


มหาวิทยาลัยราชภัฏนครินทร์
คณะวิศวกรรมศาสตร์

สอบปลายภาค: ภาคการศึกษาที่ 2

ปีการศึกษา: 2554

วันที่สอบ: 23 กุมภาพันธ์ 2555

เวลาสอบ: 9.00 – 12.00 น.

ห้องสอบ: A400, R201, ห้องหัวหุ่นยนต์, S101, S102, S103, S104, S201, S203, S817

ผู้สอน: อ.รัชชัย อ.มัลลิกา อ.อารีย์ อ.สกุณา อ.เสกสรรค์ อ.วรพต ภาควิชาวิศวกรรมคอมพิวเตอร์

รหัสและชื่อวิชา: 242-101, 241-101 Introduction to Computer Programming

แนะนำการเขียนโปรแกรมคอมพิวเตอร์

ทฤษฎีในการสอบมีโทษขั้นต่ำคือ ปรับตกในรายวิชาที่ทฤษฎีและพักการเรียน 1 ภาคการศึกษา

คำสั่ง: อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

อนุญาต: เครื่องเขียนต่างๆ เช่น ปากกา หรือดินสอ เข้าห้องสอบ

ไม่อนุญาต: หนังสือ หรือเครื่องคิดเลขเข้าห้องสอบ และเอกสารใดๆ เข้าและออกห้องสอบ

เวลา: 3 ชั่วโมง (180 นาที)

คำแนะนำ

- ข้อสอบมี 11 หน้า (รวมหน้าปก) แบ่งออกเป็น 3 ตอน คะแนนรวม 90 คะแนน (คิดเป็น 30%)
- เขียนคำตอบในข้อสอบ คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด
- อ่านคำสั่งในแต่ละข้อให้เข้าใจก่อนลงมือทำ
- หากข้อใดเขียนคำตอบไม่พอ ให้เขียนเพิ่มเติมด้านหลังของหน้านั้นเท่านั้น

ตอน	1 (30)	2 (30)	3 (30)	รวม (90)
คะแนน				

นักศึกษารับทราบ ลงชื่อ

ตอนที่ 1 อาร์เรย์ (30 คะแนน)

1. จงเขียนฟังก์ชัน `max_Matrix` ซึ่งเปรียบเทียบค่าแต่ละอีลิเมนต์ของแมตริกซ์จำนวนเต็มสองแมตริกซ์ที่ขนาดเท่ากัน แล้วนำค่าที่มากกว่าไปใส่ในแมตริกซ์ผลลัพธ์ แมตริกซ์มีขนาดสูงสุด 10×10 กำหนดต้นแบบฟังก์ชันดังนี้

```
void max_Matrix(int a[10][10], int b[10][10], int c[10][10], int m, int n);
```

ตัวแปร `c` เก็บผลลัพธ์จากค่าแมตริกซ์อินพุต `a` และ `b` ซึ่งมีขนาดเป็น $m \times n$ (5 คะแนน)

```
void max_Matrix(int a[10][10], int b[10][10], int c[10][10], int m, int n)
{
    int i,j;
    for(i=0; _____ ; i++)
        for(j=0; _____ ; j++)
            if ( _____ )
                _____
            else
                _____
}

```

2. จงเขียนโปรแกรมที่รับค่าคะแนนของนักศึกษาจำนวน 50 คน เพื่อหาค่าคะแนนของนักศึกษาลำดับที่เท่าไรที่ได้คะแนนสูงสุด แล้วแสดงคะแนนสูงสุดออกจากจอภาพ โดยแบ่งการทำงานออกเป็น 3 ฟังก์ชันดังนี้

