

PRINCE OF SONGKLA UNIVERSITY
FACULTY OF ENGINEERING



Final Examination: ภาคการศึกษาที่ 1/2555

Date: 7 ตุลาคม 2555 เวลา 13.30-16.30

Subject Number: 241-207, 242-207

Room: S203, A400, R201

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

ทุจริตในการสอบ มีโทษขั้นต่ำ คือ ปรับตกในรายวิชาที่ทุจริต และพักการเรียน 1 ภาคการศึกษา

อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

รายละเอียดของข้อสอบ:

ชื่อ

เวลา 2 ชั่วโมง

รหัสนักศึกษา

เอกสารมีทั้งหมด 7 หน้า (ไม่รวมหน้านี้) คำตามจำนวน 4 ข้อ

สำหรับอาจารย์

สิ่งที่สามารถนำเข้าห้องสอบได้:

อนุญาต: เครื่องเขียน

ไม่อนุญาต: เอกสาร อุปกรณ์คำนวนอิเล็กทรอนิกส์

ข้อ	คะแนน
1	
2	
3	
4	

คำแนะนำ:

- พยายามทำทุกข้อ
- คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด
- อ่านคำสั่งในแต่ละข้อให้ชัดเจนว่า เขียนโปรแกรมบางส่วน เขียนเพิ่มเติม และหมายเหตุในข้อนั้นๆ
- การเขียน code จะต้องตั้งชื่อตัวแปรให้เหมาะสม และมี comment ในจุดสำคัญต่างๆ โดยให้หังนมเป็นไปตามหลักการเขียนโปรแกรมที่ดี

ชื่อ รหัสนักศึกษา.....

ข้อที่ 1 ลิงค์ลิสต์

(40 คะแนน)

จากโครงสร้างข้อมูลที่กำหนดตอบคำถามข้อ 1.1-1.3

```
struct listnode
{
    int data;
    struct listnode *next;
};

typedef struct listnode ListNode;
typedef ListNode *LNP;
void insert_before(LNP head, int data, LNP target);
int is_sequence(LNP head); // 2 3 4 => 1, 2 3 5 => 0
```

- 1.1 จงเขียนนิยามของฟังก์ชัน `insert_before()` เพื่อเพิ่ม data ลงไปในลิงค์ลิสต์ก่อนโน้นด้วย target ทั้งนี้ให้อ้วว่า ฟังก์ชันนี้จะไม่ถูกเรียกเมื่อลิงค์ลิสต์ว่าง หรือ target เท่ากับ head (15 คะแนน)

ชื่อ

รหัสนักศึกษา.....

1.2 จงเขียนແພນກາພແສດງໜ່າຍຄວາມຈຳ ເມື່ອມີກາຣ execute ປຸດຄຳສັ່ງ (ເປັນຄຳຕອບຕ່ອນູ່ອງ)

(5 ຄະແນນ * 2)

```
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;
```

1.3 ຈົນຍາມພັງກົນ `is_sequence()` ເພື່ອຕຽບສອບວ່າຂ້ອມລິລິກົດລິສົດທີ່ກຳຫນດໄທ້ ເປັນເລຂເຮີຍກັນ (ເພີ່ມຂຶ້ນທີ່ລະ 1)
ຫຼືອໍາໄນ ໂດຍຄືນ 1 ພາກເປັນເລຂເຮີຍ ແລະ ຄືນ 0 ໃນການຜົດຮັງກັນຂ້າມ

(15 ຄະແນນ)

ชื่อ _____ รหัสนักศึกษา _____

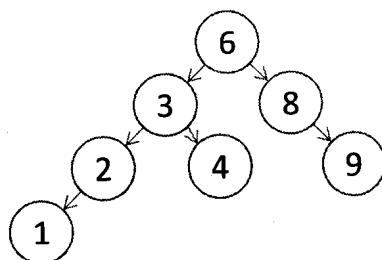
ข้อที่ 2 ต้นไม้

(30 คะแนน)

จาก BST ที่กำหนดให้ ตอบคำถามข้อ 2.1-2.3

2.1 จงตอบคำถามเกี่ยวกับคุณลักษณะของ Tree ตามที่กำหนดให้ (5 คะแนน)

គុណភាពកម្មណ៍	គំពាប
Root	
Children ទី 3	
Descendants ទី 3	
Leaf	
គរាយ	



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

(10 คะแนน)

Pre-order:

In-order:

Post-order:

ชื่อ

รหัสนักศึกษา.....

