

เลขที่ห้องสอบ

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

ข้อสอบกลางภาคการศึกษาที่ 2/2550

วันพุธที่ 19 ธันวาคม 2550

เวลา 09.00 - 12.00 น.

วิชา CPE 130 Algorithms and Data Structures.

น.ศ. วิศวกรรมคอมพิวเตอร์ชั้นปีที่ 1A,B

คำสั่ง

1. ข้อสอบมีทั้งสิ้น 5 ข้อ จำนวน 7 แผ่น(รวมแผ่นนี้) ทำทุกข้อลงในข้อสอบที่เว้นช่องไว้ให้
2. ไม่อนุญาตให้นำเครื่องคำนวณใดๆเข้าห้องสอบ
3. ห้ามนำเอกสารใดๆ เข้าห้องสอบ
4. เขียนชื่อ และ รหัสประจำตัว ลงในกระดาษคำตอบทุกแผ่น

.....
(อ.พิพัฒน์ ศุภศิริสันต์)

ผู้ออกข้อสอบ

ข้อสอบนี้ได้ผ่านการประเมินจากภาควิชาวิศวกรรมคอมพิวเตอร์แล้ว

ชื่อรหัสประจำตัว.....ภาควิชา/ชั้นปี.....

ชื่อ รหัสประจำตัว.....เลขที่นั่งสอบ.....

กำหนดให้ โครงสร้างนี้ใช้ได้ตั้งแต่ข้อ 1 ถึงข้อ 4

50211501	Mr.KRIT	BOONYARANG	3.38	41
----------	---------	------------	------	----

โครงสร้างข้อมูลสำหรับการประกาศผลการเรียนประกอบด้วย รหัสประจำตัว ชื่อ นามสกุล เกรดเฉลี่ย และลำดับที่ ตามลำดับ โดยมีการกำหนดโครงสร้างไว้แล้วดังนี้

typedef struct { long int id ;

char firstname[20], lastname[30];

float gpa;

int order } st_type ;

โครงสร้างนี้ใช้สำหรับเก็บข้อมูลนักเรียน 1 คน

typedef struct st_tag { st_type info;

struct st_tag *next; } st_link ;

โครงสร้างนี้ใช้สำหรับสร้างโหนดของลิงค์ลิสต์

typedef struct link_tab { st_link *first , *last } table_type ;

โครงสร้างนี้ใช้สำหรับสร้างเป็นตารางตัวเลขเพื่อทำ radix sort

ชื่อ รหัสประจำตัว.....เลขที่นั่งสอบ.....

1. สมมุติให้ข้อมูลทั้งหมดเก็บอยู่ในตัวแปรอาร์เรย์ชื่อ student[] ตั้งแต่ตำแหน่งที่ 1 .. stcount

จงสร้างฟังก์ชันชื่อ Binary_Search_ID เพื่อค้นหาว่า ตัวแปร idkey ที่รับเข้ามาทางพารามิเตอร์ว่าสอดคล้องกับรหัสประจำตัวของนักเรียนที่มีอยู่ในอาร์เรย์หรือไม่ โดยใช้เทคนิคของ Binary Search ถ้าค้นเจอ ให้ Return ตัวเลขตำแหน่งของอาร์เรย์ที่ค้นเจอ แต่ถ้าค้นไม่เจอให้ return ค่าเป็น 0 โดยมีตัวอย่างการเรียกใช้ฟังก์ชันดังนี้ (10 points)

```
found = Binary_Search_ID (student , stcount, idkey);
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

ชื่อ..... รหัสประจำตัว..... เลขที่นั่งสอบ.....

2) จงเขียนฟังก์ชัน Quick Sort (พร้อมทั้ง function swap ที่ใช้สลับตำแหน่งข้อมูล) เพื่อเรียงลำดับข้อมูลที่อยู่ในอาร์เรย์ student[] ตั้งแต่ตัวที่ 1.. stcount โดยใช้วิธีแบ่งกลุ่มข้อมูลแบบ Partitioning โดยมีวิธีเรียกใช้ดังนี้ (14 points)

```
QuickSort (student , 1 , stcount );
```

ข้อสอบวิชา CPE 130 Algorithms and Data Structures กลางภาคการศึกษาที่ 2/2550 หน้า ที่ 4

ชื่อ รหัสประจำตัว..... เลขที่นั่งสอบ.....

3) ถ้าต้องการนำข้อมูลทั้งหมดที่อยู่ในอาร์เรย์ student[] ตั้งแต่ 1.. stcount มาสร้างเป็น linked list โดยมีฟังก์ชันชื่อ makelink() ทำหน้าที่นำข้อมูลที่เก็บอยู่ในอาร์เรย์ มาสร้างเป็น linked list โดยมีส่วนของการเรียกใช้ดังนี้ (10 points)

```
for (i = 1; i <= stcount; i++)  
    makelink (data[i], first, last) ;
```

จงสร้างฟังก์ชัน makelink()

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

5. กำหนดให้ ฟังก์ชัน $f(x,y)$ มีเงื่อนไขในการหาคำตอบดังนี้

$$f(x,y) = \begin{cases} f(x-y, y) & \text{เมื่อ } x > y \\ f(y-x, x) & \text{เมื่อ } x < y \\ x & \text{เมื่อ } x = y \end{cases}$$

5.1 จงสร้างฟังก์ชันชื่อ GCD() โดยมี input parameter เป็นเลขจำนวนเต็ม x และ y แล้ว

return คำตอบเป็นตัวเลขจำนวนเต็ม ตามเงื่อนไขที่กำหนดไว้(5 points)

.....

.....

.....

.....

.....

.....

.....

.....

5.2 จงแสดงตัวอย่างการหาคำตอบของ GCD(24,16) (1 points)

.....

.....

.....

.....

.....

.....

5. กำหนดให้ ฟังก์ชัน $f(x,y)$ มีเงื่อนไขในการหาคำตอบดังนี้

$$f(x,y) = \begin{cases} f(x-y, y) & \text{เมื่อ } x > y \\ f(y-x, x) & \text{เมื่อ } x < y \\ x & \text{เมื่อ } x = y \end{cases}$$

5.1 จงสร้างฟังก์ชันชื่อ GCD() โดยมี input parameter เป็นเลขจำนวนเต็ม x และ y แล้ว return คำตอบเป็นตัวเลขจำนวนเต็ม ตามเงื่อนไขที่กำหนดไว้

```
int GCD(int x, int y)

{ if (x > y) return GCD(x-y, y);

    else if (x < y) return GCD(y-x, x);

        else return x ;

}
```

5.2 จงแสดงตัวอย่างการหาคำคำตอบของ GCD(24,16)

$$\mathbf{GCD(24, 16) = GCD(24-16, 16) = GCD(8, 16)}$$

$$\mathbf{GCD(8, 16) = GCD(16-8, 8) = GCD(8, 8)}$$

$$\mathbf{GCD(8, 8) = 8}$$