2.1 ฟังก์ชัน `findIndexMax` เป็นฟังก์ชันที่รับค่าตัวแปรอาร์เรย์ของคะแนนทั้งหมด 50 คะแนนมา แล้วคำนวณหาว่าอีลิเมนต์ที่เท่าไร (หรือ `index`) ที่เป็นค่าคะแนนสูงสุด (กรณีที่มีคะแนนเท่ากัน ให้ถือว่า อีลิเมนต์แรกที่เจอเป็นค่าคะแนนสูงสุด) โดยฟังก์ชัน `findIndexMax` จะส่งค่ากลับเป็นตัวเลข `index` ของอีลิเมนต์ที่มีค่าคะแนนสูงสุดนั้น (3 คะแนน)

2.2 ฟังก์ชัน `displayMax` เป็นฟังก์ชันที่รับค่าอาร์เรย์เข้ามา 1 อีลิเมนต์แล้วแสดงค่าอาร์เรย์นั้นออกจากจอภาพ (2 คะแนน)

2.3 ฟังก์ชัน main เป็นส่วนหลักของโปรแกรมที่ใช้รับค่าคะแนนทั้งหมด 50 คะแนนจากผู้ใช้ แล้วเรียกใช้งานฟังก์ชัน findIndexMax เพื่อหาว่าคะแนนลำดับที่เท่าไรมีค่าคะแนนมากที่สุด จากนั้นส่งค่าอาร์เรย์ในลำดับที่มีคะแนนมากที่สุดให้กับฟังก์ชัน displayMax เพื่อแสดงว่าค่าคะแนนสูงสุดมีค่าเท่าไร (5 คะแนน)

3. จงเติมโปรแกรมต่อไปนี้ให้สมบูรณ์ โปรแกรมค้นหาตำแหน่งสินค้าในชั้นเก็บสินค้าขนาด 5 แถว 10 คอลัมน์ ระบุตำแหน่งแรกของชั้นเป็นแถวที่หนึ่ง คอลัมน์ที่หนึ่ง ตำแหน่งสุดท้ายเป็นแถวที่ 5 คอลัมน์ที่ 10 รหัสสินค้าเป็นข้อความ 3 ตัวอักษร โปรแกรมจะรับรหัสสินค้าใส่ลงในอาเรย์สองมิติของข้อความจนครบทุกตำแหน่ง จากนั้นจะวนรับรหัสสินค้ามาแสดงตำแหน่งที่เก็บไปเรื่อยๆ จนกว่าจะได้ข้อความรหัส zzz จึงจะออกจากลูปและจบโปรแกรม การค้นหารหัสสินค้าจะเปรียบเทียบกับรหัสสินค้าที่ผู้ใช้ป้อนกับรหัสที่เก็บไว้ หากตรงกันจะแสดงตำแหน่งที่เก็บออกมา แต่หากไม่ตรงกัน จะแสดงข้อความ "Not Found" (15 คะแนน)

```
#include <stdio.h>
#include <string.h>
int main()
{
    char goods[5][10][4]; // ตัวแปรเก็บรหัสสินค้าในชั้นเก็บสินค้า
    int i, j;
    char code[4];
    for( i = 0 ; _____ ; i++) { // วนลูปค่า i ตามจำนวนแถวชั้นเก็บสินค้า
        for( j = 0 ; _____ ; j++){ // วนลูปค่า i ตามจำนวนแถวชั้นเก็บสินค้า
            printf("Enter code of goods row %d column %d : ", _____ ,
                _____ );
            scanf("%s", _____ ); // รับรหัสสินค้าเก็บลงในอาเรย์ goods ตำแหน่ง i, j
        }
    }
}
```

