



สอบปลายภาค: ภาคการศึกษาที่ 1

ปีการศึกษา: 2557

วันสอบ: 13 ธันวาคม 2557

เวลาสอบ: 13.30 – 16.30 น.

ห้องสอบ: (01) S817, (02) A400, (03) R200, (04) A400, (05) R200, (06) A401, (07) R201,
(08) S817, (09) A401, (10) หัวหิน, (241-101) R201

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

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

แนะนำการเขียนโปรแกรมคอมพิวเตอร์

ทูลจิดในการสอบมีโทษขั้นต่ำคือ ปรับตกในรายวิชาที่ทูลจิดและหักการเรียน 2 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 60 คะแนน (คิดเป็นคะแนนเก็บ 30%)
- ☐ เขียนคำตอบในข้อสอบ คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด
- ☐ อ่านคำสั่งในแต่ละข้อให้เข้าใจก่อนลงมือทำ
- ☐ เวลาที่ใช้เวลาทำตอนให้เหมาะสม ตามคำแนะนำ
- ☐ หากข้อใดเขียนคำตอบไม่พอ ให้เขียนเพิ่มเติมด้านหลังของหน้านั้นเท่านั้น

ตอน	1 (20) 10%	2 (20) 10%	3 (20) 10%	รวม (60) 30%
คะแนน				

นักศึกษาที่รับทราบ ลงชื่อ

ตอนที่ 1 (20 คะแนน, 10%, 50 นาที)

ฟังก์ชัน(Function)

1. จงเขียนนิยามของฟังก์ชัน \maxValue ที่ค้นหาค่าที่มากที่สุด(max)ในอาร์เรย์สองมิติ ขนาด 4x3 และคืนค่า max ที่หาได้

กำหนดต้นแบบฟังก์ชัน(prototype) ดังนี้ int maxVal (int Matrix[4][3]) (5 คะแนน)

2. จงเขียนโปรแกรม รับค่าสตริงมาหนึ่งตัว (ความยาวสูงสุดไม่เกิน 50 ตัวอักษร) และเรียกใช้ฟังก์ชัน

`printReverse` (ฟังก์ชันสร้างขึ้นเอง) เพื่อพิมพ์ผลตรงนั้นกลับจากหลังไปหน้า

ตัวอย่างการรันโปรแกรม (ตัวเอียงหน้าคือค่าที่ผู้ใช้ใส่) (5 คะแนน)

```
Enter a string Hello World
Reverse string dlroW olleH
```

<p>จงเขียนโปรแกรมที่สมบูรณ์</p>	<p>โค้ดโปรแกรม (ต่อ)</p>
---------------------------------	--------------------------

3. จงเขียนโปรแกรม รับค่าจำนวนเต็ม 10 จำนวน เก็บไว้ในอาร์เรย์ แล้วเรียกใช้ฟังก์ชัน `re_arrange` เพื่อจัดข้อมูลในอาร์เรย์ใหม่ โดยให้เลื่อนข้อมูลขึ้นไปหนึ่งตำแหน่ง และข้อมูลตัวสุดท้ายกลับมาอยู่ตำแหน่งเริ่มต้น ให้ส่วนของการรับค่าและแสดงข้อมูลของอาร์เรย์ ก่อนและหลังการเลื่อนข้อมูลอยู่ในโปรแกรมหลัก และให้เรียกใช้ฟังก์ชัน `re_arrange` 3 ครั้ง (10 คะแนน)

กำหนดต้นแบบของฟังก์ชันดังนี้ `void re_arrange(int num[])`

ตัวอย่างการรันโปรแกรม (ตัวเลขในวงเล็บคือค่าที่ผู้ใช้ใส่)

```
Please enter 10 integers: 5 2 5 1 4 6 7 9 0 8
Initial array: 5 2 5 1 4 6 7 9 0 8
After re-arranging the array 3 times
New array: 9 0 8 5 2 5 1 4 6 7
```

จงเขียนโปรแกรมที่สมบูรณ์

/****** จบตอนที่ 1 *****/

ตอนที่ 2 (30 คะแนน, 10%, 60 นาที)

โครงสร้างข้อมูล (Structure)

1. จากสตรักเจอร์ที่กำหนด คือ

```
struct employee  
{ char name[30], surname[50];  
  int age;  
  float salary;  
};
```

จงเขียนบางส่วนของโปรแกรมตามคำถามต่อไปนี้ (5 คะแนน)

- 1.1 ประกาศตัวแปร a และ b เป็นสตรักเจอร์แบบ employee

ตอบ....

