

ชื่อ-สกุล _____

รหัส _____

คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร

ข้อสอบปลายภาคการศึกษาต้น ปีการศึกษา 2557

ข้อสอบวิชา 517 111 การเขียนโปรแกรมคอมพิวเตอร์ 1

สอบวันพฤหัสบดีที่ 18 ธันวาคม 2557 เวลา 9.00-12.00 น. ห้อง ร.วท.1 และ 1240 ว.1

คำสั่ง

- ข้อสอบมีทั้งหมด 2 ตอน 11 หน้า (รวมหน้าคำสั่ง) คะแนนรวม 105 คะแนน
- ให้ตอบคำถามลงในข้อสอบ โดยเติมลงในช่องคำตอบที่เตรียมไว้ให้ในแต่ละข้อ
- ข้อสอบประกอบด้วยสองส่วน แต่ละส่วนมีคำสั่งที่ใช้เฉพาะในส่วนนั้น ๆ
- ห้ามนำสมุด หนังสือ เอกสาร ทุกชนิดเข้าไปในบริเวณที่นั่งสอบ
- ห้ามใช้เครื่องมือสื่อสารและเครื่องคิดเลขทุกชนิด
- ห้ามฉีกข้อสอบหรือนำข้อสอบออกนอกห้องสอบ
- อนุญาตให้ใช้ดินสอ 2B ขึ้นไปในการเขียนคำตอบได้

คะแนน

ข้อ	คะแนน	ข้อ	คะแนน
1	/ 3	7	/ 8
2	/ 7.5	8	/ 5
3	/ 6	9	/ 24
4	/ 9	10	/ 12.5
5	/ 8	11	/ 13
6	/ 9		
รวมข้อ 1 - 6	/ 42.5	รวมข้อ 7 - 11	/ 62.5
		รวมทั้งหมด	/ 105

ชื่อ-สกุล _____ รหัส _____

ตอนที่ 1 พื้นฐานการคำนวณ ความเข้าใจโครงสร้างและการทำงานของภาษาซี**คำสั่ง**

จากโปรแกรมหรือส่วนของโปรแกรมที่กำหนดให้ จงระบุว่าโปรแกรมจะพิมพ์ค่าใดออกมา (อาจมีมากกว่า 1 ค่า) โดยให้เขียนคำตอบลงในช่องว่างสำหรับคำตอบที่อยู่ในบรรทัดเดียวกันกับคำสั่ง printf หมายเหตุ ในปัญหาแต่ละข้อในส่วนนี้ได้สมมติว่ามีการ #include ไลบรารีมาตรฐานที่เหมาะสมมาเรียบร้อยแล้ว จึงไม่มีการแสดงคำสั่ง #include ในโปรแกรมหรือส่วนของโปรแกรมอีก นอกจากนี้ ในกรณีที่ printf มีการพิมพ์ \n ที่ด้านท้ายของการพิมพ์ ผู้เข้าสอบไม่จำเป็นต้องแสดงการขึ้นบรรทัดใหม่ในคำตอบแต่อย่างใด

1. ปัญหา พื้นฐานการดำเนินการทางตัวเลขในภาษาซี (3 คะแนน)

```
double d = 34.56789;
```

```
double e = 67.89012;
```

```
printf("%.1f", -d / 2);
```

ตอบ _____

```
printf("%d", (int)e);
```

ตอบ _____

```
printf("%.3f", e);
```

ตอบ _____

หมายเหตุ การแสดงผลยึดตาม C99 ที่ใช้กับ MinGW GCC ซึ่งพ่วงมากับ Code::Block 13.12

ชื่อ-สกุล _____ รหัส _____

2. ปัญหา พื้นฐานของอาเรย์ (7.5 คะแนน)

```
int A[100];    int i;
int s = 0;
for(i = 0; i < 100; ++i) {
    A[i] = i;
    s = i + 1;
}
```

```
for(i = 0; i < 3; ++i) {
    A[i] = s + A[(i * i) % 100];
    s = s + A[2 * i + 1];
}
```

```
printf("%d\n", A[2]);    ตอบ _____
```

```
printf("%d\n", A[3]);    ตอบ _____
```

```
printf("%d", s);        ตอบ _____
```

3. ปัญหา อาเรย์คู่เก็บตัวเลข [ไล่ลำดับการทำงานให้ดี ๆ ดูให้ชัด ๆ ว่าตอน printf ค่า i มีค่าเท่าใด] (6 คะแนน)

```
int A[] = {1, 2, 5, 4, 3, 7, 8};
int B[] = {1, 3, 2, 4, 6, 5};
```

```
int i = B[3];
printf("%d\n", B[i]+A[6-i]);    ตอบ _____
```

```
i = A[A[i] + 2];
printf("%d\n", i);            ตอบ _____
```

```
printf("%d\n", A[i/2]+B[i-3]);    ตอบ _____
```

