

PRINCE OF SONGKLA UNIVERSITY  
FACULTY OF ENGINEERING



**Final Examination:** ภาคการศึกษาที่ 1/2554

**Date:** 9 ตุลาคม 2554 13.30-16.30 น.

**Subject Number:** 241-207

**Room:** A401, A400, A201, R200, S103

**Subject Title:** Data Structure and Computer Programming Techniques

ทฤษฎีในการสอบ มีโทษขั้นต่ำ คือ ปรับตกในรายวิชาที่ทฤษฎี และพักการเรียน 1 ภาคการศึกษา

อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

รายละเอียดของข้อสอบ:

ชื่อ .....

เวลา 3 ชั่วโมง

รหัสนักศึกษา .....

เอกสารมีทั้งหมด 8 หน้า (ไม่รวมหน้านี้) คำถามจำนวน 5 ข้อ

สำหรับอาจารย์

สิ่งที่สามารถนำเข้าห้องสอบได้:

อนุญาต: เครื่องเขียน

ไม่อนุญาต: เอกสาร อุปกรณ์คำนวณอิเล็กทรอนิกส์

คำแนะนำ:

- พยายามทำทุกข้อ
- คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด
- อ่านคำสั่งในแต่ละข้อให้ชัดเจนว่า เขียนโปรแกรมบางส่วน เขียนเพิ่มเติม และหมายเหตุในข้ออื่นๆ
- การเขียน code จะต้องตั้งชื่อตัวแปรให้เหมาะสม และมี comment ในจุดสำคัญต่างๆ โดยให้ทั้งหมดเป็นไปตามหลักการเขียนโปรแกรมที่ดี

ข้อ	คะแนน
1	
2	
3	
4	
5	

ชื่อ ..... รหัสนักศึกษา.....

ข้อที่ 1 ไฟล์

(15 คะแนน)

1.1 จงเขียนส่วนของโค้ดเพื่อเปิดไฟล์ไบนารีชื่อ `game.sav` และทำการแก้ไขตัวเลขจำนวนเต็ม (`int`) ในตำแหน่ง 2440 ให้มีค่าเป็น 1.2 เท่าของค่าเดิม โดยให้สมมติฐานว่าสามารถแก้ไขไฟล์ได้โดยไม่ติดปัญหาใดๆ (9 คะแนน)

.....

.....

.....

.....

.....

.....

1.2 จงบอกว่าโปรแกรมในแต่ละข้อย่อย ควรจะประมวลไฟล์แบบใด ระหว่าง ไฟล์ข้อความ ไฟล์ไบนารี หรือ ใช้ได้ทั้งสองชนิด เพราะเหตุใด (6 คะแนน)

a. โปรแกรมสุ่มแสดงคติประจำวัน

.....

.....

b. โปรแกรมเก็บข้อมูลของพนักงานในบริษัท โดยมีการแก้ไขจำนวน ชม.ทำงานในแต่ละวัน ทุกวัน

.....

.....

c. โปรแกรมที่ทำการประมวลผลไฟล์ HTML

.....

.....

ชื่อ ..... รหัสนักศึกษา .....

## ข้อที่ 2 ลิงค์ลิสต์

(40 คะแนน)

จากโครงสร้างข้อมูลที่กำหนดตอบคำถามข้อ 2.1-2.3

```

struct listnode
{
    int data;
    struct listnode *next;
    struct listnode *before; //ใช้เฉพาะข้อ 2.1
};
typedef struct listnode ListNode;
typedef ListNode *LNP;
void delete_node(LNP *head, LNP target);
int is_ordered(LNP head);

```

2.1 จงเขียนนิยามของฟังก์ชัน `delete_node` เพื่อลบโหนด `target` ออกจากลิสต์ ทั้งนี้ให้ใช้ประโยชน์ของ `Doubly Linked List` อย่างเหมาะสม (ไม่ต้องใช้การวนซ้ำ) (10 คะแนน)

.....

.....

.....

.....

.....

.....

.....

.....

ชื่อ ..... รหัสนักศึกษา .....

