



สอบกลางภาค: ภาคการศึกษาที่ 1

ปีการศึกษา: 2557

วันสอบ: 14 ตุลาคม 2557

เวลาสอบ: 13.30 – 16.30 น.

ห้องสอบ: (01)R200, (02)A 203, (03)S101, (04)S817, (05)R201, (06)S102, (07) A201, (08)A401,
(09)A400, (10)Robot, (241-101)S104,

ผู้สอน: อ.มัลลิกา อ.สกุณา อ.อารีย์ อ.อนันท์ อ.เสกสรรค์ อ.นิคม ภาควิชาวิศวกรรมคอมพิวเตอร์

รหัสและชื่อวิชา: 242-101, 241-101 Introduction to Computer Programming

แนะนำการเขียนโปรแกรมคอมพิวเตอร์

ทุจริตในการสอบมีโทษขั้นต่ำคือ ปรับตกในรายวิชาที่ทุจริตและพักการเรียน 1 ภาคการศึกษา

คำสั่ง: อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

อนุญาต: เครื่องเขียนต่างๆ เช่น ปากกา หรือดินสอ เข้าห้องสอบ

ไม่อนุญาต: หนังสือ หรือเครื่องคิดเลขเข้าห้องสอบ และเอกสารใดๆ เข้าและออกห้องสอบ

เวลา: 3 ชั่วโมง (180 นาที)

คำแนะนำ

- ข้อสอบมี 14 หน้า (รวมหน้าปก) แบ่งออกเป็น 4 ตอน คะแนนรวม 80 คะแนน (คิดเป็นคะแนนเก็บ 40%)
- เขียนคำตอบในข้อสอบ คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด
- อ่านคำสั่งในแต่ละข้อให้เข้าใจก่อนลงมือทำ
- เวลาที่ใช้เวลาทำตอนให้เหมาะสม ตามคำแนะนำ
- หากข้อใดเขียนคำตอบไม่พอ ให้เขียนเพิ่มเติมด้านหลังของหน้านั้นเท่านั้น

	1	2	3	4	รวม
ตอน	(20) 10%	(16) 8%	(24) 12%	(20) 10%	(80) 40%
คะแนน					

นักศึกษากรรรับทราบ ลงชื่อ

ตอนที่ 1 (20 คะแนน, 10%, 30 นาที)

ระบบคอมพิวเตอร์เบื้องต้น

1. ในยุคสมัยที่มีคอมพิวเตอร์ มนุษย์ได้ประโยชน์อะไรบ้างจากการนำคอมพิวเตอร์มาใช้งาน ให้นักศึกษาเขียนมา 2 คำตอบที่แตกต่างกันอย่างชัดเจน (1 คะแนน)
2. ในยุคของข้อมูล Information, จงบอกชื่อโปรแกรมประยุกต์บนคอมพิวเตอร์ที่เกี่ยวกับการจัดการหรือการสื่อสารข้อมูลมา 2 โปรแกรม (1 คะแนน)
3. จงบอกความแตกต่างของ main memory และ secondary storage และจงยกตัวอย่าง (1 คะแนน)
4. เพื่อการติดต่อกับคอมพิวเตอร์ มนุษย์จะต้องใช้ส่วน input และ output unit จงยกตัวอย่าง input และ output unit มาอย่างละ 3 ตัวอย่าง (1 คะแนน)

5. ข้อมูล 1 KB เท่ากับข้อมูลจำนวนเท่าไรในหน่วยไบต์

(1 คะแนน)

6. DVD-ROM สามารถเก็บข้อมูลได้ 4.7 GB หมายถึง DVD-ROM สามารถเก็บข้อมูลได้เท่าไรในหน่วยไบต์
หมายเหตุ เขียนคำตอบให้อยู่ในรูปการคูณได้

(1 คะแนน)

7. ข้อมูลจำนวน 2048 KB เท่ากับเท่าไรข้อมูลจำนวนเท่าไรในหน่วย MB

(1 คะแนน)

8. จงบอกหน้าที่ของ Main board หรือ Mother board

(1 คะแนน)

9. ถ้าหากคอมพิวเตอร์ขาดส่วน CPU จะเกิดอะไรขึ้น ให้อธิบายว่าเพราะเหตุใด

(1 คะแนน)

10. จงบอกความแตกต่างของกลุ่มซอฟต์แวร์ประเภท MS-Windows, Linux Ubuntu กับกลุ่มของซอฟต์แวร์
เช่น Paint, Photoshop, MS-Word, MS-Excel เป็นต้น

(1 คะแนน)

