

CP 111 Computer Programming, 2/2554, Section B01/B02

Programming Homework 08

กำหนดส่งภายในวันจันทร์ที่ 20 ก.พ. 2555 เวลา 23.59 น.

คำเตือน

นิสิตสามารถจับกลุ่มปรึกษาหารือกันได้ แต่นิสิตแต่ละคนในกลุ่มจะต้องลงมือเขียนโปรแกรมด้วยตนเองและส่งโปรแกรมของตนเองโดยไม่ขึ้นอยู่กับสมาชิกในกลุ่ม และอย่านำโปรแกรมของตนเองให้กับผู้อื่นดู
ถ้าตรวจพบว่านิสิตนำงานของคนอื่นมาส่งหรือนิสิตก๊อปปี้ แก้ไข คัดแปลงโปรแกรมของผู้อื่น จะมีโทษคือปรับตก (ได้เกรด E) ในวิชานี้ทันที

หมายเหตุ

- ข้อกำหนดการตั้งชื่อไฟล์การบ้านคือ xxx_HWyy_Qzz.* โดยที่ xxx = รหัสสามตัวท้ายของเลขประจำตัว, yy = หมายเลข homework, zz = หมายเลขข้อในแต่ละ homework (ตัวอย่างเช่น ไฟล์โปรแกรม ข้อ 1 การบ้านครั้งที่ 1 ของนิสิตรหัส 000 คือ 000_HW01_Q01.c) และในแต่ละข้อ
- การส่งการบ้าน ให้ส่งไฟล์โปรแกรมไปตรวจกับระบบ Café-grader ที่ <http://cslab.swu.ac.th/grader>
- การเขียนโปรแกรมที่คำนวณกับตัวเลขที่มีจุดทศนิยม ให้ประกาศตัวแปรที่เกี่ยวข้องเป็นแบบ double (Double-precision floating point)

¹ เรียบเรียงจากหนังสือเริ่มเรียนเขียนโปรแกรม ฉบับวจาจาวา ของ สมชาย ประสิทธิ์จิตรกุล

A[0]	A[1]	A[2]	A[3]	A[4]
13	20	42	67	27

ในรอบถัดไปลดขนาดของอาร์เรย์
ย่อยลง 1 กลายเป็น A[2]..A[4]

A[0]	A[1]	A[2]	A[3]	A[4]
13	20	42	67	27

ค่าน้อยสุดใน A[2]..A[4] คือ 27 ให้
สลับ A[4] (20) กับ A[2] (ซ้ายสุด)

A[0]	A[1]	A[2]	A[3]	A[4]
13	20	27	67	42

ในรอบถัดไปลดขนาดของอาร์เรย์
ย่อยลง 1 กลายเป็น A[3]..A[4]

A[0]	A[1]	A[2]	A[3]	A[4]
13	20	27	42	67

ในรอบถัดไปลดขนาดของอาร์เรย์
ย่อยลง 1 กลายเป็น A[4] เสร็จ

```

#include <stdio.h>

void SelectionSort(int *A,int n);

int main()
{
    int data[100];
    int N,i;
        scanf("%d",&N);
        for(...)
            scanf(...);

        SelectionSort(...);

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

void SelectionSort(int *A,int n)
{
    int i,j,min_index,min_val,tmp;

/*
- ในแต่ละรอบ i โดยที่ i จะมีค่าเป็น 0, 1, ... n-2 จะทำการหาค่าที่น้อยที่สุดในบรรดาข้อมูลที่อยู่ใน subarray A[i] ... A[n-1]
- สมมติว่าในแต่ละรอบ i หลังจากหาค่าน้อยสุดเราได้ min_val ค่าน้อยสุดและ min_index คือค่าอาร์เรย์ index ของตัวที่เป็นค่าน้อยสุด
- เราจะทำการทำการสลับค่าระหว่าง A[i] กับ A[min_index] หลังจากนั้นจะเพิ่มค่า i ด้วย 1 แล้วเริ่มต้นในรอบถัดไป
*/

        for(i=...;i<...;...)
        {
            // find min among A[i] ... A[n-1]
            min_val=...
            min_index=...
            for(j=...;j<...;...)
            {
                if(...)
                {
                    min_val=....
                    min_index=...
                }
            }
            // Swap A[i] <-> A[min_index]
            tmp=...
            ...
            ...
        }
}

