอ

ตัวอย่างที่ 5.5 การประกาศโปรโตไทป์ของฟังก์ชันที่สร้างขึ้นก่อนโปรแกรมหลัก

```
#include <stdio.h>
float  multi_num(float , float );          // ประกาศโปรโตไทป์
void main()                    // โปรแกรมหลัก
{
  float  a, b,c;               //  สร้างตัวแปร
  printf("This is Multiple Function  Z = X * Y");
  printf("\nPlease Enter X  =");
  scanf("%f",&a);
  printf("\nPlease Enter Y =");
  scanf("%f",&b);
  c = multi_num(a,b);
  printf("\nResult  Z = %.2f x %.2f  = %.2f",a,b,c);
  printf("END");
}
//    ฟังก์ชันย่อย
float  multi_num(float x, float y)       // ฟังก์ชันย่อยที่สร้างขึ้น โดยรับข้อมูล 2 ตัวแปร
{
  float z;
  z = x * y;
  return(z);                   // ส่งค่าออกจากฟังก์ชัน
 }
```

ซึ่งผลการทำงานของโปรแกรมจะเหมือนกับผลลัพธ์ในตัวอย่างที่ 5.4 เพียงแต่แตกต่างกันตรงที่ ในตัวอย่างนี้ การสร้างฟังก์ชันย่อยอยู่ลำดับหลังฟังก์ชัน main ( )

5.3 ฟังก์ชันในไลบรารีมาตรฐาน

นอกจากฟังก์ชันต่างๆที่นักพัฒนาเขียนขึ้นแล้ว ภาษาซีได้เตรียมฟังก์ชันมาตรฐานซึ่งจัดเก็บอยู่ในลักษณะที่เรียกว่าคลังโปรแกรมหรือไลบรารี (Library) ตัวอย่างของฟังก์ชันในไลบรารีมาตรฐานที่ได้มีการยกตัวอย่างการใช้งานไปแล้วคือ ฟังก์ชัน scanf() สำหรับรับข้อมูล และฟังก์ชัน printf() สำหรับแสดงผลข้อมูลซึ่งเป็นฟังก์ชันอยู่ในไลบรารี stdio.h ดังนั้นการเรียกใช้ฟังก์ชันดังกล่าวจะต้องมีการประกาศ #include < stdio.h> ในส่วนหัวโปรแกรม ก่อนเรียกใช้ฟังก์ชันดังกล่าว นอกจากนี้ยังมีฟังก์ชันต่างๆ ที่อยู่ในไลบรารีมาตรฐานที่น่าสนใจดังนี้

5.3.1 ฟังก์ชันเกี่ยวกับอักขระ

ฟังก์ชันเกี่ยวกับอักขระ เป็นฟังก์ชันต่างๆที่เกี่ยวกับการตรวจสอบตัวอักษร ซึ่งอยู่ในไลบรารีมาตรฐาน ctype.h ดังนั้นหากต้องการเรียกใช้ฟังก์ชันต่างๆที่อยู่ในไลบรารีนี้ จะต้องประกาศ #include <ctype.h> ที่ส่วนหัวโปรแกรมเสียก่อน ฟังก์ชันต่างๆเกี่ยวกับตัวอักขระที่น่าสนใจมีดังตารางที่ 5.1

