

คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร

เฉลย ข้อสอบปลายภาคการศึกษาปลาย ปีการศึกษา 2554

ข้อสอบชุด

3

ข้อสอบวิชา 517 111 การเขียนโปรแกรมคอมพิวเตอร์ 1

สอบวันเสาร์ที่ 24 มีนาคม 2555 เวลา 13.30-16.30 น.

คำสั่ง

1. ข้อสอบมีทั้งหมด 2 ตอน 10 หน้า (รวมหน้าคำสั่ง) คะแนนรวม 110 คะแนน
2. ให้ตอบคำถามลงในข้อสอบ โดยเติมลงในช่องคำตอบที่เตรียมไว้ให้ในแต่ละข้อ
3. ข้อสอบประกอบด้วยสองส่วน แต่ละส่วนมีคำสั่งที่ใช้เฉพาะในส่วนนั้น ๆ
4. ห้ามนำสมุด หนังสือ เอกสาร ทุกชนิดเข้าไปในบริเวณที่นั่งสอบ
5. ห้ามใช้เครื่องมือสื่อสารและเครื่องคิดเลขทุกชนิด
6. ห้ามฉีกข้อสอบหรือนำข้อสอบออกนอกห้องสอบ
7. อนุญาตให้ใช้ดินสอ 2B ขึ้นไปในการเขียนคำตอบได้

คะแนน

ข้อ	คะแนน	ข้อ	คะแนน
1		8	
2		9	
3		10	
4		11	
5		12	
6		รวมตอนที่ 2	
7		คำถามพิเศษ	
รวมตอนที่ 1		รวมทั้งหมด	

ตอนที่ 1 ความเข้าใจโครงสร้างและการทำงานของภาษาซี**คำสั่ง**

จากโปรแกรมหรือส่วนของโปรแกรมที่กำหนดให้ จงระบุว่าโปรแกรมจะพิมพ์ค่าใดออกมา (อาจมีมากกว่า 1 ค่า) โดยให้เขียนคำตอบลงในช่องว่างสำหรับคำตอบที่อยู่ในบรรทัดเดียวกันกับคำสั่ง printf หมายเหตุ ในปัญหาแต่ละข้อในส่วนนี้ได้สมมติว่ามีการ #include ไลบรารีมาตรฐานที่เหมาะสมมาเรียบร้อยแล้ว จึงไม่มีการแสดงคำสั่ง #include ในโปรแกรมหรือส่วนของโปรแกรมอีก

1. (2 คะแนน)

```
float x = 23.234567890;
```

```
printf("%.0f", x);      ตอบ  23
```

```
printf("%.5f", x);      ตอบ  23.2346
```

2. (4 คะแนน)

```
int A[5];      int i;
```

```
int s = 0;
```

```
for(i = 0; i < 5; ++i) {
```

```
    A[i] = 2*i * i;
```

```
    s = s + A[i];
```

```
}
```

```
printf("%d %d", A[1], A[3]);      ตอบ  2  18
```

```
printf("%d", s);                  ตอบ  60
```

3. (6 คะแนน)

```
int fx (int x) {
```

```
    if (x % 4 == 0) return 1;
```

```
    else return 0;
```

ชื่อ-สกุล รหัส

```

}

int fy (int x) {
    return ! fx( x );
}

void main() {

    printf("%d %d", fx( 0 ), fy( 0 ) );   ตอบ _____ 1  0 _____

    printf("%d %d",  fx( 3 ), fy( 2 ) );   ตอบ _____ 0  1 _____

    int a = fy( 0 );  int b = fx( 5 );

    printf("%d %d", fx( a ),  fy( b ) );   ตอบ _____ 1  1 _____

}

```

4. (9 คะแนน)

```

int g (int x, int y) {
    x = x + 1;
    y = y - 1;
    return (x * x) + (x - y);
}

void main() {

    printf( "%d", g(1, 2) );               ตอบ _____ 5 _____

    printf( "%d", g(3, 4) );               ตอบ _____ 17 _____

    int x = 1;  int y = 0;

    printf( "%d", g(x, y) );               ตอบ _____ 7 _____

    printf( "%d %d", x, y );               ตอบ _____ 1  0 _____

```

ชื่อ-สกุล รหัส

```
printf( "%d", g(y, x) );
```

ตอบ 2

```
printf( "%d %d", x, y );
```

ตอบ 1 0

```
}
```

5. ในข้อนี้กำหนดให้ผลการพิมพ์ในฟังก์ชัน fz ขึ้นอยู่กับค่า x และ y ที่ส่งจาก main ดังแสดงในปัญหานี้ คือให้ตอบเป็นตัวเลขที่สอดคล้องกับค่าที่ส่งไปจาก main

