



สอบปลายภาค: ภาคการศึกษาที่ 1

ปีการศึกษา: 2555

วันที่สอบ: 5 ตุลาคม 2555

เวลาสอบ: 13.30 – 16.30 น.

ห้องสอบ: S203(01), S103(02), S203(03), R200(04), S201(05), R200(06), S201(07),  
S101(08), A201(09), R201(10), A401(01)

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

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

แนะนำการเขียนโปรแกรมคอมพิวเตอร์

ทฤษฏีในการสอบมีโทษขั้นต่ำคือ ปรับตกในรายวิชาที่ทฤษฏีและพักการเรียน 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 3 ตอน คะแนนรวม 90 คะแนน (คิดเป็น 30%)
- ☐ เขียนคำตอบในข้อสอบ คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด
- ☐ อ่านคำสั่งในแต่ละข้อให้เข้าใจก่อนลงมือทำ
- ☐ ควรใช้เวลาทำตอนละ ไม่เกิน 1 ชั่วโมง (คะแนนแต่ละข้อ  $\times 2 =$  จำนวนนาทีที่ควรใช้)
- ☐ หากข้อใดเขียนคำตอบไม่พอ ให้เขียนเพิ่มเติมด้านหลังของหน้านั้นเท่านั้น

ตอน	1 (30)	2 (30)	3 (30)	รวม (90)
คะแนน				

นักศึกษากรรรับทราบ ลงชื่อ .....

**ตอนที่ 1 (30 คะแนน)**

อาร์เรย์ (Array) และสตริง (String)

1. จงตอบคำถามต่อไปนี้ (6 คะแนน)

1.1 จากการประกาศตัวแปร `int A[3][5] = {10,5,67,8,9,45,2,0,4,6,88,99,1,5};`

- `A[2][1]` = .....

- ตัวแปร `A[2][1]` ต้องใช้หน่วยความจำขนาดเท่าไร.....

1.2 จากตัวแปร `float B[ ][4] = {{2.3},{5.2,3.4,5.0}};` ให้เขียนค่าของตัวแปร B ทั้งหมด

เช่น `B[0][0] = ?` , `B[0][1] = ?` ... จนกระทั่งถึงตัวแปร B ตัวสุดท้าย

.....

.....

.....

.....

1.3 ในกรณีที่เป็นการใช้ตัวแปรอาร์เรย์ 3 มิติ จะต้องเขียนการรับค่าจากคีย์บอร์ดอย่างไร

โดยกำหนดตัวแปรดังนี้ `float volume[2][3][4];`

.....

.....

.....

.....

.....

.....

.....

1.4 กำหนดตัวแปร `char name[3][10]={ "abcdef", "ghijklmn"};`

จงหาค่าต่อไปนี้

`name[0][3]` = \_\_\_\_\_

`name[0][6]` = \_\_\_\_\_

`name[1][4]` = \_\_\_\_\_

`name[1][8]` = \_\_\_\_\_

.....

## 2. จงเขียนส่วนของโปรแกรมต่อไปนี้ (เฉพาะส่วนที่โจทย์กำหนดให้ทำ)

(8 คะแนน)

หมายเหตุ สามารถเรียกใช้ฟังก์ชันจาก String Library ได้

คำถาม	คำตอบ
<p>2.1 กำหนดให้ตัวแปรชนิดจำนวนเต็มเป็นอาร์เรย์ 2 มิติ ได้แก่ ตัวแปร A[2][2] และ B[2][2]</p> <p>* จงแสดงค่าผลบวกของสองตัวแปร (ผลบวกของคู่อันดับที่ตรงกันของ A กับ B)</p>	
<p>2.2 กำหนดตัวแปรชนิดจำนวนเต็มชื่อ</p> <p>scores[3][7] = { {84, 71, 96, 65, 79}, {90, 55, 83, 68, 96}, {61, 77, 82, 94, 59} };</p> <p>* จงคำนวณผลรวมในแต่ละแถวและแสดงผลลัพธ์</p>	
<p>2.3 กำหนดให้ตัวแปรชื่อ char x[10] = "ABCDE"; และ char y[20] = "A1B2C3";</p> <p>* ให้แสดงค่าความยาวของสตริง x และ y ออกทางจอภาพ แล้วกำหนดค่าใหม่กับตัวแปร y เป็นให้เหมือนกับตัวแปร x</p>	
<p>2.4 จากการประกาศตัวแปร char Rose[10] = "WATER"; และ char Orchid[15] = "soil";</p> <p>* จงเขียนคำสั่งกำหนดค่าให้ตัวแปร Orchid มีค่าเท่ากับ "soilwater" แล้วแสดงค่าตัวแปรทั้งสองตัวออกทางจอภาพ</p>	