ตารางที่ 5.1 ฟังก์ชันต่างๆที่เกี่ยวกับอักขระ

ฟังก์ชัน	หน้าที่
int isalnum(c)	ตรวจสอบตัวอักขระนั้นว่าเป็นตัวอักขระ 0-9 หรือ a-z หรือ A-Z หรือไม่ ถ้าไม่ใช่จะคืนค่า 0 ถ้าใช่จะคืนค่าที่ไม่ใช่ 0
int isalpha(c)	ตรวจสอบตัวอักขระนั้นว่าเป็นตัวอักขระ a-z หรือ A-Z ถ้าไม่ใช่จะคืนค่า 0 ถ้าใช่จะคืนค่าที่ไม่ใช่ 0
int isdigit(c)	ตรวจสอบตัวอักขระนั้นว่าเป็นตัวอักขระ 0-9 ถ้าไม่ใช่จะคืนค่า 0 ถ้าใช่จะคืนค่าที่ไม่ใช่ 0
int islower(c)	ตรวจสอบตัวอักขระนั้นว่าเป็นตัวอักขระ a-z ไม่ใช่จะคืนค่า 0 ถ้าใช่จะคืนค่าที่ไม่ใช่ 0
int isupper(c)	ตรวจสอบตัวอักขระนั้นว่าเป็นตัวอักขระ A-Z ไม่ใช่จะคืนค่า 0 ถ้าใช่จะคืนค่าที่ไม่ใช่ 0
int tolower(c)	แปลงค่าอักขระปัจจุบันให้เป็นอักขระเดียวกันแต่เป็นตัวพิมพ์เล็ก
int toupper(c)	แปลงค่าอักขระปัจจุบันให้เป็นอักขระเดียวกันแต่เป็นตัวพิมพ์ใหญ่

5.3.2 ฟังก์ชันคณิตศาสตร์

ฟังก์ชันคณิตศาสตร์ เป็นฟังก์ชันที่เกี่ยวกับการหาผลลัพธ์ทางคณิตศาสตร์ต่างๆ เช่น การหาค่ายกกำลัง ค่าเอ็กโปเนนเชียล เป็นต้น การเรียกใช้ฟังก์ชันคณิตศาสตร์จะต้องประกาศ #include <math.h> ที่ส่วนหัวโปรแกรม โดยมีฟังก์ชันคณิตศาสตร์ต่างๆดังตารางที่ 5.2

ตารางที่ 5.2 ฟังก์ชันคณิตศาสตร์ต่างๆ

ฟังก์ชัน	หน้าที่
acos(x)	หาค่า Arc Cosine ของ x
asin(x)	หาค่า Arc Sine ของ x
atan(x)	หาค่า Arc Tangent ของ x
atan2(x,y)	หาค่า Arc Tangent ของ x/y
cell(x)	หาค่าจำนวนจริง x ที่มีการปัดเศษทิ้งทั้งหมด
cos(x)	หาค่า Cosine ของ x
exp(x)	หาค่าเอ็กโปเนนเชียล (Exponential) ของ x
floor(x)	หาค่าจำนวนจริง x ที่มีการปัดเศษขึ้นทั้งหมด
log(x)	หาค่า Logarithm ของ x
log10(x)	หาค่า Logarithm ฐาน 10 ของ x
pow(x,y)	หาค่า x ยกกำลัง y
sin(x)	หาค่า Sine ของ x
sqrt(x)	หาค่า Square Root ของ x
tan(x)	หาค่า Tangent ของ x

ตัวอย่างที่ 5.6 การใช้ฟังก์ชันคณิตศาสตร์คำนวณหาค่า Sine

```
#include <stdio.h>
#include <math.h>
float ang2radial(int ang)
{
  float rad;
  rad = ang*3.14159/180;
```

```
  return(rad);
}
void main()
{
   float result,ang,radial;
   printf("Please Enter Angle : " );
   scanf("%f",&ang);
   radial = ang2radial(ang);
   result = sin(radial);
   printf("\n The result Sine(%.2f) = %f",ang,result);
   getch();
}
```

การใช้ฟังก์ชันคณิตศาสตร์หาค่าตรีโกณมิติจะต้องแปลงค่ามุมองศาให้อยู่ในรูปของความถี่เชิงมุม (Radial) เสียก่อน ดังนั้นจึงต้องสร้างฟังก์ชัน ang2radial( ) เพื่อแปลงค่าองศาให้เป็นหน่วยเรเดียน แล้วจึงส่งค่าเข้าฟังก์ชัน sin(x) เพื่อหาค่าผลลัพธ์ของ Sine ดังแสดงในภาพที่ 5.6

```
Please Enter Angle : 45
The result Sine(45.00) = 0.7071
```