11. อะไรคือการแปลเป็นภาษาเครื่อง จงยกตัวอย่างโปรแกรมที่สามารถแปลโปรแกรมเป็นภาษาเครื่อง (1 คะแนน)

12. กรณีศึกษา ในระหว่างที่นักศึกษาใช้โปรแกรมติดต่อสื่อสารทั้งทางภาพและเสียงเช่น Skype จากหอพัก เพื่อพูดคุยกับแม่ที่บ้านโดยผ่านทางเครื่องคอมพิวเตอร์ จงอธิบายว่า (3 คะแนน)

12.1 โปรแกรมที่ใช้มีส่วนประกอบ input อะไรบ้าง (จงเขียนมาให้หมดสำหรับกรณีข้างต้น)

12.2 โปรแกรมที่ใช้มีส่วนประกอบ output อะไรบ้าง (จงเขียนมาให้หมดสำหรับกรณีข้างต้น)

12.3 เราจะเรียกโปรแกรม skype ว่าเป็นซอฟต์แวร์ประเภทระบบ (system software) ใช่หรือไม่ ถ้าไม่ใช่ จงระบุประเภทที่ถูกต้อง

13. นักศึกษาได้เขียนโปรแกรมหาพื้นที่วงกลมโดยมีสูตร $A = \pi r^2$ โดยมีขั้นตอนดังข้างล่างนี้
 ให้อธิบายว่าการทำงานในแต่ละขั้นตอนเกิดขึ้นที่ส่วนประกอบใดของคอมพิวเตอร์
 [ตัวอย่างเช่น ขั้นตอนที่ 3 การกำหนดตัวแปร A จะเกิดขึ้นใน RAM เป็นต้น] (6 คะแนน)

ขั้นตอนการทำงาน	ส่วนของคอมพิวเตอร์ที่เกี่ยวข้อง
1. สร้างไฟล์โปรแกรม area.c (ซึ่งมีการทำงานขั้นตอน 2-7)	
2. กำหนดค่าคงที่ π	
3. กำหนดตัวแปร A	หน่วยความจำ RAM
4. กำหนดตัวแปร R	
5. รับค่าตัวเลขมาจากผู้ใช้ เก็บไว้ใส่ตัวแปร R	
6. คำนวณพื้นที่ (A)	
7 แสดงผลลัพธ์บนจอคอมพิวเตอร์	

/****** จบตอนที่ 1 *****/

ตอนที่ 2 (16 คะแนน, 8%, 30 นาที)

พื้นฐานภาษาซี

1. จงตอบคำถามต่อไปนี้ (ข้อละ 1 คะแนน)

1.1 จาก source code โปรแกรม program.c จะต้องผ่านขั้นตอนใดบ้าง เพื่อให้ได้ program.exe

.....

1.2 ฟังก์ชันที่จำเป็นที่สุดของโปรแกรมภาษา C คือฟังก์ชันใด และ เพราะเหตุใด

.....

1.3 หลักการในการตั้งชื่อตัวแปรภาษา C มีอะไรบ้าง ให้ออกมาอย่างน้อย 2 ข้อ

.....

.....

2 จงประกาศตัวแปรที่จำเป็นสำหรับเก็บข้อมูลในโจทย์ปัญหาต่อไปนี้ (2.1-2.2 ข้อละ 1 คะแนน)

2.1 โปรแกรมรับค่าจำนวนเต็ม 3 จำนวนจากผู้ใช้ แล้วหาค่ามากที่สุดและน้อยที่สุดจากทั้ง 3 จำนวน

.....

2.2 โปรแกรมตรวจสอบว่าเป็นสี่เหลี่ยมผืนผ้าหรือไม่ โดยรับด้านกว้างและด้านยาวของสี่เหลี่ยมจากผู้ใช้ ถ้าเป็นสี่เหลี่ยมผืนผ้าให้แสดงข้อความ "Rectangle" ถ้าไม่ใช่ให้พิมพ์ข้อความ "Not rectangle"

.....

2.3 โปรแกรมตรวจนับจำนวนกล่องในตู้สินค้าซึ่งบรรจุได้มากที่สุด 100 กล่อง โดยเริ่มต้นให้ตู้ว่างเปล่า การนับกล่อง พนักงานจะกดเครื่องหมาย + ทุกครั้งที่นำกล่องใส่ในตู้ และกดเครื่องหมาย - เมื่อนำกล่องออกจากตู้ เมื่อสิ้นสุดการขนย้ายกล่องเข้าหรือออกจากตู้ พนักงานจะกด q เป็นการจบโปรแกรม เมื่อจบการขนย้าย โปรแกรมจะแสดงจำนวนกล่องทั้งหมดที่มีอยู่ในตู้ให้พนักงานทราบ และเมื่อตู้เต็มก่อนที่พนักงานจะกด q โปรแกรมก็จะหยุดอัตโนมัติ พร้อมแสดงข้อความ FULL (2 คะแนน)

.....

