การทดลองที่ 7

ฟังก์ชันเบื้องต้น

วัตถุประสงค์

1. เข้าใจหลักการของฟังก์ชัน
2. สามารถใช้งานฟังก์ชันสำเร็จรูปได้
3. เข้าใจขอบเขตของตัวแปลชนิดโกลบอลและโลคัล
4. เข้าใจการส่งค่าระหว่างฟังชันแบบ pass by value
5. สามารถสร้างฟังก์ชันขึ้นมาใช้งานได้

ทฤษฎีโดยย่อ

ฟังก์ชันในภาษาซีจะมีอยู่ 2 ลักษณะคือ ฟังก์ชันสำเร็จรูปและฟังก์ชันที่สร้างขึ้นเอง โดยฟังก์ชันสำเร็จรูปนั้นจะเป็นฟังก์ชันสำหรับใช้งานพื้นฐานทั่ว ๆ ไป ซึ่งภาษาซีจะมีให้เรียกใช้ได้ หากแต่การใช้งานฟังก์ชันสำเร็จรูปนั้นจะมีสิ่งที่จะต้องทราบคือ ชื่อฟังก์ชันและไฟล์ .h ที่ต้อง #include เพื่อที่จะใช้ฟังก์ชันนั้นได้ สำหรับฟังก์ชันที่สร้างขึ้นเองจะเป็นฟังก์ชันสำหรับใช้งานเฉพาะอย่างซึ่งไม่มีฟังก์ชันในภาษาซีที่รองรับทำงานนั้นได้

สำหรับฟังก์ชันที่สร้างขึ้นเองนั้น สามารถจัดวางตัวฟังก์ชันที่สร้างขึ้นเองไว้ได้ใน 2 ตำแหน่ง ได้แก่ ก่อนหน้าฟังก์ชันหลัก (main()) และหลังฟังก์ชันหลัก ทั้งนี้หากวางฟังก์ชันที่สร้างขึ้นเองไว้ที่ตำแหน่งหลังฟังก์ชันหลักจะต้องทำการประกาศฟังก์ชันโปรโตไทป์ด้วย มิฉะนั้นจะไม่สามารถเรียกใช้งานฟังก์ชันที่สร้างขึ้นเองนั้นได้

ตอนที่ 1 ศึกษาการใช้งานฟังก์ชันสำเร็จรูป

1.1 ฟังก์ชันสำเร็จรูปทางคณิตศาสตร์พื้นฐาน

1) ใช้โปรแกรมต่อไปนี้ตอบคำถามด้านล่าง

```
#include<stdio.h>  //01
int main()         //02
{
     float x;
     x = sin(30);
     printf("sin(30 degree)=%f\n", x);
     x = sqrt(2);
     printf("sqrt(2)=%f\n", x);
     x = log(10);
     printf("log(10)=%f\n", x);

     return 0;
}
```

ก) โปรแกรมนี้ compile ผ่านหรือไม่ เพราะเหตุใด compile ผ่านแต่ก็ไม่ถูกต้อง เพราะ sin sqrt log อยู่ใน <math.h>

ข) ให้แทรก #include<math.h> ระหว่างบรรทัด //01 และ //02 เมื่อแทรกแล้ว compile ผ่านหรือไม่ ผ่าน

ค) เมื่อสั่ง run ได้ผลลัพธ์บนหน้าจอคือ

```
PS D:\67010411> cd "d:\67010411\Lab7\" ; if ($?) { gcc Test.c -o Test } ; if ($?) { .\Test }
sin(30 degree)=-0.988032
sqrt(2)=1.414214
log(10)=2.302585
PS D:\67010411\Lab7>
```

ง) ผลลัพธ์ที่ได้ในข้อ ค) ถูกต้องหรือไม่ ให้ตรวจสอบโดยใช้เครื่องคิดเลข ไม่

ค่าที่ไม่ถูกต้องคือ sin , log

เพราะ sin, log รับค่ามาผิดตำแหน่ง output ไม่ตรง

สามารถแก้ไขให้ถูกต้องได้โดย x= sin(30 * M_PI /180);

x= log10(10);

1.2 ฟังก์ชันสำเร็จรูปสำหรับใช้กับข้อมูลชนิดข้อความ

1) ใช้โปรแกรมต่อไปนี้ตอบคำถามด้านล่าง

```
#include<stdio.h>
int main()
{
      char w[10]= “hello”;
      char x[20]= “Hello”;
      char y[20];
      char z[20]= “ World”;
      int v1,v2;

      strcpy(y,x);
      printf(“after strcpy x=%s, y=%s\n”, x, y);

      strcat(y,z);
      printf(“after strcat y=%s, z=%s\n”, y, z);

      v1=strcmp(w,x);
      if (!v1)
            printf(“v1 = true\n”);
      else
            printf(“v1 = false\n”);

      v2=strcmpi(w,x);
      if (!v2)
            printf(“v2 = true\n”);
      else
            printf(“v2 = false\n”);

      return 0;
}
```

