

CP 111 Computer Programming

2/2554 Section B01/B02

Lab 3

หมายเหตุ การเขียนโปรแกรมที่คำนวณกับตัวเลขที่มีจุดทศนิยม ให้ประกาศตัวแปรที่เกี่ยวข้องเป็นแบบ double (Double-precision floating point)

1. Problem name: Mean_Min_Max

จงเขียนโปรแกรมหาค่าเฉลี่ย (Mean) ค่าน้อยสุด (Minimum value) และค่ามากที่สุด (Maximum value) ของข้อมูล M ตัว ($M > 0$) โดยข้อมูลที่ป้อนเข้ามา (input) มีรูปแบบ (format) ดังนี้

- บรรทัดแรกบอกจำนวนของข้อมูลทั้งหมด สมมติ M
- ตั้งแต่บรรทัดที่ 2 ไปเป็นจำนวน M บรรทัดเป็นข้อมูลแต่ละตัว โดยกำหนดให้ข้อมูลเป็นตัวเลขแบบจำนวนเต็ม

เมื่อได้รับ input ครบแล้ว ให้แสดงค่าเฉลี่ย (แสดงเลขทศนิยม 2 หลัก) ค่าน้อยที่สุด และค่ามากที่สุด ออกมาที่หน้าจอที่ละบรรทัด

ตัวอย่างการทำงานของโปรแกรม

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
5 325 732 -520 983 881	480.20 -520 983

2. Problem name: Hailstones

ถ้ากำหนดให้ n_0 เป็นเลขจำนวนเต็มบวก (Positive Integer) ตัวหนึ่ง เราสามารถสร้างชุดของตัวเลข (a sequence of numbers) สำหรับ $i=0,1,2,\dots$ ตามสูตรข้างล่าง

ถ้า n_i เป็นจำนวนคู่ (Even)

$$n_{i+1} \leftarrow n_i / 2$$

ถ้า n_i เป็นจำนวนคี่ (Odd)

$$n_{i+1} \leftarrow 3n_i + 1$$

ถ้า n_i เท่ากับ 1

หยุดการสร้างชุดของตัวเลข

ชุดของตัวเลขที่ถูกสร้างโดยวิธีนี้เรียกว่า hailstones ตัวอย่างเช่น ถ้า $n_0=13$ จะได้ชุดของตัวเลข

13 40 20 10 5 16 8 4 2 1

ดังรายละเอียดข้างล่าง

เริ่มต้น $n_0=13$ (เลขคี่)

$$n_1 = 3 \cdot n_0 + 1 = 3 \cdot 13 + 1 = 40 \text{ (เลขคู่)}$$

$$n_2 = n_1 / 2 = 40 / 2 = 20 \text{ (เลขคู่)}$$

$$n_3 = n_2 / 2 = 20 / 2 = 10 \text{ (เลขคู่)}$$

$$n_4 = n_3 / 2 = 10 / 2 = 5 \text{ (เลขคี่)}$$

$$n_5 = 3 \cdot n_4 + 1 = 3 \cdot 5 + 1 = 16 \text{ (เลขคู่)}$$

$$n_6 = n_5 / 2 = 16 / 2 = 8 \text{ (เลขคู่)}$$

$$n_7 = n_6 / 2 = 8 / 2 = 4 \text{ (เลขคู่)}$$

$$n_8 = n_7 / 2 = 4 / 2 = 2 \text{ (เลขคู่)}$$

$$n_9 = n_8 / 2 = 2 / 2 = 1 \text{ หยุด}$$

Note: ปัญหาดังข้างบนเรียกว่า Collatz problem ซึ่งถึงแม้ยังไม่ได้ถูกพิสูจน์ แต่สามารถบอกได้ว่าตัวเลขทุกตัวสามารถนำมาสร้าง sequence ซึ่งสุดท้ายจะลงท้ายด้วย 1

(<http://projecteuler.net/index.php?section=problems&id=14>)

คำสั่ง: จงเขียนโปรแกรม เพื่อทำการแสดงชุดของตัวเลข hailstones ออกสู่หน้าจอตัวเลขละบรรทัด โดยที่ค่า n_0 จะถูกป้อนเข้ามาโดยผู้ใช้
ตัวอย่างการทำงานของโปรแกรม เมื่อผู้ใช้ป้อน $n_0=3$

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
3	10 5 16 8 4 2 1

```

#include <stdio.h>
int main()
{
    int n;

    scanf("%d",&n);

    while(          ) /* condition that makes the computation continue */
    {
        if(          ) /* n is even */
        {
            n=
        }
        else /* n is odd */
        {
            n=
        }
        printf("%d\n",n);
    }

    return 0;
}

