PRINCE OF SONGKLA UNIVERSITY

FACULTY OF ENGINEERING

Final Examination: ภาคการศึกษาที่ 1/2558 **Date:** 14 ธันวาคม พ.ศ. 2558 (13.30-15.30 น.)

Subject Number: 241-207, 242-207 **Room:** A201, A401, S817

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

ทุจริตในการสอบ มีโทษขั้นต่ำ คือ ปรับตกในรายวิชาที่ทุจริต และพักการเรียน 1 ภาคการศึกษา

อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

รายละเอียดของข้อสอบ:

เวลา 2 ชั่วโมง (120 คะแนน: 120 นาที)

เอกสารมีทั้งหมด 7 หน้า (ไม่รวมหน้านี้)

คำถามจำนวน 4 ข้อ

สิ่งที่สามารถนำเข้าห้องสอบได้:

อนุญาต: เครื่องเขียน

ไม่อนุญาต: หนังสือ และเครื่องคิดเลข

รหัสนักศึกษา .....

ชื่อ-สกุล .....

สำหรับอาจารย์

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

อ.สุธน: ผู้ออกข้อสอบ

ข้อที่ 1 **Warm Up** **(30 คะแนน)**

1.1 จงเลือกโครงสร้างข้อมูลที่เหมาะสมกับโปรแกรมประยุกต์แต่ละข้อ โดยโครงสร้างข้อมูลที่มีให้เลือกใช้ มีดังนี้

a) Array with dynamic memory allocation
b) Linked List
c) Hash Table
d) Binary Search Tree (6 คะแนน)

ลักษณะการทำงาน	คำตอบ (เลือกที่ถูกที่สุด)
เน้นการเข้าถึงข้อมูลแบบสุ่ม โดยระบุตำแหน่ง (index) ที่ต้องการเข้าถึง	
เน้นการแทรกและลบข้อมูล	
เน้นการอ่านข้อมูล โดยต้องการประสิทธิภาพในการค้นหาข้อมูลสูงสุด	

1.2 จากโครงสร้างข้อมูลที่กำหนด จงเขียนนิยามของฟังก์ชัน **update_score** เพื่อเปลี่ยนคะแนนของนักเรียนที่ถูกเก็บไว้ใน **node** ที่ส่งเป็นพารามิเตอร์ (8 คะแนน)

```
struct student {
  char code[5];
  float score;
}
struct listnode {
   struct student s;
   struct listnode *next;
};
void update_score(struct listnode* node, float new_score);
```

.....

.....

.....

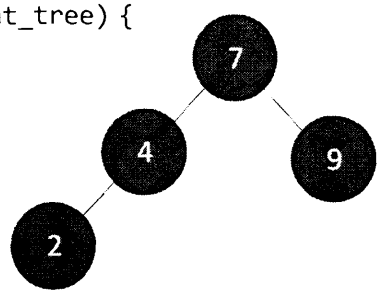
.....

1.3 จากฟังก์ชัน mktree ที่กำหนดให้ จงเรียกใช้ฟังก์ชันดังกล่าว เพื่อสร้างทรีดังรูปข้างล่าง (8 คะแนน)

```
TREE mk_tree(int x, TREE left_tree, TREE right_tree) {
  TREE temp;

  temp = (TREE) malloc(sizeof(TREENODE));
  temp->data = x;
  temp->left = left_tree;
  temp->right = right_tree;

  return temp;
}
```



1.4 จากกราฟที่กำหนดให้ จงเขียนโครงสร้างข้อมูลแทนกราฟ โดยใช้เทคนิค Adjacency Matrix และ Adjacency Lists (เขียนทั้งสองอย่าง) (8 คะแนน)

ข้อที่ 2 Linked List (35 คะแนน)

จงใช้โครงสร้างข้อมูลข้างล่าง เขียนโปรแกรมตามข้อกำหนดในข้อ 2.1 และ 2.2

```
struct listnode {
   char c;
   struct listnode *next;
};
typedef struct listnode LN;
```

