



มหาวิทยาลัยสงขลานครินทร์  
คณะวิศวกรรมศาสตร์

สอบกลางภาค: ภาคการศึกษาที่ 2

ปีการศึกษา: 2556

วันที่สอบ: 13 มกราคม 2557

เวลาสอบ: 09.00 – 12.00 น.

รหัสวิชา: 241-207 / 242-207

ห้องสอบ: R201

ชื่อวิชา: Computer Programming Techniques / Programming Fundamentals I

ทุจริตในการสอบ มีโทษขั้นต่ำ คือ ปรับตกในรายวิชาที่ทุจริต และพักการเรียน 1 ภาคการศึกษา

คำสั่ง: อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

อนุญาต: เครื่องเขียนต่างๆ เช่น ปากกา หรือดินสอ เข้าห้องสอบ

ไม่อนุญาต: หนังสือ หรือเครื่องคิดเลขเข้าห้องสอบ และเอกสารใดๆ เข้าและออกห้องสอบ

เวลา: 3 ชั่วโมง (180 นาที)

คำแนะนำ

- ข้อสอบมี 13 หน้า (รวมใบปะหน้า) รวม 180 คะแนน คิดเป็นคะแนนเก็บ 30 %
- คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด
- อ่านคำสั่งในแต่ละข้อให้ชัดเจนว่า เขียนโปรแกรม เขียนฟังก์ชัน หรือเขียนบางส่วนของโปรแกรม รวมไปถึงข้อกำหนดเพิ่มเติม และหมายเหตุในข้อนั้นๆ
- หากข้อใดเขียนตอบไม่พอ ให้เขียนเพิ่มเติมที่ด้านหลังของหน้านั้นเท่านั้น

ชื่อ \_\_\_\_\_ รหัส \_\_\_\_\_ กลุ่ม \_\_\_\_\_

ตอน 1 (30)	ตอน 2 (30)	ตอน 3 (60)	ตอน 4 (60)	รวม 180

ทุจริตในการสอบ โทษขั้นต่ำคือปรับตกในรายวิชานี้และพักการเรียน 1 ภาคการศึกษา

## ตอนที่ 1 ความรู้พื้นฐาน

(30 คะแนน 30 นาที)

1.1 จากอาร์เรย์ a จงเขียนส่วนของโค้ด เพื่อนับจำนวนของตัวเลขจำนวนเต็มลบ (5 คะแนน)

```
int a[5] = {2, -4, 0, -8, 7}
```

1.2 จากข้อมูลที่กำหนดให้ จงเขียนคำสั่งและตอบคำถามตามที่กำหนดในแต่ละข้อย่อย (5 คะแนน)

```
int x = 5, y[5] = {1, 2, 3, 4, 5};  
float z = 2.0;  
int *ptr;
```

1.2.1 ให้ตัวแปร ptr อ้างอิงไปยังตัวแปร x

1.2.2 เพิ่มค่าตัวแปร x เป็น 2 เท่าจากค่าเดิม โดยเพิ่มค่าผ่านตัวแปร ptr

1.2.3 หลังจากทำคำสั่งจากข้อ 1.2.1 และ 1.2.2 ค่าของตัวแปร ptr คืออะไร

1.2.4 ให้ตัวแปร ptr อ้างอิงไปยังตัวแปร z

1.2.5 ให้ตัวแปร ptr อ้างอิงไปยังสมาชิกตัวแรกของอาร์เรย์ y

- 1.3 จงเขียนฟังก์ชัน printOct สำหรับพิมพ์ค่าจำนวนเต็ม (เลขฐานสิบ) เป็นเลขฐานแปด โดยกำหนดให้ฟังก์ชันต้นแบบเป็นดังนี้

```
void printOct (int num);
```

ตัวอย่างเช่น

```
printOct (8);           //แสดงผลลัพธ์ออกทางหน้าจอคือ 10
```

```
printOct (15);        //แสดงผลลัพธ์ออกทางหน้าจอคือ 17
```

\*\* กำหนดให้ค่าตัวแปร num เป็นค่า > 0 เสมอ (นักศึกษาไม่ต้องเขียนตรวจสอบค่า) และให้ใช้ฟังก์ชันเรียก

