

มหาวิทยาลัยราชภัฏนครินทร์
คณะวิทยาศาสตร์

สอบปลายภาค: ภาคการศึกษาที่ 1

ปีการศึกษา: 2550

วันที่สอบ: 5 ตุลาคม 2550

เวลาสอบ: 13.30 – 16.30 น.

รหัสวิชา: 241-101

ห้องสอบ: R300, R201, R200

ชื่อวิชา: Introduction to Computer Programming

คำสั่ง: อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

อนุญาต: เครื่องเขียนต่างๆ เช่น ปากกา หรือดินสอ

ไม่อนุญาต: หนังสือ, เอกสารใดๆ และเครื่องคิดเลข

เวลา: 3 ชั่วโมง (180 นาที)

คำแนะนำ

- ข้อสอบมี 14 หน้า (ไม่รวมใบปะหน้า) มี 3 ตอน คะแนนรวม 100 คะแนน
- คำตอบทั้งหมดจะต้องเขียนลงในข้อสอบ รวมทั้งเขียนชื่อและรหัสให้ชัดเจนทุกแผ่น หากหน้าใดไม่มีชื่อหรือรหัสจะให้คะแนนเป็น 0 ในหน้านั้น
- คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด
- อ่านคำสั่งในแต่ละข้อให้เข้าใจก่อนลงมือทำ
- หากข้อใดเขียนตอบไม่พอ ให้เขียนเพิ่มเติมที่ด้านหลังของหน้านั้นเท่านั้น

ทูลกระหม่อมเอกในการสอบ โทษขั้นต่ำคือปรับตกในรายวิชานี้และพักการเรียนหนึ่งภาคการศึกษา

ตอนที่ 1 (30 คะแนน)

1. จงเขียนต้นแบบของฟังก์ชัน (function prototype) ดังต่อไปนี้ (8 คะแนน)

1.1 ฟังก์ชัน area รับพารามิเตอร์เป็นทศนิยม 2 จำนวน ส่งค่ากลับเป็นจำนวนทศนิยม

`float area(float x, float y);`

1.2 ฟังก์ชัน date ไม่รับพารามิเตอร์ แต่มีการส่งค่ากลับเป็นจำนวนเต็ม

`int date();`

1.3 ฟังก์ชันชื่อ hola ไม่มีการรับพารามิเตอร์และไม่มีการส่งค่ากลับ

`void hola();`

1.4 ฟังก์ชันชื่อ sorting มีการรับพารามิเตอร์จำนวนเต็มจำนวน 3 ค่า แต่ไม่มีการส่งค่ากลับ

`void sorting(int arr[3]);`

2. จงแสดงการเรียกใช้ฟังก์ชันตามที่กำหนดซึ่งมีต้นแบบของฟังก์ชันดังนี้ เพื่อให้ได้ผลลัพธ์ที่ต้องการ โดยแสดงคำสั่งในการใช้เรียกเท่านั้น ไม่ต้องประกาศตัวแปรใดๆ (4 คะแนน)

`int subtract(int a , int b);` // ฟังก์ชัน subtract ให้ค่ากลับเป็นผลลบของ a และ b

`char cal_grade(int score);` // ฟังก์ชัน cal_grade ให้ค่ากลับเป็นระดับของเกรด

2.1 แสดงการเรียกใช้ฟังก์ชันเพื่อหาผลลบของของตัวแปร num1 และ num2 โดยเก็บค่าผลลบไว้ในตัวแปร result

`int result = subtract(num1, num2);`

2.2 แสดงการเรียกใช้ฟังก์ชัน cal_grade เพื่อหาว่าระดับคะแนน 50 อยู่ในระดับเกรดใด โดยเก็บค่าเกรดไว้ในตัวแปร grade

`char grade = cal_grade(50);`

3. จงเพิ่มเติมส่วนของโปรแกรมต่อไปนี้เพื่อให้คอมไพล์ผ่าน และทำงานได้อย่างถูกต้อง (3 คะแนน)

```
#include<stdio.h>

#include <math.h> (1)
float Log_xy(int x, int y);
int main()
{
    float ans;
    int x,y;
    printf("Enter x value: ");
    scanf("%d",&x);
    printf("Enter y value: ");
    scanf("%d",&y);
    ans = Log_xy(x,y); (2)
    return 0;
}

float Log_xy(int x, int y)
{
    float ans;
    ans = 10 * log10(pow(x,y));
    return ans;
}
```