2.1 จงเขียนฟังก์ชัน string_to_list เพื่อสร้างลิงค์ลิสต์จากข้อความที่กำหนดให้ โดยแต่ละโหนดของลิงค์ลิสต์ จะเก็บตัวอักษรแต่ละตัวของข้อความตามลำดับ (ไม่ต้องเก็บ \0 ไว้ในลิงค์ลิสต์) (15 คะแนน)

```
LN *string_to_list(char *s){  //return head of linked list

   LN *head = NULL;
   LN *tail = NULL;
   LN *current = NULL;

   return head;

}
```

2.2 จงเขียนฟังก์ชัน `str_end` เพื่อตัดข้อความ (ลิงค์ลิสต์) ให้มีขนาดสั้นลง โดยฟังก์ชันจะตัดอักขระทั้งหมดที่อยู่หลังอักขระ `c` (พารามิเตอร์ของฟังก์ชัน) (20 คะแนน)

- หากไม่ปรากฏอักขระ `c` การเรียกใช้ฟังก์ชันนี้ จะไม่มีผลกับข้อความเดิม
- หากอักขระ `c` เป็นตัวอักษรแรกของข้อความ เทียบเท่ากับการทำให้ลิงค์ลิสต์ว่าง (Empty)
- หากข้อความเดิม คือ `"hello"` หลังการเรียกใช้ฟังก์ชัน `str_end(&head, 'l')` แล้ว ข้อความในลิงลิสต์ จะเหลือเพียง `"he"`

```
void str_end(LN **hptr, char c){

   LN *current = *hptr;
   LN *previous = NULL;
   LN *tmp;









}
```

ข้อที่ 3 More Structures (20 คะแนน)

3.1 จงอธิบายด้วยภาพ เราสามารถนำ stack มาคำนวณนิพจน์ Postfix 397+* ได้อย่างไร (10 คะแนน)

3.2 หากเราต้องการเก็บเลขจำนวนเต็มจำนวนหนึ่งลงใน hash table โดยกำหนดให้ hash value เกิดจากการนำค่าของตัวเลขจำนวนเต็มมอดูลัสกับขนาดของอาเรย์ (n = 10) (10 คะแนน)

เลขจำนวนเต็มที่ต้องการเก็บใน hash table มีดังนี้

5 12 2 8 7 15 16

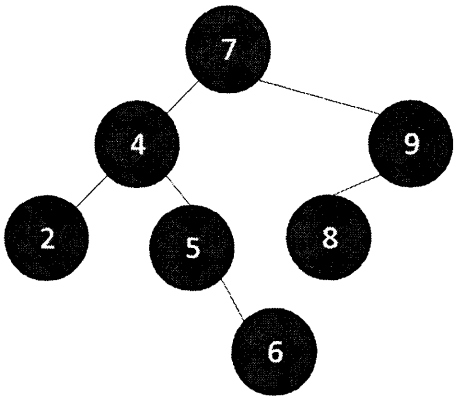
จงแสดงผลลัพธ์ของ hash table

ข้อที่ 4 Tree (35 คะแนน)

จาก BST ที่กำหนดให้ ตอบคำถามข้อ 4.1-4.2

4.1 จงตอบคำถามเกี่ยวกับคุณลักษณะของทรี ตามที่กำหนดให้ (5 คะแนน)

คุณลักษณะ	**คำตอบ**
Root	
Children ของ 4	
Descendants ของ 4	
Leaf	
ความสูง	



4.2 จงแสดงข้อมูลในแบบ pre-order, in-order และ post-order (9 คะแนน)

Pre-order: .....

In-order: .....

Post-order: .....

4.3 จงอธิบายถึงคุณลักษณะของ BST และระบุข้อดีของ BST โดยให้กล่าวถึง Big O Notation ด้วย (6 คะแนน)

.....

.....

.....

.....

.....

4.4 จากโครงสร้างข้อมูลที่กำหนดให้ (15 คะแนน)

```
struct treenode{
   struct treenode *left;
   int data;
   struct treenode *right;
};
typedef struct treenode TREENODE;
typedef TREENODE *TREE;
```

จงเขียนฟังก์ชัน **sum_leaf** เพื่อหาผลรวมของ data ใน leaves ของทรี