- 1.2 ใส่ข้อมูลพนักงานลงในตัวแปร a ว่า ชื่อ John นามสกุล Smith อายุ 30 ปี และมีเงินเดือน

20,000 บาท

ส่วนพนักงาน b ชื่อ Miranda นามสกุล Brook อายุ 25 ปี และมีเงินเดือน 18,000 บาท

ตอบ...

- 1.3 นำข้อมูลทุกอย่างในตัวแปร a สลับกับในตัวแปร b (นศ. สามารถประกาศตัวแปรเพิ่มได้)

ตอบ...

2. สมมติสถานการณ์ว่าร้านค้าแห่งหนึ่งต้องการจะกรอกประวัติสมาชิก 100 คน ลงในระบบที่คุณกำลังจะพัฒนา จากสตรัคเจอร์ Date และ Data ที่กำหนด จงประกาศตัวแปร Client[100] ซึ่งเป็นอาร์เรย์แบบสตรัคเจอร์ Data แล้วทำการเขียนโปรแกรมเพื่อวนลูปรับค่าต่างๆ ของสมาชิกทั้ง 100 คน โดยเก็บค่าดังกล่าวไว้ในตัวแปร Client[100] (5 คะแนน)

```
struct Date
```

```
{ int Day, Month, Year;  
};
```

```
struct Data
```

```
{ char NameSurname[60];  
  struct Date IssueDate, ExpireDate;  
};
```

ตัวอย่างผลการรันโปรแกรม:

Client 1:

Please enter Name and Surname: **John Smith**

Please enter Issue Date: **12 5 2014**

Please enter Expire Date: **11 5 2015**

Client 2:

Please enter Name and Surname: **Mandy Scott**

Please enter Issue Date: **21 9 2014**

Please enter Expire Date: **20 9 2017**

.

.

.

Client 100:

Please enter Name and Surname: **Julia Brook**

Please enter Issue Date: **12 5 2014**

Please enter Expire Date: **11 5 2015**

จงเติมโปรแกรมนี้ให้สมบูรณ์

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

```
#include<string.h>
```

```
int main()
```

```
{ struct Date
```

```
{ int Day, Month, Year;  
};
```

```
struct Data
```

```
{ char NameSurname[60];  
  struct Date IssueDate, ExpireDate;  
};
```

```
_____ Client[100];
```

```

int i;
for(i=0; _____ )
{   printf("Client %d:\n", i+1);
    printf("Please enter Name and Surname: ");
    gets( _____ );
    printf("Please enter Issue Date: ");
    scanf(" _____", &Client[i].IssueDate.Day,
          _____,
          &Client[i].IssueDate.Month,
          _____,
          &Client[i].IssueDate.Year );

    fflush(stdin);
    printf("Please enter Expire Date: ");
    scanf("%d %d %d", _____,
          _____,
          _____ );

    fflush(stdin);
    printf("\n");
}
getch();
return 0;
}

```

3. อนุบาลหมีน้อยวิทยาคม ต้องการให้ นศ. พัฒนาโปรแกรมสำหรับเก็บข้อมูลคะแนนสอบของ นร. ในโรงเรียน โดย

- โรงเรียนมีทั้งหมด 3 ระดับชั้น คือ อนุบาล1, 2 และ 3
- แต่ละชั้นมี นร. 50 คน
- แต่ละคนเรียน 3 วิชา คือ Math, English และ Thai

ข้อกำหนดของการเขียนโปรแกรมมีดังนี้ (10 คะแนน)

- นศ. ต้องใช้งานตัวแปรต่อ.ปนี้ในการเขียนโปรแกรม

```

typedef struct
{   float Math, English, Thai, Total;
} score;

```

```

typedef struct
{   score Student[50];
} level;

```

- ในฟังก์ชัน main ให้ นศ. ประกาศตัวแปรอาเรย์ของ level ชื่อ MeeNoi[3] เพื่อเก็บข้อมูลคะแนนสอบของ นร. ทั้ง 3 ระดับชั้นในโรงเรียน

2. ให้เขียนนิยามของฟังก์ชันสำหรับรับค่าข้อมูลสภาพอากาศรายวัน รับค่าวันที่ ปริมาณน้ำฝน อุณหภูมิต่ำสุด และสูงสุด โดยให้บันทึกค่าเรียงตามวันที่เป็นระยะเวลา 1 ปี (365 หรือ 366 วัน โดยจำนวนวันรับค่าผ่านพารามิเตอร์ n) และบันทึกค่าเก็บไว้ในอาร์เรย์ M

ฟังก์ชันไม่มีการส่งค่ากลับ (4 คะแนน)

ต้นแบบของฟังก์ชัน void inputYearTable(MeteoRecord M[], int n)

ตัวอย่างการทำงาน เช่น

Day#1

DATE: 01 01 2013

RAIN: 0

MIN TEMP: 20.5

MAX TEMP: 33.2

Day#2

DATE: 02 01 2013

... ใส่ข้อมูลไปเรื่อยๆ จนครบวันสุดท้าย เช่นเมื่อ n = 365 ...

