

PRINCE OF SONGKLA UNIVERSITY  
FACULTY OF ENGINEERING



Midterm Examination: ภาคการศึกษาที่ 1/2552

Date: 28 กรกฎาคม 2552 Time : 9.00- 12.00

Subject Number: 241-207 ( 2A0- 20A )

Room: R200, R300

Subject Title: Data Structures and Computer Programming Techniques

ทฤษฎีในการสอบ มีโทษขั้นต่ำ คือ ปรับตกในรายวิชาที่ทฤษฎี และพักรับการเรียน 1 ภาคการศึกษา

อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

รายละเอียดของข้อสอบ:

เวลา 3 ชั่วโมง (180 คะแนน: 180 นาที)

เอกสารมีทั้งหมด 10 หน้า (ไม่รวมหน้านี้) คำถามจำนวน 4 ข้อ

สิ่งที่สามารถนำเข้าห้องสอบได้:

ไม่อนุญาต: หนังสือ และเครื่องคิดเลข

คำแนะนำ:

- พยายามทำทุกข้อ
- คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด
- อ่านคำสั่งในแต่ละข้อให้ชัดเจนว่า เขียนโปรแกรมบางส่วน เขียนฟังก์ชัน หรือเขียนทั้งโปรแกรม รวมไปถึงข้อกำหนดเพิ่มเติม และหมายเหตุในข้อนั้นๆ
- การเขียนโปรแกรมในแต่ละข้อ อาจจะไม่ต้องเขียนตามคำสั่งย่อยทั้งหมด แต่คะแนนจะลดลงตามส่วน และหากในข้อใหญ่หนึ่งข้อ นักศึกษาไม่สามารถทำข้อย่อยข้อแรกๆ ได้ นักศึกษาสามารถทำข้อย่อยหลังๆ โดยให้อ้างอิงเหมือนนักศึกษาทำข้อย่อยข้อแรกๆ ได้
- การเขียน code จะต้องตั้งชื่อตัวแปรให้เหมาะสม และมี comment ในจุดสำคัญต่างๆ โดยให้ทั้งหมดเป็นไปตามหลักการเขียนโปรแกรมที่ดี

ข้อที่ 1 คำถามวัดความรู้พื้นฐาน

(40 คะแนน)

1.1 จากอาเรย์ a จงเขียนส่วนของโค้ด เพื่อพิมพ์เฉพาะตัวเลขที่เป็นค่าบวกเท่านั้น (3 คะแนน)

```
int a[5] = {3, 2, 7, -4, 6};
```

.....

.....

.....

1.2 จากตัวแปรพอยเตอร์ ptr และตัวแปรเลขจำนวนเต็ม x จงใช้พอยเตอร์ ptr เปลี่ยนค่าของ x ให้เป็น 10 (3 คะแนน)

```
int x = 0;  
int *ptr;
```

.....

.....

1.3 จงเขียนต้นแบบของฟังก์ชันที่เหมาะสมกับการใช้งานดังต่อไปนี้ (6 คะแนน)

a. หาว่าตัวอักษรที่กำหนดอยู่ในลำดับใดจาก a ถึง z

.....

b. นับจำนวนของอีลีเมนต์ที่มีค่าติดลบในอาเรย์ของ int

.....

c. หาตำแหน่งในหน่วยความจำของ สมาชิกตัวแรก ในอาเรย์ของ int ที่มีค่าเป็น 0

.....

1.4 โค้ดข้างล่างเป็นการค้นหาตำแหน่งของเลข 3 (พารามิเตอร์ตัวสุดท้ายของฟังก์ชัน find\_data) แล้วเปลี่ยนให้เป็นเลข 4 จงเขียนนิยามฟังก์ชัน find\_data เพื่อหาตำแหน่งในหน่วยความจำของข้อมูลที่ต้องการค้นหา โดยให้เขียนสอดคล้องกับโค้ด และผลลัพธ์การรันดังแสดงข้างล่าง (8 คะแนน)

```
int data[5] = {2, 5, 1, 3, 8};
int *ptr = find_data(data, 5, 3);
*ptr = 4;
for(i = 0; i < 5; i++)
    printf("%d ", data[i]);

----Result----
2 5 1 4 8
```

.....

.....

.....

.....

.....

.....

.....

1.5 จงเขียนส่วนของโค้ดที่ใช้พอยเตอร์ cptr และ Dereferencing แทน [] ที่ให้ผลแบบเดียวกันกับโค้ดข้างล่าง (10 คะแนน)

```
int i; //char *cptr;
char str[] = "Hello";
for(i = 0; str[i] != '\0'; i++)
    if(str[i] >= 'a' && str[i] <= 'z')
        str[i] += ('A' - 'a');
```

.....

.....

.....

.....

.....

.....

1.6 จากโค้ดที่กำหนดให้ จงเขียนส่วนของโค้ด เพื่อใช้พอยเตอร์ ptr กำหนดค่าเริ่มต้นให้กับ e โดยให้ name มีค่าเป็น "John Doe" และมีค่าเงินเดือนเป็น 25000 บาท (10 คะแนน)

```
typedef struct {
    char name[128];
    double salary;
} Employee;
Employee e;
Employee *ptr;
```

.....

.....

.....

.....

.....

.....