ตัวเองซ้ำ (recursive) ห้ามนักศึกษาใช้ while, for, do-while และห้ามใช้ฟังก์ชันใน math.h

คำแนะนำ: วิธีการแปลงเลขฐานสิบให้เป็นเลขฐานแปด สามารถแปลงโดยการนำ 8 มาหารเรื่อยๆ จนทำให้ผลหารเป็น 0 แล้วจึงเอาเศษที่ได้มาเขียนเรียงกัน โดยเริ่มจากเศษตัวสุดท้าย อย่างเช่น ต้องการแปลง 15 ให้เป็นเลขฐานแปด สามารถทำได้ดังนี้

$$15 = 17_8$$

(20 คะแนน)

	ผลหาร	เศษ
$15 \div 8$	1	7
$1 \div 8$	0	1

ตอนที่ 2 ข้อความและตัวอักษร

(30 คะแนน 30 นาที)

2. จงเขียนฟังก์ชัน `toUpperLower(word, mode)` ซึ่งเป็นฟังก์ชันที่แปลงตัวอักษรในข้อความ ที่ส่งผ่านทางพารามิเตอร์ ให้เป็นตัวพิมพ์เล็กหรือตัวพิมพ์ใหญ่ทั้งหมด โดยพิจารณาจากค่า `mode` ที่ส่งไปตอนเรียกใช้ฟังก์ชัน กำหนดฟังก์ชันต้นแบบ

```
char* toUpperLower( char str[ ], int mode);
```

ฟังก์ชันจะทำงานดังนี้

แปลงเป็นตัวอักษรตัวพิมพ์ใหญ่ทั้งหมด เมื่อ `mode` มีค่าเท่ากับ 0

แปลงเป็นตัวอักษรตัวพิมพ์เล็กทั้งหมด เมื่อ `mode` มีค่าเท่ากับ 1

ทั้งนี้ฟังก์ชันพิจารณาเฉพาะตัวอักษรภาษาอังกฤษเท่านั้น ตัวเลขและสัญลักษณ์อื่นๆให้คงตามเดิมไว้

ตัวอย่างเช่น

main()	ผลการรัน
<pre>char a[]="Hello World 1234"; printf("%s\n", a); printf("%s\n", toUpperLower(a,0)); printf("%s\n", toUpperLower(a,1));</pre>	<pre>Hello World 1234 HELLO WORLD 1234 hello world 1234</pre>

### ตอนที่ 3 Pointer (60 mins)

1. จากส่วนของโปรแกรมต่อไปนี้ จงเขียนส่วนของโค้ดใหม่ ตั้งแต่บรรทัดที่ 3 เป็นต้นไป โดยใช้พอยเตอร์ begin, end และ \* แทนการอ้างข้อมูลแบบ str[ ] โดยที่ผลการทำงานยังคงเดิม (15 คะแนน)

```
char str[20] = "radar";
int i = 0, j = strlen(str)-1;
char *begin, *end;
while(str[i]==str[j] && i<j)
{ i++; j--; }
if(i>=j) printf("Palindrom");
```

คำตอบ

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. จงเติมโปรแกรมต่อไปนี้ให้สมบูรณ์ และ เขียนฟังก์ชัน setTime () ซึ่งรับข้อมูลเวลาจาก stdin และเก็บไว้ในตัวแปรชนิด Time โดยตัวแปรดังกล่าวจะคืนค่ากลับไปที่ฟังก์ชัน main () แบบ Call-by-reference (**return-type** ของฟังก์ชัน setTime () เป็น **void**) (15 คะแนน, 15 นาที)
- ตัวอย่างผลการทำงาน

```
Enter hour: 15
Enter minute: 10
Time = 15:10
```

```
#include<stdio.h>
typedef struct {
    int hour;
    int minute;
} Time;

.....// prototype setTime()

int main() {
    Time t;

    .....// Call function setTiime
```

```

    printf("Time = %d:%d\n", t.hour, t.minute);
    return 0;
}
// Function definition setTime()

