

PRINCE OF SONGKLA UNIVERSITY
FACULTY OF ENGINEERING



Midterm Examination: ภาคการศึกษาที่ 1/2555

Date: 31 กรกฎาคม 2555 09.00-11.00 น.

Subject Number: 241-207, 242-207

Room: A400, A403, S203

Subject Title: Data Structures and Computer Programming Techniques, Fundamental Programming I

ทฤษฎีในการสอบ มีโทษขั้นต่ำ คือ ปรับตกในรายวิชาที่ทฤษฎี และพักการเรียน 1 ภาคการศึกษา

อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

รหัสนักศึกษา

รายละเอียดของข้อสอบ:

ชื่อ-สกุล

เวลา 2 ชั่วโมง (120 คะแนน: 120 นาที)

เอกสารมีทั้งหมด 7 หน้า (ไม่รวมหน้านี้)

สำหรับอาจารย์

คำถามจำนวน 4 ข้อ

สิ่งที่สามารถนำเข้าห้องสอบได้:

อนุญาต: เครื่องเขียน

ไม่อนุญาต: หนังสือ และเครื่องคิดเลข

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

| ข้อ | คะแนน |
|-----|-------|
| 1 | |
| 2 | |
| 3 | |
| 4 | |

อ.สุธน: ผู้ออกข้อสอบ

ข้อที่ 1 ความรู้พื้นฐาน

(20 คะแนน)

1.1 จากอาร์เรย์ a จงเขียนส่วนของโค้ด เพื่อนับจำนวนตัวเลขที่เป็นเลขคู่

(5 คะแนน)

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

.....

.....

.....

.....

1.2 จากข้อมูลที่กำหนดให้ จงเขียนคำสั่งตามที่กำหนดในแต่ละข้อย่อย

(5 คะแนน)

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

ให้ ptr ช้างอิงไปยังตัวแปร x

.....

เพิ่มค่าตัวแปร x ขึ้นอีก 1 ผ่านตัวแปร ptr

.....

1.3 จงเขียนแผนภาพแสดงค่าของตัวแปร และการเชื่อมโยงของตัวแปรทุกตัวในหน่วยความจำ เมื่อโปรแกรมทำงานจนสิ้นสุด บรรทัดสุดท้ายของโค้ดข้างล่าง

(10 คะแนน)

```
int a = 7, *p, b[5] = {2, 1, 1, 3, 5}, *s[2];
s[0] = &a;
s[1] = &b[1];
*(s[1] + 2) = 6;
p = b + 3;
(*p)++;
```

ข้อที่ 2 ข้อความและตัวอักษร

(40 คะแนน)

จงเขียนฟังก์ชัน `ssub` เพื่อตัดส่วนของข้อความจาก ข้อความ `s` โดยใช้ตัวอักษร `a` และ ตัวอักษร `b` เป็นขอบเขตในการตัดข้อความ โดยมีข้อกำหนดพื้นฐานดังนี้

- ห้ามใช้เครื่องหมาย `[]` ในโค้ด (หากผิดเงื่อนไขนี้ หัก 10 คะแนน)
- ต้นแบบของฟังก์ชัน

```
char* ssub(char *s, char a, char b);
```

- ฟังก์ชัน `ssub` จะต้องมีการเรียกใช้ฟังก์ชัน `cfind` อย่างเหมาะสม เพื่อหาพอยเตอร์ที่ชี้ไปยังตัวอักษรที่ต้องการค้นหา (คืนค่าเป็น `NULL` หากหาไม่เจอ)

```
char* cfind(char *s, char key);
```

- ฟังก์ชัน `ssub` คืนค่าเป็นข้อความที่เก็บอยู่บนหน่วยความจำที่เกิดจากการ `malloc` เท่าที่จำเป็น โดยข้อความผลลัพธ์คือข้อความ ระหว่างตัวอักษร `a` ถึง `b` เท่านั้น (ไม่รวมตัว `a` และ ตัว `b` ในผลลัพธ์)
 - หากไม่เจอ `a` ในข้อความ ให้ผลลัพธ์คือ `NULL`
 - หากเจอ `a` แล้วไม่เจอ `b` ในข้อความ ให้ผลลัพธ์การตัดคือข้อความเริ่มต้นจากตัวอักษร `a` ไปจนถึงสิ้นสุดข้อความเดิม
 - `ssub("Hello+world", '+', 'l') => "wor"` `ssub("Hello+world", 'w', 'x') => "orld"`
 - `ssub("Hello+world", 'o', 'o') => "+w"` `ssub("Hello+world", 'x', 'd') => ""`