3. จงเติมโปรแกรมให้สมบูรณ์ (5 คะแนน)

```
#include <stdio.h>
#define NUM_STUDENTS _____
#define NUM_TESTS _____

int get_highest(int a[][NUM_TESTS], int row, int col);

int main()
{
    int grades[NUM_STUDENTS][NUM_TESTS] = { {85, 50, 75},
                                                {80, 91, 95} };

    int num_students = 2;
    int num_tests = 3;
    int high_test;

    high_test = _____

    printf("The highest score is %d.\n", high_test );
    getch();
    return 0;
}

int get_highest(int a[][NUM_TESTS], int row, int col)
/* Assumes that there is at least one element */
{
    int i, j;
    int highest = _____

    for( i = 0; _____; i++)

        for( j = 0; _____; j++)

            if ( _____ )

                _____

    return _____
}
```

4. จงเขียนโปรแกรมทรานสโพสเมตริกส์ ดังตัวอย่างผลการรันโปรแกรมด้านล่าง (8 คะแนน)

```

Input of matrix elements :
Input element [0][0] : 1
Input element [0][1] : 2
Input element [0][2] : 3
Input element [1][0] : 4
Input element [1][1] : 5
Input element [1][2] : 6
Input element [2][0] : 7
Input element [2][1] : 8
Input element [2][2] : 9

Matrix before transposing :
1 2 3
4 5 6
7 8 9

Matrix after transposing :
1 4 7
2 5 8
3 6 9
    
```

ตัวอักษรเอียงหนา  
คือค่าที่ผู้ใช้ป้อน  
ให้กับโปรแกรม

โค้ดโปรแกรมข้อ 4.

โค้ดโปรแกรมข้อ 4. (ต่อ)

/\*\*\*\*\*\* จบตอนที่ 1 \*\*\*\*\*/

ตอนที่ 2 (30 คะแนน)

โครงสร้างข้อมูล (Structure)

1. **จงนิยาม** Structure สำหรับเก็บรายละเอียดรถยนต์โดยให้ชื่อว่า car มีสมาชิก 4 ตัวคือ brand สำหรับเก็บยี่ห้อรถยนต์ (20 ตัวอักษร) model สำหรับเก็บชื่อรุ่นของรถ (30 ตัวอักษร) color สำหรับเก็บสีรถ (30 ตัวอักษร) และ year สำหรับเก็บปีค.ศ.ที่ผลิต หลังจากนิยามเรียบร้อยแล้ว ให้**ประกาศตัวแปร** mycar สำหรับเก็บรายละเอียดรถยนต์ (4 คะแนน)

.....

.....

.....

.....

.....

.....

2. จากข้อ 1 ให้เขียนส่วนของคำสั่งที่ใช้ในการ**เก็บค่าข้อมูล**ของสมาชิกทุกตัวของตัวแปร mycar (2 คะแนน)

.....

.....

.....

.....

3. **จงประกาศตัวแปร** carList สำหรับเก็บรายละเอียดรถยนต์จำนวน 30 คัน (2 คะแนน)

.....

4. จงเขียนคำสั่งในการกำหนดให้ปี ค.ศ. ของรถทุกคันใน carList มีค่าเท่ากับ 2011 (2 คะแนน)

.....

.....

.....

5. จากโครงสร้างข้อมูล car ตัวแปร mycar และ carList จะมี**ขนาด**กี่ไบต์ (2 คะแนน)

.....

6. จงเติมส่วนของโปรแกรมที่กำหนด โดยโปรแกรมมีตัวแปร point 2 ตัว ตัวหนึ่งกำหนดค่าเริ่มต้นตอนประกาศตัวแปร (โดยให้ตำแหน่งพิกัด (x,y) เป็น 10 และ 20 ตามลำดับและตั้งชื่อว่าจุด B1) อีกตัวหนึ่งกำหนดค่าจากการเรียกใช้งานฟังก์ชัน setPoint() จากนั้นทดสอบว่าทั้ง 2 จุดมีค่าตำแหน่งพิกัด (x,y) เหมือนกันหรือไม่ ถ้าตำแหน่งพิกัด (x,y) ตรงกันจะคืนค่าเป็น 1 ถ้าต่างกันจะคืนค่า 0 (12 คะแนน)

```
#include <stdio.h>
#include <string.h>

_____ { // นิยามโครงสร้างข้อมูลใหม่ให้มีชื่อว่า point

    int x,y;           // เก็บค่าตำแหน่งจุด (x,y)
    char pointName[20]; // เก็บชื่อจุดเช่น A1,B1
} point;

_____ setPoint(int a, int b, char name[]);

int main()
{
    point A,B = _____;
    A = setPoint(10,15,"A1");
    if (isSamePoint(A,B) == 1)
        printf("A and B are the same point.\n");
    else
        printf("A and B are not the same point.\n");
    return 0;
}

_____ setPoint(int a, int b, char name[]){

    _____

    _____

    _____

    _____

    _____

    _____

    _____

    _____

    _____

    _____

}

int isSamePoint ( _____ ){

    _____

    _____

    _____

    _____

    _____

    _____

    _____

    _____

    _____

    _____

}
```



