

มหาวิทยาลัยสงขลานครินทร์
คณะวิศวกรรมศาสตร์

สอบปลายภาค: ภาคการศึกษาที่ 2

ปีการศึกษา: 2554

วันที่สอบ: 23 กุมภาพันธ์ 2555

เวลาสอบ: 9.00 – 12.00 น.

ห้องสอบ:

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

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

แนะนำการเขียนโปรแกรมคอมพิวเตอร์

ทุจริตในการสอบมีโทษขั้นต่ำคือ ปรับตกในรายวิชาที่ทุจริตและพักการเรียน 1 ภาคการศึกษา

คำสั่ง: อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

อนุญาต: เครื่องเขียนต่างๆ เช่น ปากกา หรือดินสอ เข้าห้องสอบ

ไม่อนุญาต: หนังสือ หรือเครื่องคิดเลขเข้าห้องสอบ และเอกสารใดๆ เข้าและออกห้องสอบ

เวลา: 3 ชั่วโมง (180 นาที)

คำแนะนำ

- ข้อสอบมี 10 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				

นักศึกษากรรรับทราบ ลงชื่อ

ตอนที่ 1 อาร์เรย์ (30 คะแนน)

1. จงตอบคำถามต่อไปนี้ (10 คะแนน)

```

int arrayMystery(int array[])
{
    int x = 0;
    for (int i = 0; i < 6; i++)
        { if (array[i] > array[i + 1])
            x++;
        }
    return x;
}

```

ค่าเริ่มต้น	ผลลัพธ์ที่ return จากฟังก์ชัน arrayMystery
int a1[8] = {8}; int result1 = arrayMystery(a1);	
int a2[8] = {14, 7}; int result2 = arrayMystery(a2);	
int a3[8] = {7, 1, 3, 2, 0, 4}; int result3 = arrayMystery(a3);	
int a4[8] = {10, 8, 9, 5, 6}; int result4 = arrayMystery(a4);	
int a5[8] = {8, 10, 8, 6, 4, 2}; int result5 = arrayMystery(a5);	

2. จากค่าเริ่มต้น ให้ตอบคำถามดังนี้ (5 คะแนน)

```

char s1[]="big sky country";
char s2[20]="blue moon";
char s3[]="then falls Caesar";

```

ฟังก์ชัน	ผลลัพธ์ที่ได้
strlen(s1)	
strlen(s2)	
strcmp(s1,s2)	
strcmp(s3,s2)	
strcat(s2," tonight")	(สตริงค่าใหม่ของตัวแปร s2)

3. จงเขียนโปรแกรมที่รับค่าคะแนนของนักศึกษาจำนวน 50 คน เพื่อหาคะแนนของนักศึกษาลำดับที่เท่าไรที่ได้คะแนนสูงสุด แล้วแสดงคะแนนสูงสุดออกทางจอภาพ โดยแบ่งการทำงานออกเป็น 3 ฟังก์ชันดังนี้

3.1 ฟังก์ชัน `findIndexMax` เป็นฟังก์ชันที่รับค่าตัวแปรอาร์เรย์ของคะแนนทั้งหมด 50 คะแนนมา แล้วคำนวณหาว่าในอีลีเมนต์ (หรือ `index`) ที่เท่าไร ที่เป็นค่าคะแนนสูงสุด (กรณีที่คะแนนเท่ากัน ให้ถือว่า อีลีเมนต์แรกเจอเป็นค่าคะแนนสูงสุด) โดยฟังก์ชัน `findIndexMax` จะส่งค่ากลับเป็นตัวเลข `index` ที่มีค่าคะแนนสูงสุดนั้น (5 คะแนน)

3.2 ฟังก์ชัน `displayMax` เป็นฟังก์ชันที่รับค่าอาร์เรย์เข้ามา 1 อีลีเมนต์แล้วแสดงค่าอาร์เรย์นั้นออกทางจอภาพ (3 คะแนน)

3.3 ฟังก์ชัน `main` เป็นส่วนหลักของโปรแกรมที่ใช้รับค่าคะแนนทั้งหมด 50 คะแนนจากผู้ใช้ แล้วเรียกใช้งานฟังก์ชัน `findIndexMax` เพื่อหาคะแนนลำดับที่เท่าไรมีค่าคะแนนมากที่สุด จากนั้นส่งค่าอาร์เรย์ในลำดับที่มีคะแนนมากที่สุดให้กับฟังก์ชัน `displayMax` เพื่อแสดงว่าค่าคะแนนสูงสุดมีค่าเท่าไร (7 คะแนน)

/****** จบตอนที่ 1 *****/

ตอนที่ 2 สตริงเจอร์ (30 คะแนน)

1. จากส่วนหนึ่งของโปรแกรมต่อไปนี้ ซึ่งเป็นการนิยามโครงสร้าง จงเขียน**ส่วนของโปรแกรม** เพื่อให้ทำงานตามคำสั่งในข้อ 1.1-1.10 (สามารถใช้ฟังก์ชันจาก 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 เพื่อแสดงข้อมูลของ**ทุกสมาชิก**ของตัวแปร student3

1.10 เรียกใช้ฟังก์ชัน printf เพื่อแสดงข้อมูลทุกสมาชิกของตัวแปร 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	2
มีความสามารถด้านดนตรีอย่างน้อย 1 อย่าง	2

กำหนดให้การเก็บข้อมูลของผู้สมัคร (candidate) ใช้โครงสร้างข้อมูล 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 ;
```

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

```
Candidate candid1;
```

จงเติมข้อมูลของผู้สมัครที่เก็บไว้ในตัวแปร `candid1` ให้สมบูรณ์ (5 คะแนน)

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

จงเขียนโปรแกรมเพื่อรับข้อมูลของผู้สมัครแต่ละคน แล้วคำนวณแต้มคะแนนที่ได้ จากนั้นรับค่าเกณฑ์แต้มคะแนนที่โรงเรียนต้องการ แล้วให้แสดงรายชื่อนักเรียนที่ได้แต้มคะแนนผ่านเกณฑ์

ลำดับการทำงานของโปรแกรมเป็นดังนี้

- (1) รับค่าจำนวนผู้สมัคร (n) (สมมุติว่ามีผู้สมัครไม่เกิน 100 คน, $n < 100$)
- (2) วนรับข้อมูลของผู้สมัครแต่ละคนจนครบ ประกอบด้วย ชื่อ(name) , คะแนนสอบวิชาคณิตศาสตร์ (math) ภาษาไทย (thai) และ ภาษาอังกฤษ (eng) , ประเภทความใกล้ไกล (distance) , จำนวนกีฬา (sport) และ ดนตรี (music) หลังจากได้ข้อมูลครบแล้ว ให้คำนวณแต้มคะแนน (point) ของแต่ละคนเก็บไว้
- (3) คำนวณค่าเฉลี่ยของแต้มคะแนนของผู้สมัครทั้งหมด แล้วแสดงค่าเฉลี่ยนั้น
- (4) รับค่าเกณฑ์คะแนนการผ่าน (pass point) เช่น หากกำหนดเป็น 10 แสดงว่าผู้สมัครที่มีแต้มคะแนน มากกว่า หรือเท่ากับ 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 *****/