ชื่อ-นามสกุล_____ รหัสนักศึกษา_____

PRINCE OF SONGKLA UNIVERSITY

FACULTY OF ENGINEERING

Final Examination: Semester 2 **Academic Year:** 2008-2009

Date: 19 February 2009 **Room:** R300

Time: 9.00 – 12.00 am. **Lecturers**: อ.อารีย์ ธีรภาพเสรี, อ.อัมรินทร์ ดีมะการ

Subject: 241-207 Data Structure and Computer Programming Techniques

---

ทุจริตในการสอบ มีโทษขั้นต่ำ คือ ปรับตกในรายวิชาที่ทุจริต และพักการเรียน 1 ภาคการศึกษา

อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

รายละเอียดของข้อสอบ:

เวลา 3 ชั่วโมง (80 คะแนน: 180 นาที)

ข้อสอบมีทั้งหมด 9 หน้า (รวมใบปะหน้า) ประกอบด้วยคำถามจำนวน 6 ข้อ

สิ่งที่สามารถนำเข้าห้องสอบได้:

อนุญาต: เครื่องเขียนต่าง ๆ เช่น ปากกา หรือดินสอ

ไม่อนุญาต: หนังสือ และเครื่องคิดเลข

คำแนะนำ:

- พยายามทำทุกข้อ
- คำตอบทั้งหมดจะต้องเขียนลงในช่องว่างที่เว้นไว้ในข้อสอบ
- <u>คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด</u>
- อ่านคำสั่งในแต่ละข้อให้ชัดเจนว่า เขียนโปรแกรมบางส่วน เขียนฟังก์ชัน หรือเขียนทั้งโปรแกรม รวมไปถึงข้อกำหนดเพิ่มเติม และหมายเหตุในข้อนั้นๆ
- การเขียน code จะต้องตั้งชื่อตัวแปรให้เหมาะสม และมี comment ในจุดสำคัญต่างๆ โดยให้ทั้งหมดเป็นไปตามหลักการเขียนโปรแกรมที่ดี

	1	2	3	4	5	6	Total (80)
คะแนน							

ต้นแบบของฟังก์ชันที่สำคัญ

Functions declared in stdio.h

```
int fclose(FILE *stream);
FILE *fopen(const char *filename, const char *opentype);
int fprintf(FILE *stream, const char *template,…);
int fputc(int c, FILE *stream);
int fputs(const char *s, FILE *stream);
int fscanf(FILE *stream, const char *template,…);
size_t fread(void *data, size_t datasize, size_t nb2read,
FILE *stream);
size_t fwrite(const void *data, size_t datasize, size_t
nb2write, FILE *stream);
fseek(FILE *stream, long int offset, int whence);
```

Function declared in string.h

```
int strcmp(const char *target, const char *source);
char *strcpy(char *target, const char *source);
size_t strlen(char *str);
```

Function declared in stdlib.h

```
void *malloc(size_t size);
void *calloc(size_t num, size_t size);
void free(void *ptr);
```

1. สมมติว่านักศึกษาได้รับคำสั่งให้สร้าง Linked list สำหรับเก็บเวลาที่ทำได้ในการแข่งขันของนักกีฬาแต่ละคน ในการแข่งขันวิ่งรายการหนึ่ง โดย Linked list นี้เก็บข้อมูลโดย**เรียงลำดับโหนดจากนักกีฬาที่ใช้เวลาน้อยที่สุด ไปยังนักกีฬาที่ใช้เวลามากที่สุด** จากข้อสมมตินี้ ให้นักศึกษาตอบคำถามต่อไปนี้

1.1 **จงออกแบบโครงสร้างข้อมูล `athlete`** ที่รองรับโครงสร้างข้อมูล Linked list ที่ต้องการ โดยให้**แต่ละโหนด**ประกอบไปด้วย ชื่อนักกีฬา (`name`) และ เวลาที่นักกีฬาคนนั้นทำได้ในการแข่งขัน (`completion_time` มีหน่วยเป็นวินาที) (2 คะแนน)

```
struct athlete {




};
```

1.2 จงเขียน**ส่วนของโปรแกรม**สำหรับ**สร้างโหนดใหม่** เพื่อเก็บข้อมูลของนักกีฬาชื่อ Winnie ที่มีผลการแข่งขันเป็นเวลา 14.38 วินาที (ใช้โครงสร้าง `athlete` ในข้อ 1.1) (4 คะแนน)

กำหนดให้ `struct athlete` ***alist** เป็น linked list ที่เก็บข้อมูลของนักกีฬาในการแข่งขันรายการหนึ่ง จงใช้ตัวแปรนี้ในการตอบคำถามข้อ 1.3 – 1.6 ต่อไป

1.3 ถ้ามี `struct athlete *new_node` ที่เก็บข้อมูลนักกีฬาคนใหม่ ซึ่งมีผลการแข่งขันที่ใช้เวลาน้อยที่สุด กว่าทุกข้อมูลใน `alist` จงเขียน**ส่วนของโปรแกรม เพื่อเพิ่ม `new_node` เข้าไปใน `alist`** (2 คะแนน)

1.4 นักกีฬาชื่อ Winnie ถูกตัดสินว่าใช้สารกระตุ้นในการแข่งขัน และต้องถูกตัดสิทธิ์ออกจากรายชื่อนักกีฬา **จงเขียน<u>ส่วนของโปรแกรม</u>**เพื่อลบโหนดของนักกีฬาคนนี้ ออกจาก `alist` (8 คะแนน)

