

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

### Programming Homework 09

กำหนดส่งภายในวันศุกร์ที่ 23 ก.พ. 2555 เวลา 23.59 น.

#### คำเตือน

นิสิตสามารถจับกลุ่มปรึกษาหารือกันได้ แต่นิสิตแต่ละคนในกลุ่มจะต้องลงมือเขียนโปรแกรมด้วยตนเองและส่งโปรแกรมของตนเองโดยไม่ขึ้นอยู่กับสมาชิกในกลุ่ม และอย่านำโปรแกรมของตนเองให้กับผู้อื่นดู  
ถ้าตรวจพบว่านิสิตนำงานของคนอื่นมาส่งหรือนิสิตก๊อปปี้ แก้ไข ดัดแปลงโปรแกรมของผู้อื่น 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. Problem name: Recur\_GCD\_Func

ขั้นตอนในการหาตัวหารร่วมมาก (ห.ร.ม.) หรือ Greatest Common Divisor (GCD) แบบ Euclid สามารถอธิบายโดยใช้ Recurrence equation ข้างล่าง

$$\text{gcd}(m,n) = \begin{cases} m; & \text{if } n = 0 \\ n; & \text{if } m = 0 \\ \text{gcd}(n, m \% n) & \text{if } m \neq 0 \text{ and } n \neq 0 \end{cases}$$

จงทำโปรแกรมข้างล่างให้สมบูรณ์ โดยการเขียนโค้ดในฟังก์ชัน Recur\_GCD ในรูปแบบ Recursive function (ฟังก์ชันเรียกตัวเอง) ซึ่งจะรับอาร์กิวเมนต์ 2 ตัวคือ m และ n โดยที่จะคืนค่า gcd ของ m และ n

```
#include <stdio.h>
int Recur_GCD(int m, int n);
int main()
{
    int a,b,gcd;
    scanf("%d %d",&a,&b);
    gcd=Recur_GCD(a,b);
    printf("%d\n",gcd);
    return 0;
}
int Recur_GCD(int m, int n)
{
}
```

ตัวอย่างการทำงานของโปรแกรม

| ข้อมูลนำเข้า (Input) | ข้อมูลส่งออก (Output) |
|----------------------|-----------------------|
| 49 84                | 7                     |
| 125 75               | 25                    |

## 2. Problem name: Recur\_Maximum

จงใช้เทคนิค Recursion เพื่อเขียนโปรแกรมทำการหาข้อมูลที่มากที่สุดในอาร์เรย์ โดยกำหนดให้ข้อมูลนำเข้าในบรรทัดแรกคือ n ซึ่งแทนจำนวน element ในอาร์เรย์ หลังจากนั้น n บรรทัดถัดมา จะเป็นข้อมูลแบบ int แทนข้อมูลในแต่ละ element ของอาร์เรย์ กำหนดให้ข้อมูลในอาร์เรย์มีไม่เกิน 100 ตัว และ ให้เขียนและเรียกใช้ฟังก์ชัน Recur\_Maximum ดังนิยามข้างล่าง

```
int Recur_Maximum(int *A, int n)
```

ซึ่งฟังก์ชันนี้จะทำหน้าที่ในการหาค่าน้อยสุดในอาร์เรย์ A ที่มีขนาด n ตัว (A[0], A[1], ...A[n-1])

```
#include <stdio.h>
int Recur_Maximum(int *A, int n);
int main()
{
    int data[100],i,n,max;
        scanf("%d",&n);
        for(i=0;i<n;i++)
            scanf("%d",&data[i]);
        max= Recur_Maximum(data,n);
        printf("%d\n",max);
        return 0;
}
int Recur_Maximum(int *A, int n)
{
}
}
```

ตัวอย่างการทำงานของโปรแกรม

| ข้อมูลนำเข้า (Input)                  | ข้อมูลส่งออก (Output) |
|---------------------------------------|-----------------------|
| 5<br>325<br>732<br>-520<br>983<br>881 | 983                   |

### 3. Problem name: Recur\_Power\_Func

จงทำโปรแกรมข้างล่างให้สมบูรณ์ เพื่อให้โปรแกรมข้างล่างทำการคำนวณค่าของ  $x^p$  โดยฟังก์ชัน Recur\_Power จะต้องเขียนในรูปแบบ Recursive function (ฟังก์ชันเรียกตัวเอง) และในฟังก์ชัน main ค่า x และ p จะถูกป้อนเข้ามาและนำมาใช้เป็นอาร์กิวเมนต์ของฟังก์ชัน Recur\_Power

Note: Recurrence equation of  $x^p$  สามารถนิยามได้ดังนี้

$$x^p = 1 \text{ if } p=0$$

$$x^p = x * x^{p-1} \text{ if } p=1,2,\dots$$

```
#include <stdio.h>

double Recur_Power(double x,int  p);

int main()
{
    int pp;
    double xx,value;

        scanf("%lf %d",&xx,&pp);
        value=Recur_Power(xx,pp);
        printf("%.2f\n",value);
        return 0;
}

double Recur_Power(double x,int  p)
{
}
```

