

CP 111 Computer Programming

2/2554 Section B01/B02

Lab 8

หมายเหตุ การเขียนโปรแกรมที่คำนวณกับตัวเลขที่มีจุดทศนิยม ให้ประกาศตัวแปรที่เกี่ยวข้องเป็นแบบ double (Double-precision floating point)

1. Problem name: Extract_Date

(Pointer: Pass-By-Reference)

จงทำโปรแกรมข้างล่างให้สมบูรณ์โดยการเขียนโค้ดในฟังก์ชัน Extract_Date ซึ่งมีฟังก์ชันอาร์กิวเมนต์ทั้งหมด 4 ตัว และฟังก์ชันจะไม่คืนค่าอะไร

- ฟังก์ชันอาร์กิวเมนต์ตัวแรก เป็นค่าแบบ int ซึ่งเป็นตัวเลขจำนวนเต็ม 8 หลักที่เก็บวันเดือนปีในรูปแบบ mmddyyyy โดยที่ dd แทนวันที่ (00-31), mm แทนเดือน (01-12) และ yyyy แทนปี (0000-9999) เช่น 01282554 หมายถึง วันที่ 28 เดือน 1 ปี 2554

- ฟังก์ชันอาร์กิวเมนต์ตัวที่ 2, 3 และ 4 จะเป็นการส่งค่าแบบ pass by reference ซึ่งจะมีชนิดข้อมูลแบบ pointer to an int ซึ่งจะเป็นค่าของ address ของตัวแปรแบบ int ที่ใช้เก็บวันที่, ค่าของ address ของตัวแปรแบบ int ที่ใช้เก็บเดือน และ ค่าของ address ของตัวแปรแบบ int ที่ใช้เก็บปี ที่จะส่งมาจากฟังก์ชันตามลำดับ

```
#include <stdio.h>
void Extract_Date(      );
int main()
{
    int i_date, dd, mm, yyyy;
    scanf("%d", &i_date);
    Extract_Date (i_date, &dd, &mm, &yyyy);
    printf("%d\n", dd);
    printf("%d\n", mm);
    printf("%d\n", yyyy);
    return 0;
}

void Extract_Date(int date, int *day, int *month, int *year)
{
}
```

ตัวอย่างการทำงานของโปรแกรม

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
01282554	28 1 2554
12302010	30 12 2010

2. Problem name: Quadratic_equation

(Pointer: Pass-By-Reference)

จงทำโปรแกรมข้างล่างให้สมบูรณ์โดยการเขียนโค้ดในฟังก์ชัน Quad_Eq ซึ่งจะทำการหาค่า x ที่ทำให้สมการ $ax^2 + bx + c$ มีค่าเท่ากับ 0 โดยกำหนดให้ฟังก์ชันคืนค่าคำตอบ 2 ค่าซึ่งเป็นจำนวนจริง (real number) เป็นแบบ pass-by-reference โดยที่คำตอบที่เก็บอยู่ในตัวแปรที่แอดเดรสกำหนดโดย *sol1 จะต้องมีค่าน้อยกว่าหรือเท่ากับ คำตอบที่เก็บอยู่ในตัวแปรที่แอดเดรสกำหนดโดย *sol2

ตัวอย่างการทำงานของโปรแกรม

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
-12 40 1	-0.02 3.36
1 0 -36	-6.00 6.00

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

void Quad_Eq(double a, double b, double c, .....);

int main()
{
    double a,b,c,x1,x2;

    scanf("%lf %lf %lf",&a,&b,&c);
    Quad_Eq(a,b,c,.... .., ...);
    printf("%.2f\n",x1);
    printf("%.2f\n",x2);
    return 0;
}

void Quad_Eq(double a, double b, double c,.....)
{

}
```

3. Problem name: Min_Max_Avg_Std_Func

จงทำโปรแกรมข้างล่างให้สมบูรณ์โดยการเขียนโค้ดในฟังก์ชัน Min_Max_Avg_Std_Func และ main เพื่อทำให้โปรแกรมนี้คำนวณหาค่าเฉลี่ย (Average) และค่าเบี่ยงเบนมาตรฐาน (Standard deviation) ของข้อมูลจำนวนเต็มบวก โดยข้อมูลนำเข้าจะเป็นจำนวนเต็มบวกหรือศูนย์บรรทัดละตัว และสิ้นสุดด้วยค่าลบ เมื่อสิ้นสุดข้อมูลนำเข้าให้แสดงค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานออกสู่หน้าจอที่ละบรรทัด โดยให้แสดงเป็นเลขจุดทศนิยม 2 ตำแหน่ง และกำหนดให้ข้อมูลที่เข้ามามีไม่เกิน 100 ตัว

อาร์กิวเมนต์ของฟังก์ชัน Min_Max_Avg_Std_Func เป็นดังข้างล่าง โดย pData คือแอดเดรสของ element แรกของอินพุตอาร์เรย์ ซึ่งมีขนาด N element และให้ผลิตผลลัพธ์กับไปยังตัวแปรที่แอดเดรสส่งมาที่ pMin, pMax, pAvg และ pStd ตามลำดับ

```
void Min_Max_Avg_Std_Func(int *pData, int N,
                          int *pMin, int *pMax,
                          double *pAvg, double *pStd)
```

Note: ค่าเบี่ยงเบนมาตรฐาน (Standard deviation) ของข้อมูล $x_1, x_2, \dots, x_N$ คำนวณโดยใช้สูตรข้างล่าง

$$STD = \sqrt{\frac{1}{N} (x_i - \bar{x})^2}$$

ตัวอย่างการทำงานของโปรแกรม

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
12 31 10 1 41 32 -100	1 41 21.17 14.28
5 0 10 -1	0 10 5.00 4.08

```

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

void Min_Max_Avg_Std_Func(.....);

int main()
{
    int N,x,i,min,max;
    double avg,std;
    int data[100];

    N=0;
    while(1)
    {
        scanf("%d",&x);
        if(.....)
            break;

        .....