4. จากโปรแกรมที่กำหนดให้จงหาผลลัพธ์ของโปรแกรม (3 คะแนน)

```
#include <stdio.h>
double local;
void my_func();
int main()
{
    double global ;
    local = 1.1;
    my_func();
    printf("local = %.2f \n", local);
    return 0;
}
void my_func()
{
    double global ;
    global = 1.11
    local = 10*1.1;
    printf("global= %.2f \n, global);
}
```

global = 1.11

local = 11

5. จงเขียนส่วนของการนิยามฟังก์ชันแบบเรียกตัวเองซ้ำ เพื่อหาค่าสมการดังต่อไปนี้

$F(x) = 2 * F(x-1) + 2.5 * F(x-2)$ ถ้า x มีค่ามากกว่า 1

= 1 ถ้า x มีค่าเท่ากับ 1

= 0 ถ้า x มีค่าน้อยกว่า 1 (5 คะแนน)

int F(x)

{

if(x == 1){ return 1; }

else if(x < 1){ return 0; }

else { return 2*F(x-1) + 2.5*F(x-2); }

}

6. จงเขียนโปรแกรมคำนวณหาค่าเลขยกกำลัง (2^x) กำหนดให้เลขฐานมีค่าเท่ากับ 2 เลขชี้กำลังรับผ่านจากคีย์บอร์ด โดยต้องสร้างฟังก์ชันขึ้นมาเองไม่ให้ใช้ไลบรารีฟังก์ชัน pow() ของภาษาซี ข้อกำหนดคือต้องส่งผ่านค่าให้กับฟังก์ชันเป็นเลขจำนวนเต็ม 1 จำนวน ตัวเลขที่ส่งเป็นเลขชี้กำลัง โดยก่อนที่จะเรียกใช้ฟังก์ชันเพื่อคำนวณ ต้องการตรวจสอบเลขชี้กำลังก่อน เลขชี้ต้องมีค่าเป็นจำนวนเต็มบวกเท่านั้น (7 คะแนน)

```
#include <stdio.h>

int pow2(int x)
{
    int a=1,i;
    for(i=0; i<x; i++)
        a *= 2;

    return a;
}

int main()
{
    int x; scanf(" %d", &x);
    printf("2^%d = %d", x, pow2(x));
    return 0;
}
```

/**/ อย่าลืมเขียนชื่อ รหัส และ Section ลงในข้อสอบทุกแผ่น**/

ตอนที่ 2 (30 คะแนน)

1. จากข้อมูลที่กำหนดให้ จงเติมค่าในช่องว่าง เพื่อให้ประโยคสมบูรณ์ (6 คะแนน)

```
int a[10] = {2, 3, -1, 4}; // [ 2 3 -1 4 0 0 0 0 0 0]
```

```
int b[3];
```

- ตัวแปรอาร์เรย์ชื่อ `a` ที่สามารถเก็บเลขจำนวนเต็มได้ทั้งหมด **10** ตัว โดยตัวแปร `a` มีขนาดในหน่วยความจำเป็น **$4 \times 10 = 40$** ไบต์
- ค่าของ `a[2]` คือ **-1**
- ค่าของ `a[7]` คือ **0**
- ค่าของ `a[a[1]]` คือ **`a[3] = 4`**
- เราสามารถกำหนดให้อีลีเมนต์แรกของ `b` เป็นค่าของ `a[3]` โดยใช้คำสั่ง **`b[0] = a[3];`**
- เราสามารถรับค่าจำนวนเต็มจากผู้ใช้ ใส่ในอีลีเมนต์ที่สองของ `b` โดยใช้คำสั่ง **`scanf("%d", &b[1]);`**

2. จากโปรแกรมข้างล่าง จงเติมส่วนที่ขาดหายไปอย่างเหมาะสม เพื่อให้โปรแกรมทำการสำเนาค่าของอีลีเมนต์ทั้งหมดในอาร์เรย์ `a` ไปยังอาร์เรย์ `b` (4 คะแนน)

```
#include<stdio.h>
#define LEN 5
```

```
int main(){
    int a[LEN] = {1, 3, -1, 4, 2};
    int b[LEN];
```

```
    int i;
```

```
    for(i=0; i<len; i++){
```

```
        b[i] = a[i];
```

```
    }
```

```
}
```