1.5 จงเขียนฟังก์ชัน **`void save2file`**`(struct athlete *list)` เพื่อเก็บข้อมูลโครงสร้าง (`athlete2`) ของนักกีฬาแต่ละคน (ชื่อและเวลาที่ใช้ในการแข่งขัน) ที่อ่านได้จาก `struct athlete *list` ลงใน `Binary file` ชื่อ `competition.dat` (8 คะแนน)

```
struct athlete2 {




};
```

```
void save2file(struct athlete *list){
```

--}

1.6 จงเขียนฟังก์ชัน `void display(float` **`time`**`)` ที่ใช้ในการอ่านข้อมูลจาก `Binary file` ชื่อ `competition.dat` (ที่ได้มาจากข้อ 1.5) แล้ว**<u>พิมพ์ชื่อนักกีฬา</u>**ทุกคนที่มีเวลาที่ใช้ในการแข่งขัน**<u>น้อยกว่าหรือเท่ากับค่า time</u>** ที่รับเข้ามา (7 คะแนน)

----

2. จากข้อมูลที่กำหนดให้เป็นผลของการท่องไปใน Binary Tree (BT) แบบ inorder traversal และ postorder traversal โดยที่ BT มีทั้งหมด 10 โหนดแต่ละโหนดมีค่าตั้งแต่ 0 ถึง 9

ผลของ **inorder traversal**: 4 1 5 6 2 0 8 3 9 7

ผลของ **postorder traversal**: 4 6 5 2 1 8 9 7 3 0

จงตอบคำถามต่อไปนี้

2.1 **จงวาดรูปของ Tree** จากข้อมูลของการ traversal ที่กำหนดให้ (6 คะแนน)

2.2 จากภาพของ Tree ที่วาดได้ในข้อ 2.1 จงหาค่าต่อไปนี้ (4 คะแนน)

a) leave nodes ของ tree

---

b) root node ของ tree

---

c) children ของ root node

---

d) ความสูงของ tree (height of binary tree)

---

3. จากโครงสร้างข้อมูลและต้นแบบฟังก์ชันที่กำหนดให้ จงเขียนฟังก์ชันสำหรับการทำงานดังต่อไปนี้

```
struct treenode
{
      int data;
      struct treenode *leftptr,*rightptr;
};
typedef struct treenode TREENODE;
typedef TREENODE *TREE;
int isBST(TREE t);
int maxNode(TREE t);
int max(int a, int b); //if(a>b) return a; else return b;
```

(หมายเหตุ: ฟังก์ชัน `max()` เป็นฟังก์ชันที่สามารถเรียกใช้ได้เลย โดยไม่ต้องสร้างขึ้นเองใหม่)

3.1 **จงเขียนฟังก์ชัน** `isBST(TREE t)` ในการตรวจสอบว่า `t` เป็น `Binary Search Tree` หรือไม่ ถ้าใช่ให้ส่งค่ากลับเป็น 1 ถ้าไม่ใช่ให้ส่งค่ากลับเป็น 0 (8 คะแนน)

3.2 **จงเขียนฟังก์ชัน** `maxNode(TREE t)` ในการค่า data สูงสุดใน `TREE t` (7 คะแนน)

4. จงตอบคำถามต่อไปนี้

4.1 จงหา Big-O ของและอัลกอริธึมดังต่อไปนี้ (4 คะแนน)

a) อัลกอริธึมในการค้นหาแบบที่ 1 ใช้เวลา **n + nlogn**

---

b) อัลกอริธึมในการค้นหาแบบที่ 2 ใช้เวลา **n^3+2n+5**

4.2 จงเขียนลำดับของขั้นตอนในการค้นหาแบบ Binary Search Algorithm (4 คะแนน)

---

5. จงเขียนผลลัพธ์ที่เปลี่ยนไปในแต่ละรอบของการเรียงลำดับข้อมูลด้วย Insertion sort และ Quick sort

ตัวอย่าง การแสดงผลลัพธ์ในแต่ละรอบ
ข้อมูล: 3 5 2 7 4 2
รอบที่ 1: 3 2 5 4 2 7
รอบที่ 2: 2 3 4 2 5 7
รอบที่ 3: 2 3 2 4 5 7
รอบที่ 4: 2 2 3 4 5 7
หยุด

5.1 Insertion sort (4 คะแนน)

ข้อมูลที่ต้องการเรียงคือ: **5 6 1 7 2 3**

5.2 Quick sort (6 คะแนน)

กำหนดให้ Pivot เป็นข้อมูลทางขวามือสุด และ

ให้การจบของแต่ละรอบคือการสลับค่า pivot ไปยังตำแหน่งที่ถูกต้อง

ข้อมูลที่ต้องการเรียงคือ: **5 6 9 1 7 2 3 8 10**

---

6. จงเปรียบเทียบประสิทธิภาพ ในแง่ของเวลา (Best-case, Worst-case และ Average-case) ที่ใช้ในการทำงาน ในรูปแบบ O() ของ sorting algorithm ดังต่อไปนี้ (6 คะแนน)

- Bubble sort
- Quick sort

Best-case	
Worst-case	
Average-case	