ชื่อ-สกุล _____ รหัส _____

4. ปัญหา รูปสองชั้นกับอาร์เรย์สองมิติ (9 คะแนน)

```
int A[3][3];
for(j = 0; j < 3; ++j)
    for(i = 0; i < 3; ++i)
        A[j][i] = i + 2 * j;
```

```
int S[3];
for(j = 0; j < 3; ++j) {
    S[j] = 0;
    for(i = 0; i < 3; ++i) {
        S[j] += A[j][i];
    }
}
```

```
printf("%d\n", S[2]);          ตอบ _____
```

```
printf("%d\n", S[1]);          ตอบ _____
```

```
printf("%d\n", S[0]);          ตอบ _____
```

```
printf("%d\n", A[2][1]);       ตอบ _____
```

```
printf("%d\n", A[1][1]);       ตอบ _____
```

```
printf("%d\n", A[0][1]);       ตอบ _____
```

5. ปัญหา ฉันทสร้างฟังก์ชันด้วยตัวเอง แล้วก็งกับตัวเอง (8 คะแนน)

```
int f(int x, int y, int z) {
    x = x * 2;
    y = x + y + z;
    z = z - x - y;
    return z + y;
}
```

```
int main() {
    int x = 2;
    int y = -2;
    int z = 1;
    printf("%d\n", f(1, 2, -1));   ตอบ _____
    z = f(z, 0, z + x);
    y = f(x, y, -z);
```

ชื่อ-สกุล _____ รหัส _____

```

printf("%d\n", f(x, y, z));    ตอบ _____
printf("%d\n", z);           ตอบ _____
printf("%d\n", y);           ตอบ _____
return 0;
}

```

6. ปัญหา อาเรย์อลวน ปัญญาชนอลเวง (9 คะแนน)

```

int A[1000][1000];
for(int row = 0; row < 1000; ++row) {
    for(int col = 0; col < 1000; ++col) {
        A[row][col] = 1;
    }
}
int sum1 = 0;
int sum2 = 0;
for(int row = 0; row < 1000; ++row) {
    for(int col = 0; col < 1000; ++col) {
        int p = row - 50;    int q = col - 40;
        if(p < 0) p = 0;
        if(q < 0) q = 0;
        A[row][col] = A[p][q] + 2;
        sum1 += 1;
    }
    sum2 -= 2;
}
printf("%d\n", sum1+sum2);    ตอบ _____
printf("%d\n", A[20][3]);    ตอบ _____
printf("%d\n", A[700][800]);  ตอบ _____
printf("%d\n", A[801][90]);   ตอบ _____

```

หมายเหตุ 1 ข้อนี้สมมติว่าการประกาศใช้อาเรย์ขนาดใหญ่ในที่นี่ ไม่มีปัญหาเรื่องพื้นที่หน่วยความจำ**หมายเหตุ 2** ข้อนี้ถ้าใครคิดตรง ๆ แบบที่คอมพิวเตอร์ทำ สงสัยจะไม่สำเร็จง่าย ๆ ผู้เข้าสอบควรจะใช้ความฉลาดอย่างมนุษย์เพื่อคำนวณคำตอบที่เทียบเท่ากันของข้อนี้ให้ได้อย่างรวดเร็ว**คำแนะนำ** ถ้าเราหาค่าของ A[0][0] ได้แล้ว เราจะสามารถหาค่าของ A[10][5] ได้ทันทีหรือไม่ หรือว่าเราต้องทำการหาไล่เรียงค่าในอาเรย์ไปเรื่อย ๆ

ชื่อ-สกุล _____ รหัส _____

7. ปัญหา สตรีคเก็บข้อมูลและฉันทึที่พยายามเก็บคะแนน (8 คะแนน)

```

struct data {
    int x[5];
    int y[10];
    int z;
} typedef Data;

int main() {
    Data A[5];
    for(int i = 0; i < 5; ++i) {
        for(int s = 0; s < 5; ++s) {
            A[i].x[s] = i + s;
            A[i].y[2*s] = A[i].x[s] + 1;
            A[i].y[2*s + 1] = A[i].y[s] + 1;
        }
        A[i].z = 30 - A[i].x[4];
    }

    printf("%d\n", A[3].x[4]);    ตอบ _____
    printf("%d\n", A[4].y[9]);    ตอบ _____

    int what = 0;
    for(int i = 0; i < 5; ++i) {
        A[i].z = 0;
        for(int s = 0; s < 5; ++s) {
            A[i].z += A[i].x[s] + A[i].y[s];
        }
        what += A[i].z + A[i].y[i + 5];
    }

    printf("%d\n", A[2].z);    ตอบ _____
    printf("%d", what);    ตอบ _____

    return 0;
}