3. จากโค้ดข้างล่างจงตอบคำถามต่อไปนี้

```
int len = strlen(a);
int i;
for(i = 0; i < strlen(b); i++){
    a[len - 1] = b[i];
    len++;
}
a[len] = '\0';
```

- a. ส่วนของโค้ดที่กำหนดให้ ทำหน้าที่อะไร จงอธิบายสั้นๆ (3 คะแนน)

ต่อข้อความ b ไปยังข้อความ a

- b. จงปรับปรุงโค้ดดังกล่าว ให้อยู่ในรูปแบบของฟังก์ชัน โดยให้เขียนเฉพาะส่วนนิยามฟังก์ชัน ทั้งนี้ ฟังก์ชันจะต้องคืนค่าความยาวของสตริง a หลังการประมวลผลเสร็จสิ้น (5 คะแนน)

```
int concat(char a[], char b[]){
    int len = strlen(a), i;
    for(i=0;i<strlen(b);i++){
        a[len-1] = b[i];
        len++;
    }
    a[len] = '\0';
    return strlen(a);
}
```

4. จากต้นแบบของฟังก์ชันที่กำหนดให้ จงเขียนนิยามของฟังก์ชัน เพื่อทำการแสดงค่าที่มากที่สุด น้อยที่สุด และค่าเฉลี่ยของข้อมูลในอาร์เรย์ a โดยอาร์เรย์ a มีจำนวนข้อมูลทั้งหมด len ตัว

(6 คะแนน)

```
void stat(int a[], int len);

void stat(int a[], int len){
    int max = a[0], min = a[0], sum = 0, i;
    for(i=0; i<len; i++){
        sum += a[i];
        if(a[i] > max) max = a[i];
        if(a[i] < min) min = a[i];
    }
    printf("Max = %d\n", max);
    printf("Min = %d\n", min);
    printf("Average = %f\n", sum/len);
}
```


5. จากโค้ดที่กำหนดให้ จงระบุผลลัพธ์ของโปรแกรม

(6 คะแนน)

```
#include<stdio.h>
#define LEN 3
int main(){

    int i,j;
    int a[LEN][LEN] ={{3,2,1},{4,5,6}}; //  $\begin{bmatrix} 3 & 2 & 1 \\ 4 & 5 & 6 \\ 0 & 0 & 0 \end{bmatrix}$ 

    for(i = 0; i < LEN - 2; i++){ // Run คำสั่งครั้งเดียว
        for(j = 0; j < LEN -1; j++){ // Run คำสั่ง 2 ครั้ง
            printf("A * B = %d\n", a[i][j] * a[j][i]); // 3 * 3 = 9
                                                    // 2 * 4 = 8
        }
        printf("A + B = %d\n", a[i][j] + a[j][i]); // 1 + 0 = 1
    }
    return 0;
}
```

A * B = 9

A * B = 8

A + B = 1

ตอนที่ 3 (40 คะแนน)

1. จงนิยามโครงสร้างข้อมูลโดยใช้ struct ตามข้อความที่กำหนดให้ (12 คะแนน)

- a. โครงสร้างชื่อ author ประกอบด้วย ชื่อนักเขียน ความยาวไม่เกิน 120 ตัวอักษร หมายเลขประจำตัวผู้เขียนเป็นเลขจำนวนเต็ม

```
struct author {
    char name[121];
    int id;
};
```

```
typedef struct {
    char name[121];
    int id;
} author;
```

- b. โครงสร้างชื่อ chapter ประกอบด้วย ชื่อบทความยาวไม่เกิน 120 ตัวอักษร ผู้เขียน (author จากข้อ a) และปีที่พิมพ์

```
struct chapter {
    char name[121];
    struct author auth;
    int yearOfPublish;
};
```

```
typedef struct {
    char name[121];
    author auth;
    int yearOfPublish;
} chapter;
```

- c. โครงสร้างชื่อ book ประกอบด้วย ชื่อหนังสือความยาวไม่เกิน 120 ตัวอักษร บทจำนวนไม่เกิน 12 บท (chapter จากข้อ b) และจำนวนบท

```
struct book {
    char name[121];
    struct chapter ch[12];
    int nChapters;
};
```

```
typedef struct {
    char name[121];
    chapter ch[12];
    int nChapters;
} book;
```