.....

3 กำหนดให้ตัวแปร sum กับ num มีชนิดข้อมูลเป็น int ทั้งคู่ โดยมีค่า 55 และ 10 ตามลำดับ และตัวแปร avg มีชนิดข้อมูลเป็นแบบ float ถ้าเขียนคำสั่ง $avg = sum/num$; จะได้ค่าผลลัพธ์ avg มีค่าเท่าไร และจะแก้ไขให้ข้อมูล avg มีค่าที่ถูกต้องตามความเป็นจริงได้อย่างไร (2 คะแนน)

.....

.....

- 4 จงหาค่าตัวแปรหลังจากจบคำสั่งในแต่ละข้อ โดยให้ถือว่า คำสั่งแต่ละข้อ ไม่ต่อเนื่อง กัน
กำหนดให้ใช้ตัวแปรแต่ละตัวมีค่าเริ่มต้นดังนี้ `int a=5, b=3, c=2;` (ข้อละ 1 คะแนน)

คำสั่ง	ค่าตัวแปร a	ค่าตัวแปร b	ค่าตัวแปร c
<code>a = ++b + c;</code>			
<code>c = a + b * 5 / 3;</code>			
<code>b += a + c -- ;</code>			
<code>a = !b (a && c);</code>			

- 5 จงเขียนคำสั่งในการรับค่าหรือแสดงค่าตัวแปร ตามข้อกำหนดในแต่ละข้อต่อไปนี้ (ข้อละ 1 คะแนน)

5.1 เขียนคำสั่งในการรับค่า วัน เดือน ปี โดยผู้ใช้กรอกข้อมูลในรูปแบบ dd/mm/yy แล้วจึงกด Enter

เช่น 14/10/57

`int dd, mm, yy;`

.....

5.2 เขียนคำสั่งในการรับค่าเพศ อายุ และ น้ำหนัก จากผู้ใช้

`char sex; int age, float weight;`

.....

5.3 เขียนคำสั่งแสดงค่าข้อมูลตัวแปรในข้อ 5.2 ออกทางจอภาพ โดยมีรูปแบบตามตัวอย่างผลลัพธ์ดังนี้

`sex = M`

`age = 19, weight = 46.5`

.....

.....

/****** จบตอนที่ 2 *****/

ตอนที่ 3 (24 คะแนน, 12%, 60 นาที)

โครงสร้างควบคุม

1. จงเขียน ผลลัพธ์ของส่วนของโปรแกรม ต่อไปนี้

(10 คะแนน)

ข้อ	นิพจน์	ผลลัพธ์
1.1	<pre>int a = -1, b = 1; if((a++) && 5*(--b)+1) printf("TRUE"); else printf("FALSE");</pre>	
1.2	<pre>int i,j; for(j=0,i=-5; i<j; i++) printf("O"); printf("H");</pre>	
1.3	<pre>int x=0; x += 1; while (x<5) { x++; switch(x) { case 0: printf("O"); break; case 2: printf("M"); break; default: printf("G"); } }</pre>	
1.4	<pre>int i,j; for (i=1; i<4; i++) { for(j=1; j<=i; j++) printf("%d", i+j); printf("\n"); i++; }</pre>	
1.5	<pre>int i = 1; do { if(i<=4) printf("Yeah"); else printf("!"); if(i>5) break; i+=5; } while(i<=10);</pre>	

2. จงเขียนโปรแกรมรับจำนวนเต็มบวก a, d และ n จากผู้ใช้ 3 จำนวน แล้วทำการหาค่าผลรวมของอนุกรมต่อไปนี้

$a, a+d, a+2d, \dots, a+i*d, \dots, a+(n-1)d$ กำหนดให้ $i = 0, \dots, n$ และ d คือผลต่างของสองพจน์ที่ติดกัน ค่าของ a, d และ n ให้โปรแกรมรับจากผู้ใช้

ผลลัพธ์ของโปรแกรมคือผลบวกของอนุกรมนี้ ตั้งแต่พจน์แรกจนถึงพจน์ที่ n

เช่น ผู้ใช้ป้อนค่า 1, 2 และ 3 ($a=1, d=2, n=3$) ผลลัพธ์คือผลบวกตั้งแต่พจน์ 1 ถึง 3 ของอนุกรม 1, 1+2, 1+4

