

CP 111 Computer Programming

2/2554 Section B01/B02

Lab 5

หมายเหตุ การเขียนโปรแกรมที่คำนวณกับตัวเลขที่มีจุดทศนิยม ให้ประกาศตัวแปรที่เกี่ยวข้องเป็นแบบ double (Double-precision floating point)

1. Problem name: PolyTwo_Func

จงทำโปรแกรมข้างล่างให้สมบูรณ์ เพื่อให้โปรแกรมข้างล่างทำการคำนวณหาค่าของฟังก์ชัน f เมื่อค่า x (จำนวนเต็ม) ถูกป้อนเข้ามาทางคีย์บอร์ด โดยกำหนดให้ฟังก์ชัน f ถูกนิยามในรูปของฟังก์ชันประกอบ (composite function) $f(z(y(x)))$ ดังนี้

$$f(z) = 3z^2 + 8z + 7, \quad z(y) = 2y^2 - 5y + 4 \quad \text{and} \quad y(x) = -3x^2 + 2x - 1$$

ตัวอย่างการทำงานของโปรแกรม

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
2	135258
-5	695615043

```
#include <stdio.h>
_____/ * Function declaration */ ;
int main()
{
    int x,y,z,f;
    scanf("%d",&x);
    y=PolyTwo_Func(____);
    z=PolyTwo_Func(____);
    f=____;
    printf("%d\n",f);
    return 0;
}
int PolyTwo_Func(int a,int b, int c, int x)
{
    /* ฟังก์ชัน PolyTwo มีอาร์กิวเมนต์แบบ int ทั้งหมด 4 ตัว ได้แก่ a, b, c และ x ตามลำดับ a */
    /* ฟังก์ชันจะคำนวณและคืนค่า  $ax^2+bx+c$  */
}
```

2. Problem name: Exp_2

จงเขียนโปรแกรมเพื่อคำนวณหาค่า $\sum_{k=-N}^N e^k$ โดยที่ค่า N จะถูกป้อนเข้ามาทางคีย์บอร์ด โดยกำหนดให้ค่า e^x คำนวณโดยใช้สูตร

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!} \quad \text{where } n = 50$$

หลังจากนั้นให้แสดงค่า $\sum_{k=-N}^N e^k$ ที่คำนวณได้ในบรรทัดถัดไป โดยแสดงเป็นตัวเลขทศนิยม 2 ตำแหน่ง

ตัวอย่างการทำงานของโปรแกรม

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
5	234.78
10	34845.36
0	1.00

Hint: นำโค้ดที่ได้เขียนไว้ในโปรแกรมชื่อ **Exp_Approx** มาเขียนเป็นฟังก์ชัน โดยนิยามอาร์กิวเมนต์และ

ค่าที่คืนให้เหมาะสม หลังจากนั้นให้เรียกใช้ฟังก์ชันนี้ในการคำนวณ $\sum_{k=-N}^N e^k$

Hint:

```
#include <stdio.h>

-----/* Function declaration */
int main()
{
double val,sum;
int k,N;

scanf("%d",&N);
sum=
for(      ;      ;      )
{
val=
sum=
}
printf("%.2f\n",sum);
return 0;
}

-----/* Function definition */
{ /* คำนวณและคืนค่า e^x */

}

}
```

3. Problem name: X_Figure

จงเขียนโปรแกรมเพื่อแสดงเครื่องหมายกากบาทโดยใช้ตัวอักษร 'O' บนพื้นที่หลังที่แสดงโดยตัวอักษร '.'
แสดงดังตัวอย่างข้างล่าง ซึ่งเป็นรูปของเครื่องหมายกากบาทขนาด 11x11

```
0.....0
.O.....O
..O.....O
...O.....O
....O.....O
.....O.....O
.....O.....O
.....O.....O
.....O.....O
.....O.....O
0.....0
```