ก) โปรแกรมนี้ compile ผ่านหรือไม่ ไม่ผ่าน

เพราะเหตุใด strcpy, strcat อยู่ใน #include<string.h>

แก้ไขโดย เพิ่ม #include<string.h> ด้านบน int main

ข) หลังจากแก้ไขโปรแกรมให้ compile ได้แล้ว เมื่อ run จะได้ผลลัพธ์

```
PS D:\67010411\Lab7> cd "d:\67010411\Lab7\" ; if ($?) { gcc L02.c -o L02 } ; if ($?) { .\L02 }
after strcpy x=Hello, y=Hello
after strcat y=Hello World, z= World
v1 = false
v2 = true
PS D:\67010411\Lab7>
```

ค) เหตุใดเงื่อนไขของ if จึงใช้เป็น !v1 (ทำไมไม่ใช้เป็น v1) เพราะโค้ด ต้องการ หา ไม่เท่ากับ สองอันจึงใช้ !v1

ง) ให้บอกความแตกต่างของ strcmp กับ strcmpi

เปรียบเทียบข้อความแบบสนใจตัวใหญ่ตัวเล็ก (strcmp) และ ไม่สนใจ (strcmpi)

1.3 การประกาศฟังก์ชันโปรโตไทป์

1) ใช้โปรแกรมต่อไปนี้ตอบคำถามด้านล่าง

```
#include<stdio.h>
int main()
{
      int x;
      printf("Input number: ");
      scanf("%d",&x);
      showstar(x);
}
void showstar(int x)
{
      int i;
      for(i=0;i<x;i++)
            printf("*");
}
```

ก) โปรแกรมนี้ compile ผ่านหรือไม่ ไม่ผ่าน

เพราะเหตุใด ต้องประกาศ void showstar(int) บรรทัดแรก

ข) ให้แก้ไขโปรแกรมโดยย้ายฟังก์ชัน showstar ไปวางไว้ก่อนหน้าฟังก์ชัน main เมื่อ compile จะผ่านหรือไม่ ผ่าน

ค) ย้ายฟังก์ชัน showstar กลับที่เดิม แล้วให้ประกาศฟังก์ชันโปรโตไทป์ของฟังก์ชัน showstar ไว้ก่อนฟังก์ชัน main

ฟังก์ชันโปรโตไทป์ที่ประกาศเพิ่มในโปรแกรมคือ void showstar(int);

เมื่อ compile จะผ่านหรือไม่ ผ่าน

ง) หากประกาศฟังก์ชันโปรโตไทป์ และในขณะเดียวกันก็วางฟังก์ชันเอาไว้ก่อนหน้าฟังก์ชัน main เหมือนในข้อ ข) จะ compile ผ่านหรือไม่ ___ผ่าน_____

จ) ผลลัพธ์ของโปรแกรมนี้คือ

```
PS D:\67010411\Lab7> cd "d:\67010411\Lab7\" ; if ($?) { gcc L03.c -o L03 } ; if ($?) { .\L03 }
Input number: 20
********************
PS D:\67010411\Lab7>
```

1.4 ขอบเขตของตัวแปร

1) ใช้โปรแกรมต่อไปนี้ตอบคำถามด้านล่าง

```
#include<stdio.h>
int x=10;                          //00

void showx(void);

int main()
{
     int x=99;                     //01
     printf("main: x=%d\n",x);
     showx();
     return 0;
}

void showx()
{
     printf("showx: x=%d",x);
}
```

ก) โปรแกรมนี้ compile ผ่านหรือไม่ _____ผ่าน_____

ข) ผลลัพธ์ของโปรแกรมนี้คือ

```
PS D:\67010411\Lab7> cd "d:\67010411\Lab7\" ; if ($?) { gcc L04.c -o L04 } ; if ($?) { .\L04 }
main: x=99
showx: x=10
PS D:\67010411\Lab7>
```

ค) เหตุใดค่า x ที่แสดงผลโดยฟังก์ชัน main จึงมีค่าแตกต่างจาก x ที่แสดงผลโดยฟังก์ชัน showx เพราะ ในฟังก์ชั่น main จะมีค่า x ของตนเองเป็น 99

ง) เหตุใดฟังก์ชัน showx จึงแสดงผลค่า x ได้ ทั้งที่ไม่มีการประกาศตัวแปร x ในฟังก์ชัน showx แต่อย่างใด เพราะมีค่าของ x ที่ประกาศเป็นตัวแปร gobal ทำให้สามารถใช้ได้ทั้งโปรแกรม