2.2 จงเขียนแผนภาพแสดงหน่วยความจำ เมื่อมีการ execute ชุดคำสั่ง (เป็นคำตอบต่อเนื่อง)

(5 คะแนน \* 3)

```
int i, data;
LNP head, tail, tmp;

head = tail = (LNP)malloc(sizeof(ListNode));
head->data = 0;
head->next = NULL;
```

```
for(i = 0; i < 4; i++){
    tmp = (LNP)malloc(sizeof(ListNode));
    tmp->data = tail->data + 2;
    tmp->next = NULL;
    tail->next = tmp;
    tail = tmp;
}
head->next->next->data = 7;
```

```
tmp = head->next;
head->next = tmp->next->next;
```



ชื่อ ..... รหัสนักศึกษา .....

3.2 จงแสดงข้อมูลในแบบ pre-order, in-order และ post-order

(10 คะแนน)

..... Pre-order: .....

..... In-order: .....

..... Post-order: .....

ข้อที่ 4 การเรียงลำดับ

(20 คะแนน)

4.1 จงแสดงขั้นตอน การเรียงลำดับข้อมูล 5 1 2 3 4 6 จากน้อยไปหามาก โดยใช้อัลกอริทึม Selection Sort และ Insertion Sort

(10 คะแนน)

Selection Sort	Insertion Sort

ชื่อ ..... รหัสนักศึกษา.....

4.2 จากตารางการวัดประสิทธิภาพที่ให้ จงแสดงความคิดเห็นว่า เราควรเลือกใช้อัลกอริทึมใดในการเรียงลำดับข้อมูล เมื่อ

- ข้อมูลเรียงลำดับแล้ว
- จำนวนข้อมูลไม่เกิน 100 ตัว
- การสลับตำแหน่งข้อมูล สร้างภาระให้กับระบบ 100 เท่า เมื่อเทียบกับการเปรียบเทียบค่าข้อมูล

(10 คะแนน)

	Selection	Insertion	Bubble	Quicksort
<b>Best</b>		$C = O(n)$ $X = O(n)$	$C = O(n)$ $X = O(1)$	$C = O(n \log n)$ $X = O(1)$
<b>Worst</b>	$C = O(n^2)$ $X = O(n)$	$C = O(n^2)$ $X = O(n^2)$	$C = O(n^2)$ $X = O(n^2)$	$C = O(n^2)$ $X = O(n^2)$
<b>Average</b>		$C = O(n^2)$ $X = O(n^2)$	$C = O(n^2)$ $X = O(n^2)$	$C = O(n \log n)$ $X = O(n \log n)$

best best case  
good average

.....

.....

.....

.....

.....

.....

.....

ชื่อ ..... รหัสนักศึกษา .....

## ข้อที่ 5 ประยุกต์ใช้

(25 คะแนน)

นิยามฟังก์ชันต่างๆ ของโปรแกรม เพื่อรับข้อความซึ่งเป็นการเขียนนิพจน์ทางคณิตศาสตร์แบบ Postfix โดยให้ตั้งสมมติฐานว่านิพจน์ดังกล่าว จะประกอบด้วย ตัวเลขเฉพาะหลักหน่วย และเครื่องหมาย บวก ลบ และคูณเท่านั้น แต่อาจมีเว้นวรรคหรือไม่เว้นวรรคก็ได้ ทั้งนี้จะต้องมีการจัดการตัวแปร size อย่างเหมาะสม

หลังจากนั้นทำการคำนวณหาค่าผลลัพธ์โดยใช้ stack ในการประมวลผล

หากนิพจน์ที่ผู้ใช้ป้อน มี **operator** หรือ **operand** มากไป หรือน้อยไป โปรแกรมจะหยุดการประมวลผล และแสดงข้อความ **Invalid expression!**

```
#include<stdio.h>
struct listnode{
    int data;
    struct listnode *next;
};
typedef struct listnode LN;
typedef LN *LNP;
typedef struct{
    int size;
    LNP head;
} Stack;

int main(){
    Stack stack = {0, NULL};
    char s[128]; char *cptr;
    printf("Enter: ");
    gets(s);
    tmp = s;
    while(*cptr!= '\0'){
        if(*cptr>= '0' && *cptr<= '9'){
            push(&stack, *cptr - '0');
        }
        else if(*cptr == '+' || *cptr == '-' || *cptr == '*'){
            if(cal(&stack, *cptr) == -1){
                printf("Invalid expression!\n");
                break;
            }
        }
        cptr++;
    }
    printf("= %d", pop(&stack));

    return 0;
}
```

รูปแบบการแสดงผล

```
Enter: <921+->
= 6
```