(5 คะแนน)

```
void fz (int x, int y);
```

```
void main() {
```

```
int x = 4; int y = 8;
```

```
fz( x, y );
```

```
printf("%d", x );
```

ตอบ 4

```
printf("%d", y );
```

ตอบ 8

```
}
```

```
void fz (int x, int y) {
```

```
int temp = x;
```

```
x = y;
```

```
y = x;
```

```
printf("%d", x );
```

ตอบ 8

```
printf("%d", y );
```

ตอบ 8

```
}
```

ชื่อ-สกุล รหัส

6. (12 คะแนน)

```
int A[3][3];   int i, j;
for(j = 0; j < 3; ++j)
    for(i = 0; i < 3; ++i)
        A[j][i] = 2*i + 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", S[2]);           ตอบ  12 
```

```
printf("%d", S[1]);           ตอบ  9 
```

```
printf("%d", S[0]);           ตอบ  6 
```

```
printf("%d", A[2][1]);        ตอบ  4 
```

```
printf("%d", A[1][1]);        ตอบ  3 
```

```
printf("%d", A[0][1]);        ตอบ  2 
```

```
for(j = 0; j < 3; ++j) {
    S[j] = 0;
    for(i = 0; i < 3; ++i) {
        A[j][2-i] = A[j][i];
        S[j] += A[j][2-i];
    }
}
```

ชื่อ-สกุล รหัส

}

printf("%d", S[2]); ตอบ 8 printf("%d", S[1]); ตอบ 5 printf("%d", S[0]); ตอบ 2 printf("%d", A[2][1]); ตอบ 4 printf("%d", A[1][1]); ตอบ 3 printf("%d", A[0][1]); ตอบ 2

7. (5 คะแนน)

int A[10]; int B[5]; int i;

for(i = 0; i < 10; ++i) {

A[i] = i - 5;

B[i / 2] = i;

}

int sum = 0;

for(i = 0; i < 10; ++i)

sum += B[i/2] - A[i];

printf("%d", sum); ตอบ 55

sum = 0;

for(i = 0; i < 10; ++i)

sum += B[i / 2] + A[i];

printf("%d", sum); ตอบ 45

ชื่อ-สกุล รหัส

ตอนที่ 2 ทักษะการเขียนโปรแกรมภาษาซี**คำสั่ง**

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

นอกจากนี้ ข้อสอบไม่อนุญาตให้นักศึกษาเขียนโปรแกรมใหม่ทั้งหมด นักศึกษาจะต้องอ่านโปรแกรมหรือส่วนของโปรแกรมที่ให้ไปและเข้าใจจุดประสงค์ของมันได้ จากนั้นต้องเติมข้อความเพื่อให้โปรแกรมหรือส่วนของโปรแกรมทำงานตรงตามวัตถุประสงค์ของปัญหาได้

8. **ปัญหา** ต้องการประกาศสตรัค (struct) เพื่อเก็บข้อมูลนักศึกษา โดยในสตรัคประกอบด้วยข้อมูลต่อไปนี้ (เรียงตามลำดับการปรากฏในสตรัค) 1. รหัสนักศึกษามีชื่อตัวแปรคือ ID เป็นเลขจำนวนเต็มแบบ int 2. ชื่อนักศึกษาเป็นข้อความมีความยาวไม่เกิน 25 ตัวอักษร (ยังไม่รวม null character ปิดท้าย) โดยมีชื่อตัวแปรคือ first_name 3. นามสกุลนักศึกษาเป็นข้อความมีความยาวไม่เกิน 45 ตัวอักษร (ยังไม่รวม null character ปิดท้าย) โดยมีชื่อตัวแปรคือ last_name 4. อาจารย์เก็บเกรดเฉลี่ยแต่ละภาคการศึกษาของนักศึกษา มีความยาว 25 ช่องข้อมูล แต่ละช่องเป็นเลขทศนิยมแบบ single precision อาจารย์นี้มีชื่อว่า arGrades (10 คะแนน)

```

__struct_____ Student {

    int ID;

    char first_name _____[26]_____ ;

    char last_name _____[46]_____ ;

    __float_____ arGrades __[25]____ ;

};

```

ชื่อ-สกุล รหัส

9. **ปัญหา** ต้องการเขียนฟังก์ชันหาค่าเฉลี่ยของข้อมูลในอาร์เรย์ arData ซึ่งอาร์เรย์นี้มีความยาวเป็นเลขจำนวนเต็ม N และค่าเฉลี่ยที่ได้คำนวณได้จะถูกคืนมาเป็นเลขทศนิยมแบบ single precision (14 คะแนน)

```

__float__ average ( int ___N___ , int* ___arData___ ) {

    int i;

    int sum = 0;

    for( i = 0; ___i < N___; ++i ) {

        sum += ___arData[i]___;

    }

    return sum / ( __float__ ) ___N___;

}