จ) หากลบคำว่า int ในบรรทัด //01 ให้เหลือเพียง x=99; โปรแกรมนี้จะ compile ผ่านหรือไม่ ผ่าน

ฉ) จากข้อ ง) เมื่อ run จะได้ผลลัพธ์คือ

```
PS D:\67010411\Lab7> cd "d:\67010411\Lab7\" ; if ($?) { gcc L04.c -o L04 } ; if ($?) { .\L04 }
main: x=99
showx: x=99
PS D:\67010411\Lab7>
```

ช) เหตุใดค่า x ที่แสดงผลโดยฟังก์ชัน showx จึงให้ผลเป็นเลข 99 ทั้งที่ในฟังก์ชัน showx ไม่มีการใช้คำสั่งใดเพื่อให้เกิดการเปลี่ยนแปลงค่า x เพราะใช้ค่า x เดียวกันทั้งโปรแกรม

ซ) หากลบบรรทัด //01 จะทำให้ compile ผ่านหรือไม่ ผ่าน

ฌ) หากลบบรรทัด //01 เมื่อ run จะได้ผลลัพธ์คือ

```
PS D:\67010411\Lab7> cd "d:\67010411\Lab7\" ; if ($?) { gcc L04.c -o L04 } ; if ($?) { .\L04 }
main: x=10
showx: x=10
PS D:\67010411\Lab7>
```

2) จาก 1) ให้สรุปเกี่ยวกับ x ในบรรทัด //00 กับ x ในบรรทัด //01 _____

บรรทัด //00 เป็นตัวแปรแบบ โกบอล

บรรทัด //01 เป็นตัวแปรแบบ โลคอล

1.5 การส่งค่าตัวแปรแบบ pass by value

1) ใช้โปรแกรมต่อไปนี้ตอบคำถามด้านล่าง

```
#include<stdio.h>

void test(int x)
{
     x+=10;
     printf("test: x=%d\n", x);
}

int main()
{
     int x;
     x=10;
     printf("main (before): x=%d\n", x);    //00
     test(x);
     printf("main (after): x=%d\n", x);     //01

     return 0;
}
```

ก) โปรแกรมนี้ compile ผ่านหรือไม่ ผ่าน

ข) เมื่อ run โปรแกรมนี้แล้วจะได้ผลลัพธ์คือ

```
PS D:\67010411\Lab7> cd "d:\67010411\Lab7\" ; if ($?) { gcc L05.c -o L05 } ; if ($?) { .\L05 }
main (before): x=10
test: x=20
main (after): x=10
PS D:\67010411\Lab7>
```

ค) เหตุใดค่า x ที่แสดงผลโดย printf ในบรรทัด //00 และบรรทัด //01 จึงยังคงให้ค่าเท่ากัน ค่าไม่เปลี่ยนเพราะเป็นการส่งค่า x โดยก็อปปี้ค่าไปเท่านั้น (Pass by value)

ง) ให้สรุปความสัมพันธ์ระหว่าง x ในฟังก์ชัน main และ x ในฟังก์ชัน test _____

เป็น x คนละตัว

_____

_____

1.6 การส่งค่ากลับ

1) ใช้โปรแกรมต่อไปนี้ตอบคำถามด้านล่าง

```
#include<stdio.h>

int test(int x)
{
      x+=10;
      printf("test: x=%d\n", x);
      return x;
}

int main()
{
      int x;
      x=10;
      printf("main (before): x=%d\n", x);    //01
      test(x);                               //02
      printf("main (after): x=%d\n", x);     //03

      return 0;
}
```

ก) โปรแกรมนี้ compile ผ่านหรือไม่ _____ผ่าน_____

ข) ผลลัพธ์ที่ได้บนหน้าจอคือ

```
PS D:\67010411\Lab7> cd "d:\67010411\Lab7\" ; if ($?) { gcc L06.c -o L06 } ; if ($?) { .\L06 }
main (before): x=10
test: x=20
main (after): x=10
PS D:\67010411\Lab7>
```

ค) เปลี่ยนบรรทัด //02 เป็น x=test(x); เมื่อ compile & run จะได้ผลลัพธ์คือ

```
PS D:\67010411\Lab7> cd "d:\67010411\Lab7\" ; if ($?) { gcc L06.c -o L06 } ; if ($?) { .\L06 }
main (before): x=10
test: x=20
main (after): x=20
PS D:\67010411\Lab7>
```

ง) จากข้อ ค) เหตุใดค่า x ที่แสดงจึงแตกต่างจากข้อ ข) _____

มีการส่งค่าผล `x` กลับมาทับค่า `x` ใน `main` โปรแกรม ทำให้ได้ค่าใหม่

_____

_____

_____

2) จากการทดลองในหัวข้อนี้

ก) ให้สรุปเกี่ยวกับตัวแปรที่อยู่คนละฟังก์ชันกัน (เช่น ตั้งชื่อเดียวกันจะมีความเกี่ยวข้องกันหรือไม่) _____