ตัวอย่างการทำงานของโปรแกรม

| ข้อมูลนำเข้า (Input) | ข้อมูลส่งออก (Output) |
|----------------------|-----------------------|
| 2.7 3                | 19.68                 |
| -1.5 5               | -7.59                 |

#### 4. Problem name: Recur\_ReverseStr

จงใช้เทคนิค Recursion เพื่อเขียนโปรแกรมทำการสลับตัวอักษรในสตริงจากหน้าไปท้ายให้เรียงจากท้ายไปหน้า เช่น ถ้าอินพุตสตริง คือ "12abc" ผลลัพธ์ที่ได้คือ "cba12" โดยกำหนดให้เขียนและเรียกใช้ฟังก์ชันแบบเรียกตัวเอง (Recursive function) ชื่อ Recur\_ReverseStr() ซึ่งนิยามดังนี้

```
void Recur_ReverseStr(char *str,int len)
```

ซึ่งจะทำการสลับตัวอักษรที่เก็บอยู่สตริงที่ระบุอาร์กิวเมนต์ str โดยที่จำนวนตัวอักษรในสตริง (ความยาวของสตริง) จะถูกระบุโดยอาร์กิวเมนต์ len

นอกจากนี้ให้เขียนฟังก์ชัน main() ทำการรับข้อความจากผู้ใช้โดยใช้ฟังก์ชัน gets() มาเก็บในตัวแปรแบบอาร์เรย์ที่เหมาะสม แล้วทำการนับจำนวนตัวอักษรของข้อความที่ผู้ใช้ป้อนเข้ามา หลังจากนั้นให้เรียกฟังก์ชัน Recur\_ReverseStr() โดยส่งอาร์กิวเมนต์ที่เหมาะสม เมื่อฟังก์ชันทำงานเสร็จแล้วให้แสดงข้อความในสตริงที่ส่งเป็นอาร์กิวเมนต์ของฟังก์ชัน ซึ่งลำดับของตัวอักษรจะต้องถูกสลับจากท้ายไปหน้า

โดยสมมติว่าจำนวนตัวอักษรที่ป้อนเข้ามาจะไม่เกิน 1000 ตัวอักษร

ตัวอย่างการทำงานของโปรแกรม

| ข้อมูลนำเข้า (Input) | ข้อมูลส่งออก (Output) |
|----------------------|-----------------------|
| A1B2C3               | 3C2B1A                |
| Hi, my name is C.    | .C si eman ym ,iH     |
| live not on evil     | live no ton evil      |

Hint:

```
#include <stdio.h>
void Recur_ReverseStr(char *str,int len);
int main()
{
    int len,i;
    char str[1001];
    gets(str);
    i=
    while(    )
        i=
    len=i;
    ReverseStr(    ,    );
    printf("%s\n",str);
    return 0;
}
void Recur_ReverseStr(char *str,int len)
{
}
```

## 5. Problem name: Recur\_Even\_Factorial

นิยาม Factorial เลขคู่ (Even factorial) ของ  $n$  คือ ผลคูณของจำนวนคู่ทั้งหมดที่น้อยกว่าหรือเท่ากับ  $n$  โดยในกรณีที่  $n < 2$  Even factorial) ของ  $n$  จะนิยามให้มีค่าเท่ากับ 1 เช่น

ถ้า  $n=6$  แล้ว Even factorial ของ  $n$  คือ  $6 \times 4 \times 2 = 48$

ถ้า  $n=9$  แล้ว Even factorial ของ  $n$  คือ  $8 \times 6 \times 4 \times 2 = 384$

ถ้า  $n=0$  หรือ  $n=1$  แล้ว Even factorial ของ  $n$  คือ 1

จงใช้เทคนิค Recursion เพื่อเขียนโปรแกรมทำการคำนวณหาค่า Even factorial ของ  $n$  โดยกำหนดให้เขียนและเรียกใช้ฟังก์ชันแบบเรียกตัวเอง (Recursive function) ชื่อ Recur\_eFact() ซึ่งนิยามดังนี้

```
int Recur_eFact(int n)
```

จะทำการคำนวณและคืนค่า even factorial ของ  $n$

กำหนดให้ทำโปรแกรมข้างล่างให้สมบูรณ์ โดยการเขียนโค้ดในฟังก์ชัน Recur\_eFact ในรูปแบบ Recursive function (ฟังก์ชันเรียกตัวเอง)

```
#include <stdio.h>
int Recur_eFact(int n);
int main()
{
    int n, val;
    scanf("%d ", &n);
    val = Recur_eFact(n);
    printf("%d\n", val);
    return 0;
}
int Recur_eFact(int n)
{
}
```

ตัวอย่างการทำงานของโปรแกรม

| ข้อมูลนำเข้า (Input) | ข้อมูลส่งออก (Output) |
|----------------------|-----------------------|
| 12                   | 46080                 |
| 15                   | 645120                |

## 6. Problem name: Recur\_Palindrome

กำหนดให้ข้อมูลนำเข้าประกอบด้วยตัวอักษร 'a'-'z', ตัวเลข '0'-'9' และ Space

นิยาม Palindrome คือ สตริงที่มีตัวอักษรจากซ้ายไปขวาเหมือนกับขวาไปซ้ายโดยไม่สนใจ space เช่น "civic", "was it a rat i saw", "11 02 2011" เป็นต้น