ชื่อ _____ รหัส _____ section _____

```
// ส่วนทำซ้ำวนลูปหาตำแหน่งสินค้า
printf("Enter code : ");
scanf("%s", _____ ); // เก็บรหัสสินค้าที่ต้องการค้นหา ลงในตัวแปร code

while( _____ ) { // วนซ้ำจนกว่า code จะเป็น "zzz"
    for( i = 0 ; _____ ; i++)
        for( j = 0 ; _____ ; j++)
            // หากพบสินค้า ให้แสดงตำแหน่งสินค้า และกำหนดค่า i, j ให้ออกจากลูป
            if( _____ ) {
                printf("found %s at row %d column %d ", _____
                    , _____ , _____ );
                i=12; j=12; // ถ้าหาพบกำหนดให้ i และ j เป็น 12 เพื่อจะได้ออกจากลูป
            }

    if( _____ ) // ไม่พบสินค้า
        printf("The goods with code %s is not found\n" , code);
    printf("Enter code : ");
    scanf("%s", _____ ); // เก็บรหัสสินค้าที่ต้องการค้นหา ลงในตัวแปร code
}
return 0;
}
```

***** จบตอนที่ 1 *****

ตอนที่ 2 สตริคเจอร์ (30 คะแนน)

1. จากส่วนหนึ่งของโปรแกรมต่อไปนี้ ซึ่งเป็นการนิยามโครงสร้าง จงเขียนส่วนของโปรแกรม เพื่อให้ทำงานตามคำสั่งในข้อ 1.1-1.10 (สามารถเรียกใช้ฟังก์ชันจาก string.h ได้) (14 คะแนน)

```
typedef struct {
    char name[50], ID[20] , sex ;
    int age ;
    float GPA ;
} student;
student student1, ENG1[500] ;
```

- 1.1 ประกาศตัวแปร student2 เป็นตัวแปรชนิด student

- 1.2 ประกาศตัวแปร ENG2 เป็นตัวแปรอาร์เรย์ชนิด student ที่เก็บข้อมูลได้ 300 ชุด

- 1.3 รับข้อความเก็บลงในสมาชิก ID ของตัวแปร student1

- 1.4 รับตัวเลขเก็บลงในสมาชิก GPA ของตัวแปร student1

- 1.5 รับอักขระ 1 ตัว เก็บลงในสมาชิก sex ของตัวแปร student2

- 1.6 กำหนดค่า 19 ลงในสมาชิก age ของอีลีเมนต์แรกของตัวแปร ENG1

- 1.7 กำหนดค่าให้สมาชิก name ของอีลีเมนต์สุดท้ายของตัวแปร ENG1 ให้เก็บข้อความ "Sandra"

- 1.8 ประกาศตัวแปร student3 เป็นตัวแปรชนิด student โดยมีค่าเริ่มต้นของสมาชิกคือ "Erik" "45D55512345" 'M' 20 และ 3.50 ตามลำดับ

- 1.9 เรียกใช้ฟังก์ชัน printf เพื่อแสดงข้อมูลของทุกสมาชิกของตัวแปร student3

1.10 เรียกใช้ฟังก์ชัน printf เพื่อแสดงข้อมูลสมาชิกของตัวแปร ENG1

2. จากส่วนหนึ่งของโปรแกรมในการประกาศอาร์เรย์ของโครงสร้าง เพื่อเก็บข้อมูลนักฟุตบอลแต่ละทีม ให้เขียนคำสั่งเพื่อวนรับค่าต่างๆ เข้ามาเก็บในอาร์เรย์โครงสร้างของทีม Liverpool (6 คะแนน)

```
struct football_player {  
    char name[100];  
    char position[50];  
    int number;  
};
```

```
struct football_player ManU[100], Liverpool[100], Arsenal[100];
```

ชื่อ _____ รหัส _____ section _____

3. โปรแกรมข้างล่าง มีการทำงานดังนี้ เริ่มต้นโดยการวนรับข้อมูลต่างๆ ของนักศึกษา 100 คน ในฟังก์ชัน main แล้ว
ค้นหานักศึกษาที่มีคะแนน GPA สูงสุด โดยเรียกใช้ฟังก์ชัน Max_GPA จากนั้นจึงพิมพ์ชื่อนักศึกษาค้นดังกล่าวใน
ฟังก์ชัน main จงเติมโปรแกรมนี้ให้สมบูรณ์ (10คะแนน)

```
#include <stdio.h>
#define MAX 100
typedef struct {
    char stcode [10] ;
    char name [30] ;
    float gpa ;
} student ;
student Max_GPA(student st [ ] ); //find student with max GPA
int main() {
    int i ;
    student ENG_PSU[MAX] , Best_student ;
    printf ("\nLoop for filling student data\n") ;
```