7. จากโปรแกรมด้านล่าง จงตอบคำถามข้อ 7.1 – 7.3

```
#include <stdio.h>

typedef struct {
    int day,month,year;
} date;

void edit(date a);

int main() {
    date d1 = {1,10,2012};
    edit(d1);
    printf("%d/%d/%d\n",d1.day,d1.month,d1.year);
    return 0;
}

void edit(date a)
{
    a.year = a.year + 10;
}
```

7.1 จงแสดงผลลัพธ์ของโปรแกรม (1 คะแนน)

.....

7.2 จงปรับปรุงฟังก์ชัน edit() เพื่อให้ค่าของตัวแปร d1.year มีค่าเพิ่มขึ้นจากเดิม 20 ปี เมื่อเรียกใช้งาน edit(d1) (4 คะแนน)

.....

.....

.....

.....

.....

.....

.....

.....

7.3 เขียนวิธีการเรียกใช้งานฟังก์ชัน edit( ) ที่ปรับปรุงใหม่ ภายในฟังก์ชัน main (1 คะแนน)

.....

/\*\*\*\*\*\* จบตอนที่ 2 \*\*\*\*\*/

## ตอนที่ 3 (30 คะแนน)

## โปรแกรมประยุกต์

## Simple Sudoku เกมซูโดกุอย่างง่าย

ตารางขนาด 4x4 แต่ละช่อง เต็มด้วยค่าตัวเลข 1-4 โดยในแต่ละแถว และ แต่ละคอลัมน์ จะมีตัวเลขไม่ซ้ำกัน เช่น

1	2	3	4
3	4	1	2
2	1	4	3
4	3	2	1

3	4	2	1
4	2	1	3
2	1	3	4
1	3	4	2

การเล่นเกม Simple Sudoku จะให้ตาราง พร้อมตัวเลขและเว้นไว้ 1 ช่อง เพื่อให้หาคำตอบว่าตัวเลขที่หายไปคืออะไร และให้ตรวจสอบด้วยว่าตารางนั้นมีคุณสมบัติเป็นซูโดกุหรือไม่ เช่น

Table A

1	2	3	4
	4	1	2
2	1	4	3
4	3	2	1

Table B

3	4	1	2
4	2		3
2	1	3	4
1	3	4	1

จากตัวอย่าง Table A มีคุณสมบัติเป็น Sudoku เมื่อเติม ช่องว่างด้วยเลข 3

ส่วน Table B ไม่เป็นตาราง Sudoku เพราะแถวที่ 4 มีตัวเลขซ้ำกัน

จงเขียนโปรแกรมเล่นเกม Simple Sudoku โดยให้เขียนส่วนต่างๆของโปรแกรกดังต่อไปนี้

1. จงนิยามโครงสร้างข้อมูล(struct) สำหรับเก็บตาราง Sudoku ให้โครงสร้างข้อมูลมีชื่อว่า sudokuTable ซึ่งมีสมาชิก 2 ตัว ประกอบด้วย table เป็นอาร์เรย์ 2 มิติ ไว้เก็บค่าตัวเลขในตาราง, และ isSudoku เก็บค่า 1 หรือ 0 (จริงหรือเท็จ) เมื่อได้ทดสอบแล้วว่าเป็นตาราง Sudoku หรือไม่ (5 คะแนน)

```
typedef struct {
```

2. จงเขียนนิยามฟังก์ชัน inputSudokuTable ที่สอบถามให้ผู้ใช้ใส่ค่าตัวเลข 1-4 ในตาราง สำหรับช่องที่เว้นว่างให้ใส่ค่า 0 แล้วเก็บไว้ในตัวแปรชนิด sudokuTable จากนั้นฟังก์ชัน return ข้อมูลนั้น (6 คะแนน)