Day#365

DATE: 31 12 2013

RAIN: 1

MIN TEMP: 19.9

MAX TEMP: 28.5

นิยามของฟังก์ชัน

3. ให้เขียนนิยามของฟังก์ชันสำหรับคำนวณค่าอุณหภูมิเฉลี่ยรายวัน(`avgTemp`) ซึ่งคำนวณจาก $(\text{minTemp} + \text{maxTemp}) / 2$ เก็บไว้ในอาร์เรย์ `M` (ซึ่งรับค่าอุณหภูมิสูงสุดและต่ำสุดมาแล้วก่อนหน้านี้) และให้แสดงค่าอุณหภูมิเฉลี่ยรายวันตามลำดับ ฟังก์ชันไม่มีการส่งค่ากลับ (4 คะแนน)

ต้นแบบของฟังก์ชัน `void calAvgTemp(MeteoRecord M[], int n)`

ตัวอย่างการทำงาน เช่น

```
DAY#1 DATE 01/01/2013 AVG TEMP = 26.85
DAY#2 DATE 02/01/2013 AVG TEMP = 26.95
... คำนวณค่าและแสดงผลไปเรื่อยๆ จนครบวันสุดท้าย เช่นเมื่อ n = 365 ...
DAY#365 DATE 31/12/2013 AVG TEMP = 24.20
```

นิยามของฟังก์ชัน

4. ให้เขียนนิยามของฟังก์ชันสำหรับหาค่าอุณหภูมิต่ำที่สุดในข้อมูลที่เก็บในอาร์เรย์ `M` ซึ่งเก็บค่าข้อมูลสภาพอากาศเป็นจำนวน `n` วัน ฟังก์ชันส่งค่ากลับเป็นค่าอุณหภูมิต่ำสุดที่พบ ไม่มีการพิมพ์ผลลัพธ์ใดๆ ในฟังก์ชันนี้ (3 คะแนน)

ต้นแบบของฟังก์ชัน `float findMinTemp(MeteoRecord M[], int n)`

นิยามของฟังก์ชัน

5. ให้เขียนนิยามของฟังก์ชันสำหรับคำนวณค่าปริมาณน้ำฝนสะสมรายเดือนทั้ง 12 เดือน จากข้อมูลสภาพอากาศรายปีซึ่งเก็บอยู่ในอาร์เรย์ M (มีจำนวน n วัน) และให้แสดงค่าปริมาณน้ำฝนสะสม (Monthly accumulated rain) รายเดือนทั้ง 12 เดือน ตามลำดับ ฟังก์ชันไม่มีการส่งค่ากลับ (4 คะแนน)

ต้นแบบของฟังก์ชัน void accumRainMonthly(MeteoRecord M[], int n)

ตัวอย่างการทำงาน เช่น

```
MONTH#1 ACCUMULATED RAIN = 96.5 mm
MONTH#2 ACCUMULATED RAIN = 80.5 mm
MONTH#3 ACCUMULATED RAIN = 36.3 mm
MONTH#4 ACCUMULATED RAIN = 7.5 mm
MONTH#5 ACCUMULATED RAIN = 47.7 mm
MONTH#6 ACCUMULATED RAIN = 126.4 mm
MONTH#7 ACCUMULATED RAIN = 130.9 mm
MONTH#8 ACCUMULATED RAIN = 394.2 mm
MONTH#9 ACCUMULATED RAIN = 200.0 mm
MONTH#10 ACCUMULATED RAIN = 316.5 mm
MONTH#11 ACCUMULATED RAIN = 320.5 mm
MONTH#12 ACCUMULATED RAIN = 123.5 mm
```

นิยามของฟังก์ชัน

6. ให้เขียนนิยามของฟังก์ชันสำหรับ ค้นหาข้อมูลสภาพอากาศของวันใดวันหนึ่ง ในข้อมูลที่เก็บในอาร์เรย์ M ซึ่งเก็บค่าข้อมูลสภาพอากาศเป็นจำนวน n วัน และวันที่ซึ่งต้องการค้นหาคือ sd รับผ่านพารามิเตอร์ของฟังก์ชัน และฟังก์ชันนี้ส่งค่ากลับเป็นชนิด MeteorRecord คือข้อมูลสภาพอากาศของวันที่ที่ค้นเจอ (หากค้นไม่เจอวันที่ดังกล่าว จะกำหนดให้ค่าวันที่เป็น 0/0/0) ไม่มีการพิมพ์ผลลัพธ์ใดๆในฟังก์ชันนี้ (2 คะแนน)