```

ชื่อ-สกุล _____ รหัส _____

8. ปัญหา ตัวชี้แจงง ค่าวิ่งวนไปมา (5 คะแนน)

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

```
void myFunc(int* x, int y) {  
    *x = *x + 1;  
    y = y + 1;  
    printf("%d\n", y + *x);  
}
```

ตอบ _____

```
int main() {  
    int x = 5;  
  
    myFunc(&x, x);  
    printf("%d", x);  
  
    return 0;  
}
```

ตอบ _____

ชื่อ-สกุล _____ รหัส _____

ตอนที่ 2 ทักษะการเขียนโปรแกรมภาษาซี**คำสั่ง**

จงเติมส่วนของโปรแกรมภาษา C ต่อไปนี้ให้สมบูรณ์ โดยสามารถใช้ทุกอย่างจากชุดคำสั่งมาตรฐานภาษา C ได้ ทั้งนี้ในข้อที่เป็นส่วนของโปรแกรมไม่จำเป็นต้อง #include ไฟล์เฮดเดอร์ (พวกไฟล์ .h) เข้ามาก่อน เว้นเสียแต่มีการถามในประเด็นดังกล่าวโดยตรง ทั้งนี้คำตอบที่ถูกไม่ขึ้นกับขนาดของช่องว่างที่เว้นไว้ แต่ช่องว่างแต่ละอันจะไม่มีการใช้เครื่องหมาย semi-colon (เครื่องหมาย ;) กล่าวคือสิ่งที่นักศึกษาต้องเติมลงไปจะอยู่ภายในขอบเขตข้อความสมบูรณ์ (complete statement) เพียงข้อความเดียวเท่านั้น นอกจากนี้คำตอบที่ถูกต้องอาจมีได้มากกว่าหนึ่งแบบ

นอกจากนี้ ข้อสอบไม่อนุญาตให้นักศึกษาเขียนโปรแกรมใหม่ทั้งหมด นักศึกษาจะต้องอ่านโปรแกรมหรือส่วนของโปรแกรมที่ให้ไปและเข้าใจจุดประสงค์ของมันได้ จากนั้นต้องเติมข้อความเพื่อให้โปรแกรมหรือส่วนของโปรแกรมทำงานตรงตามวัตถุประสงค์ของปัญหาได้

การให้คะแนนจะขึ้นอยู่กับความยากของแต่ละช่อง โดยช่องที่ยากกว่ากรณีทั่วไปหรือช่องที่นักศึกษามักจะตอบผิดจะมีช่องว่างเป็นเส้นประ เช่น _ _ _ _ _ ส่วนช่องที่มีความยากในเกณฑ์ปกติจะเป็นเส้นทึบ เช่น _____

9. ปัญหา ข้อความที่สั้นที่สุด จงเขียนโปรแกรมสมบูรณ์ที่หาข้อความที่สั้นที่สุดจากข้อความที่ผู้ใช้ใส่เข้ามา โดยที่

ข้อความแต่ละข้อความจะไม่มีช่องว่างปนอยู่เลย แต่จะเป็นข้อความที่มีตัวอักษรเรียงต่อกันไปไม่ขาดออกจากกัน

โปรแกรมนี้แสดงผลเป็นความยาวของข้อความที่สั้นที่สุดออกมาและขึ้นบรรทัดใหม่ แล้วจึงแสดงข้อความที่สั้น

ที่สุดดังกล่าวออกมา ในกรณีที่ข้อความที่สั้นที่สุดมีมากกว่าหนึ่งข้อความ ให้แสดงข้อความที่สั้นที่สุดอันสุดท้ายใน

ข้อมูลเข้า โดยโปรแกรมนี้จะเริ่มรับจำนวนข้อความที่ต้องการอ่านเข้ามาจากผู้ใช้เก็บไว้ในตัวแปรชื่อ N ก่อน จากนั้น

จึงเริ่มอ่านข้อความทีละข้อความเข้ามาตรวจสอบ ซึ่งข้อความจะยาวไม่เกิน 1024 ตัวอักษร (ความยาวนี้รวมอักขระ

ศูนย์แล้ว) ทั้งนี้อาเรย์ที่เก็บข้อความต้องมีความยาวที่ไม่พุ่มเพี้ยงเกินจากนี้ (24 คะแนน)