```

2. Problem name: What_in_parentheses

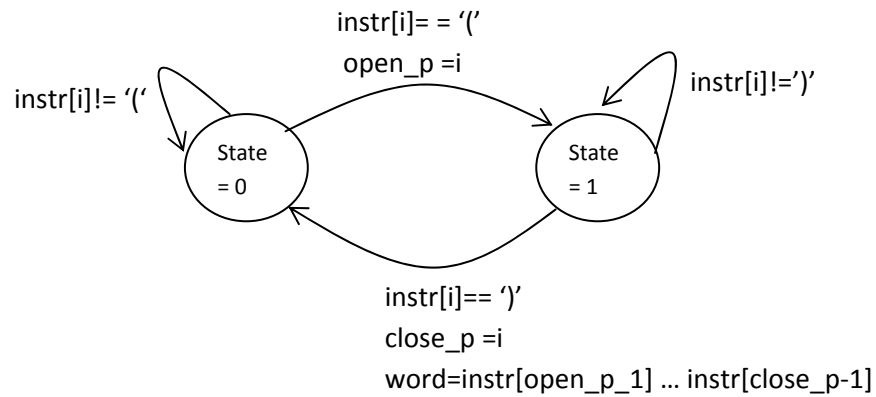
จงเขียนโปรแกรมเพื่อดึงเฉพาะข้อความที่อยู่ภายในเครื่องหมายวงเล็บ '(' และ ')' ออกมาจากสตริงนำเข้า โดยกำหนดให้สตริงนำเข้ามีความยาวไม่เกิน 500 ตัวอักษร และจะมีข้อความที่อยู่ในวงเล็บอย่างน้อย 1 ข้อความ โดยให้แสดงผลลัพธ์ในแต่ละบรรทัดเป็นจำนวนตัวอักษรในข้อความและข้อความที่ค้นเจอเรียงตามลำดับ

หมายเหตุ กำหนดให้ภายใน () ของสตริงทดสอบจะมีตัวอักษรอย่างน้อย 1 ตัว และไม่มีสตริงทดสอบใดที่มีวงเล็บซ้อนกัน เช่น (a+(b+c)) หรือ (Computer Sci (CS)) เป็นต้น และไม่มีสตริงทดสอบใดที่มีวงเล็บที่ไม่สมบูรณ์ เช่นสตริง CP111 (Computer Programming หรือสตริง a+b*c) เป็นต้น

ตัวอย่างการทำงานของโปรแกรม

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
I am a lecturer at Srinakharinwirot U (SWU). I live in Bangkok (BKK)	3 SWU 3 BKK
*(p[1])=(a+b)*2-(c*d-e/f);	4 p[1] 3 a+b 7 c*d-e/f
(pointer)	7 pointer
(a)(b2c)(d)	1 a 3 b2c 1 d

Hint: สามารถดัดแปลงโปรแกรม Extract_word ใน lab 9 โดย State diagram



```
#include <stdio.h>

int main()
{
    char instr[501];
    int i, k, state, open_p, close_p;

    gets(instr);

    i = ...
    state = ..
    while(1)
    {
        if(state == )
        {
            if( )
            {
                state =
                open_p =
            }
        }
        else if(state == )
        {
            if( )
            {
                close_p =
                word in ( ) is instr[open_p_1] to instr[close_p-1]
                state =
            }
        }
        if( )
            break;
        i =
    }
    return 0;
}
```

3. Problem name: Max_consecutive_sum

จงเขียนโปรแกรมเพื่อหาผลรวมของข้อมูลที่อยู่ติดกันในอาร์เรย์หนึ่งๆ ที่มีค่ามากที่สุด เช่น อาร์เรย์ คือ {2, -1, 3, -5, 4} ผลรวมของข้อมูลที่อยู่ติดกันที่เป็นไปได้ คือ

2	-1	3	-5	4
2+-1=1	-1+3=2	3+-5=-2	-5+4=-1	
2+-1+3=4	-1+3+-5=-3	3+-5+4=2		
2+-1+3+-5=-1	-1+3+-5+4=1			
2+-1+3+-5+4=3				

ค่าผลรวมของข้อมูลที่อยู่ติดกันที่มีค่ามากที่สุดคือ $2+-1+3=4$

กำหนดให้อาร์เรย์มีรูปแบบของข้อมูลนำเข้า คือ บรรทัดที่ 1 เป็นตัวเลขจำนวนเต็มสมมติว่าเท่ากับ N โดย N จะมีค่าไม่เกิน 100 ส่วนบรรทัดที่ 2 ถึง บรรทัดที่ N+1 จะเป็นข้อมูลแบบจำนวนเต็มบรรทัดละตัว โปรแกรมจะแสดงผลคือค่าผลรวมของข้อมูลที่อยู่ติดกันที่มีค่ามากที่สุดที่ตรวจเจอ