ภาพที่ 5.6 ผลลัพธ์บนจอภาพของโปรแกรมตัวอย่างที่ 5.7

5.3.3 ฟังก์ชันการรับและแสดงผล

ฟังก์ชันการรับและแสดงผล จะเกี่ยวกับการรับข้อมูลผ่านทางคีย์บอร์ดและการแสดงผลทางจอภาพ ซึ่งเป็นฟังก์ชันที่อยู่ในไลบรารีมาตรฐาน stdio.h ดังนั้นการจะเรียกใช้ฟังก์ชันเหล่านี้ต้องประกาศ #include <stdio.h> ในส่วนหัวโปรแกรม และฟังก์ชันการรับและแสดงผลมีดังตารางที่ 5.3

ตารางที่ 5.3 ฟังก์ชันการรับและแสดงผล

ฟังก์ชัน	หน้าที่
int getchar()	อ่านข้อมูล 1 อักขระจากอุปกรณ์รับข้อมูลมาตรฐาน
char *gets(const char *buffer)	อ่านข้อมูลสตริงจากอุปกรณ์รับข้อมูลมาตรฐานมาเก็บในตัวแปร buffer จนกว่าจะมีการขึ้นบรรทัดใหม่ หากมีความผิดพลาดจะคืนค่า NULL
int printf(const char *format,..)	ใช้แสดงผลข้อมูลทางอุปกรณ์แสดงผลมาตรฐานในรูปแบบที่กำหนด
int putchar(int c)	ส่งอักขระ c ไปแสดงผลทางอุปกรณ์แสดงผลมาตรฐาน
int scanf(const char *format,...)	ใช้อ่านข้อมูลทางอุปกรณ์รับข้อมูลมาตรฐานในรูปแบบที่กำหนด
int sprintf(char *buffer,const char *format,...)	ทำหน้าที่คล้ายกับ printf( ) แต่จะส่งผลของข้อมุลในรูปแบบที่กำหนดไปเก็บไว้ในสตริง buffer

ตัวอย่างที่ 5.7 การรับข้อมูลเลขจำนวนจริงจากผู้ใช้และแปลงค่าเป็นเลขจำนวนเต็มโดยการปัดเศษ หากเศษมากกว่า .50 ให้ปัดขึ้น แต่หากเศษน้อยกว่าให้ปัดลง

```
#include <stdio.h>
#include <conio.h>
void main()
{
  float f;
  char s[21];
  int i;
  clrscr();
  printf("Enter number :");
  scanf("%f",&f);
  sprintf(s,"%.0f",f);
  i = atoi(s);
  printf("Integer number is %d",i);
```

```
    getchar();
}
```

5.3.4 ไลบรารีมาตรฐาน

ไลบรารีมาตรฐาน (Standard Library) เป็นไฟล์นามสกุล .h เก็บฟังก์ชันต่างๆที่ทางคอมไพเลอร์ของแต่ละบริษัทได้ผลิตขึ้นมาให้สะดวกต่อการประยุกต์ใช้งาน ซึ่งมีอยู่ด้วยกันหลายไฟล์หรือไลบรารี ที่ให้การสนับสนุนการทำงานเกียวกับอักขระ สตริง กราฟฟิก และคณิตศาสตร์ ดังที่ได้กล่าวมาแล้ว นอกจากยังมีไลบรารีอื่นๆอีกที่ไม่ได้ยกตัวอย่างซึ่งสามารถไปทำความเข้าใจจากคู่มือของคอมไพเลอร์ของบริษัทผู้ผลิตต่างๆ และไลบรารีมาตรฐานมีดังตารางที่ 5.4

