


มหาวิทยาลัยสงขลานครินทร์
คณะวิศวกรรมศาสตร์

สอบปลายภาค: ภาคการศึกษาที่ 1

ปีการศึกษา: 2554

วันที่สอบ: 7 ตุลาคม 2554

เวลาสอบ: 13.30 – 16.30 น.

ห้องสอบ: A401 R200 S201 S203 S817 หิวหุ่นยนต์

ผู้สอน: อ.เสกสรรค์ อ.สุกญา อ.นิคม อ.อารีย์ อ.อนันท์ อ.มัลลิกา ภาควิชาวิศวกรรมคอมพิวเตอร์

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

แนะนำการเขียนโปรแกรมคอมพิวเตอร์ (รหัสหลักสูตรเก่า 240-101)

ทิวจริตในการสอบมีโทษขั้นต่ำคือ ปรับตกในรายวิชาที่ทิวจริตและพักการเรียน 1 ภาคการศึกษา

คำสั่ง: อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

อนุญาต: เครื่องเขียนต่างๆ เช่น ปากกา หรือดินสอ เข้าห้องสอบ

ไม่อนุญาต: หนังสือ หรือเครื่องคิดเลขเข้าห้องสอบ และเอกสารใดๆ เข้าและออกห้องสอบ

เวลา: 3 ชั่วโมง (180 นาที)

คำแนะนำ

- ข้อสอบมี 13 หน้า (รวมหน้าปก) แบ่งออกเป็น 3 ตอน คะแนนรวม 105 คะแนน (คิดเป็น 35%)
- เขียนคำตอบในข้อสอบ คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด
- อ่านคำสั่งในแต่ละข้อให้เข้าใจก่อนลงมือทำ
- หากข้อใดเขียนคำตอบไม่พอ ให้เขียนเพิ่มเติมที่ด้านหลังของหน้านั้นเท่านั้น

ตอน	1 (30)	2 (30)	3 (35)	รวม (105)
คะแนน				

นักศึกษากรับทราบ ลงชื่อ

ตอนที่ 1 อาร์เรย์ (30 คะแนน)

1. จงตอบคำถามต่อไปนี้ (5 คะแนน)

```

1. //labarrayex1.c
2. #include<stdio.h>
3.
4. int main()
5. {   int n, prize[3]={30, 20, 50};
6.
7.     for(n=0;n<3;n++)
8.         printf("prize %d is %d\n", n+1, prize[n]);
9.     return 0;
10. }
```

1.1) ถ้ามีการเปลี่ยนจำนวนของข้อมูลของตัวแปร prize จาก 3 เป็น 6 นั่นคือ

```
int prize[3]={30,20,50}; แก้ไขเป็น int prize[6]={31,21,51};
```

ดังนั้นจากโปรแกรมข้างต้นควรแก้ไขโปรแกรมในบรรทัดที่ 7 อย่างไร

1.2) จงเขียนผลลัพธ์ของโปรแกรมที่ได้ก่อนเปลี่ยนค่าตัวแปร และหลังเปลี่ยนค่าตัวแปร

ผลลัพธ์ที่ได้ก่อนเปลี่ยนค่าตัวแปร	ผลลัพธ์ที่ได้หลังเปลี่ยนค่าตัวแปร

2. ให้นักศึกษาแก้ไขโปรแกรม โดยแยกส่วนการคำนวณค่าเฉลี่ยของอาร์เรย์ออกเป็นฟังก์ชัน ชื่อ average ซึ่งรับอาร์เรย์เป็นพารามิเตอร์และคืนค่าเป็นค่าเฉลี่ยที่คำนวณได้ และส่วนการรับค่าข้อมูลอาร์เรย์และแสดงผลอยู่ในฟังก์ชัน main เหมือนเดิม (5 คะแนน)

```

1. //labarrayex3.c
2. #include <stdio.h>
3. int main()
4. {   int temp[3], total = 0, count;
5.     float avg;
6.     for(count = 0; count <3; ++count)
7.     {   printf("Enter element %d ", count);
8.         scanf("%d", &temp[count]);
9.         total+=temp[count];
10.    }
11.    avg = total/3.0;
12.    for(count = 0; count <3; ++count)
13.        printf("\n Element #%d = %d", count, temp[count]);
14.    printf("\nThe average is %f\n", avg);
15.    return 0;
16. }
```

โค้ดโปรแกรมข้อ 2.