ตัวอย่างการทำงานของโปรแกรม

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

Hint: ดู paper homework

```
#include <stdio.h>

int main()
{
    int X[100];
    int N,i,maxsum,first,last;

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

    Max_Consecutive_Sum(X,N,...)

    printf("%d\n",maxsum); // first และ last ไม่ได้ใช้
    return 0;
}

void Max_Consecutive_Sum(int *A,int N, int *max_sum,int *first, int *last)
{
    int i,j,nC,sum;

    // i ระบุ index ของตัวแรกในข้อมูลที่ติดกันที่จะนับรวมใน sum
    for(i=.....;.....;.....)
    {
        if(.....)
        {
            *max_sum=.....
            *first=.....
            *last=.....
        }
        // nC ระบุจำนวนข้อมูลที่ติดกันที่จะนับรวมใน sum
        for(nC=0; .....; ..... )
        {
            sum= .....
            // j ระบุ index ของตัวสุดท้ายในข้อมูลที่ติดกันที่จะนับรวมใน sum
            for(j=0; .....; ..... )
            {
                sum=.....
                if(.....)
                {
                    .....
                    .....
                    .....
                }
            }
        }
    }
}
```

ตัวอย่างการทำงานของโปรแกรม (ต่อ)

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
10 -1 -2 -3 -4 11 -5 5 4 3 2	20
6 -1 1 -1 3 -3 2	3
10 -1 -2 1 2 -3 6 -4 3 4 -5	9
8 10 2 3 -5 4 5 5 -4	24

ตัวอย่างการทำงานของโปรแกรม

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
24 10 -1 -2 1 2 -3 6 -4 3 10 -1 -2 1 2 -3 6 -4 3 4 -5 -6 4 -5 5	28
16 1 2 -5 -4 -3 -7 4 3 8 -2 50 -40 50 10 -100 80	83

4. Problem name: Reverse_word

จงเขียนโปรแกรมเพื่อทำการสลับลำดับของตัวอักษรของคำหนึ่งๆ ที่อยู่ในสตริงจากท้ายไปหน้า โดยที่ลำดับของคำในสตริงผลลัพธ์ยังเหมือนเดิม กำหนดให้ข้อความนำเข้าจะประกอบด้วยตัวอักษรที่ไม่ใช่ space ที่เรียงต่อกันเป็นคำ และแต่ละคำจะคั่นด้วยตัวอักษร space อย่างน้อย 1 ตัว ตัวอย่างเช่น

"abcdefg hijklmnop qrs tuv wux y z " หลังจากผ่านการประมวลผลของโปรแกรมแล้วจะได้สตริงผลลัพธ์เป็น "gfedcba ponmlkjih srq vut xuw y z "

ตัวอย่างการทำงานของโปรแกรม

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
how can you tell an extrovert	woh nac uoy llet na trevortxe
I want to know how tough you can be	I tnaw ot wonk woh hguot uoy nac eb
a bb c de fgh	a bb c ed hgf
All truths are easy to understand	llA shturt era ysae ot dnatsrednu
once they are discovered;	ecno yeht era ;derevocsid
the point is to discover them.	eht tniop si ot revocsid .meht

Hint: ดัดแปลงโค้ดจาก Extract_word และ Reverse_String

5. Problem name: Substring

สตริง B เป็นสตริงย่อย (substring) ของ A ก็ต่อเมื่อมีช่วงของตัวอักษรใน A ที่เรียงติดกันแล้วเท่ากับ B ตัวอย่างเช่น ถ้า A = hellosmallworld และ B = small กรณีนี้ B จะเป็นสตริงย่อยของ A โดยตำแหน่งของสตริงย่อยคือตำแหน่งที่ 5 (ตำแหน่งแรกคือ 0) โดยกำหนดให้การเปรียบเทียบเป็นแบบ case sensitive นั่นคือ ตัวอักษรเล็กและใหญ่จะพิจารณาเป็นคนละตัวกัน เช่น 'H' ไม่เท่ากับ 'h'

กำหนดให้ข้อมูลนำเข้ามี 2 บรรทัด บรรทัดแรกเป็นสตริง A และบรรทัดที่สองเป็นสตริง B โดยที่แต่ละสตริงมีความยาวไม่เกิน 500 ตัวอักษร จงเขียนโปรแกรมเพื่อคำนวณหาตำแหน่งของสตริงย่อย B ใน A