ต้นแบบของฟังก์ชัน MeteorRecord findDateMeteo(MeteorRecord M[], int n, Date sd)

ตัวอย่างการเรียกใช้งานฟังก์ชันนี้

```
. . .
Date sd = {12, 5, 2013} ;
MeteorRecord m;
m = findDateMeteo(M, 365, sd);
if (m.date.day == 0)
    printf("Not found meteo record for date %d/%d/%d\n",
           sd.day, sd.month, sd.year);
else
{ printf("Meteo for date %d/%d/%d\n",
        m.date.day, m.date.month, m.date.year);
  printf("RAIN = %.2f mm\n", m.rain);
  printf("MIN TEMP = %.2f mm\n", m.minTemp);
  printf("MAX TEMP = %.2f mm\n", m.maxTemp);
}
. . .
```

นิยามของฟังก์ชัน

/***** จบตอนที่ 3 *****/

- เขียนและใช้งานฟังก์ชัน `input_data` เพื่อให้รู้ทำการกรอกเฉพาะคะแนนทั้ง 3 วิชา ได้แก่ Math, English, Thai ของ นร. ทุกชั้น ทุกคน (โดยไม่ต้องกรอกคะแนนรวม) กำหนดฟังก์ชัน `input_data` ถูกเรียกใช้งานจาก ฟังก์ชัน `main` และฟังก์ชันโปรโตไทป์ของ `input_data` คือ `void input_data (level C[])`; ซึ่งเป็นการส่งค่าแบบ Call-by-reference
- เขียนและใช้งานฟังก์ชัน `calculate_total` ในการคำนวณคะแนนรวม (Total) ของ นร. **ครั้งละ 1 คน** โดยกำหนดให้ ฟังก์ชัน `calculate_total` ถูกเรียกใช้งานจาก ฟังก์ชัน `main` และฟังก์ชันโปรโตไทป์ของ `calculate_total` คือ `float calculate_total (score S)`; ซึ่งเป็นการส่งค่าแบบ Call-by-value

ตัวอย่างผลรัน.....

```
Enter scores of MeeNoi[1] Student[1]: 80 90 90
Enter scores of MeeNoi[1] Student[2]: 76 88 87
Enter scores of MeeNoi[1] Student[3]: 99 90 90
.
.
.
Enter scores of MeeNoi[2] Student[1]: 81 70 92
Enter scores of MeeNoi[2] Student[2]: 80 90 90
.
.
.
Enter scores of MeeNoi[3] Student[1]: 80 65 90
Enter scores of MeeNoi[3] Student[2]: 76 68 94
.
.
.
Enter scores of MeeNoi[3] Student[50]: 80 66 90
```

จงเขียนโค้ดโปรแกรมให้สมบูรณ์ (10 คะแนน)

```
#include<stdio.h>
typedef struct
{    float Maths, English, Thai, Total;
}score;

typedef struct
{    score Student[50];
} level;

void input_data (level C[ ]);
float calculate_total (score S);
```

```
int main()
{
```

```
void input_data (level C[ ])
{
```

```
float calculate_total (score S)
{
```

```
/****** จบตอนที่ 2 *****/
```

ตอนที่ 3 (20 คะแนน, 10%, 60 นาที)

โปรแกรมประยุกต์

จงพิจารณาข้อมูลจากสถานีตรวจวัดสภาพอากาศ (meteorology station) มีการบันทึกค่าข้อมูลที่ตรวจวัดจากสภาพอากาศรายวัน โดยข้อมูลที่สนใจในที่นี้ประกอบด้วย ปริมาณน้ำฝน (rain, หน่วยวัด mm) อุณหภูมิสูงสุดและต่ำสุด (Min/Max Temperature, องศาเซลเซียส)

ตัวอย่างข้อมูล

Station: SKA17

Period: 01/01/2013 – 31/12/2013

Date	Rain (mm)	Min Temp (° C)	Max Temp (° C)
01/01/2013	0.0	20.5	33.2
02/01/2013	5.0	21.5	32.4
03/01/2013	12.3	19.8	30.5
...
29/12/2013	34.7	21.5	32.5
30/12/2013	15.0	22.0	30.8
31/12/2013	1.0	19.9	28.5

1. ให้กำหนดชนิดข้อมูล(โดยใช้ typedef) Date และ MeteoRecord แบบโครงสร้างข้อมูล โดยโครงสร้างข้อมูล Date (วันที่) มีสมาชิกประกอบด้วย day, month, year และ โครงสร้างข้อมูล MeteoRecord ประกอบด้วย date, rain, minTemp, maxTemp และ avgTemp (อุณหภูมิเฉลี่ย) ให้กำหนดชนิดข้อมูลอย่างเหมาะสม (3 คะแนน)