



มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
การสอบปลายภาคเรียนที่ 1 ปีการศึกษา 2559

วิชา CPE 100 Computer Programming for Engineers.

วศ.คอมพิวเตอร์ ปีที่ 1 กลุ่มที่ 1, 2

สอบวันศุกร์ที่ 25 พฤศจิกายน 2559

เวลา 13.00 -16.00 น.

คำชี้แจง

1. ข้อสอบแบ่งเป็น 6 ข้อ จำนวน 8 แผ่น ประกอบด้วยใบปะหน้าข้อสอบ 1 แผ่น ตัวข้อสอบ 7 แผ่น (หน้า 2 - 8) มีคะแนนรวม 40 คะแนน
2. เขียนคำตอบลงในส่วนที่กำหนดไว้ท้ายข้อ
3. ไม่อนุญาตให้นำ เอกสาร หรือ เครื่องคำนวณใดๆ เข้าห้องสอบ
4. เขียนชื่อ และ รหัสประจำตัว ลงในกระดาษคำตอบทุกแผ่น (และแผ่นนี้)

ชื่อ-สกุล..... รหัสนักศึกษา.....

เมื่อนักศึกษาทำข้อสอบเสร็จ ต้องยกมือบอกกรรมการคุมสอบ

เพื่อขออนุญาตออกนอกห้องสอบ

ห้ามนักศึกษานำข้อสอบและกระดาษคำตอบออกนอกห้องสอบ

นักศึกษาซึ่งทุจริตในห้องสอบ อาจถูกพิจารณาโทษสูงสุดให้พ้นสภาพการเป็นนักศึกษา

(อ.พิพัฒน์ สุภศิริสันต์)

ผู้ออกข้อสอบ (086 770 7033)

ข้อสอบนี้ได้ผ่านการประเมินจากคณะกรรมการภาควิชาวิศวกรรมคอมพิวเตอร์

(รศ.ดร.พิรพล ศิริพงศ์วุฒิกุล)

ประธานหลักสูตร

วันที่ 11 เดือน พ.ย. พศ. 2559

กำหนดโครงสร้างข้อมูลสำหรับการตัดเกรดไว้ สำหรับเขียนโปรแกรมข้อ 3-4 ดังนี้

```
typedef struct grade_info {    char id[14] ;  
                                char name[40] ;  
                                double mid, final, atten, total , gpoint ;  
                                char grade[3] ;  
                                } gradetype;  
  
gradetype x, st[100] ;  
  
int stcount , success ;
```

3. สมมติว่ามีการอ่านข้อมูลนักศึกษา และตัดเกรดไว้แล้วเก็บอยู่ในตัวแปรชื่อ st จำนวน stcount ต้องการ ค้นหา และแสดงผลนักศึกษาที่มีคะแนนสอบปลายภาคน้อยกว่าคะแนนสอบกลางภาค

Show_weak_student(st, stcount);

โดยการแสดงผลในแต่ละบรรทัดจะมี 2 ส่วน คือส่วนที่เกิดจากการนับจำนวน และส่วนที่เกิดจากการเรียกใช้ ฟังก์ชัน ซึ่งจะได้ตัวอย่างผลลัพธ์ดังนี้

1) 57070501006	CHAYAWUT EANGCHA-UM	10	7	10	27	F
2) 57070501008	CHUTIPON WONGSIRIVIPAT	16	9	11	36	D
3)						

จงสร้างฟังก์ชันที่ทำงานสอดคล้องกับวิธีเรียกใช้ ที่กำหนด (6 คะแนน)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

6. คุณ SID อยากนับจำนวนครั้งของการแชทของตัวเองในกลุ่ม "CPE100EZ" จึงส่งข้อมูลประวัติการแชทออกมาเป็นไฟล์ไฟล์ชื่อ"[LINE] แชทใน CPE100EZ" แล้วนำไปเปิดในเครื่องคอมพิวเตอร์ พบว่า ไฟล์มีคุณสมบัติเป็น text โดยบรรทัดที่มีข้อมูลการแชทจะมี 3 ฟิลด์ คือ เวลาบันทึก ชื่อในไลน์(อาจมีเว้นวรรคป้อนอยู่ในได้) และข้อความที่แชท และใช้ตัวคั่นข้อมูลคือ '\t' (TAB delimited) ซึ่งเป็น white space ชนิดหนึ่ง(มองไม่เห็น และปิดบรรทัดด้วยรหัส '\n') ส่วนบรรทัดที่ไม่มีรหัส '\t' จะไม่ใช่ข้อมูลการแชท ดังตัวอย่าง

[LINE] ประวัติการสนทนากับ CPE100EZ

วันเวลาบันทึก : 25/11/2559 13:35

.....

ศ. 25/11/2559

07:10	SID	[สติ๊กเกอร์]
07:11	SID	สวัสดีเพื่อนๆ ทุกคน วันนี้มีสอบ CPE100 นะ
07:15	DIEGO	หวัดดี SID ดื่นเช้าจัง สอบวันนี้จะทำได้มัย
07:20	SHIRA	ดีจ้าทุกคน SID ยังไงเชอก็ A อยู่แล้วใช้มัย EZ มั๊ยจ

.....

กำหนดให้มีฟังก์ชันชื่อ is_chat(char *buff) ทำหน้าที่ตรวจสอบว่า buff ใชข้อความแชทหรือไม่ เช่น

"ศ. 25/11/2559" // จะรีเทิร์นค่าเป็น 0

"07:10 SID [สติ๊กเกอร์]" // จะรีเทิร์นค่าเป็น 1

```
int is_chat(char *buff)
{ char *s;
  if (strchr(buff, '\t') == NULL)
    return 0;
  else
    return 1;
}
```

จงเขียนโปรแกรมเพิ่มเติมเพื่อแยกข้อมูล"ชื่อในไลน์"ออกจากสตริงบรรทัดที่มีข้อมูล แล้วนำมาเปรียบเทียบกับชื่อ user ที่ส่งมา เพื่อนับว่าในไฟล์ที่กำหนดให้มีการแชทของ user คนนั้นอยู่รวมทั้งหมดกี่ครั้ง แล้วรีเทิร์นคำตอบเป็นจำนวนครั้งที่นับได้ (8 คะแนน)

จากตัวอย่าง ถ้าใช้คำสั่ง Count_Line_User ("[LINE] แชทใน "CPE100EZ", "SID") จะต้องรีเทิร์นค่าเป็น 2

คำแนะนำ เปิดไฟล์อ่านทีละบรรทัด แล้วเช็คบรรทัดนั้นว่าใช้ได้หรือไม่ แล้วจึงดึงข้อมูลมาเปรียบเทียบ และนับ

```
int Count_Line_User (char *filename, char *user)
```

```
{ FILE *fp;
```

```
char timestr[10], linename[100] ;
```

```
int count ;
```

}

กำหนดให้ รูปแบบ function prototype ที่อาจเกี่ยวข้องกับข้อสอบชุดนี้

```
int fclose(FILE *stream)
```

```
int feof(FILE *stream)
```

FILE *fopen(const char *filename, const char *mode)

```
int fprintf(FILE *stream, const char *format, ...)
```

```
int fscanf(FILE *stream, const char *format, ...)
```

```
char *fgets(char *str, int n, FILE *stream)
```

```
int sscanf(const char *str, const char *format, ...)
```

```
char *strchr(const char *str, int c)
```

```
int strcmp(const char *str1, const char *str2)
```

```
char *strcpy(char *dest, const char *src)
```

```
char *strstr(const char *haystack, const char *needle)
```

char *strtok(char *str, const char *delim)