

เลขที่นั่งสอบ

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

ข้อสอบปลายภาคการศึกษาที่ 2/2550

วันจันทร์ที่ 10 มีนาคม 2551

เวลา 9.00 -12.00 น.

วิชา CPE 100 Computer Programming for Engineers. น.ศ. วศ.เคมี ปีที่ 1 กลุ่มที่ 1, 2

น.ศ. วศ.เคมี 2 ภาษา ปีที่ 1 กลุ่มที่ 21

คำสั่ง

1. ข้อสอบมีทั้งสิ้น 5 ข้อ จำนวน 8 แผ่น(รวมแผ่นนี้ และใบแนบ)
2. ให้ทำข้อสอบทุกข้อลงในตัวข้อสอบที่เว้นช่องไว้ให้
2. ไม่อนุญาตให้นำเครื่องคำนวณใดๆเข้าห้องสอบ
4. เขียนชื่อ และ รหัสประจำตัว ลงในกระดาษคำตอบทุกแผ่น (และแผ่นนี้)

(อ.พิพัฒน์ สุภศิริสันต์)

ผู้ออกข้อสอบ (9082)

ข้อสอบนี้ได้ผ่านการประเมินจากภาควิชาวิศวกรรมคอมพิวเตอร์แล้ว

ชื่อ รหัสประจำตัว..... ภาควิชา/ชั้นปี.....

โจทย์ 1 - 2

กำหนดให้ สมมุติให้ มีการเก็บค่าอุณหภูมิที่เวลา 07.00 น. ของแต่ละวันใน 1 ปีไว้ในไฟล์ชื่อ "TEMP.TXT" โดยค่าที่เก็บเป็นตัวเลขจำนวนจริงดังตัวอย่าง

32.0

31.8

.....

ต้องการเขียนโปรแกรม เพื่อหาค่าต่ำสุด สูงสุด ค่าเฉลี่ย และค่าเบี่ยงเบนมาตรฐานของอุณหภูมิ ในไฟล์ดังกล่าว โดยเขียนฟังก์ชัน main () ดังนี้

```
void main(void)
```

```
{ double data[1000] , min, max, mean , sd ;
```

```
int count;
```

```
Read_Temperature_File ("TEMP.TXT", data, &count );
```

```
Find_Statistical_Data(data, count, &min, &max , &mean, &sd);
```

```
printf("\nMinimum temperature = %6.2lf \nMaximum temperature " , min , max);
```

```
printf("\nAverage temperature = %6.2lf \nStandard Deviation " , mean , sd);
```

```
}
```

สูตรในการหาค่าเฉลี่ย

$$mean = \frac{\sum_{i=1}^n x_i}{n}$$

สูตรในการหาค่าส่วนเบี่ยงเบนเฉลี่ย

$$sd = \sqrt{\frac{\sum_{i=1}^n x_i^2}{n} - mean^2}$$

ชื่อ.....รหัสประจำตัว.....ภาควิชา/ชั้นปี.....

1. จงสร้างฟังก์ชัน `Read_Temperature_File()` ซึ่งมีพารามิเตอร์ที่สอดคล้องกับการเรียกใช้ ในฟังก์ชัน `main()` เพื่อให้โปรแกรมนี้ทำงานได้

[illegible]

ชื่อ.....รหัสประจำตัว.....ภาคเรียนนี้.....

2. จงสร้างฟังก์ชัน **Find_Statistical_Data** () ซึ่งมีพารามิเตอร์ที่สอดคล้องกับการเรียกใช้ ในฟังก์ชัน **main()** เพื่อให้โปรแกรมนี้ทำงานได้

[illegible]

3. จงสร้างฟังก์ชัน เพื่อหาผลบวกของ MATRIX ซึ่งเป็นอาร์เรย์ 2 มิติ ของตัวเลขจำนวนเต็ม พร้อมขนาดที่ส่งมาให้ทางพารามิเตอร์ 2 ชุดแรก แต่ละชุดประกอบด้วย ตัวแปรเก็บค่า และขนาดของ row, col ตามลำดับ เมื่อบวกกันแล้วให้ส่งคำตอบกลับคืนโดยใช้พารามิเตอร์ตัวที่ 3 โดยมีตัวอย่าง การเรียกใช้งานดังนี้

```
Add_Matrix(MA , row_A , col_A , MB , row_B , col_B, MC , &row_C, &col_C);
```

กำหนดให้ สมาชิกของแต่ละเมตริกซ์ เป็นตัวเลขจำนวนเต็ม มีขนาดไม่เกิน 20 x 20

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

5. จงสร้างฟังก์ชัน Search_Data () เพื่อค้นหา และ แสดงผล นักเรียนคนที่ได้เกรดเฉลี่ยมากกว่าหรือเท่ากับ ค่าที่กำหนดในพารามิเตอร์ตัวที่ 3

(เช่น ถ้าต้องการให้แสดงผลนักเรียนที่มีเกรดเฉลี่ย มากกว่าหรือเท่ากับ 3.00)

This image shows a full page of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page, providing a template for handwriting practice. There are no margins, text, or other markings on the page.