```

3. จงเขียนโปรแกรมในการเก็บค่าคะแนนของนักศึกษาในรายวิชาหนึ่ง ซึ่งมีนักศึกษาทั้งหมด 10 section แต่ละ section มีจำนวนนักศึกษาไม่เท่ากัน โดยให้รับค่าจำนวนนักศึกษาแต่ละ section จากผู้ใช้ สมมติว่าผู้ใช้ใส่ค่า > 0 เสมอ จากนั้นให้แสดงค่าคะแนนเฉลี่ยของแต่ละ section ว่ามีค่าเท่าไร โดยมีข้อกำหนดดังนี้
- ให้แต่ละ section เก็บคะแนนสุดท้ายเป็น -1 เพิ่มจากจำนวนนักศึกษาที่มีอยู่ เช่น section 1 มีน.ศ 32 คน ก็ให้เก็บคะแนน 32 ค่า และเก็บค่าที่ 33 เป็น -1 อัตโนมัติ
  - โปรแกรมจะต้องจองและใช้หน่วยความจำ ในการเก็บค่าคะแนนของนักศึกษาแต่ละ section เท่ากับจำนวนข้อมูลที่ต้องการเก็บจริง เช่น section 1 มีน.ศ. 32 คน ให้ใช้พื้นที่เก็บตัวเลข 33 ค่า (รวม -1)
  - `void *malloc(int memory_size);` ฟังก์ชันในการจองหน่วยความจำ สามารถเรียกใช้งานได้เลย

ตัวอย่างผลการทำงาน

(30 คะแนน, 30 นาที)

```

Section 1
Enter number of stds: 32
  Enter score of std 1: 70.5
  Enter score of std 2: 65.0
  :
  Enter score of std 32: 35.8
Section 2
Enter number of stds: 58
  Enter score of std 1: 70.5
  Enter score of std 2: 65.0
  :
  Enter score of std 58: 50.8
...
// รับค่าคะแนนต่อเนื่องไปเรื่อยๆ จนถึง 32
// รับค่าคะแนนต่อเนื่องไปเรื่อยๆ จนถึง 58
// รับค่าคะแนนใน section 3-9
Section 10
Enter number of stds: 45
  Enter score of std 1: 85.0
  :
  Enter score of std 45: 50.8

Average scores
Section 1 = 57.1
Section 2 = 60.5
...
// แสดงค่าคะแนนใน section 3-9
Section 10 = 51.6

```



## ตอนที่ 4 File (50 mins)

### 1. การประมวลผลไฟล์แบบข้อความ (25 คะแนน, 25 นาที)

ในปัจจุบันนี้ บ้านเมืองมีปัญหาทางการเมืองเป็นอย่างมาก โดยเฉพาะอย่างยิ่ง เรื่อง พรบ. นิรโทษกรรม ฉบับปี 2556 (Amnesty Bill 2013) ที่มีความพยายามที่จะยกเลิกความผิดทางการเมืองต่าง ๆ และมีการ แอบแก้ไข เพื่อยกเลิกคดี จากการทุจริต คอร์รัปชันฯ ต่าง ๆ จากเหตุการณ์นี้เอง ทำให้อาจารย์มหาวิทยาลัย ชื่อตั้งท่านหนึ่ง อยากได้ข้อมูลเฉพาะนักการเมืองที่มีระดับการทุจริตสูงสุด และ ต่ำสุด ในรูปแบบไฟล์ สำเร็จรูป เขาไว้เพื่อเป็นข้อมูลประกอบการตัดสินใจในการเลือกตั้งในอนาคต แต่ท่านไม่อยากจะเสียเวลา ค้นข้อมูลจากไฟล์ที่มีข้อมูลต่าง ๆ ทั้งหมดด้วยตนเอง และ อยากได้โปรแกรมสำเร็จรูปที่สามารถค้นหา ข้อมูลนี้ได้อย่างรวดเร็ว และสามารถนำไฟล์ข้อมูลนี้ ไปใช้งานอื่น ๆ ต่อไป โดยอาจารย์ได้เตรียมไฟล์ข้อมูล (politician.txt) สำหรับให้นำเข้าในโปรแกรม ให้นักศึกษาแล้วดังนี้

#### ตัวอย่าง Input ไฟล์ politician.txt

```
10
Yingluck 70
Apisit 85
Taksin 93
Sutep 35
Chuwit 38
Korn 53
Samak 30
Chalerm 87
Kittirat 20
Sirichoke 40
```