```

10. **ปัญหา** ขนาดของเวกเตอร์ (14 คะแนน)

ปัญหานี้ต้องการสร้างฟังก์ชัน vector_norm เพื่อคำนวณขนาดของเวกเตอร์ $\vec{v} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$ ซึ่งสามารถคำนวณได้จาก $\sqrt{x^2 + y^2 + z^2}$ เมื่อกำหนดค่า x , y และ z มาให้กับฟังก์ชันผ่าน main ในโปรแกรมสมบูรณ์ (ในปัญหานี้ค่า x , y และ z ถูกกำหนดไว้ตายตัวในฟังก์ชัน main)

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

```
#include <___math.h___>
```

```
double vector_norm(double x, double y, double z) {
```

คำตอบด้านล่างนี้ที่จริงจะเปลี่ยนลำดับก็ได้ หรือจะใช้วิธีที่ต่างจากนี้ก็ได้อะ ตราบใดที่ให้ผลที่ถูกต้อง

```
___double___ sum = ___x * x___;
```

```
sum += ___y * y___;
```

```
sum += ___z * z___;
```

ชื่อ-สกุล รหัส

```

return ____sqrt(sum)____;

}

void main() {

double x, y, z;

scanf(" ____%lf %lf %lf____", &x, &y, &z);

double norm = ____vector_norm(x, y, z)____;

//ถ้าเป็นมาตรฐาน C เวอร์ชันที่ใหม่กว่า ตรง printf ตรงนี้จะเป็น %f ไม่ใช่ %lf

printf("%lf\n", norm);

}

```

11. ปัญหา สตริงที่มาก่อนตามหลักพจนานุกรม (12 คะแนน)

ในปัญหานี้ เราต้องการเขียนฟังก์ชัน `word_order` ที่รับพารามิเตอร์มาเป็นสตริงสองตัวที่ประกอบด้วยตัวอักษร A ถึง Z ซึ่งอาจจะมีตัวเล็กและตัวใหญ่ปนกัน โดยพารามิเตอร์ทั้งสองคือ `str1` และ `str2` ตามลำดับ หาก `str1` มาก่อน `str2` ตามหลักพจนานุกรม (ในที่นี้คือไม่คำนึงถึงความเป็นตัวเล็กตัวใหญ่) ฟังก์ชัน `word_order` จะคืน 1 แต่ถ้า `str2` มาก่อน `str1` ฟังก์ชัน `word_order` จะคืน 2 แต่ถ้า `str1` และ `str2` มีลำดับที่เทียบเท่ากัน ฟังก์ชัน `word_order` จะคืน 0 สุดท้ายฟังก์ชันนี้ถูกเรียกใช้จาก `main` ในโปรแกรมสมบูรณ์อีกครั้งหนึ่ง

```

int word_order ( ____char____ * str1, ____char____ * str2 ) {

int x = strcmpi ( ____str1____ , ____str2____ );

if ( ____x == 0____ )

return 0;

else if ( ____x < 0____ )

```

ชื่อ-สกุล รหัส

```

        return 2;

    else

        return 1;

}

```

12. ปัญหา

อาเรย์ที่มีค่ามากกว่า (12 คะแนน)

ในปัญหานี้เราต้องการเขียนฟังก์ชัน `better_array` ที่รับอาเรย์ `A1` และ `A2` ที่เก็บจำนวนเต็มไว้ `N` ตัว โดย `A1`, `A2`, และ `N` ต่างเป็นอินพุตที่ส่งมากับพารามิเตอร์ของฟังก์ชัน หากผลรวมของตัวเลขในอาเรย์ `A1` มีค่ามากกว่าตัวเลขใน `A2` ฟังก์ชัน `better_array` จะคืนเลข 1 แต่ถ้าผลรวมของตัวเลขใน `A2` มากกว่า `A1` ฟังก์ชันจะคืนเลข 2 แต่ถ้าผลรวมของตัวเลขใน `A1` และ `A2` เท่ากัน ฟังก์ชันจะคืนเลข 0

```

int better_array(int* A1, int* A2, int _____ N _____) {

    int sum1 = 0;    int sum2 = 0;

    int _____ k _____;

    for (k = 0; k < N; ++k) {

        sum1 += A1[k];

        sum2 += A2[k];

    }

    if ( _____ sum1 == sum2 _____ )

        return 0;

    else if ( _____ sum2 > sum1 _____ )

        return _____ 2 _____;

    else

        return 1;

}

```