

PRINCE OF SONGKLA UNIVERSITY
FACULTY OF ENGINEERING



Final Examination: ภาคการศึกษาที่ 1/2553

Date: 10 ตุลาคม 2553 13.30-15.30 น.

Subject Number: 241-207

Room: S201, S203, A400

Subject Title: Data Structure and Computer Programming Techniques

ทฤษฎีในการสอบ มีโทษขั้นต่ำ คือ ปรับตกในรายวิชาที่ทฤษฎี และพักการเรียน 1 ภาคการศึกษา

อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

รายละเอียดของข้อสอบ:

ชื่อ

เวลา 2 ชั่วโมง (120 คะแนน: 120 นาที)

รหัสนักศึกษา

เอกสารมีทั้งหมด 6 หน้า (ไม่รวมหน้านี้) คำถามจำนวน 4 ข้อ

สำหรับอาจารย์

สิ่งที่สามารถนำเข้าห้องสอบได้:

อนุญาต: เครื่องเขียน กระดาษ A4 จำนวน 1 แผ่น เขียนด้วยลายมือ

(เขียนชื่อ รหัสนักศึกษา และแนบมากับข้อสอบด้วย)

ไม่อนุญาต: อุปกรณ์คำนวณอิเล็กทรอนิกส์

คำแนะนำ:

- พยายามทำทุกข้อ
- คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด
- อ่านคำสั่งในแต่ละข้อให้ชัดเจนว่า เขียนโปรแกรมบางส่วน เขียนฟังก์ชัน หรือเขียนทั้งโปรแกรม รวมไปถึงข้อกำหนดเพิ่มเติม และหมายเหตุในข้อนั้นๆ
- การเขียน code จะต้องตั้งชื่อตัวแปรให้เหมาะสม และมี comment ในจุดสำคัญต่างๆ โดยให้ทั้งหมดเป็นไปตามหลักการเขียนโปรแกรมที่ดี

ข้อ	คะแนน
1	
2	
3	
4	

ชื่อ รหัสนักศึกษา

ข้อที่ 1 ไฟล์

(15 คะแนน)

1.1 จงเขียนส่วนของโค้ดเพื่อเปิดไฟล์ `game.sav` ซึ่งเป็นไฟล์ไบนารี และทำการแก้ไขตัวเลขจำนวนเต็ม (`int`) ในตำแหน่ง 2440 ให้มีค่าเป็น 80 โดยให้สมมติฐานว่าสามารถแก้ไขไฟล์ได้โดยไม่ติดปัญหาใดๆ (9 คะแนน)

.....

.....

.....

.....

.....

.....

.....

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

a. โปรแกรมเข้ารหัสไฟล์

.....

.....

b. โปรแกรมเก็บข้อมูลของพนักงานในบริษัท โดยมีการแก้ไขจำนวน ชม.ทำงานในแต่ละวัน ทุกวัน

.....

.....

c. โปรแกรมที่ทำการประมวลผลไฟล์ HTML

.....

.....

ชื่อ วิทยาลัยศึกษา

ข้อที่ 2 Linked List**(40 คะแนน)**

2.1 จงสร้าง struct ชื่อ intdl ที่สามารถนำไปใช้ในการสร้าง Doubly Linked List ที่แต่ละโหนดสามารถเก็บตัวเลขจำนวนเต็มได้หนึ่งตัว (4 คะแนน)

.....

.....

.....

.....

.....

2.2 จงเลือกโครงสร้างข้อมูลที่เหมาะสมกับโปรแกรมประยุกต์แต่ละข้อ (6 คะแนน)

โดยโครงสร้างข้อมูลที่มีให้เลือกใช้ มีดังนี้

- Array with static memory allocation (ระบุจำนวนข้อมูลใน ซอร์สโค้ด)
- Array with dynamic memory allocation (ระบุจำนวนโดยการใช้ malloc)
- Linked List
- Linked List of Array

ลักษณะการทำงาน	คำตอบ (เลือกที่ถูกต้องที่สุด)
ผู้ใช้ระบุจำนวนของข้อมูลเมื่อรันโปรแกรม เน้นการอ่านข้อมูลเพียงอย่างเดียว	
รับข้อมูลจากผู้ใช้ โดยไม่จำกัดจำนวน เน้นการอ่านข้อมูล โดยต้องการประสิทธิภาพในการค้นหาข้อมูลสูงสุด	
เน้นการแทรกและลบข้อมูล	

ชื่อ รหัสนักศึกษา

จากโครงสร้างข้อมูลที่กำหนดตอบคำถามข้อ 2.3-2.5

```
struct listnode
{
    int data;
    struct listnode *next;
};
typedef struct listnode ListNode;
typedef ListNode *LNP;
void modify_data(LNP node, int new_value);
```

2.3 จงเขียนนิยามของฟังก์ชัน `modify_data` เพื่อเปลี่ยน `data` ของ `node` ให้เป็น `new_value` หาก `node` ดังกล่าว ไม่ใช่โหนดสุดท้ายของลิสต์ (5 คะแนน)

.....

.....

.....

.....

.....

2.4 จงเขียนแผนภาพแสดงหน่วยความจำ เมื่อมีการ `execute` ชุดคำสั่ง (เป็นคำตอบต่อเนื่อง) โดยข้อมูลที่ผู้ใช้ทำการป้อนให้กับโปรแกรมเป็นตัวเลขตามลำดับดังนี้ 5 2 3 6 (5 คะแนน * 3)

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

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

ชื่อ..... รหัสนักศึกษา.....

```
for(i = 0; i < 4; i++){
    scanf("%d", &data);
    tmp = (LNP)malloc(sizeof(ListNode));
    tmp->data = data;
    tmp->next = NULL;
    tail->next = tmp;
    tail = tmp;
}
head->next->next->next->data = 7;
```

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

2.5 จงเขียนส่วนของโค้ดเพื่อทำการคืนค่าหน่วยความจำ **ทั้งหมด** ที่ทำการจองไว้ในข้อ 2.4 (10 คะแนน)

.....