2. จากโปรแกรมข้างล่าง จงเติมส่วนที่ขาดหายไปอย่างเหมาะสม เพื่อให้โปรแกรมทำการคำนวณหาผลบวกของจำนวนเชิงซ้อน 2 ตัว โดยตัวที่ 1 กำหนดค่าภายในโปรแกรม และตัวที่ 2 รับค่าจากผู้ใช้ (6 คะแนน)

```
#include<stdio.h>

typedef struct{
    float real;
    float imagine;
} Complex.....; (1) // ใน main มี ตัวแปร Complex

int main(){
    Complex a = { 2, 4 }.....; (1)
    Complex b, c;
    printf("Enter real: ");
    scanf("%f", &b.real.....); (0.5)
    printf("Enter imagine: ");
    scanf("%f", &b.imagine.....); (0.5)

    c.real = a.real + b.real.....; (1)

    c.imagine = a.imagine + b.imagine.....; (1)

    printf("result = %.1f + %.1fi\n", c.real, c.imagine.....); (1)
    return 0;
}
```

ผลลัพธ์ของโปรแกรม (ตัวเฉียงคือข้อมูลที่ผู้ใช้ป้อนเข้าไป)

```
Enter real: 3
Enter imagine: 2
result = 5.0+6.0i
```

3. โปรแกรมข้างล่างเป็นโปรแกรมที่ยังไม่ได้ผ่านการคอมไพล์ โดยพบว่ามีข้อผิดพลาดของโปรแกรม ทั้งข้อผิดพลาดของโครงสร้างภาษา หรือเป็นข้อผิดพลาดของการออกแบบโปรแกรม (อาจมีข้อผิดพลาดมากกว่า 1 จุด)

จงเขียนโปรแกรมนี้ใหม่โดยให้คงโครงสร้างเดิมไว้มากที่สุด เพื่อให้ได้ผลลัพธ์ตามที่ระบุไว้

(10 คะแนน)

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

```
struct Employee{
    char name[50];
    float salary;
};
```

```
void up_salary(Employee e){
    e.salary = e.salary * 1.1;
}
```

```
struct Employee(struct Employee e) {
    e.salary = e.salary * 1.1;
    return e;
}
```

```
int main(){
    Employee e;
    e.name = "Kyle XY";
    e.salary = 1000;
    up_salary(e);
    printf("%s has %.1f baht\n", e.name, e.salary);
    return 0;
}
```

```
e = up_salary(e);
```

ผลลัพธ์ของโปรแกรม

Kyle XY has 1100.0 baht

4. จากโครงสร้างของโค้ดที่กำหนดให้ข้างล่าง จงเติมโค้ดให้สมบูรณ์เพื่อรับข้อมูลการสอบของนักเรียนจำนวน 10 คนจากผู้ใช้งานป้อนข้อมูลในอาร์เรย์ และเขียนฟังก์ชันในการหาผู้ที่ได้คะแนนสูงสุด (12 คะแนน)

```
#include<stdio.h>

typedef struct{
    char name[50];
    float score;
}Student;

Student findMax(Student students[], int num);

int main(){
    Student max;    //เก็บข้อมูลนักเรียนที่ได้คะแนนสูงสุด

    Student arrStd[10];
    int i;
    for(i=0; i<10; i++){
        fflush(stdin);
        printf("Student #%d\n", i+1);
        printf("Enter Name> ");
        gets(arrStd[i].name);

        fflush(stdin);
        printf("Enter Score> ");
        scanf("%f", &arrStd[i].score);
    }
    max = findMax(arrStd, 10);

    printf("Max => %s gets %.1f\n", max.name, max.score);
}
```

```
Student findMax(Student students[], int num){
```

```
//ค้นหาข้อมูลของนักเรียนที่ได้คะแนนสูงจาก โดย students คืออาร์เรย์ของนักเรียนทั้งหมด
```

```
//และ num คือจำนวนของนักเรียน
```

```
    int max = 0, i; // max เก็บตำแหน่งน.ร. ที่มีคะแนนสูงสุด
```

```
    for(i=0; i<num; i++){
```

```
        if(students[i].score > students[max].score){
```

```
            max = i;
```

```
        }
```

```
    }
```

```
    return students[max];
```

```
}
```