จะได้คำตอบคือ 9 ($1+3+5 = 9$) (4 คะแนน)

ตัวอย่างการรันโปรแกรม (*ตัวหนาเอียง* คือ ค่าที่ผู้ใช้ป้อนผ่านคีย์บอร์ด)

ตัวอย่างที่ 1

```
Enter a d and n: 1 2 3  
Summation from series is 9
```

ตัวอย่างที่ 2

```
Enter a d and n: 2 3 5  
Summation from series is 40
```

3. จงเขียนโปรแกรมเกมทายเลข โดยให้ผู้ใช้สองคนเล่น เริ่มต้นจากผู้เล่นคนแรกกำหนดตัวเลขที่จะให้ทาย ผู้เล่นคนที่สองเป็นคนทาย(สมมุติว่าผู้เล่นคนที่สองไม่สามารถตัวเลขที่ผู้เล่นคนแรกกำหนดไว้) โปรแกรมจะวนรับค่าจากผู้เล่นคนที่สองจนกว่าจะทายถูก โดยแต่ละครั้งที่ผู้เล่นคนที่สองทาย หากทายไม่ถูกต้องจะมีคำใบ้บอกให้ว่าค่าที่ถูกต้องนั้น มีค่ามากหรือน้อยกว่าค่าที่ทาย โดยเป็น 4 ระดับ ดังนี้คือ

- | | |
|---|--|
| - Much bigger!!! (คำตอบมีค่ามากกว่านี้มากมาย) | ถ้า 10 เท่าของค่าที่ทาย ยังน้อยกว่าค่าที่ถูกต้อง |
| - Bigger! (คำตอบมีค่ามากกว่านี้) | ถ้าค่าที่ทาย น้อยค่าที่ถูกต้อง |
| - Much smaller!!! (คำตอบมีค่าน้อยกว่านี้มากมาย) | ถ้าค่าที่ทายมากกว่า 10 เท่าของค่าที่ถูกต้อง |
| - Smaller! (คำตอบมีค่าน้อยกว่านี้) | ถ้าค่าที่ทาย มากกว่าค่าที่ถูกต้อง |

และแสดงความยินดีเมื่อตอบถูกต้อง และจบโปรแกรม

(6 คะแนน)

ตัวอย่างการรันโปรแกรม (ตัวหนาเอียง คือ คำที่ผู้ใช้ป้อนผ่านคีย์บอร์ด)

```
First Player, please enter your number: 60
Let's start!
Second Player, please enter your number: 1000
  Much smaller!!!
Second Player, please enter your number: 100
  Smaller!
Second Player, please enter your number: 5
  Much Bigger!!!
Second Player, please enter your number: 50
  Bigger!
Second Player, please enter your number: 60
  Congratulations!
```

โปรแกรมข้อ 3 (ต่อ) (ถ้ามี)

4. จงเขียนโปรแกรมรับค่าจำนวนเต็ม 10 จำนวน แล้วสรุปว่า เป็นเลขคู่ (even number) กี่จำนวน และ เลขคี่ (odd number) กี่จำนวน (4 คะแนน)

ตัวอย่างการรันโปรแกรม (ตัวหนาเอียง คือ ค่าที่ผู้ใช้ป้อนผ่านคีย์บอร์ด)

Enter 10 integers: **60 12 -3 9 5 17 1103 8 9 10**
There are 4 even numbers and 6 odd numbers.

/****** จบตอนที่ 3 *****/

ตอนที่ 4 (20 คะแนน, 10%, 50 นาที)

อาร์เรย์

1. จงตอบคำถามต่อไปนี้ให้ถูกต้อง (5 คะแนน)

1.1 ประกาศตัวแปรเพื่อเก็บข้อมูลอายุของนักเรียน 20 คน

.....

1.2 กำหนดให้ `int x[3] = {0,1,3}, 5, 7, 9`; ตัวแปร `x` ต้องใช้หน่วยความจำขนาดกี่ไบต์

.....

1.3 กำหนดให้ `float y[3][2]`; ตัวแปร `y` สามารถเก็บจำนวนจริงได้กี่จำนวน

.....

1.4 กำหนดให้ `int z[2][3] = {{4},{5,1},2}`; ตัวแปร `z` มีค่าเท่ากับ 0 ณ ตำแหน่งอาร์เรย์ใดบ้าง

.....

1.5 กำหนดให้ `char name[][14] = {"Manee", "Mana", "Chujai"}`; ให้ใช้คำสั่ง `printf` เพื่อแสดงค่า Mana ออกทางจอภาพ

.....