จงใช้เทคนิค Recursion เพื่อเขียนโปรแกรมทำการตรวจสอบว่าสตริงที่ป้อนเข้ามาเป็น Palindrome หรือไม่ ถ้าใช่ให้แสดง YES ถ้าไม่ใช่ให้แสดง NO โดยกำหนดให้ข้อมูลนำเข้า 1 test case ประกอบด้วย สตริงทดสอบจำนวน 5 สตริง บรรทัดละ 1 สตริง และกำหนดให้ความยาวของแต่ละสตริงไม่เกิน 500 ตัวอักษร

โดยกำหนดให้เขียนและเรียกใช้ฟังก์ชันแบบเรียกตัวเอง (Recursive function) ชื่อ Recur\_  
Recur\_IsPalindrome() ซึ่งนิยามดังนี้

```
int Recur_IsPalindrome(char *str,int len)
```

จะทำการตรวจสอบว่า str ที่มีความยาว len ตัวอักษรนั้นเป็น palindrome หรือไม่ ถ้าใช่ให้คืนค่า 1  
ถ้าไม่ใช่ให้คืนค่า 0

กำหนดให้ทำโปรแกรมข้างล่างให้สมบูรณ์ โดยการเขียนโค้ดในฟังก์ชัน Recur\_IsPalindrome ในรูปแบบ Recursive function (ฟังก์ชันเรียกตัวเอง)

ตัวอย่างการทำงานของโปรแกรม

| ข้อมูลนำเข้า (Input)                                                                                                | ข้อมูลส่งออก (Output)          |
|---------------------------------------------------------------------------------------------------------------------|--------------------------------|
| was it a rat i saw<br>civic<br>25 52<br>11 02 2011<br>was it a rat i see                                            | YES<br>YES<br>YES<br>YES<br>NO |
| a man a plan a canal panama<br>a man a plan a canal banama<br>madam i m adam<br>madam i m a d a m<br>madam i m adan | YES<br>NO<br>YES<br>YES<br>NO  |

หมายเหตุ

- was it a rat i saw ถ้าไม่สนใจ space เมื่อเรียงจากซ้ายมาหน้าจะได้สตริงเดียวกัน
- a man a plan a canal panama ถ้าไม่สนใจ space เมื่อเรียงจากซ้ายมาหน้าจะได้สตริงเดียวกัน

## 7. Problem name: Iter\_Palindrome

กำหนดให้ข้อมูลนำเข้าประกอบด้วยตัวอักษร 'a'-'z', ตัวเลข '0'-'9' และ Space

นิยาม Palindrome คือ สตริงที่มีตัวอักษรจากซ้ายไปขวาเหมือนกับขวาไปซ้ายโดยไม่สนใจ space เช่น "civic", "was it a rat i saw", "11 02 2011" เป็นต้น

จงใช้เทคนิค Recursion เพื่อเขียนโปรแกรมทำการตรวจสอบว่าสตริงที่ป้อนเข้ามาเป็น Palindrome หรือไม่ ถ้าใช่ให้แสดง YES ถ้าไม่ใช่ให้แสดง NO โดยกำหนดให้ข้อมูลนำเข้า 1 test case ประกอบด้วย สตริงทดสอบจำนวน 5 สตริง บรรทัดละ 1 สตริง และกำหนดให้ความยาวของแต่ละสตริงไม่เกิน 500 ตัวอักษร

จงเขียนโปรแกรมทำการตรวจสอบว่าสตริงที่ป้อนเข้ามาเป็น Palindrome หรือไม่ ถ้าใช่ให้แสดง YES ถ้าไม่ใช่ให้แสดง NO โดยกำหนดให้ข้อมูลนำเข้า 1 test case ประกอบด้วยสตริงทดสอบจำนวน 5 สตริง บรรทัดละ 1 สตริง และกำหนดให้ความยาวของแต่ละสตริงไม่เกิน 500 ตัวอักษร

โดยกำหนดให้เขียนและเรียกใช้ฟังก์ชันแบบไม่ได้เรียกตัวเอง (Non-Recursive function) ชื่อ Iter\_IsPalindrome() ซึ่งนิยามดังนี้

```
int Iter_IsPalindrome(char *str)
```

จะทำการตรวจสอบว่า str นั้นเป็น palindrome หรือไม่ ถ้าใช่ให้คืนค่า 1 ถ้าไม่ใช่ให้คืนค่า 0

ตัวอย่างการทำงานของโปรแกรม

| ข้อมูลนำเข้า (Input)        | ข้อมูลส่งออก (Output) |
|-----------------------------|-----------------------|
| was it a rat i saw          | YES                   |
| civic                       | YES                   |
| 25 52                       | YES                   |
| 11 02 2011                  | YES                   |
| was it a rat i see          | NO                    |
| a man a plan a canal panama | YES                   |
| a man a plan a canal banama | NO                    |
| madam i m adam              | YES                   |
| madam i m a d a m           | YES                   |
| madam i m adan              | NO                    |