โดยโปรแกรมจะรับค่าจำนวนเต็มบวก 1 ค่าสำหรับระบุค่าความกว้าง/ยาวของรูปนี้ (ค่าที่ป้อนเข้ามาจะเป็นเลขคี่เสมอ)

ตัวอย่างการทำงานของโปรแกรม

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
7↵	0.....0 .O.....O ..O.....O ...O.....OO.....OO.....OO.....O 0.....0

Hints:

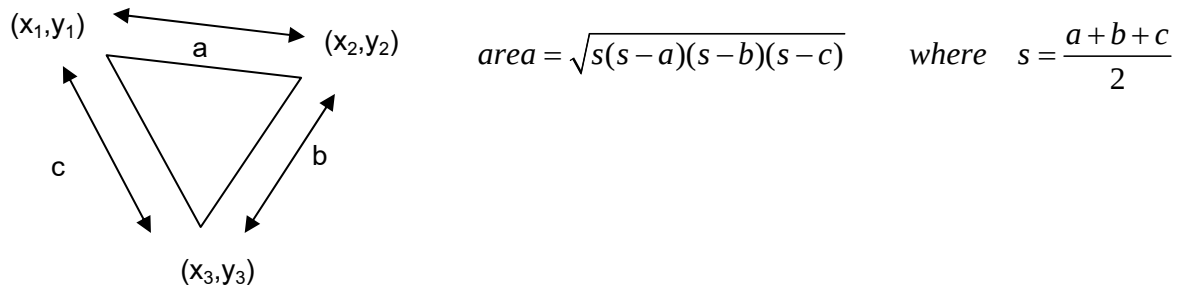
1. พิจารณากราฟของ $|x| = |y|$
- 2.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int x,y,N,Abx,Aby;
    scanf("%d",&N);
    for(_____;_____;_____)
    {
        for(_____;_____;_____)
        {
            Abx=
            Aby=
            if(_____)

                printf(____);
            else
                printf(____);
        }
        printf("____");
    }
    return 0;
}
```

4. Problem name: Tri_Area_Func

จงทำโปรแกรมข้างล่างให้สมบูรณ์ เพื่อให้โปรแกรมนี้คำนวณหาพื้นที่ของสามเหลี่ยมที่ระบุโดยค่า xy-coordinates (x_1, y_1) , (x_2, y_2) และ (x_3, y_3) ดังรูปข้างล่าง และกำหนดให้ใช้ Heron's formula ตามสูตรข้างล่าง



โดยโปรแกรมจะรับค่า (x_1, y_1) , (x_2, y_2) และ (x_3, y_3) ซึ่งเป็นค่าแบบ double คู่ละบรรทัด หลังจากนั้น โปรแกรมค่าพื้นที่ของสามเหลี่ยมที่คำนวณได้ออกสู่หน้าจอ โดยให้แสดงเป็นเลขทศนิยม 2 ตำแหน่ง

ตัวอย่างการทำงานของโปรแกรม

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
0 0 10 0 0 10	50.0
-1.23 5.67 9.01 3.45 7.89 -10.12	70.72

Hint: การคำนวณค่า square root สามารถทำได้โดยใช้ฟังก์ชัน sqrt ซึ่ง Function prototype นิยามไว้ในไฟล์ math.h

ตัวอย่างการใช้งานฟังก์ชันแสดงดังโปรแกรมข้างล่าง

```
#include <stdio.h>
#include <math.h>
void main()
{
double a,b;
a=2;
b=sqrt(a);    // <= b is equal to 1.414...
}
```

หมายเหตุ โปรแกรมนี้จะต้องประกอบไปด้วย 3 ฟังก์ชัน ได้แก่ main และ Triangle_Area โดยที่ฟังก์ชัน Triangle_Area จะรับอาร์กิวเมนต์ 6 ตัวเป็นค่าของโคออร์ดิเนต x1, y1, x2, y2, x3, y3 ตามลำดับ

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

----- /* Function Declaration */
int main()
{
    double x1,y1,x2,y2,x3,y3;
    double a,b,c;
    double area;

    scanf("%lf %lf", &x1,&y1);
    scanf("%lf %lf", &x2,&y2);
    scanf("%lf %lf", &x3,&y3);

    ----- /* Function Call */

    printf("%.2f\n",area);

    return 0;
}

----- /* Function Definition */
{

}

}
```