2. ให้เขียนส่วนหนึ่งของโปรแกรมที่ทำงานดังต่อไปนี้ (5 คะแนน)

กำหนดตัวแปรสำหรับชื่อและคะแนนของนักศึกษา 5 คน ดังนี้

```
char name[5][20]; int score[5];
```

2.1 รับชื่อและคะแนน ของนักศึกษาทั้ง 5 คน จากแป้นพิมพ์ (1 คะแนน)

.....

.....

.....

.....

2.2 เขียนส่วนของโปรแกรมเพื่อคำนวณหาค่าคะแนนเฉลี่ย (2 คะแนน)

.....

.....

.....

2.3 เขียนส่วนของโปรแกรมเพื่อแสดงชื่อของนักศึกษาที่มีคะแนนมากกว่าคะแนนเฉลี่ย (2 คะแนน)

.....

.....

.....

3. จงเขียนโปรแกรมระบบจัดการการจอดรถ Parking Lot Management System ให้สมบูรณ์ โดยมีการทำงานของโปรแกรมเป็นดังนี้ (10 คะแนน)

- ระบบสามารถแสดงตำแหน่งที่มีการจอดรถ (X) และไม่มีการจอดรถได้ (-)
- ผู้ใช้สามารถกำหนดการเพิ่มและการลดตำแหน่งของการจอดรถได้
- ระบบสามารถแสดงสถานะของแต่ละแถวโดยแทนด้วยรหัส ดังนี้
 - รหัส E เมื่อที่จอดว่างทุกตำแหน่ง
 - รหัส A เมื่อมีที่ว่างเหลือบางตำแหน่ง
 - รหัส F เมื่อรถเต็มทุกตำแหน่ง

ตัวอย่างการทำงานของโปรแกรม

```
PARK AT : 1 1 + // 1 1 + หมายถึง มีรถเข้าจอดที่แถวที่ 1 คันที่ 1
R1 : X----- : A
R2 : -X----- : A
R3 : ----- : E
PARK AT : 2 2 - // 2 2 - หมายถึง มีรถออกจากแถวที่ 2 คันที่ 2
R1 : X----- : A
R2 : ----- : E
R3 : ----- : E
:
PARK AT : 3 3 +
R1 : X----- : A
R2 : XXXXXXXXXX : F
R3 : --X----- : A
```

จงเติมโปรแกรมต่อไปนี้ให้สมบูรณ์ (โดยไม่ต้องประกาศตัวแปรใดๆ เพิ่มเติม)

```
#include <stdio.h>
#define R 3 // ที่จอดรถมี 3 แถว
#define C 10 // แต่ละแถวสามารถจอดรถได้ 10 คัน
int main()
{
    int loop=1, i, j;
    int park[R][C] = {0}; // เก็บค่า 1 แสดงว่ามีรถจอด / ค่า 0 แสดงว่าไม่มีรถจอด
    int free[R] = {C, C, C}; // เก็บค่าจำนวนรถที่ว่างในแต่ละแถว
    int x, y; // เก็บค่าตำแหน่งการจอด ณ แถว x และ คันที่ y
    char sym; // เก็บค่า +/- แสดงถึงการจอด/ไม่จอด
    char code[R] = {'E', 'E', 'E'}; // เก็บค่ารหัสสถานะของแต่ละแถว
    do{ // แสดงค่าตำแหน่งการจอด/ไม่จอดรถ
        for(i=0; i<R; i++){
            printf("R%d : ", i+1);
            for(j=0; j<C; j++){
                if(park[i][j]) printf("X");
                else printf("-");
            }
            printf(" : %c", code[i]);
            printf("\n");
        }
    }
```

```
// รับค่าตำแหน่งที่จะจอด/ไม่จอดรถ
printf("PARK AT : ");
scanf("%d %d %c", &x, &y, &sym);

if(sym == '+') park[x-1][y-1] = 1;
else if(sym == '-') park[x-1][y-1] = 0;
else break;
```

// ให้เขียนส่วนของโปรแกรมเพื่อตรวจสอบจำนวนรถที่ว่างในแต่ละแถว (5 คะแนน)

.....
.....
.....
.....
.....
.....
.....

// ให้เขียนส่วนของโปรแกรมเพื่อกำหนดค่ารหัสของแต่ละแถว จากจำนวนรถที่ว่าง (5 คะแนน)

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

```
} while(loop);

system("PAUSE");
return 0;
}
```

***** จบตอนที่ 4 *****