

PRINCE OF SONGKLA UNIVERSITY
FACULTY OF ENGINEERING



Midterm Examination: ภาคการศึกษาที่ 2/2553

Date: 24 ธันวาคม พ.ศ. 2553

Subject Number: 241-207

Room: S817, S203

Subject Title: Data Structures and Computer Programming Techniques

ทฤษฎีในการสอบ มีโทษขั้นต่ำ คือ ปรับตกในรายวิชาที่ทฤษฎี และพักการเรียน 1 ภาคการศึกษา

อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

รหัสนักศึกษา

รายละเอียดของข้อสอบ:

ชื่อ-สกุล

เวลา 3 ชั่วโมง (180 คะแนน: 180 นาที)

เอกสารมีทั้งหมด 7 หน้า (ไม่รวมหน้านี้)

สำหรับอาจารย์

คำถามจำนวน 4 ข้อ

สิ่งที่สามารถนำเข้าห้องสอบได้:

อนุญาต: เครื่องเขียน

ไม่อนุญาต: หนังสือ และเครื่องคิดเลข

ข้อ	คะแนน
1	
2	
3	
4	

คำแนะนำ:

- เขียนชื่อ (ไม่ต้องเขียนนามสกุล) และรหัสนักศึกษา ในทุกหน้า
- พยายามทำทุกข้อ และคำตอบทั้งหมดให้ทำในข้อสอบชุดนี้
- คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด
- อ่านคำสั่งในแต่ละข้อให้ชัดเจนว่า เขียนโปรแกรมบางส่วน เขียนฟังก์ชัน หรือเขียนทั้งโปรแกรม รวมไปถึงข้อกำหนดเพิ่มเติม และหมายเหตุในข้อนั้นๆ
- การเขียนโปรแกรมในแต่ละข้อ อาจจะไม่ต้องเขียนตามคำสั่งย่อยทั้งหมด แต่คะแนนจะลดลงตามส่วน
- การเขียน code จะต้องตั้งชื่อตัวแปรให้เหมาะสม และมี comment ในจุดสำคัญต่างๆ โดยให้ทั้งหมดเป็นไปตามหลักการเขียนโปรแกรมที่ดี

ข้อที่ 1 ความรู้พื้นฐาน

(60 คะแนน)

1.1 จากอาเรย์ a จงเขียนส่วนของโค้ด เพื่อนับจำนวนตัวเลขที่ติดลบ

(10 คะแนน)

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

.....

.....

.....

.....

1.2 จากข้อมูลที่กำหนดให้ จงเขียนคำสั่งตามที่กำหนดในแต่ละข้อย่อย

(10 คะแนน)

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

ให้ ptr อ้างอิงไปยังตัวแปร x

.....

เพิ่มค่าตัวแปร x ขึ้นอีก 1 ผ่านตัวแปร ptr

.....

1.3 จงเขียนส่วนของโค้ดที่ใช้พอยเตอร์ cptr และ * 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');
```

.....

.....

.....

.....

.....

.....

1.4 จากโค้ดที่กำหนดให้ จงเขียนส่วนของโค้ด เพื่อใช้พอยเตอร์ ptr กำหนดค่าให้กับ e โดยให้ name มีค่าเป็น "Jane Doe" และมีค่าเงินเดือนเป็น 20000 บาท (10 คะแนน)

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

.....
.....
.....

1.5 จงเขียนแผนภาพแสดงค่าของตัวแปร และการเชื่อมโยงของตัวแปรทุกตัวในหน่วยความจำ เมื่อโปรแกรมทำงานจนสิ้นสุด บรรทัดสุดท้ายของโค้ดข้างล่าง (20 คะแนน)

```
int a = 7, *p, b[5] = {2, 1}, *s[2];  
s[0] = &a;  
s[1] = b;  
*(s[1] + 2) = 7;  
p = &b[3];  
(*p)++;
```

ข้อที่ 2 ข้อความและตัวอักษร

(40 คะแนน)

จงเขียนฟังก์ชันเพื่อทำการตรวจสอบว่า ข้อความ s2 คล้ายคลึงกับข้อความ s1 หรือไม่ โดยมีข้อกำหนดพื้นฐานดังนี้

- ต้นแบบของฟังก์ชัน