3. จงตอบคำถามต่อไปนี้ (5 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);	

4. 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)

5. จงเติมส่วนที่หายไปของโปรแกรมการบวกเมตริกซ์ ตั้งแต่หมายเลข 1-7 (10 คะแนน)

```
//labMatrix.c
#include <stdio.h>

#define MAX 10
void showMatrix(int M[MAX][MAX], int m, int n);
void inputMatrix(int M[MAX][MAX], int m, int n);
void addMatrix(int A[MAX][MAX], int B[MAX][MAX], int C[MAX][MAX],
               int m, int n);

int main()
{
    int MatA[MAX][MAX]={}, MatB[MAX][MAX]={}, MatC[MAX][MAX]={};
    int row,col;
    printf("Matrix A\n");
    printf("Number of rows = ");      scanf("%d",&row);
    printf("Number of columns = ");   scanf("%d",&col);

    (1) _____

    (2) _____
    printf("Matrix B\n");
    inputMatrix(MatB,row,col);
    showMatrix(MatB,row,col);
    printf("Matrix C = A + B \n");

    (3) _____

    (4) _____

    return 0;
}
```

```
// Show elements of Matrix M(m x n)
void showMatrix(int M[MAX][MAX], int m, int n)
{ (5)

}

// Input elements of Matrix M(m x n)
void inputMatrix(int M[MAX][MAX], int m, int n)
{ (6)

}

// Matrix C = A + B
void addMatrix(int A[MAX][MAX], int B[MAX][MAX], int C[MAX][MAX],
               int m, int n)
{ (7)

}

}
```

/****** จบตอนที่ 1 *****/

ตอนที่ 2 โครงสร้างข้อมูล Structure (30 คะแนน)

1. จาก Structure ที่กำหนดให้ จงเขียนโค้ดของโปรแกรมเพื่อให้ทำงานตามคำสั่งในข้อ 1.1-1.10

(10 คะแนน)

```
#include<stdio.h>
#include<stdlib.h>
#include <string.h>
struct soccer_player{
    char name[60];
    char team[50];
    int age;
    float weight;
    float height;
} player1;
```

1.1 ประกาศตัวแปรชื่อ player2 เป็นตัวแปรสตรัคเจอร์ soccer_player

1.2 รับค่าข้อความเก็บลงในสมาชิก name ของ player2

1.3 รับค่าอายุ(จำนวนเต็ม)เก็บลงในสมาชิก age ของ player2

1.4 กำหนดค่าสมาชิก team ของ player2 ให้มีข้อความเป็น "Thailand"

1.5 ประกาศตัวแปรชื่อ player3 โดยให้มีค่าเริ่มต้นเป็น "Pele" "Brazil" 20 70.5 165.5 ตามลำดับ

1.6 เรียกใช้ฟังก์ชัน printf เพื่อแสดงข้อมูลสมาชิก team ของ player3

1.7 ประกาศ myplayer_list เป็นตัวแปรอาร์เรย์ของสตรัคเจอร์แบบ soccer_player ที่เก็บข้อมูลได้ 100 ชุด

1.8 กำหนดค่าของสมาชิก ของตัวแปร myplayer_list ที่ดัชนี 0 ให้มีค่าเท่ากับ player3

1.9 กำหนดค่าสมาชิก weight ของตัวแปร myplayer_list ที่ดัชนีสุดท้ายให้มีค่าเป็น 80.7

1.10 เรียกใช้ฟังก์ชัน printf เพื่อแสดงข้อมูลสมาชิก team ของตัวแปร myplayer_list ที่ดัชนีหมายเลข 10

ชื่อ _____ รหัส _____ section _____

2. จงนิยามโครงสร้างข้อมูล (struct) ตามความเหมาะสมดังนี้ (8 คะแนน)

2.1) นิยามโครงสร้างชื่อ address มีสมาชิก คือ (3 คะแนน)

- เลขที่บ้าน (num) เก็บเป็นตัวเลขและตัวอักษร ไม่เกิน 20 ตัว
- อำเภอ (amphur) เป็นตัวอักษรไม่เกิน 50
- จังหวัด (province) เป็นตัวอักษรไม่เกิน 50
- รหัสไปรษณีย์ (zip) เป็นตัวเลขจำนวนเต็ม 5 ตัว