ตัวอย่าง

ข้อมูลเข้า	ผลลัพธ์
4 Lenovo HTC Oppo Samsung	3 HTC

ชื่อ-สกุล _____ รหัส _____

3 Pokemon Naruto Bleach	6 Bleach
----------------------------------	-------------

```

#include <stdio.h>

#include <_____>          [2 คะแนน]

int main() {
    int min = _____;      [2 คะแนน]

    char input[_____];      [1 คะแนน]

    char best[_____];       [1 คะแนน]

    int N;
    scanf("%d", &N);
    for(int i = 0; i < N; ++i) {
        scanf("_____", _____); [2 + 2 คะแนน]

        int length = _____(input); [2 คะแนน]

        if(length _____) { [3 คะแนน]
            min = length;
            for(int i = 0; _____; ++i) { [4 คะแนน]
                _____; [3 คะแนน]
            }
        }
    }

    printf("_____", min); [1 คะแนน]

    printf("_____", best); [1 คะแนน]
}

```


ชื่อ-สกุล _____ รหัส _____

11. ปัญหา ออกแบบสตรัคสำหรับเก็บข้อมูล (13 คะแนน)

นายมิกต้องการเก็บเกรดของนักศึกษาในภาคการศึกษาหนึ่ง ๆ ซึ่งเกรดแต่ละค่าจะเก็บในรูปของเลขทศนิยมแบบ double เช่น B+ จะแทนด้วย 3.5 และ W แทนด้วย -1 เป็นต้น ทว่านายมิกก็ได้ตระหนักดีว่าข้อมูลเกรดแต่ละตัวจะมาลอย ๆ ไม่ได้ แต่จะต้องประกอบไปด้วย (1) รหัสวิชา เป็นสายอักขระยาวไม่เกิน 10 ตัวอักษร เก็บไว้ในตัวแปรชื่อ code, (2) ชื่อวิชา เป็นสายอักขระยาวไม่เกิน 50 ตัวอักษร เก็บไว้ในตัวแปรชื่อ name, (3) จำนวนหน่วยกิต เป็นจำนวนเต็ม int เก็บไว้ในตัวแปรชื่อ credit และ (4) ตัวเกรดที่ได้เก็บไว้ในตัวแปรชื่อ val ซึ่งความยาวสายอักขระเหล่านี้ยังไม่รวมอักขระศูนย์สำหรับปิดท้ายข้อความ และเพื่อให้ข้อมูลเหล่านี้อยู่ด้วยกันอย่างเป็นระบบ นายมิกจึงเก็บรวมข้อมูลเกรดของวิชาหนึ่งไว้ในสตรัคชื่อ grade

แต่นายมิกก็พบว่าข้อมูลเกรดในแต่ละเทอมนั้นไม่ได้มีเกรดแค่ตัวเดียว แต่มีเกรดจากวิชาหลายวิชา แต่ก็ทราบดีว่าจะไม่เกิน 15 วิชาใน 1 ภาคการศึกษาแน่นอน นายมิกจึงสร้างอาร์เรย์ของสตรัคสำหรับเก็บเกรดขึ้นมาด้วย และเพื่อที่จะบอกได้ว่าในเทอมนี้ลงรายวิชาไปกี่รายวิชากันแน่ นายมิกจึงได้สร้างตัวแปร N ขึ้นมาเพื่อเก็บจำนวนรายวิชาที่นักศึกษาเจ้าของเกรดลงทะเบียนไว้ สุดท้ายเพื่อให้รู้ได้ว่าเป็นเกรดใครจึงต้องมีการเก็บรหัสนักศึกษาไว้ด้วยซึ่งเป็นสายอักขระมีความยาวไม่เกิน 10 ตัวอักษร ไม่รวมอักขระศูนย์ เช่นเดิม เพื่อให้ข้อมูลเกรดนักศึกษาแต่ละคนอยู่ด้วยกันอย่างเป็นระบบ นายมิกจึงสร้างสตรัคชื่อ semester ขึ้นมาเพื่อรองรับการเก็บข้อมูลนี้

จงเติมโค้ดลงไปในช่วงที่กำหนดให้ เพื่อให้ได้สตรัคที่เก็บข้อมูลตามข้อกำหนดข้างต้น

```

struct grade {

    char code[_____];           [1 คะแนน]

    char _____ [_____];     [1 + 1 คะแนน]

    _____ _____;         [1 + 1 คะแนน]

    _____ credit;            [1 คะแนน]

} typedef Grade;

struct semester {

    char ID[_____];             [1 คะแนน]

    _____ _____;         [1 + 1 คะแนน]

    _ _ _ _ _ A[_____];        [3 + 1 คะแนน]

} typedef Semester;

```