```
int is_similar(char *s1, char *s2);
```

- ฟังก์ชันคืนค่า 1 หากข้อความคล้ายกัน แต่คืนค่า 0 ในทางตรงกันข้าม
- สามารถนิยามฟังก์ชันเพิ่มเติมได้
- ห้ามเรียกใช้ฟังก์ชันมาตรฐานเกี่ยวกับข้อความ ไม่ว่าจะเป็น strlen หรือ strcmp (หากผิดเงื่อนไขนี้ หัก 10 คะแนน)
- ห้ามใช้เครื่องหมาย [] ในโค้ด (หากผิดเงื่อนไขนี้ หัก 10 คะแนน)
- ถือว่าคล้ายกัน เมื่อข้อความมีความยาวเท่ากัน และมีอักขระต่างกันไม่เกิน 1 ตัว
 - abc คล้ายกับ abd เพราะความยาวเท่ากัน และมีตัวอักษรต่างกันเพียง 1 ตัว
 - abc ไม่คล้ายกับ ab เพราะความยาวไม่เท่ากัน

ให้ตอบในหน้าถัดไป

ข้อที่ 3 อาร์เรย์ พอยเตอร์และหน่วยความจำ

(30 คะแนน)

โค้ดข้างล่าง เป็นโค้ดที่แสดงการประมวลผลกับอาร์เรย์ data จงใช้ข้อมูลในโค้ดตอบคำถามข้อ 3.1 และ 3.2

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

#define LEN 5

int* foo(int data[], int len){
    int *r = (int*) malloc(sizeof(int));
    int *p = r;
    int i;
    for(i = 0; i < len; i++, p++){
        *p = data[i] + (data[i] % 2);
    }
    return r;
}

int main(){
    int a[5] = {1, 4, 3, 0, 9};
    int i;
    int *r = foo(a, LEN);

    for(i = 0; i < LEN; i++){
        printf("%d ", r[i]);
    }
    printf("\n");

    return 0;
}
```

3.1 โค้ดข้างต้นมีความผิดพลาดในการจัดการหน่วยความจำหรือไม่ หากมีให้แก้ไข หรือเพิ่มเติม ให้เขียนไว้ร่วมกับโค้ดด้านบน พร้อมคำอธิบายประกอบ

3.2 จงเขียนผลลัพธ์ที่ถูกต้องเมื่อรันโปรแกรมข้างต้น (หลังจากแก้ไขข้อผิดพลาดเกี่ยวกับการจัดการหน่วยความจำแล้ว)

.....

.....

ข้อที่ 4 โครงสร้างข้อมูล

(50 คะแนน)

จากโค้ดของที่กำหนดให้ จงเขียนฟังก์ชันที่ขาดหายไป โดยมีข้อกำหนดเพิ่มเติมว่า ฟังก์ชัน `total_distance` ต้องเรียกใช้ฟังก์ชัน `distance` อย่างเหมาะสม

```
typedef struct {
    int x;
    int y;
} Point;

float distance(Point *p1, Point *p2){
    return sqrt(pow(p1->x - p2->x, 2) +
                pow(p1->y - p2->y, 2));
}

int main(){
    Point p[5];
    int i;
    for(i = 0; i < 5; i++){
        printf("P%02d:", i + 1);
        input_point(&p[i]);
    }
    printf("=%.2f", total_distance(p, 5));
    return 0;
}
```

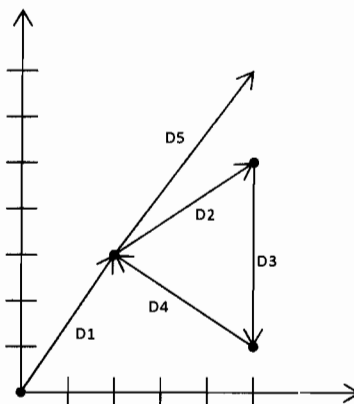
ได้ข้างต้นจะรับพิกัดจุด x, y จำนวน 5 จุด แล้วหาระยะทางการเดินทางระหว่างจุดตามลำดับ

ทั้งนี้ นศ. สามารถเลือกระดับความสมบูรณ์ของฟังก์ชันได้ตามความต้องการ โดยแต่ละระดับจะมีคะแนนสูงสุดต่างกัน

- ระดับที่ 1 (MAX 30) เป็นการหาระยะทาง โดยไม่มีเงื่อนไข
- ระดับที่ 2 (MAX 50) เป็นการหาระยะทาง โดยมีเงื่อนไขว่า หากจุดใหม่ซ้ำกับจุดก่อนหน้า จุดใดก็ตาม ให้ reset ระยะทางการเดินทางเป็น 0 อีกครั้ง

ตัวอย่างข้อมูลที่ป้อน

P01: <0 0>
P02: <2 3>
P03: <5 5>
P04: <5 1>
P05: <2 3>
P06: <5 7>



ให้ตอบในหน้าถัดไป

ระดับที่ 1 คำตอบคือ

$D1 + D2 + D3 + D4 + D5$

ระดับที่ 2 คำตอบคือ

$D5$ (เนื่องจากมีการย้อนกลับมาที่จุด (2, 3))