2.3 จากฟังก์ชัน `insert_node()` ในสไลด์ที่ 53-54 จะเขียนส่วนของโคดเพื่อสร้างต้นไม้ตามข้อ 2.1 โดยมีขั้นตอนสำคัญดังนี้ต่อไปนี้

- สร้างอาร์เรย์ของเลขจำนวนเต็ม และกำหนดให้ข้อมูลเป็นตัวเลขที่จะนำไปสร้างต้นไม้ โดยจะต้องมีลำดับของตัวเลขที่เหมาะสม
- ทำการวนซ้ำเพื่อเรียกใช้ฟังก์ชัน `insert_node()` อย่างเหมาะสม

```
BST insert_node(BST t, int value){
    if (t == NULL) {
        t = malloc(sizeof(BSTNODE));
        t->data = value;
        t->leftptr = NULL;
        t->rightptr = NULL;
    } else if (value < t->data)
        t->leftptr = insert_node(t->leftptr, value);
    else if (value > t->data)
        t->rightptr = insert_node(t->rightptr, value);
    else
        printf("duplicate node\n");
    return t;
}
```

ชื่อ รหัสนักศึกษา

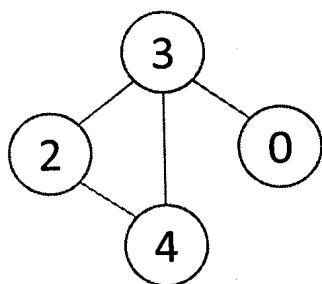
ข้อที่ 3 ตารางແຍ່ງ และกราฟ

(25 คะแนน)

- 3.1 จงอธิบายว่า เพราะเหตุใดการค้นหาข้อมูลที่ต้องการผ่านการใช้ตารางແຍ່ງมีความเร็วสูงกว่าการใช้อาร์ทัวร์ไปใช้ภาพประกอบคำอธิบาย
(15 คะแนน)

- 3.2 จงใช้โครงสร้างข้อมูลที่กำหนดให้ เพื่อเขียนแผนภาพ Adjacency Lists ของกราฟข้างล่าง (10 คะแนน)

```
struct node {
    int vertex;
    struct node *link;
};
```



ชื่อ รหัสนักศึกษา

ข้อที่ 4 ประยุกต์ใช้

(25 คะแนน)

นิยามฟังก์ชัน cal() และ result() ที่จะทำให้โปรแกรมรับข้อมูลซึ่งเป็นการเขียนนิพจน์ทางคณิตศาสตร์แบบ Postfix โดยให้ตั้งสมมติฐานว่า นิพจน์ดังกล่าว จะประกอบด้วย ตัวเลขเฉพาะหลักหน่วย และเครื่องหมาย บวก ลบ และคูณเท่านั้น แต่อาจมี เว้นวรรคหรือไม่วนวรรคก็ได้

หลังจากนั้นทำการคำนวนหาค่าผลลัพธ์โดยใช้ stack ในการประมวลผล

หากนิพจน์ที่ผู้ใช้ป้อน มี **operand** มากเกินไป ให้แสดงข้อความ **Too many operands.** แต่หากมี **operator** มาเกินไป ให้ แสดงข้อความ **Too many operators.**

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

int main(){
    LNP stack = 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("Too many operators.\n");
                break;
            }
        }
        cptr++;
    }
    result(&stack); //show the result

    return 0;
}
```

รูปแบบการแสดงผล

Enter: <921+->
= 6