5. Problem name: Max_Tri_Area

กำหนดให้ข้อมูลนำเข้ามีรูปดังนี้ (i) บรรทัดแรกระบุจำนวนชุดของข้อมูลสมมติว่าเท่ากับ N (ii) บรรทัดที่ 2 ถึง บรรทัดที่ N+1 ในแต่ละบรรทัดจะประกอบไปด้วยตัวเลขแบบ double ทั้งหมด 6 จำนวนแต่ละตัวแทนค่า x_1 y_1 x_2 y_2 x_3 y_3 ที่ระบุ xy-coordinate ของสามเหลี่ยมดังในเช่นชื่อ **Tri_Area_Func**

จงเขียนโปรแกรมเพื่อคำนวณหาพื้นที่ของสามเหลี่ยมที่มีขนาดมากที่สุด ในบรรดาสามเหลี่ยม N ตัวที่ถูกป้อนเข้ามา หลังจากนั้นให้แสดงค่าพื้นที่สามเหลี่ยมที่มากที่สุดที่หามาได้ออกสู่หน้าจอในบรรทัดถัดไป โดยแสดงเป็นตัวเลขทศนิยม 2 ตำแหน่ง

ตัวอย่างการทำงานของโปรแกรม

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
4 0 0 10 0 0 10 -1.23 5.67 9.01 3.45 7.89 -10.12 -1 -1 1 1 0 20 10 10 10 0 0 10	70.72

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

/* TBD */

int main()
{
    int i,N;
    double x1,y1;

    scanf("%d",&N);
    max_area=_____ ;

    for(_____)
    {
        scanf(_____) ;

        _____/* Function call to Triangle_Area() */

        if(_____)
        {
            _____ ;
        }
        printf("%.2f\n",max_area);
        return 0;
    }
double Triangle_Area()
{
/* Reuse from Tri_Area_Func */
}
```

6. Problem name: Is_Prime_Func

จงทำโปรแกรมข้างล่างให้สมบูรณ์เพื่อให้โปรแกรมนี้ทำการแสดงค่าตัวเลขที่เป็นจำนวนเฉพาะ (Prime number) ที่อยู่ในช่วง 2 ถึง N โดยที่ N เป็นค่าจำนวนเต็มที่ถูกป้อนเข้ามาทางคีย์บอร์ด

ตัวอย่างการทำงานของโปรแกรม

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
20	2 3 5 7 11 13 17 19

หมายเหตุ โปรแกรมนี้จะต้องประกอบไปด้วย 2 ฟังก์ชัน ได้แก่ main และ Is_Prime โดยที่ฟังก์ชัน Is_Prime จะรับอาร์กิวเมนต์ 1 ตัวคือตัวเลขที่จะตรวจสอบ และฟังก์ชันจะคืนค่า 1 ถ้าค่าตัวเลขที่ส่งมาเป็นอาร์กิวเมนต์เป็นจำนวนเฉพาะ มิฉะนั้นจะคืนค่า 0

```

#include <stdio.h>

int main()
{
    int i,N,a;

    scanf("%d",&N);
    for( _____; _____; _____)
    {
        _____ /* Function call to Is_Prime */

        if(_____)
            printf("%d\n",i);
    }
    return 0;
}

int Is_Prime(int x)
{ /* คืนค่า 1 ถ้าตัวเลขที่ส่งมาเป็นอาร์กิวเมนต์เป็นจำนวนเฉพาะ มิฉะนั้นจะคืนค่า 0 */
    int flag;

    flag=0;
    for(i=____; _____; _____)
    {
        if(_____)
        {
            flag=_____;
            _____;
        }
    }

    if( _____ )
        return( _____ );
    else
        return( _____ );
}

```