```

3. Problem name: Prime_number

นิยาม Prime number หรือ จำนวนเฉพาะ คือ จำนวนเต็มบวกที่ 1 และตัวมันเองเท่านั้นที่สามารถหารตัวเลขนี้ลงตัว ตัวอย่างเช่น 2 3 5 7 11 13 17 19 เป็นต้น

จงเขียนโปรแกรมตรวจสอบว่าจำนวนที่ได้รับเข้ามาเป็น Prime number หรือไม่

Input	Output
จำนวนเต็มบวก N	ถ้า N เป็น prime number ให้พิมพ์ว่า N is a prime number. ไม่เช่นนั้นให้พิมพ์ N is not a prime number

ตัวอย่างการทำงานของโปรแกรม

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
89	89 is a prime number.
100	100 is not a prime number.

```
#include <stdio.h>
int main()
{
    int num,i,is_prime;

    scanf("%d",&num); /* num = number to be checked if it is prime. */
    is_prime=1;      /* is_prime = 1 means num is prime
                       is_prime = 0 means num is not prime */
    for(      ;      ;      )
    {
        if(      )
        {
            is_prime=0;
        }
    }
    if(is_prime==1)
    {
        printf(      );
    }
    else
    {
        printf(      );
    }
    return 0;
}
```

4. Problem name: Min_k_gt_N

จงเขียนโปรแกรมเพื่อทำการรับค่าตัวเลขจำนวนเต็มบวก 1 ตัวจากคีย์บอร์ด สมมติว่าเท่ากับ N โดยกำหนดให้ $N > 10$ หลังจากนั้นให้ทำการคำนวณหา ค่า k ที่น้อยที่สุด ที่ทำให้

$$1^2 + 2^2 + 3^2 + \dots + k^2 \geq N$$

หลังจากนั้นให้แสดงค่า k และผลรวม $1^2 + 2^2 + 3^2 + \dots + k^2$ ที่โปรแกรมหามาได้ออกสู่หน้าจอที่ละบรรทัด

ตัวอย่างการทำงานของโปรแกรม

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
20	4 30
10000	31 10416

หมายเหตุ

- กรณี Input = 20 จะเห็นได้ว่า $k=4$ คือค่าที่น้อยที่สุดที่ทำให้ $1^2 + 2^2 + 3^2 + 4^2 \geq 20$ และ $1^2 + 2^2 + 3^2 + 4^2 = 30$
- กรณี Input = 10000 จะเห็นได้ว่า $k=31$ คือค่าที่น้อยที่สุดที่ทำให้ $1^2 + 2^2 + 3^2 + 4^2 + \dots + 31^2 \geq 10000$ และ $1^2 + 2^2 + 3^2 + 4^2 + \dots + 31^2 = 10416$

```
#include <stdio.h>
int main()
{
    int N, k, sum;

    scanf("%d", &N);
    k=1;
    sum=0;
    while( sum < N )
    {
        k++;
        sum+=k*k;
    }
    printf("%d\n", k);
    printf("%d\n", sum);
    return 0;
}
```

5. Problem name: Exp_Approx

จงเขียนโปรแกรมที่ทำการคำนวณค่าประมาณของ e^x โดยใช้สูตร

$$e^x \approx 1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^k}{k!} + \dots + \frac{x^n}{n!} = \sum_{k=0}^n \frac{x^k}{k!}$$

โปรแกรมจะรับอินพุต 2 จำนวนคั่นด้วยช่องว่างในบรรทัดเดียวกัน อินพุตแรกคือค่า x ส่วนอินพุตตัวที่สองคือค่า n ที่จะระบุเทอมสุดท้าย ($\frac{x^n}{n!}$) ที่จะใช้ในการคำนวณหาค่าประมาณของ e^x นั่นคือจะใช้เทอมที่ 0 ถึงเทอมที่ n ทั้งหมด n+1 เทอมในการคำนวณหาค่า หลังจากนั้นโปรแกรมจะต้องแสดงค่าประมาณ e^x ที่คำนวณได้ออกสู่หน้าจอโดยให้แสดงเลขทศนิยม 6 ตำแหน่ง

ตัวอย่างการทำงานของโปรแกรม

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
0.5 100	1.648721
-0.5 20	0.606531
1 2	2.500000
1 100	2.718282

Hint:

```
#include <stdio.h>
int main()
{
    double e_x,x,x_k,k_fact;
    int k,n;
    scanf("%lf %d",&x,&n);
    e_x =1; /*start with 1 (term 0) */
    x_k= 1;
    k_fact=1;
    for(k=1;    ; )
    {
        x_k=          ; /* x^k */
        k_fact=        ; /* k! */
        e_x = e_x +    ;
    }
    printf("%.6f\n",val);
    return 0;
}
```