ตารางที่ 5.4 ไลบรารีมาตรฐาน

ไลบรารี	เกี่ยวกับ
blos.h	อินเตอร์รัปต์ (Interrupts) ของ BIOS
conio.h	จอภาพ คีย์บอร์ด และพอร์ต I/O
ctype.h	ฟังก์ชันอักขระ
dos.h	อินเตอร์รัปต์ ของ DOS
lo.h	ไฟล์แฮนเดิล (file Handling) และ I/O ระดับต่ำ
math.h	ฟังก์ชันคณิตศาสตร์
stdio.h	รูทีน (routines) เกี่ยวกับภาษา C
stdllb.h	รูทีน (routines) ในไลบรารีมาตรฐาน
string.h	ฟังก์ชันสตริง
time.h	วันและเวลา

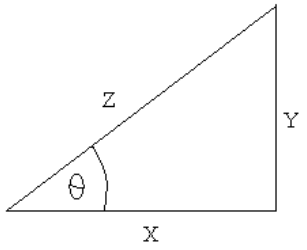
<u>แบบฝึกหัดท้ายหน่วยเรียนที่ 5</u>

1) จงเขียนโปรแกรมหาค่าพื้นที่วงกลมและเส้นรอบวงกลม โดยการคำนวณแต่ละแบบให้เป็นลักษณะของฟังก์ชัน และให้โปรแกรมรับค่ารัศมีจากการป้อน

2) จงสร้างฟังก์ชันคำนวณหาเวลาหน่วยเป็นวินาที เมื่อผู้ใช้ป้อนค่าหน่วยเป็นชั่วโมง

3) จงเขียนโปรแกรมหาผลลัพธ์ของการคำนวณจากสมการ x = 2ab+c โดยเขียนเป็นฟังก์ชันรับข้อมูลจำนวนเต็ม a b และ c ฟังก์ชันคำนวณผลลัพธ์ และฟังก์ชันแสดงผลลัพธ์

4) จงเขียนโปรแกรมโดยใช้กฎปีทากอรัท หาค่าด้าน Z และมุม θ ดังในภาพที่ โดยมีการรับค่าด้าน X และ Yจากการป้อนค่าผ่านคีย์บอร์ด



5) จงเขียนโปรแกรมหาค่า กำลังไฟฟ้า 3 เฟส และมุมเฟส เมื่อผู้ใช้ป้อนค่ากระแส แรงดัน และเพาเวอร์แฟคเตอร์

ใบงานที่ 8

การสร้างฟังก์ชัน

จุดประสงค์

1) ทดลองสร้างฟังก์ชันชนิดไม่ผ่านค่าข้อมูลและชนิดผ่านค่าข้อมูล
2) ทดลองเรียกใช้งานฟังก์ชันชนิดไม่ผ่านค่าข้อมูล

การสร้างหรือเขียนฟังก์ชันจะมีประโยชน์มากในการเขียนโปรแกรมที่มีความซับซ้อนขึ้น ทำให้การพัฒนาโปรแกรมทำได้ง่ายขึ้น ขนาดของโปรแกรมสั้นลง ทำให้การประมวลผลเร็วขึ้น การเขียนโปรแกรมมีลักษณะเป็นลำดับขั้นตอนทำให้แก้ไขโปรแกรมได้ง่ายขึ้น ซึ่งเมื่อได้สร้างฟังก์ชันย่อยขึ้นมาแล้ว ก็สามารถเรียกใช้ฟังก์ชันย่อยนั้น ณ ตำแหน่งของฟังก์ชันหลัก ซึ่งฟังก์ชันที่สร้างขึ้นนั้นจะมีด้วยกัน 3 ชนิด คือ

1) ฟังก์ชันชนิดไม่ผ่านค่าข้อมูล
2) ฟังก์ชันชนิดรับค่าข้อมูล
3) ฟังก์ชันชนิดรับและคืนค่าข้อมูล