/****** จบตอนที่ 2 *****/

ตอนที่ 3 โปรแกรมประยุกต์ (30 คะแนน)

ในการสอบคัดเลือกเข้าเรียนโรงเรียนแห่งหนึ่ง ให้ผู้สมัครสอบ 3 วิชา ได้แก่ คณิตศาสตร์(Math) ภาษาไทย(Thai) และภาษาอังกฤษ(English) แต่ละวิชาคะแนนเต็ม 100 คะแนน การเลือกนักเรียนเข้าเรียนนอกจากใช้คะแนนสอบแล้ว ยังมีปัจจัยอื่นเพิ่มเติม ได้แก่ ความใกล้ไกลของบ้านนักเรียน(distance) ความสามารถด้านกีฬา(sport) และความสามารถด้านดนตรี (Music) จากนั้นจะใช้ระบบคิดแต้มคะแนน(Point) เพื่อคัดเลือกนักเรียน ซึ่งมีหลักเกณฑ์ ดังนี้

ปัจจัย	แต้มคะแนน (Point)
คะแนนสอบรวมทั้ง 3 วิชา (score)	
คะแนนต่ำกว่า 100	1
คะแนนตั้งแต่ 100 แต่น้อยกว่า 150	2
คะแนนตั้งแต่ 150 แต่น้อยกว่า 200	4
คะแนนตั้งแต่ 200 แต่น้อยกว่า 250	7
คะแนนตั้งแต่ 250 ขึ้นไป	10
ความใกล้ไกลของบ้านนักเรียน(distance)	
ประเภท A บ้านอยู่ในรัศมี 5 กิโลเมตร	4
ประเภท B บ้านอยู่ในรัศมี 10 กิโลเมตร	3
ประเภท C บ้านอยู่ในรัศมี 20 กิโลเมตร	2
ประเภท D บ้านอยู่ไกลกว่า 20 กิโลเมตร	1
มีความสามารถด้านกีฬาน้อย 1 ชนิด (sport)	2
มีความสามารถด้านดนตรีอย่างน้อย 1 อย่าง (music)	2

กำหนดให้การเก็บข้อมูลของผู้สมัคร (candidate) ให้โครงสร้างข้อมูล ประกาศดังนี้

```
typedef struct {
    char name[50]; // ชื่อ-สกุล ของผู้สมัคร
    float math, thai, eng; // คะแนนสอบแต่ละวิชา
    char distance; // ประเภทของความใกล้ไกล 'A', 'B', 'C' หรือ 'D'
    int sport; // จำนวนชนิดกีฬาที่เล่นได้
    int music; // จำนวนเครื่องดนตรีที่เล่นได้
    int point; // แต้มคะแนน
} candidate ;
```

- มีนักเรียนผู้สมัครคนหนึ่ง มีข้อมูลดังนี้ ชื่อ ด.ช. ธงชัย ใจดี (Thongchai Jaidee) บ้านอยู่ไกลจากโรงเรียนประมาณ 8 กิโลเมตร (ประเภท B) คะแนนสอบคณิตศาสตร์ได้ 72 คะแนน ภาษาไทย 75 คะแนน ภาษาอังกฤษ 61 เป็นนักกีฬาบอลเลย์บอล ปิงปองและแบดมินตัน และเล่นกีตาร์ได้ มีการประกาศตัวแปรสำหรับเก็บข้อมูลของผู้สมัครคนนี้ได้ดังนี้

```
Candidate candid1;
```