## ข้อที่ 2 ข้อความและตัวอักษร

(30 คะแนน)

จงเขียนฟังก์ชันเพื่อเข้ารหัสข้อความ โดยหลักการของการเข้ารหัส คือตัวหนังสือจะเปลี่ยนเป็นตัวหนังสือตัวถัดๆ ไปเป็นจำนวน n ตัว โดย n จะถูกกำหนดโดย pat [] ฟังก์ชันดังกล่าวจะรับพารามิเตอร์ str แล้วจะเปลี่ยนข้อความดังกล่าวเป็นข้อความที่เข้ารหัสใหม่แล้ว ทั้งนี้ฟังก์ชันจะต้องสนับสนุนการทำงานดังแสดงข้างล่าง

- ตัวอักษรตัวถัดไปของตัว z คือ a
- หากตัวอักษรเริ่มต้นเป็นตัวอักษรที่ไม่ใช่ตัวอักษร เช่นตัวเลข หรืออักขระพิเศษ ให้คงไว้เป็นตัวอักษรเดิม
- นอกจากตัวเลข และอักขระพิเศษ ตัวอักษรใน str เป็นตัวพิมพ์เล็กทั้งหมด
- ต้นแบบของฟังก์ชันคือ

```
void encrypt(char str[], int pat[], int pat_len);
```

โดยมีตัวอย่างการเข้ารหัสดังแสดงข้างล่าง

```
str[] => "* beyondz !"  
pat[] => {1, 4, 5, 3}  
pat_len => 4
```

การประมวลผล

*		b	e	y	o	n	d	z	!
1	4	5	3	1	4	5	3	1	4
*		g	h	z	s	s	g	a	!

ผลลัพธ์คือ "\* ghzssga!"

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

### ข้อที่ 3 อาร์เรย์ พอยเตอร์และหน่วยความจำ

(30 คะแนน)

โค้ดข้างล่าง เป็นโค้ดที่แสดงการประมวลผลกับอาร์เรย์ data จงใช้ข้อมูลในโค้ดตอบคำถามข้อ 3.1 และ 3.2

```
1 #include <stdio.h>
2 int* foo(int data[], int len);
3
4 int main()
5 {
6     int data[5] = {2, 1, 3, 4, 5};
7     int i, j;
8     int *result = foo(data, 5);
9     int *ptr = result;
10
11     for(i = 0; i < 5 - 1; i++){
12         for(j = 0; j < 3; j++)
13             printf("%d ", *(ptr++));
14         printf("\n");
15     }
16
17     return 0;
18 }
19
20 int* foo(int data[], int len){
21     int *result = (int*) malloc(sizeof(int) * len);
22     int *ptr = result;
23     int i;
24     for(i = 0; i < len - 1 ; i++){
25         *(ptr++) = data[i];
26         *(ptr++) = data[i+1];
27         *(ptr++) = data[i] + data[i+1];
28     }
29     return result;
30 }
```

3.1 จงเขียนผลลัพธ์ของโปรแกรมข้างต้น หากโปรแกรมไม่มีข้อผิดพลาดใดๆ เกี่ยวกับการจัดการหน่วยความจำ

(10 คะแนน)

.....

.....

.....

.....

.....

3.2 ได้ข้างต้นมีความผิดพลาดในการจงและคินหน่วยความจำหรือไม่ หากมี ให้แก้ไข โดยอ้างอิงหมายเลขบรรทัดที่แก้ไขหรือเพิ่มเติมในคำตอบให้ชัดเจน (20 คะแนน)

.....

.....

.....

.....

.....

.....

**ข้อที่ 4 โครงสร้างข้อมูล**

**(80 คะแนน)**

โปรแกรมข้างล่าง ใช้สำหรับการคำนวณ Cumulative Point (CP) เพื่อใช้ในการคำนวณ GPA แต่ละเทอม โดย CP จะได้มาจากการหาผลรวมของผลคูณระหว่าง Credit และ Level โดยค่าของ Level จะมาจากเกรด A=2, B+=1.5, B=1, C+=0.5, C=0, D+=-0.5, D=-1, E=-2

จงเขียนนิยามโครงสร้าง Student และนิยามของของฟังก์ชันในแต่ละช้อย่อย เพื่อให้โปรแกรม แสดงผลลัพธ์ดังนี้

```
Manit Sangsan
course-id   credit   level
=====
           1         3       2.0
           2         1      -1.0
=====
Cumulative Point = 5.0
```

```

1  typedef struct{
2      int course_id;
3      int credit;
4      float level;
5  } Course;
6
7  //define Student here
8
9
10 void s_init(Student *std, char *name, int max_course);
11 void g_add(Student *std, int course_id, int credit, int level);
12 float g_cumulate(Course courses[], int amount);
13 void s_print(Student *std);
14 void s_free(Student *std);
15
16 int main(){
17     Student std;
18     int i;
19
20     s_init(&std, "Manit Sangsarn", 2);
21     g_add(&std, 1, 3, 2);
22     g_add(&std, 2, 1, -1);
23
24     s_print(&std);
25     s_free(&std);
26
27     return 0;
28
29 }

```

4.1 นิยามโครงสร้าง Student อย่างเหมาะสม (ต้องมีการใช้โครงสร้าง Course)

(10 คะแนน)

.....

.....

.....

.....

.....







4.6 ฟังก์ชัน `s_clear` ทำให้โครงสร้างข้อมูลพร้อมใช้งานกับ `nr.คนถัดไป` โดยกำหนดให้ข้อมูลชื่อนักศึกษา และ รายละเอียดวิชาเป็นว่างเปล่า พร้อมทั้งคืนหน่วยความจำที่ไม่ต้องใช้งานแล้วอย่างเหมาะสม (10 คะแนน)

---

---

---

---

---

---

---

---

---

---