กรณีที่ B เป็นสตริงย่อยของ A ให้พิมพ์ตำแหน่งเริ่มต้นของสตริงย่อยนั้นออกมา (กำหนดให้ตัวอักษรตัวแรกนับเป็นตำแหน่ง 0)

กรณีที่ B ไม่เป็นสตริงย่อยของ A ให้พิมพ์ -1

ตัวอย่างการทำงานของโปรแกรม

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
This world is lovely orld	6
Lady Gaga gaga	-1
hellosmallworld small	5
hellosmallworld d	14

Hint: 📄 paper homework

```
#include <stdio.h>

int substring(char *instr, char *t_str);

int main()
{
    char instr[501];
    char t_str[501];
    int match;

    gets(instr);
    gets(t_str);

    match=substring(instr, t_str);

    printf(...);

    return 0;
}

int substring(char *instr, char *t_str)
{
    int i, k, match;
    i=.....
    while(.....)
    {
        k=.....
        match=1;
        while(.....)
        {
            if(.....)
            {
                match=.....
                .....
            }
            .....
        }
        if(match==1)
            .....
        .....
    }
    .....
}
```

6. Problem name: *Matrix_Transpose*

กำหนดให้ A เป็น matrix ขนาด $M \times N$ ($M, N \leq 20$) แล้ว A^T หมายถึง Transpose ของเมทริกซ์ A ซึ่งนิยามคือ ถ้า $A=[a_{ij}]_{M \times N}$ แล้ว $A^T=[a_{ji}]_{N \times M}$ เช่น

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow A^T = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

จงเขียนโปรแกรมเพื่อทำการหาค่าผลคูณ $A^T * A$ ซึ่งจะเป็น matrix ขนาด $N \times N$ โดยกำหนดให้ข้อมูลนำเข้าบรรทัดแรกคือค่า M และ N หลังจากนั้น M บรรทัดถัดมาจะเป็นข้อมูลของ matrix A โดยแต่ละบรรทัดจะเป็นข้อมูลแบบจำนวนเต็มจำนวน N ตัว

ตัวอย่างการทำงานของโปรแกรม

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
2 3 1 2 3 4 5 6	17 22 27 22 29 36 27 36 45
1 4 1 2 3 4	1 2 3 4 2 4 6 8 3 6 9 12 4 8 12 16
4 1 5 6 7 8	174
3 3 1 1 1 2 2 2 3 3 3	14 14 14 14 14 14 14 14 14

7. Problem name: Straight (ไฟต้อง) (New problem for Grader 2/2554)

สมมติมีไฟอยู่ 10 ใบ ไฟแต่ละใบมีค่าที่เป็นไปได้ในช่วง 1-10 จงเขียนโปรแกรมเพื่อเลือกไฟ 3 ใบที่เป็นไฟต้อง (เรียงกัน) ที่ดีที่สุด (ให้ผลรวมมีมากที่สุด)

กำหนดให้ข้อมูลนำเข้าคือข้อมูลจำนวนเต็มที่มีค่าในช่วง 1-10 บรรทัดละ 1 ตัว จำนวน 10 บรรทัด และให้แสดงผลลัพธ์เป็นค่าของไฟ 3 ใบที่ถูกเลือก โดยเรียงจากน้อยไปมาก ในกรณีที่ไม่มีไฟต้องอยู่ในชุดข้อมูลนำเข้าให้แสดง -1

ตัวอย่างการทำงานของโปรแกรม

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

Hint: ใช้ sorting เรียงไฟจากน้อยไปมากก่อนแล้วจึงไล่หาไฟที่มีค่าเรียงติดกัน 3 โบกจากน้อยไปมาก

```
#include <stdio.h>
int main()
{
    int i,j,k,X[10];
    int X1,X2,X3,found;

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

    Use any sort function i.e. InsertionSort or SelectionSort

    found=0;
    for(i=0;i<10;i++)
    {
        for(j=i+1;j<...;...)
        {
            for(k=j+1;k<...;..)
            {
                if(X[i]+1==... && ..... == ...)
                {
                    found=...
                    X1=
                    X2=
                    X3=
                }
            }
        }
    }
    if(found!=0)
    {
        printf("%d %d %d\n",X1,X2,X3);
    }
    else
    {
        printf(    );
    }

    return 0;
}
```