#### ข้อกำหนด

- บรรทัดแรกของไฟล์ จะเก็บจำนวนรายชื่อของนักการเมืองทั้งหมด ที่ไม่ซ้ำกัน
- บรรทัดต่อมาจะเก็บ ชื่อนักการเมือง (มีความยาวไม่เกิน 20 ตัวอักษร) และ ระดับคะแนนความ ทุจริต (มีค่าอยู่ระหว่าง 0-100)
- ข้อมูลเหล่านี้ เป็นเพียงข้อมูลที่เหมาะสมขึ้นเพื่อการเรียนรู้เท่านั้น

นักศึกษาจะต้องช่วยอาจารย์ท่านนี้ เขียนโปรแกรมเพื่ออ่านข้อมูลจากไฟล์ politician.txt แล้ว แสดงข้อมูลใน ไฟล์นั้นทั้งหมด ออกทางหน้าจอดังตัวอย่างข้างล่างนี้ และกำหนดให้จัดการกับไฟล์ในรูปแบบของ text mode

#### ตัวอย่าง Output ที่แสดงออกทางหน้าจอ

```
Name = Yingluck, Level = 70
Name = Apisit, Level = 85
Name = Taksin, Level = 93
Name = Sutep, Level = 35
Name = Chuwit, Level = 38
Name = Korn, Level = 53
Name = Samak, Level = 30
Name = Chalerm, Level = 87
Name = Kittirat, Level = 20
Name = Sirichoke, Level = 40
```





## 2. การประมวลผลไฟล์แบบไบนารี

(25 คะแนน, 25 นาที)

จงเขียนโปรแกรมที่ใช้ในการบวกตัวเลขจำนวนเชิงซ้อน (Complex number) แล้วให้นำผลลัพธ์ที่ได้จากการบวก เขียนลงในไฟล์ที่ชื่อว่า Complex.dat จากนั้น ให้โปรแกรม อ่านข้อมูลจากไฟล์ Complex.dat และแสดงผลออกทางหน้าจอ ดังตัวอย่างข้างล่าง

ตัวอย่าง Output ที่ได้จากการรันโปรแกรม ที่แสดงออกทางหน้าจอ

```
Complex number c3 = 4 + 12i
```

กำหนดชนิดของตัวแปรเชิงซ้อน ดังต่อไปนี้

```
typedef struct {
    int real;
    int imag;
}Complex ;
```

กำหนดให้โปรแกรมต้องมีรายละเอียดการทำงานของฟังก์ชันดังต่อไปนี้

- `c1 = create_complex(2,3);`  
ฟังก์ชัน `create_complex()` ใช้ในการสร้างตัวเลขจำนวนเชิงซ้อน `c1` ( $c1 = 2 + 3i$ )
- `c3 = add_complex(c1,c2);`  
ฟังก์ชัน `add_complex()` ใช้ในการบวกตัวเลขจำนวนเชิงซ้อน `c1` และ `c2` ส่งค่ากลับเป็นผลบวกของจำนวนเชิงซ้อนเก็บใน `c3`
- `print_complex(c3);`  
ฟังก์ชัน `print_complex()` ใช้ในแสดงผลค่า `c3` (ตัวอย่าง output) ออกทางหน้าจอ ฟังก์ชันนี้ที่ไม่มีการส่งค่ากลับ
- `write_complex_binary_file(c3);`



```
int main() {
    Complex c1, c2, c3;
    c1 = create_complex(2,3);
    c2 = create_complex(2,9);
    c3 = add_complex(c1,c2);
    write_complex_binary_file(c3);
    read_complex_binary_file(c3);
    return 0;
}
```

ADT

(10 คะแนน, 10 นาที)

1. จงอธิบาย Abstract Data Type (ADT) พร้อมวาดรูปตัวอย่าง การใช้งานในด้านของผู้ผลิต (Manufacturer) และ ผู้ใช้ (Client) ประกอบ : (5 คะแนน, 5 นาที)

Ans:

---

---

---

---

2. จงบอกประโยชน์ของการทำ Abstract Data Type (ADT) ในด้านของผู้ผลิต (Manufacturer) และ ผู้ใช้ (Client) (5 คะแนน, 5 นาที)

Ans:

---

---

---

---

---

---

---

---