จงเติมข้อมูลของผู้สมัครที่เก็บไว้ในตัวแปร candid1 ให้สมบูรณ์ (5 คะแนน)

สมาชิกของ candid1	ข้อมูล
candid1.name	"Thongchai Jaidee"
candid1.math	
candid1.thai	
candid1.eng	
candid1.distance	'B'
candid1.sport	
candid1.music	1
candid1.point	

จงเขียนโปรแกรมเพื่อรับข้อมูลของผู้สมัครแต่ละคน แล้วคำนวณแต้มคะแนนที่ได้ จากนั้นรับค่าเกณฑ์แต้มคะแนนที่โรงเรียนต้องการ แล้วให้แสดงรายชื่อนักเรียนที่ได้แต้มคะแนนผ่านเกณฑ์ลำดับการทำงานของโปรแกรมเป็นดังนี้

- (1) รับค่าจำนวนผู้สมัคร (n) (สมมุติว่ามีผู้สมัครไม่เกิน 100 คน, $n < 100$)
- (2) วนรับข้อมูลของผู้สมัครแต่ละคนจนครบ ประกอบด้วย ชื่อ(name) , คะแนนสอบวิชาคณิตศาสตร์ (math) ภาษาไทย (thai) และ ภาษาอังกฤษ (eng) , ประเภทความไกลไกล (distance) , จำนวนกีฬา (sport) และ ดนตรี (music) หลังจากได้ข้อมูลครบแล้ว ให้คำนวณแต้มคะแนน (point) ของแต่ละคนเก็บไว้
- (3) คำนวณค่าเฉลี่ยของแต้มคะแนนของผู้สมัครทั้งหมด แล้วแสดงค่าเฉลี่ยนั้น
- (4) รับค่าเกณฑ์คะแนนการผ่าน (pass point) เช่น หากกำหนดเป็น 10 แสดงว่าผู้สมัครที่มีแต้มคะแนน มากกว่าหรือเท่ากับ 10 คือผ่านได้เข้าเรียนในโรงเรียนนี้
- (5) แสดงรายชื่อของผู้สมัครทั้งหมดที่สอบผ่านตามเกณฑ์ที่กำหนด (List of pass candidates)

2. สำหรับการทำงานในขั้นตอนที่ (3) ให้สร้างฟังก์ชันสำหรับการคำนวณค่าเฉลี่ยของแต้มคะแนนของผู้สมัครทั้งหมด โดยฟังก์ชันมีต้นแบบ (prototype) ดังนี้

```
float averagePoint(Candidate C[100], int n)
```

C คืออาร์เรย์ที่เก็บข้อมูลของผู้สมัคร และ n คือจำนวนผู้สมัคร

จงเขียนนิยามของฟังก์ชันนี้ (5 คะแนน)

3. จงเขียนโค้ดโปรแกรมที่สมบูรณ์ ดังลำดับขั้นตอน (1)-(5) ที่อธิบายไว้ข้างต้น ในโปรแกรมให้เรียกใช้ฟังก์ชัน averagePoint จากคำถามข้อ 2. ได้เลย (ไม่ต้องเขียนนิยามฟังก์ชันนี้อีก) (20 คะแนน)

```
#include <stdio.h>
typedef struct {
    char name[50]; // ชื่อ-สกุล ของผู้สมัคร
    float math, thai, eng; // คะแนนสอบแต่ละวิชา
    char distance; // ประเภทของความใกล้เคียง 'A', 'B', 'C' หรือ 'D'
    int sport; // จำนวนชนิดกีฬาที่เล่นได้
    int music; // จำนวนเครื่องดนตรีที่เล่นได้
    int point; // แต้มคะแนน
} Candidate ;

float averagePoint(Candidate C[100], int n);
// สามารถสร้างฟังก์ชันเพิ่มได้

int main() {
```

ชื่อ _____ รหัส _____ section _____

โค้ดโปรแกรม ข้อ 3. (ต่อ)

/****** จบตอนที่ 3 *****/