การทดลองที่ 8.1 การสร้างฟังก์ชันชนิดไม่ผ่านค่าข้อมูล

การทดลองนี้จะนำเอาโปรแกรมในใบงานที่ 6.1 ซึ่งเป็นการเขียนโปรแกรมแบบลำดับมาปรับรูปแบบการเขียนโปรแกรมใหม่ โดยสร้างเป็นฟังก์ชันย่อยการแปลงอุณหภูมิองศาเซลเซียสเป็นองศาฟาเรนไฮต์ และองศาฟาเรนไฮต์เป็นองศาเซลเซียส แล้วจึงเรียกใช้ฟังก์ชันย่อยดังกล่าวอีกครั้งในภายหลัง โดยมีลำดับการทดลองดังนี้

1) เปิดโปรแกรมคอมไพเลอร์ Turbo C และ สร้างไฟล์ใหม่
2) เขียนโปรแกรมลงในอีดิตเตอร์ตามตัวอย่างโปรแกรมดังนี้

```
#include <stdio.h>
#include<conio.h>
int loop,select;
char exit;
float temp,c,f;
void menu(void)
{
   printf(" ******************************\n");
   printf("*       Transform Temperature          *\n");
   printf("********************************\n");
   printf("\n Please select menu :1-3 only");
   printf("\n 1)  Transform C-> F");
   printf("\n 2)  Transform F-> C");
   printf("\n 3)  Exit");
   printf("\n You select:");
}
void c2f_temp(void)
{
do{
        printf("\nTransform C-> F");
        printf("\n Enter  Cencius Temperature : ");
        scanf("%f",&c);
        f=  ((c*9)/5)+32;
        printf("\n Farenhi Temp = %.2f",f);
        printf("\n Do you want to continue? Yes or No (Y/N) :");
        scanf("%c",&exit);
        scanf("%c",&exit);
      }while(exit != 'n');
  }  // end function
```

```
void f2c_temp(void)
{
 do{
        printf("\nTransform F-> C");
        printf("\n Enter  Farenhi Temperature : ");
        scanf("%f",&f);
        c=  (f-32)*5/9;
        printf("\n Cencius Temp = %.2f",c);
        printf("\n Do you want to continue? Yes or No (Y/N) :");
        scanf("%c",&exit);
        scanf("%c",&exit);
    }while(exit != 'n');
} // end function
void main(void)
{
  clrscr();
  do
    {
                menu()
                 scanf("%d",&select);
                 if(select ==1)

                 c2f_temp();
                 else if(select ==2)
                        f2c_temp();
                 else if(select !=3)
                 printf("\n Select Wrong Number");
     }while(select != 3);
  }
```

3) บันทึกไฟล์ในเป็นไฟล์ program8_1.c

4) เลือกคำสั่ง Compile และให้บันทึกผล

5) เลือกคำสั่ง RUN และให้บันทึกผลการทดลอง

……
……
……
……
……

การทดลองที่ 8.2 การสร้างฟังก์ชันชนิดผ่านค่าข้อมูล

ฟังก์ชันผ่านค่าข้อมูลจะเป็นการนำข้อมูลส่งให้แก่ฟังก์ชันผ่านทางตัวแปรอินพุตของฟังก์ชันนั้นๆ เพิ่มความสะดวกให้แก่การส่งข้อมูลให้แก่ฟังก์ชัน โดยที่ไม่ต้องประกาศตัวแปรนอกฟังก์ชันไว้เก็บข้อมูลอินพุต ซึ่งสามารถส่งค่าตัวเลขให้แก่ฟังก์ชันได้โดยตรง เช่นสร้างฟังก์ชันขีดเส้นบรรทัด สามารถบอกจำนวนของเส้นที่ต้องการให้แก่ฟังก์ชัน เมื่อฟังก์ชันรับค่าอินพุตผ่านทางตัวแปรและนำไปงานภายในฟังก์ชัน โดยมีลำดับการทดลองดังนี้