.....

.....

.....

.....

.....

.....

.....

.....

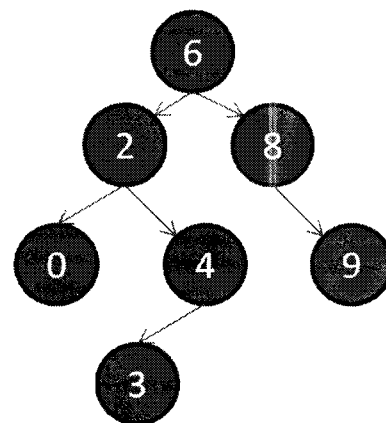
ชื่อ รหัสนักศึกษา

ข้อที่ 3 Tree**(40 คะแนน)**

จาก BST ที่กำหนดให้ ตอบคำถามข้อ 3.1-3.2

3.1 จงตอบคำถามเกี่ยวกับคุณลักษณะของ Tree ตามที่กำหนดให้ (10 คะแนน)

คุณลักษณะ	คำตอบ
Root	
Children ของ 2	
Descendents ของ 2	
Leaf	
ความสูง	



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

(9 คะแนน)

Pre-order:

In-order:

Post-order:

ใช้ข้อมูลที่กำหนดให้ตอบคำถามข้อ 3.3-3.4

```

struct treenode{
    struct treenode *leftptr;
    int data;
    struct treenode *rightptr;
};
typedef struct treenode TREENODE;
typedef TREENODE *TREE;
void mirror_tree(TREE t);
TREE join_tree(TREE lt, TREE rt, TREENODE *new_root);
  
```

ชื่อ รหัสนักศึกษา

3.3 จงเขียนฟังก์ชัน `join_tree` เพื่อเชื่อมต้นไม้สองอันเข้าด้วยกัน โดยให้โหนด `new_root` เป็นรากใหม่ของต้นไม้ และ `lt` อยู่ด้านซ้าย ในขณะที่ `rt` อยู่ด้านขวา ทั้งนี้ฟังก์ชันนี้จะคืนค่า เป็นรากใหม่ (6 คะแนน)

.....

.....

.....

.....

.....

.....

3.2 จงนิยามฟังก์ชัน `mirror_tree` เพื่อ mirror ต้นไม้ โดยโหนดที่อยู่ด้านขวาจะไปอยู่ด้านซ้าย และโหนดที่อยู่ด้านซ้ายจะไปอยู่ด้านขวา (15 คะแนน)

.....

.....

.....

.....

.....

.....

ชื่อ รหัสนักศึกษา

ข้อที่ 4 ประยุกต์ใช้**(25 คะแนน)**

ในการทดลองการเดินของหุ่นยนต์เดินบนพื้นที่ที่จัดเตรียมไว้ ข้อมูลพิกัดจุด x, y ที่หุ่นยนต์เคลื่อนที่ไป ถูกส่งอย่างต่อเนื่องผ่านสัญญาณคลื่นความถี่สูงมายังอุปกรณ์รับข้อมูลที่เชื่อมต่ออยู่กับคอมพิวเตอร์ศูนย์กลาง

เนื่องจากหุ่นยนต์มีหน่วยความจำที่จำกัดมาก จึงไม่สามารถเก็บจุดพิกัดทั้งหมดไว้ที่ตัวหุ่นยนต์เอง ท่านได้รับมอบหมายให้เพิ่มเติมโปรแกรมที่ทำงานบนคอมพิวเตอร์ศูนย์กลาง เพื่อบันทึกพิกัดจุด ทั้งหมดที่ได้เดินไปแล้ว และตรวจสอบว่าหุ่นยนต์ได้เดินไปยังจุดพิกัดที่เคยเดินผ่านไปแล้วหรือไม่

โปรแกรมเมอร์ก่อนหน้านี้ ได้วางแผนงาน เพื่อพัฒนาส่วนการบันทึกและตรวจสอบพิกัดจุดไว้ล่วงหน้าแล้ว โดยมีฟังก์ชันที่เกี่ยวข้อง 2 ฟังก์ชันคือ

- `init_record` เรียกอัตโนมัติหนึ่งครั้ง เมื่อหุ่นยนต์อยู่ที่พิกัด $(0,0)$ และพร้อมเดิน
- `record_step` จะถูกเรียกอัตโนมัติทุกครั้งเมื่อหุ่นยนต์เดิน

โดยโปรแกรมมีการนิยามต้นแบบของฟังก์ชัน และโครงสร้างข้อมูลที่เกี่ยวข้องดังนี้

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

void init_record();
int record_step(Point *p);
//p เป็นพอยต์เตอร์ที่อ้างอิงไปยังพิกัดจุดล่าสุดของหุ่นยนต์ที่เดินไป - เปลี่ยนแปลงทุกครั้งที่มีการเดิน
//ฟังก์ชันคืนค่า 0 หากไม่ได้เดินไปซ้ำจุดที่เคยเดิน แต่คืนค่า 1 ในกรณีตรงกันข้าม
```

ทั้งนี้ท่านสามารถที่จะประกาศโครงสร้างข้อมูลเพิ่มเติม หรือประกาศตัวแปร global (เฉพาะกรณีจำเป็น)

จงอธิบายพร้อมแสดงโค้ดประกอบ

.....

.....

.....

ชื่อ

รหัสนักศึกษา

Handwriting practice area consisting of multiple horizontal dashed lines.