        .....

    }

    Min_Max_Avg_Std_Func_Pointer(.....);

    printf("%d\n%d\n%.2f\n%.2f\n",min,max,avg,std);

    return 0;
}

void Min_Max_Avg_Std_Func(int *pData,int N,
                           int *pMin, int *pMax,
                           double *pAvg, double *pStd)
{

}

```

4. Problem name: Concat_String

จงเขียนโปรแกรมรับข้อความ (string) จากคีย์บอร์ดจนกว่าผู้ใช้จะกดปุ่ม Enter จำนวน 3 ข้อความหลังจากนั้นให้สร้างข้อความที่เกิดจากนั้นข้อความที่ 1 มาต่อท้ายด้วยข้อความที่ 2 และที่ 3 และสร้างอีกหนึ่งข้อความที่เกิดจากเอาข้อความที่ 3 มาต่อท้ายด้วยข้อความที่ 2 และที่ 1

กำหนดให้เขียนและเรียกใช้ฟังก์ชัน `concat_string` ซึ่งรับอาร์กิวเมนต์ดังข้างล่าง นั่นคือจะทำต่อท้ายสตริงที่ระบุโดย `str1` ด้วยสตริงระบุโดย `str2`

```
void concat_string(char *str1,char *str2)
```

เช่น เมื่อฟังก์ชัน `concat_string` ข้างล่างทำงานเสร็จสตริงที่ระบุโดย `text1` จะเปลี่ยนเป็น "ABCDEFGHJIJ" ส่วนสตริง `text2` ไม่มีการเปลี่ยนแปลง

```
{  
char text1[100]="ABCDE";  
char text2[100]="FGHIJ";  
  
    concat_string(text1, text2);  
    // After text1 is ABCDEFGHIJ  
    ...  
}
```

กำหนดให้ความยาวของสตริงที่ป้อนเข้ามาไม่เกิน 100 ตัวอักษร

ตัวอย่างการทำงานของโปรแกรม

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
S T R 1 S T R 2 S T R 3	S T R 1 S T R 2 S T R 3 S T R 3 S T R 2 S T R 1
I am Smart	IamSmart SmartamI
123 456 789	123456789 789456123

```

#include <stdio.h>
#include <string.h>

void concat_string(char *,char *);

int main()
{
    char str1[.....],str2[.....],str3[.....],outstr[.....];
    int i;

        gets(str1);
        gets(str2);
        gets(str3);
        ostr[0]=0;
        concat_string(ostr,str1);
        concat_string(.....);
        concat_string(.....);
        printf("%s\n",ostr);
        ostr[0]=...
        concat_string(.....);
        concat_string(.....);
        concat_string(.....);
        printf("%s\n",ostr);
        return 0;
}

void concat_string(char *str1,char *str2)
{

```

5. Problem name: Pick_two_cards

สมมติมีไพ่อยู่ 10 ใบ ไพ่แต่ละใบมีค่าที่เป็นไปได้ในช่วง 1-10 จงเขียนโปรแกรมเพื่อเลือกไพ่ 2 ใบที่ให้ผลรวมมีค่าน้อยที่สุดแต่ต้องมากกว่าหรือเท่ากับ 10

กำหนดให้ข้อมูลนำเข้าคือข้อมูลจำนวนเต็มที่มีค่าในช่วง 1-10 บรรทัดละ 1 ตัว จำนวน 10 บรรทัด และให้แสดงผลลัพธ์เป็นค่าของไพ่ 2 ใบที่ถูกเลือก โดยเรียงจากน้อยไปมาก และผลรวมของไพ่ทั้งสองใบให้อยู่ในบรรทัดเดียวกัน

สมมติว่าไพ่ทั้ง 10 ใบที่เป็นข้อมูลนำเข้าจะมีไพ่ 2 ใบที่ผลรวมมากกว่าหรือเท่ากับ 10 เสมอ

ตัวอย่างการทำงานของโปรแกรม

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
3 3 4 6 8 10 1 1 5 6	4 6 10
3 3 5 9 8 10 9 10 9 8	3 8 11
10 10 9 8 7 2 5 6 6 8	2 8 10

```

#include <stdio.h>

int main()
{
    int i, j, X[10];
    int X1, X2, minsum, sum;

    for(i=0; i<10; i++)
    {
        scanf("%d", &X[i]);
    }

    minsum=.....

    for(i=0; i<10; i++)
    {
        for(j=.....)
        {
            if(.....)
            {
                sum=.....

                if(.....)
                {
                    minsum=.....

                    X1=.....

                    X2==.....
                }
            }
        }

        if(.....)
            printf("%d %d %d\n", X1, X2, minsum);
        else
            printf("%d %d %d\n", X2, X1, minsum);

        return 0;
    }
}

```