2.2) นิยามโครงสร้าง ชื่อ person มีสมาชิกคือ ชื่อ ที่อยู่ (address จากข้อ 2.1) และอายุ (3 คะแนน)
(ไม่ใช้ typedef)

2.3 ใช้คำสั่ง typedef นิยามโครงสร้าง Person มีสมาชิกเหมือนข้อ 2.2 (2 คะแนน)

3. จากโค้ดโปรแกรมข้างล่างจงเติมโค้ดให้สมบูรณ์ โดยโปรแกรมมีการทำงานดังนี้ ในฟังก์ชัน main รับข้อมูลของนักฟุตบอลจำนวนไม่เกิน 100 คนจากผู้ใช้และเรียกฟังก์ชันชื่อ find_top_scorer เพื่อหาผู้ที่มีทำประตูสูงสุด ฟังก์ชัน find_top_scorer จะส่งค่าดัชนีของผู้ที่ทำประตูสูงสุดกลับมายังฟังก์ชัน main และฟังก์ชัน main จะแสดงรายละเอียดของผู้ทำประตูสูงสุดออกทางหน้าจอ (12 คะแนน)

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX 100
typedef struct {
    char name[60]; //ชื่อนักฟุตบอล
    char team[50]; //ชื่อทีม
    int num_goal; //จำนวนประตูที่ทำได้
} player;

int find_top_scorer(player players[], int num);
//num คือจำนวน element ของอาร์เรย์ players
int main(){

    _____ /** ประกาศตัวแปรอาร์เรย์ (1 คะแนน) **/
    int i, count=0 , index=0;
    printf("Enter number of players: "); scanf("%d", &count);
    for(i=0; i<count; i++) /** รับค่าข้อมูลของ player (3 คะแนน) **/
    { printf("\nPlease enter player's name: ");

        _____
        printf("\nPlease enter player's team: ");

        _____
        printf("\nPlease enter number of goals: ");

    }

    /** เรียกใช้ฟังก์ชัน find_top_scorer (2 คะแนน) **/
    index = _____

    /** แสดงผลลัพธ์ (2 คะแนน) **/
    printf("The top scorer is : %s \t %s \t %d \n", _____

    _____

    return 0;
}
```


ชื่อ _____ รหัส _____ section _____

```
int find_top_scorer(player p_arr[MAX], int arg_count)
{ /** ฟังก์ชันค้นหาผู้ทำประตูสูงสุด (3 คะแนน) **/
    int i;
    int index;
    int max = p_arr[0].num_goal;

    return _____ /** คืนค่าผลลัพธ์ของฟังก์ชันนี้ (1 คะแนน) **/
}
```

/** ***** จบตอนที่ 2 ***** */

ตอนที่ 3 โปรแกรมประยุกต์ (35 คะแนน)

Weather Data Report รายงานข้อมูลสภาพอากาศ

Songkla	Jan.	Feb.	Mar.	Apr.	May	Jun.	Jul.	Aug.	Sep.	Oct.	Nov.	Dec.
Average Temperature (°C)	27.1	27.9	28.6	28.5	28.2	28.1	28.9	28.6	28.5	27.2	27.1	26.9
Average High Temperature (°C)	28.3	30.0	31.8	31.7	32.4	32.3	31.0	31.8	31.7	30.4	28.3	28.0
Average Low Temperature (°C)	24.5	25.2	25.1	25.9	25.6	25.5	25.2	25.1	25.9	24.6	25.5	25.2
Average Precipitation (cm)	12.7	4.4	5.3	8.0	11.8	9.7	9.4	9.3	11.0	29.8	52.7	44.4

ตารางข้างต้นเป็นตัวอย่างข้อมูลอุณหภูมิเฉลี่ยและปริมาณน้ำฝนเฉลี่ยรายเดือนของจังหวัดสงขลา

- Average Temperature (AT) คืออุณหภูมิเฉลี่ยของแต่ละเดือน
- Average High Temperature (AHT) คือ ค่าเฉลี่ยของอุณหภูมิสูงสุดของแต่ละวัน ในรอบหนึ่งเดือน
- Average Low Temperature (ALT) คือ ค่าเฉลี่ยของอุณหภูมิต่ำสุดของแต่ละวัน ในรอบหนึ่งเดือน
- Average Precipitation (AP) คือ ปริมาณน้ำฝนเฉลี่ยของแต่ละเดือน

กำหนดนิยามของโครงสร้างข้อมูลสำหรับเก็บข้อมูลสภาพอากาศดังกล่าวดังนี้

