

CP 111 Computer Programming

2/2554 Section B01/B02

Lab 9

หมายเหตุ การเขียนโปรแกรมที่คำนวณกับตัวเลขที่มีจุดทศนิยม ให้ประกาศตัวแปรที่เกี่ยวข้องเป็นแบบ double (Double-precision floating point)

1. Problem name: Pos_Neg_Avg_Std_Func_Pointer_Offset_Notation

กำหนดให้เขียนโปรแกรมให้ใช้การอ้างถึง **element** ในอาร์เรย์โดยใช้ตัวแปรแบบ **pointer** ในรูปแบบ **pointer-offset notation** เช่น ***ptr** หรือ ***(ptr+i)**

จงเขียนและเรียกใช้ฟังก์ชัน Pos_Neg_Avg_Std_Func

```
void Pos_Neg_Avg_Std_Func(int *x, int N,  
                        double *pos_avg, double *pos_std,  
                        double *neg_avg, double *neg_std);
```

เพื่อสร้างโปรแกรมที่ทำการรับค่าเลขจำนวนเต็มที่มีค่าไม่เท่ากับศูนย์ (Nonzero integers) ผ่าน keyboard เข้ามาเรื่อยๆ โดยจะหยุดการรับค่าก็ต่อเมื่อผู้ใช้ป้อนค่า 0 โดยสมมติว่าข้อมูลจะมีไม่เกิน 100 จำนวน

หลังจากนั้นโปรแกรมจะแสดงค่าเฉลี่ย (average) และค่าเบี่ยงเบนมาตรฐาน (standard deviation) ที่คำนวณจากจำนวนบวกทั้งหมด (positive numbers) และค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานที่คำนวณจากจำนวนลบทั้งหมด (negative numbers) แล้วแสดงออกสู่หน้าจอ โดยให้แสดงค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานของจำนวนบวกที่บรรทัดแรก แล้วจึงตามด้วยค่าที่คำนวณได้ของจำนวนลบในบรรทัดที่สอง โดยให้แสดงจุดทศนิยม 2 ตำแหน่ง

ตัวอย่างการทำงานของโปรแกรม

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
-1 10 -5 15 -3 -7 5 0	10.00 4.08 -4.00 2.24

Hint:

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

void Pos_Neg_Avg_Std_Func(int *X, int N, double *pos_avg, double *pos_std,
                        double *neg_avg, double *neg_std);

int main()
{
    int N,x;
    int data[100];
    int *ptr;
    double pos_avg,pos_std,neg_avg,neg_std;

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

        .....

        .....

    };

    Pos_Neg_Avg_Std_Func(data,&pos_avg,&pos_std,&neg_avg,&neg_std );

    printf("%.2f %.2f\n",pos_avg,pos_std);
    printf("%.2f %.2f\n",neg_avg,neg_std);

    return 0;
}
```

กำหนดให้การเขียนโปรแกรมให้ใช้อ้างถึง
element ในอาร์เรย์โดยใช้ตัวแปรแบบ pointer ใน
รูปแบบ **pointer-offset notation** เช่น ***ptr** หรือ
***(ptr+i)**

Hint (ต่อ):

```
void Pos_Neg_Avg_Std_Func(int *X, int N, double *pos_avg, double *pos_std,
                          double *neg_avg, double *neg_std)
{
    int pos_sum, neg_sum, n_pos, n_neg, pos_sum_sq, neg_sum_sq;
    int i;
        pos_sum=0;
        neg_sum=0;
        n_pos=0;
        n_neg=0;
        for(i=0;i<N;i++)
        {
            if(.....)
            {
                pos_sum=pos_sum+      ;

                ...
            }
            else
            {
                neg_sum=neg_sum+      ;

                ...
            }
        }

        *pos_avg=...

        *neg_avg=...

        // Calculate Standard deviations
        pos_sum_sq=0;
        neg_sum_sq=0;
        for(i=0;i<N;i++)
        {
            if(.....)
            {
                pos_sum_sq = pos_sum_sq +      ;

            }
            else
            {
                neg_sum_sq = neg_sum_sq +      ;

            }
        }
        *pos_std=...;
        *neg_std=...;
}
```

กำหนดให้การเขียนโปรแกรมให้ใช้อ้างถึง
element ในอาร์เรย์โดยใช้ตัวแปรแบบ pointer ใน
รูปแบบ **pointer-offset notation** เช่น ***ptr** หรือ
***(ptr+i)**

2. Problem name: Sort_N

จงเขียนโปรแกรมเพื่อเรียงลำดับ (sort) ชุดของข้อมูลจำนวนเต็ม $\{x_i; i=1, 2, \dots, N\}$ จากค่าน้อยไปค่ามาก กำหนดให้ข้อมูลนำเข้าบรรทัดแรกคือจำนวนข้อมูล N หลังจากบรรทัดที่ 2 ถึง $N+1$ คือข้อมูล $x_i; i=1, \dots, N$ และ $N \leq 100$

ตัวอย่างการทำงานของโปรแกรม

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
10 898 235 -1234 788 -100000 10000000 789 788 0 12	-100000 -1234 0 12 235 788 788 789 898 10000000

หมายเหตุ อัลกอริทึมในการเรียงลำดับข้อมูลโดยวิธี Insertion sort แสดงได้ดัง Pseudo-code ข้างล่าง

Insertion sort Input: Array A of n elements
<pre>For j ← 1 to N-1 key ← A[j] // A[j] is added in the sorted // sequence A[1, .. j-1] i ← j - 1 While i ≥ 0 and A [i] > key A[i +1] ← A[i] i ← i -1 A [i +1] ← key</pre>

Hint:

```
#include <stdio.h>
void InsertionSort(int *A,int n);
int main()
{
    int data[100];
    int N,i;
    scanf("%d",&N);
    for(...)
        scanf(...);

    InsertionSort(...);

    for(...)
        printf(...);
    return 0;
}
```

```
void InsertionSort(int *A,int n)
{
```

```
    int i,j,key;
```

```
    // ==> ลูปนี้ j จะมีค่าเป็น 1, 2,.. n-1
```

```
    for( j=1 ; j<n ; j++ )
```

```
    {
```

```
        key=A[j]; // เช็ตค่า key ให้เท่ากับ A[j] (ไฟโบนแรกๆของกองไฟ unordered)
```

```
        i=j-1;
```

```
        while(1)
```

```
        {
```

```
            // ==> ลูปค้นหา A[i] โดยที่ i จะมีค่าเป็น j-1, j-2,... , 1, 0
```

```
            // ==> ซึ่ง A[i] < key (ไฟโบนปัจจุบันควรจะแทรกหลัง A[i])
```

```
            if(..... || ..... )
```

```
                break;
```

```
            // ==> ถ้ายังไม่เจอ A[i] ที่ A[i]<key ให้ขยับ A[i] ไปทางขวา (A[i] ==> A[i+1])
```

```
            .....
```

```
            i=i-1;
```

```
        }
```

```
        // ==> หลุดจากลูป while ข้างบนเมื่อเจอ A[i] ที่ <key หรือ i < 0 ให้นำ key ไปต่อท้าย A[i]
```

```
        .....
```

```
    }
```

```
}
```

// ==> ในแต่ละรอบ j จะได้ A[0]...A[j-1] เรียงจากน้อยไปมากแล้ว

// ==> ให้นำ A[j] ไปใส่ใน A[0]...A[j-1] ในลำดับที่ควรจะเป็น

// ==> นั่นคือ A[0]...A[j-1] = กองไฟ sorted

// ==> A[j]...A[n-1] = กองไฟ unordered

// ==> โดยที่ A[j] คือไฟโบนแรกของกองไฟ unordered

3. Problem name: *Median*

จงเขียนโปรแกรมเพื่อหาค่ามัธยฐาน (Median) ของชุดข้อมูลหนึ่ง โดยกำหนดให้ข้อมูลนำเข้าเป็นจำนวนเต็มที่อยู่ในช่วง $0 - (2^{31})-1$ ที่ถูกป้อนเข้ามาเรื่อยๆ จนกว่าจะเจอค่าลบซึ่งถือเป็นจุดสิ้นสุดของข้อมูล โดยที่ข้อมูลนำเข้าจะถูกป้อนเข้ามาไม่เกิน 100 ตัว โดยกำหนดให้แสดงผลลัพธ์เป็นเลขจำนวนเต็มไม่มีจุดทศนิยม

Note: ในกรณีที่จำนวนข้อมูลเป็นเลขคี่ เช่น 5 จำนวน สมมติว่าเป็น $\{a, b, c, d, e\}$ โดยที่ $a < b < c < d < e$ ค่า median คือ c ส่วนในกรณีที่จำนวนข้อมูลเป็นเลขคู่ เช่น 6 จำนวน สมมติว่าเป็น $\{a, b, c, d, e, f\}$ โดยที่ $a < b < c < d < e < f$ ค่า median คือ $(c+d)/2$

ตัวอย่างการทำงานของโปรแกรม

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
16 11 31 14 5 -1	14
16 11 31 14 5 17 -1	15

หมายเหตุ ค่า Median ในตัวอย่างที่สองเท่ากับ $(14+16) / 2 = 15$

Hint: ใช้โค้ดที่เขียนในโจทย์ข้อ Sort_N

```

#include <stdio.h>

void InsertionSort(int *A, int n);

int main()
{
    int x,N,med;
    int X[100];

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

        if(_____)
            break;

        _____;

        _____;

    }
    while();

    InsertionSort (X,N);

    if(_____)
    {
        med=X[_____];
    }
    else
    {
        med=(X[_____] + X[_____])/2;
    }
    printf("%d\n",med);

    return 0;
}

void InsertionSort(int *A, int n)
{

}

```

4. Problem name: Matrix_Multiplication

จงเขียนโปรแกรมเพื่อทำการคูณ matrix ขนาด $N \times N$ ($N \leq 20$) จำนวน 2 ตัวสมมติว่าคือ A และ B โดยที่สมาชิกใน matrix เป็นค่าจำนวนเต็ม และแสดงผลลัพธ์ $A*B$ ออกสู่หน้าจอ โดยกำหนดให้ข้อมูลนำเข้าบรรทัดแรกคือค่า N หลังจากนั้น N บรรทัดถัดมาจะเป็นข้อมูลของ matrix A และอีก N บรรทัดถัดมาจะเป็นข้อมูลของ matrix B โดยที่ในแต่ละบรรทัดจะประกอบไปด้วยตัวเลข N จำนวน

ตัวอย่างการทำงานของโปรแกรม

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
3 1 2 3 4 5 6 7 8 9 -1 -2 -3 -4 -5 -6 -7 -8 -9	-30 -36 -42 -66 -81 -96 -102 -126 -150
4 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8