1) เปิดโปรแกรมคอมไพเลอร์ Turbo C และ สร้างไฟล์ใหม่

2) เขียนโปรแกรมลงในอีดิตเตอร์ตามตัวอย่างโปรแกรมดังนี้

```
#include <stdio.h>
#include<conio.h>
int loop,select;
char exit;
float temp,c,f;
void line(int x)
  {
    int l;
    for(l=0;l<x;x++)
        printf(" ******************************\n");
  }
void menu(void)
```

```
{
   line(1);
   printf("*       Transform Temperature          *\n");
   line(2);
   printf("\n Please select menu :1-3 only");
   printf("\n 1)  Transform C-> F");
   printf("\n 2)  Transform F-> C");
   printf("\n 3)  Exit");
  line(1);
   printf("\n You select:");
}
void c2f_temp()
{
do{
        printf("\nTransform C-> F");
        printf("\n Enter  Cencius Temperature : ");
        scanf("%f",&c);
        f=  ((c*9)/5)+32;
        printf("\n Farenhi Temp = %.2f",f);
        line(1);
        printf("\n Do you want to continue? Yes or No (Y/N) :");
        scanf("%c",&exit);
        scanf("%c",&exit);
      }while(exit != 'n');
  }  // end function

void f2c_temp(void)
{
 do{
         printf("\nTransform F-> C");
```

```
        printf("\n Enter  Farenhi Temperature : ");
        scanf("%f",&f);
        c=  (f-32)*5/9;
        printf("\n Cencius Temp = %.2f",c);
        printf("\n Do you want to continue? Yes or No (Y/N) :");
        scanf("%c",&exit);
        scanf("%c",&exit);
    }while(exit != 'n');
} // end function
void main(void)
{
  clrscr();
  do
    {
                menu()
                 scanf("%d",&select);
                 if(select ==1)
                 c2f_temp();
                 else if(select ==2)
                        f2c_temp();
                 else if(select !=3)
                 printf("\n Select Wrong Number");
     }while(select != 3);
  }
```

3) บันทึกไฟล์ในเป็นไฟล์ program8_2.c

4) เลือกคำสั่ง Compile และให้บันทึกผล

5) เลือกคำสั่ง RUN และให้บันทึกผลการทดลอง

……
……
……
……
……

6) สรุปผลการทดลอง

……
……
……
……
……

งานที่มอบหมาย

1) จงสร้างฟังก์ชันคำนวณหาค่า กำลังไฟฟ้า 3 เฟส และมุมเฟส เมื่อรับข้อมูลเป็นค่ากระแส แรงดัน เพาเวอร์แฟคเตอร์
2) จงเขียนฟังก์ชันแปลงเลขฐานสิบไปเป็นเลขฐานสิบหก

ใบงานที่ 9

การสร้างฟังก์ชันคืนค่าข้อมูล

จุดประสงค์

1) ทดลองสร้างฟังก์ชันชนิดคืนค่าข้อมูล
2) ทดลองเรียกใช้งานฟังก์ชันคืนค่าข้อมูล

ฟังก์ชันชนิดรับและคืนค่าข้อมูลเป็นฟังก์ชันที่มีการรับค่าข้อมูลเข้าไปใช้สำหรับการคำนวณภายในฟังก์ชัน และมีการคืนค่าข้อมูลกลับออกจากฟังก์ชัน จะมีประโยชน์มากสำหรับการสร้างฟังก์ชันสำหรับการคำนวณที่ให้ผลลัพธ์ออกจากฟังก์ชันนั้นๆ โดยการสร้างฟังก์ชันคืนค่าข้อมูลมีรูปแบบดังนี้

```
ชนิดข้อมูล ชื่อฟังก์ชัน(ชนิดตัวแปร ตัวแปร )
{
 คำสั่งต่างๆ
 return(ข้อมูล)
}
```