```
typedef struct {
    float AT;
    float AHT;
    float ALT;
    float AP;
} WeatherData;

typedef struct {
    WeatherData wdata[12];
    char province[50];
} WTable;
```

มีการกำหนดค่าชื่อเดือนในตัวแปร month ซึ่งเป็นตัวแปร global ดังนี้

```
char month[12][10] = { "Jan", "Feb", "Mar", "Apr", "May", "Jun",
                      "Jul", "Aug", "Sep", "Oct", "Nov", "Dec" };
```

จงเขียนโปรแกรมที่สมบูรณ์เพื่อรับค่าข้อมูลสภาพอากาศของจังหวัดสงขลา ดังตารางข้างต้นและให้แสดงค่าข้อมูลดังกล่าว พร้อมทั้งสรุปค่าเฉลี่ยของอุณหภูมิตลอดปี และ เดือนใดมีฝนตกมากที่สุด และน้อยที่สุด โดยกำหนดให้มีฟังก์ชันต่างๆ ในโปรแกรมดังต่อไปนี้

- ฟังก์ชัน inputWTable ใช้สอบถามชื่อจังหวัดและข้อมูลสภาพอากาศทั้ง 12 เดือนจากผู้ใช้
- ฟังก์ชัน displayWTable ใช้แสดงข้อมูลสภาพอากาศดังตาราง
- ฟังก์ชัน averageTemp ใช้คำนวณค่าเฉลี่ยของอุณหภูมิตลอดทั้งปี โดยคำนวณจากค่าอุณหภูมิเฉลี่ย (AT) ของทุกเดือน
- ฟังก์ชัน driestMonth ใช้ค้นหาเดือนที่มีปริมาณน้ำฝนน้อยที่สุด ฟังก์ชัน return ค่า index ของเดือนนั้น
- ฟังก์ชัน wettestMonth ใช้ค้นหาเดือนที่มีปริมาณน้ำฝนมากที่สุด ฟังก์ชัน return ค่า index ของเดือนนั้น

จงเติมโค้ดโปรแกรมให้สมบูรณ์

```
#include <stdio.h>

typedef struct {
    float AT;
    float AHT;
    float ALT;
    float AP
} WeatherData;

typedef struct {
    WeatherData wdata[12];
    char province[50];
} WTable;

char month[12][10] = {"Jan", "Feb", "Mar", "Apr", "May", "Jun",
                    "Jul", "Aug", "Sep", "Oct", "Nov", "Dec" };

WTable inputWTable(); // (8 คะแนน)
void displayWTable(WTable t); // (8 คะแนน)
float averageTemp(WTable t); // (4 คะแนน)
int driestMonth(WTable t); // (5 คะแนน)
int wettestMonth(WTable t); // (5 คะแนน)

int main( ) // (5 คะแนน)
{
    _____ // ประกาศตัวแปร t เพื่อเก็บข้อมูลของตาราง
    // ประกาศตัวแปรอื่นๆ เท่าที่จำเป็น

    t = inputWTable();
    displayWTable(t);
    // คำนวณอุณหภูมิเฉลี่ยทั้งปี และแสดงค่า (ให้เรียกใช้ฟังก์ชัน averageTemp)

    // แสดงชื่อเดือนที่มีฝนตกมากที่สุด (ให้เรียกใช้ฟังก์ชัน driestMonth)

    // แสดงเดือนที่มีฝนตกมากที่สุด (ให้เรียกใช้ฟังก์ชัน wettestMonth)

    return 0;
}
```

ชื่อ _____ รหัส _____ section _____

/* โค้ดโปรแกรม ตอนที่ 3 (ต่อ) */

/* เขียนนิยามของฟังก์ชัน ทั้ง 5 ฟังก์ชัน ที่ได้ประกาศไว้ */

ชื่อ _____ รหัส _____ section _____

/* โค้ดโปรแกรม ตอนที่ 3 (ต่อ) */

/****** จบตอนที่ 3 *****/