```

#include <stdio.h>
#define MAX_N 20

int main()
{
    int .....;
    int N,i,j,k;

    scanf("%d",&N);
    // Reading Matrix A
    for(.....)

        for(.....)

            scanf(.....);

    // Reading Matrix B
    for(.....)

        for(.....)

            scanf(.....);

    for(.....)
    {
        for(.....)
        {
            C[i][j]=.....;
            for(k=...;k<...;k++)
                C[i][j]=C[i][j]+(A[.....][.....]*B[.....][.....]);

        }
    }

    for(i=0;i<N;i++)
    {
        for(j=0;j<N;j++)
        {
            printf("%d ",C[i][j]);
        }
        printf("\n");
    }

    return 0;
}

```

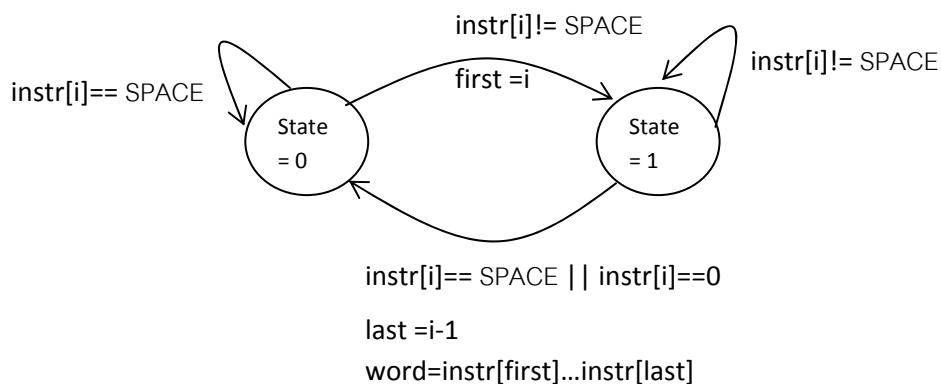
5. Problem name: Extract_word

จงเขียนโปรแกรมรับข้อความ (string) จากคีย์บอร์ดจนกว่าผู้ใช้จะกดปุ่ม Enter หลังจากนั้นให้นำข้อความมาแยกเป็นคำ (Word) โดยที่นิยามของคำคือชุดของตัวอักษรที่ติดกันโดยไม่มีช่องว่างคั่น และให้แสดงคำที่แยกออกมาได้ออกสู่หน้าจอ บรรทัดละคำ พร้อมทั้งแสดงจำนวนตัวอักษรของแต่ละคำออกมาด้วย

ตัวอย่างการทำงานของโปรแกรม

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
I am cute	I 1 am 2 cute 4
You are so beautiful	You 3 are 3 so 2 beautiful 9
A man who loves a woman	A 1 man 3 who 3 loves 5 a 1 woman 5

Hint: เราสามารถเขียนโปรแกรมที่การ scan ตัวอักษรใดๆ ของ instr ทำงานตาม state diagram ข้างล่าง



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

```
int main()
{
    char instr[501];
    int i, state;
```

```
    gets(instr);
```

```
    state=0;
    i=0;
    while(1)
    {
```

```
        if(state==0)
        {
            if(instr[i]!=' ')
            {
                first=...;
                state=...;
            }
        }
```

```
    else if(state==1)
    {
```

```
        if(.....)
        {
            last=.....;
            state=.....;
            for(k=.....;k<=.....;k++)
            {
                printf("%c",instr[k]);
            }
            printf(" %d\n",.....);
        }
```

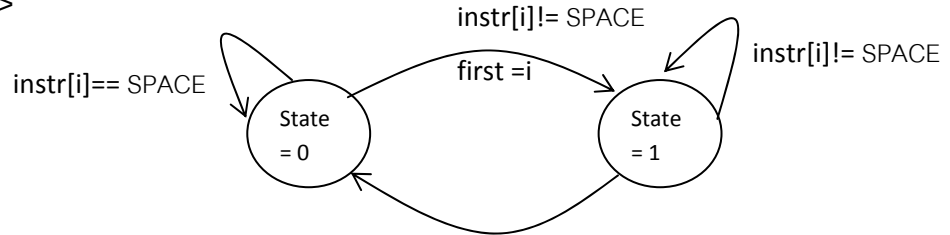
```
        if(.....)
            break;
```

```
    }
```

```
    i=.....
```

```
    }
    return 0;
```

```
}
```



```
instr[i]== SPACE || instr[i]==0
```

```
last =i-1
```

```
word=instr[first]...instr[last]
```