ตัวอย่างการทำงานของฟังก์ชัน (ตัวหนาเอียงคือค่าที่ผู้ใช้ใส่)

Input elements of 4x4 Sudoku table

row 1 : **1 2 3 4**

row 2 : **0 4 1 2**

row 3 : **2 1 4 3**

row 4 : **4 3 2 1**

นิยามของฟังก์ชัน

```
sudokuTable inputSudokuTable(void)
{
```

3. จงเขียนนิยามฟังก์ชัน showSudoku ที่แสดงตาราง Sudoku (4 คะแนน)

ตัวอย่างการทำงานของฟังก์ชัน (ตัวอย่างข้อมูลตาราง Sudoku ที่ผู้ใช้ใส่ ในข้อ 2.)

1 2 3 4

0 4 1 2

2 1 4 3

4 3 2 1

นิยามของฟังก์ชัน

```
void showSudoku(sudokuTable t)
{
```

4. จงเขียนนิยามฟังก์ชัน checkSudoku ที่รับข้อมูลชนิด sudokuTable ผ่านพารามิเตอร์ และตรวจสอบว่ามีคุณสมบัติเป็นตาราง Sudoku หรือไม่ ให้คืนค่า 1 ถ้าเป็นตาราง Sudoku และ คืนค่า 0 ถ้าไม่ใช่  
วิธีการตรวจสอบ ให้ตรวจสอบตัวเลขในแต่ละแถวและแต่ละหลัก ว่ามีค่าระหว่าง 1-4 และมีตัวเลขไม่ซ้ำกัน ให้ยกเว้นไม่ต้องตรวจสอบ แถวหรือหลัก ที่มีค่า 0 อยู่ (เพราะเป็นช่องว่างที่ต้องเติม) (8 คะแนน)

นิยามของฟังก์ชัน

```
int checkSudoku(sudokuTable t)
{
```

5. จงเขียนนิยามฟังก์ชัน solveSudoku ที่หาคำตอบของตาราง Sudoku ที่มีช่องว่างหนึ่งช่อง ฟังก์ชันจะ return ค่าตัวเลขที่จะต้องเติมในตาราง แทนที่ค่า 0 หากไม่สามารถหาค่าที่เหมาะสมได้ ให้ return ค่า -1 เช่น

Table A

1	2	3	4
0	4	1	2
2	1	4	3
4	3	2	1

Table C

3	4	2	0
4	2	1	3
2	1	3	4
1	3	4	1

Table A ค่า 0 อยู่ในแถวที่ 2 คอลัมน์ที่ 1 ในแถวที่ 2 ยังขาดเลข 3 และ จากคอลัมน์ที่ 1 ยังขาดเลข 3 ซึ่งตรงกันพอดี ได้คำตอบของตาราง Sudoku คือ 3 ฟังก์ชันจะ return ค่า 3

ส่วน Table C ค่า 0 อยู่ในแถวที่ 1 คอลัมน์ที่ 4 ในแถวที่ 1 ยังขาดเลข 1 แต่ในคอลัมน์ที่ 4 ยังขาดเลข 2 ซึ่งแตกต่างกัน ดังนั้นจึงไม่มีคำตอบที่เหมาะสมสำหรับตารางนี้ ฟังก์ชันจะ return ค่า -1 (7 คะแนน)

นิยามของฟังก์ชัน

```
int solveSudoku(sudokuTable t)
{
```

ตัวอย่างโปรแกรมหลัก ในการเล่นเกม Simple Sudoku ที่มีการเรียกใช้ฟังก์ชันต่างๆ ที่สร้างขึ้นในข้อ 1-5

```
int main()
{
    sudokuTable st;
    int r,c,x,y,z;
    st = inputSudokuTable();
    printf("Sudoku Table before solve\n");
    showSudoku(st);
    st.isSudoku = checkSudoku(st);
    if (st.isSudoku == 0)
    {
        printf("This is not Sudoku table\n");
    }
    else
    {
        for(r=0;r<4;r++)
            for(c=0;c<4;c++)
                if (st.table[r][c]==0)
                {
                    x=r; y=c; r=5; c=5;
                }
        st.table[x][y] = solveSudoku(st);
        if(st.table[x][y]!=-1)
            printf("\nSolution = %d\n",st.table[x][y]);
        printf("Sudoku Table after solve\n");
        showSudoku(st);
    }
    return 0;
}
```

/\*\*\*\*\*\* จบตอนที่ 3 \*\*\*\*\*/