ไม่เกี่ยวข้องกัน เป็นคนละตัวเลย

_____

_____

_____

ข) การนำค่าตัวแปรที่อยู่ในฟังก์ชันหนึ่ง ไปใช้ในอีกฟังก์ชันสามารถทำได้หรือไม่อย่างไร _____

ทำได้ โดยการส่งค่ากลับ `return x;`

_____

_____

_____

1.7 การประยุกต์

1) ให้เขียนโปรแกรมตามเงื่อนไขต่อไปนี้

ก) รับค่าเลขจำนวนเต็ม 2 จำนวน

ข) สร้างฟังก์ชันที่มีฟังก์ชันโปรโตไทป์ดังนี้ int findmax(int, int) โดยฟังก์ชันนี้จะนำค่าเลขจำนวนเต็ม 2 ค่าที่รับเข้ามาทางพารามิเตอร์มาเปรียบเทียบกันแล้วส่งค่าที่มากกว่ากลับออกไป

ค) เรียกใช้งานฟังก์ชันที่สร้างขึ้นในข้อ ข) แล้วนำผลที่ฟังก์ชันส่งกลับ ขึ้นแสดงผลบนหน้าจอ

```
#include <stdio.h>
#include <math.h>
int findmax(int,int);
void main(){
int a,b;
printf("\n Input Num1: ");
scanf("%d",&a);
printf(" Input Num1: ");
scanf("%d",&b);
printf("\n Maximum number : %d\n",findmax(a,b));
}
int findmax(int aa,int bb){
   if(aa>bb)
      return aa;
   else return bb;
}
```

2) ให้เขียนโปรแกรมตามเงื่อนไขต่อไปนี้

ก) รับค่าเลขจำนวนเต็ม 2 จำนวน

ข) โปรแกรมจะแสดงค่าเลขจำนวนเฉพาะที่อยู่ระหว่างตัวเลขที่รับเข้ามานั้น เช่น โปรแกรมรับตัวเลข 1 และ 10 จะได้ผลลัพธ์เป็นตัวเลข 2 3 5 7 บนหน้าจอ

ค) ในการตรวจสอบเลขจำนวนเฉพาะให้ฟังก์ชัน checkprime ซึ่งมีฟังก์ชันโปรโตไทป์ดังนี้ int checkprime(int) โดยถ้าตัวเลขที่ส่งเข้าไปในฟังก์ชันเป็นจำนวนเฉพาะ ฟังก์ชันจะส่งค่า 1 กลับออกมา หากตัวเลขไม่เป็นจำนวนเฉพาะจะส่งค่า 0 กลับออกมา

```
#include <stdio.h>
int checkprime(int);
void main(){
int a,b,i;
printf("\n Input Num start: ");
scanf("%d",&a);
printf(" Input Num stop: ");
scanf("%d",&b);
for(i=a;i<b;i++)
if(checkprime(i))
  printf("%d ",i);
printf("\n");
}
int checkprime(int number){
      int i;
      for (i=2; i<number; i++)
      {
            if (number % i == 0)
            {
                  return 0;
            }
      }
      return 1;
}
```

3) ให้เขียนโปรแกรมตามเงื่อนไขต่อไปนี้

ก) โปรแกรมรับข้อความภาษอังกฤษความยาวไม่เกิน 50 ตัวอักษร

ข) นับจำนวนสระ (a, e, i, o, u) ในข้อความที่รับเข้ามา

ค) การตรวจสอบว่าตัวอักษรเป็นสระหรือไม่ให้สร้างฟังก์ชันชื่อ checkvowel ขึ้นมาเพื่อทำงานนี้ โดยฟังก์ชันจะมีฟังก์ชันโปรโตไทป์ดังนี้ int checkvowel(char) โดยฟังก์ชันจะส่งค่ากลับเป็น 0 ถ้าตัวอักษรที่รับเข้าไปประมวลผลนั้นไม่ใช่สระ และจะส่งค่ากลับเป็น 1 เมื่อตัวอักษรที่รับเข้าไปประมวลผลเป็นสระ

```
#include <stdio.h>
void main(){
int a=0,i;
char str[50];
printf("\n Input String: ");
i=0;
scanf("%s",str);
while(str[i]!='\0')
{
      switch (str[i])
      {
            case 'a' : a++;
                        break;
            case 'e' : a++;
                        break;
            case 'i' : a++;
                        break;
            case 'o' : a++;
                        break;
            case 'u' : a++;
                        break;
            default:   break;
      }
      i++;
}
printf("\n  Number of a,e,i,o,u = : %d\n\n",a);
}
```