การทดลองที่ 9.1 การสร้างฟังก์ชันชนิดผ่านค่าข้อมูล

การทดลองนี้จะเป็นการนำฟังก์ชันในใบงานที่ 7.2 มาดัดแปลงในส่วน สมการคำนวณ จะนำมาสร้างเป็นฟังก์ชันการคืนค่าข้อมูล โดยเมื่อโปรแกรมรับข้อมูลจากผู้ใช้แล้ว จะส่งค่านั้นให้แก่ฟังก์ชัน และเมื่อฟังก์ชันคำนวณเสร็จก็จะคือค่าด้วยคำสั่ง return ออกจากฟังก์ชัน โดยมีลำดับขั้นตอนการทดลองดังต่อไปนี้

1) เปิดโปรแกรมคอมไพเลอร์ Turbo C และ สร้างไฟล์ใหม่

2) เขียนโปรแกรมลงในอีดิตเตอร์ตามตัวอย่างโปรแกรมดังนี้

```
#include <stdio.h>
#include<conio.h>
int loop,select;
char exit;
float temp,c,f;
void line(int x)
  {
    int l;
    for(l=0;l<x;x++)
          printf(" ******************************\n");
  }
float c2f_cal(float cel)
{
  float f;
   f =  ((cel*9)/5)+32;
  return(f);
}
float f2c_cal(float far)
{
  float c;
   c=  (f-32)*5/9;
  return(c);
}
void menu(void)
{
   line(1);
   printf("*       Transform Temperature          *\n");
   line(2);
```

```
   printf("\n Please select menu :1-3 only");
   printf("\n 1)  Transform C-> F");
   printf("\n 2)  Transform F-> C");
   printf("\n 3)  Exit");
  line(1);
   printf("\n You select:");
}
void c2f_temp()
{
do{
        printf("\nTransform C-> F");
        printf("\n Enter  Cencius Temperature : ");
        scanf("%f",&c);
        f= c2f_temp(c);                    //((c*9)/5)+32;
        printf("\n Farenhi Temp = %.2f",f);
        line(1);
        printf("\n Do you want to continue? Yes or No (Y/N) :");
        scanf("%c",&exit);
        scanf("%c",&exit);
      }while(exit != 'n');
  }  // end function
void f2c_temp(void)
{
 do{
         printf("\nTransform F-> C");
         printf("\n Enter  Farenhi Temperature : ");
         scanf("%f",&f);
         c= f2c_temp(f);                   //  (f-32)*5/9;
         printf("\n Cencius Temp = %.2f",c);
         printf("\n Do you want to continue? Yes or No (Y/N) :");
```

```
            scanf("%c",&exit);
            scanf("%c",&exit);
        }while(exit != 'n');
    } // end function
    void main(void)
    {
      clrscr();
      do
        {
                    menu()
                     scanf("%d",&select);
                     if(select ==1)

                     c2f_temp();
                     else if(select ==2)
                           f2c_temp();
                     else if(select !=3)
                     printf("\n Select Wrong Number");
         }while(select != 3);
      }
```

3) บันทึกไฟล์ในเป็นไฟล์ program9_1.c

4) เลือกคำสั่ง Compile และให้บันทึกผล

5) เลือกคำสั่ง RUN และให้บันทึกผลการทดลอง

……

……

……

……

……

6) สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

งานที่มอบหมาย

1) จงสร้างฟังก์ชันคืนค่าข้อมูล Sin(), Cos(), tan() และฟังก์ชันแปลงค่าองศาให้เป็นค่าเรเดียน
2) จงเขียนโปรแกรม เลือกเมนูคำนวณหาค่า Sin , Cos, Tan โดยรับค่าข้อมูลเป็นองศาแล้วใช้ฟังก์ชันจากข้อ 1 หาผลลัพธ์แล้วแสดงผลทางจอภาพ