3.1 ส่วนนิยามฟังก์ชัน `cfind`

(10 คะแนน)

```
char* cfind(char *s, char key){
```

```
}
```

2.2 ส่วนนิยามฟังก์ชัน ssub

(30 คะแนน)

```

char* ssub(char *s, char a, char b){

    char *result, *ap, *bp;
    //ประกาศตัวแปรอื่นๆ ที่จำเป็นต้องใช้

    result = ap = bp = NULL;
    //หาพอยเตอร์ชี้ไปยังตัวอักษรหน้าสุดของข้อความที่จะเป็นผลลัพธ์ (5 คะแนน)

    if(ap != NULL){

        //หาพอยเตอร์หลังสุดของข้อความที่จะเป็นผลลัพธ์ (10 คะแนน)

        //จองหน่วยความจำเท่าที่จำเป็นต้องใช้ (10 คะแนน)
        //สามารถคำนวณขนาดที่ต้องจอง ได้จากการลบกันของพอยเตอร์

        //สำเนาข้อความไปยังหน่วยความจำที่จองไว้ (5 คะแนน)

    }

    return result;
}

```

ข้อที่ 3 โครงสร้างข้อมูล

(35 คะแนน)

จงเติมส่วนที่ขาดหายไป เพื่อรับเส้นทาง (Path) จำนวนหนึ่ง มากกว่า Polygon รูปใดมีระยะทางสั้นที่สุด

โดยทั้งนี้ฟังก์ชันต่างๆ ต้องถูกเรียกใช้อย่างเหมาะสม

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#define PATH_NUM 5

typedef struct {
    int x, y;
} Point;

typedef struct {
    char title[128];
    int num;
    Point *points;
} Path;

void input_path(Path *p);
float line_distance(Point *p1, Point *p2);
float path_distance(Path *p);
Path *shortest_path(Path ps[], int len);

int main(){
    Path ps[PATH_NUM];
    Path *p;
    int i;

    for(i = 0; i < PATH_NUM; i++)
        input_path(ps + i);

    p = shortest_path(ps, PATH_NUM);
    printf("Shortest Path = %s\n", p->title);
    //คืนหน่วยความจำที่จองจากการเรียกใช้ input_path (5 คะแนน)

    return 0;
}
```

```

void input_path(Path *p){
    int i;
    Point *tmp;

    printf("Title: ");
    scanf("%s", p->title);
    printf("Number: ");
    scanf("%d", &p->num);

    //จงหน่วยความจำอย่างเหมาะสม (10 คะแนน)

    for(i = 0; i < p->num; i++){
        tmp = p->points + i;
        scanf(" %d %d", &(tmp->x), &(tmp->y));
    }
}

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

float path_distance(Path *p){
    float distance = 0.0;
    int i;
    //หาระยะทางรวมของเส้นทาง (10 คะแนน)

    return distance;
}

```

```
Path *shortest_path(Path ps[], int len){
    //หาเส้นทางที่สั้นที่สุด
```

(10 คะแนน)

}

ข้อที่ 4 ไฟล์

(25 คะแนน)

นายดำขำกำลังติดเกมออนไลน์ RPG ตัวหนึ่งอย่างมาก โดยเกมนี้สามารถสร้างตัวละครหลักได้หลายตัว นายดำขำทราบมาว่าไฟล์เซฟเกมที่บ้านที่กข้อมูลของตัวละคร คือไฟล์ save.bin โดยเพื่อนของนายดำขำแจ้งว่า ไฟล์นี้ไม่ได้มีการเข้ารหัส และเก็บข้อมูลเป็นแบบไฟล์ไบนารีธรรมดา

ไฟล์ save.bin เริ่มต้นด้วยตัวเลขจำนวนเต็มหนึ่งตัว บอกจำนวนตัวละครที่ถูกบันทึกในไฟล์ ตามด้วยโครงสร้างข้อมูลที่เก็บข้อมูลของตัวละครทีละตัวละตามจำนวน นายดำขำ เริ่มต้นเขียนโค้ดไว้บางส่วนเพื่อแก้ไข max_hp ของตัวละครเป็น 9999 ชื่อ "Lucy" ขอให้ นศ. ช่วยนายดำขำเพิ่มเติมโค้ดให้สมบูรณ์

```

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

typedef struct {
    char name[128];
    ...
    int max_hp;
} Character;

int main(){
    int num, i, done = 0;
    Character c;

    FILE *fp = fopen("save.bin", "rwb");

    for(i = 0; i < num && !done; i++){

        if(strcmp(c.name, "Lucy") == 0){

            done = 1;
        }
    }

    fclose(fp);
    return 0;
}

```

```

size_t fread(void *ptr, size_t sz, size_t n, FILE *fp)
size_t fwrite(const void *ptr, size_t sz, size_t n, File *fp);
int fseek ( FILE * stream, long int offset, int origin );

```