ARRAY(Statistics)

- ARRAY 1 นิติใช้เมื่อต้องการเก็บข้อมูลจำนวนมากไว้ในตัวแปร
- การจองตัวแปร ARRAY ให้กำหนดขนาดสูงสุดที่โปรแกรมสามารถทำงานได้ และจำนวนข้อมูลที่เก็บอยู่จริง

```
double data[200];
int count = 0;
```
- การประมวลผลข้อมูล ใช้การวนรอบเพื่อกระทำกับข้อมูลตั้งแต่ตัวแรกจนถึงตัวสุดท้าย แล้วประมวลผลรอบละตัวจนครบทุกตัว

```
int i;
for (i=1; i<= count; i++)
{ให้ประมวลผลตัวแปร data[i]}
```
- การสร้างฟังก์ชันที่ต้องส่งตัวแปรเป็นพารามิเตอร์ ไม่ต้องกำหนดขนาดข้อมูลในวงเล็บ และใช้เหมือนกันทั้ง input , output

```
void Read_Data (double data[], int *count)
void Process_Data (double data[], int count)
double Find_Answer(double data[], int count)
```
- การเรียกใช้ฟังก์ชันที่สร้างขึ้นให้ใช้ชื่อตัวแปรอาร์เรย์โดยตรง ไม่ต้องกำหนดขนาด

```
Read_Data ( data, &count);
Process_Data ( data, count);
ans = Find_Answer(data, count);
```

1

MATRIX

- การจองตัวแปร ARRAY 2 มิติ เช่น matrix ให้จองเป็นอาร์เรย์ 2 มิติของตัวเลขสูงสุดที่ทำงานได้ และข้อมูลที่เก็บอยู่จริงในแต่ละมิติ

```
int M[10][10];
int row, col;
```
- การประมวลผลใช้การวนรอบซ้อน 2 ชั้น เพื่อกระทำกับข้อมูลทีละตัวจนครบ

```
int i, j;
for (i=1; i<= row; i++)
{ รอบการประมวลผลของ i ทีละแถว(มิติแรก)
for (j = 1; j<= col; j++)
{รอบของการประมวลผล j ทีละหลัก (มิติหลัง)
ประมวลผลข้อมูล M[i][j] }
}
```
- การสร้างฟังก์ชันที่ต้องส่งตัวแปรเป็นพารามิเตอร์ ไม่ต้องกำหนดขนาดเฉพาะข้อมูลในวงเล็บแรกเท่านั้น และใช้เหมือนกันทั้ง input , output

```
void Process_Matrix (int M[][10], int row, int col)
```
- การเรียกใช้ฟังก์ชันที่ต้องส่งค่าอาร์เรย์ ให้ใช้ชื่อตัวแปรอาร์เรย์ โดยไม่ต้องกำหนดขนาด

```
Process_Matrix ( M, rowa, cola);
```

2

Text File

- ภาษาซี ถ้าต้องการใช้ TEXT File ต้องทำตามขั้นตอนดังนี้
 - จองตัวแปรสำหรับใช้เป็นไฟล์
 - FILE *fp; /* fp คือชื่อตัวแปรไฟล์ที่ต้องการนำไปใช้ */
 - คำสั่งกำหนดชื่อไฟล์ในดิสก์ แล้วเปิดไฟล์
 - fp = fopen("filename.ext", "r"); /* เปิดไฟล์ fp เพื่ออ่าน */
 - fp = fopen("filename.ext", "w"); /* เปิดไฟล์ fp เพื่อเขียน */
 - ใช้วนรอบ while ในการอ่านข้อมูลจากไฟล์ ที่ไม่ทราบจำนวนเก็บไว้ในอาร์เรย์

```
while (fscanf(fp, "%d", &a) == 1) /*วนรอบขณะที่อ่านข้อมูลได้ 1 ตัว*/
{ *count = *count + 1; /* เพิ่มค่าตัวนับ */
data[*count] = a; } /* นำค่าที่อ่านได้ไปเก็บ*/
```
 - ถ้ารู้จำนวนข้อมูลที่ต้องอ่าน อาจใช้วนรอบ for ในการอ่านได้

```
fscanf(fp, "%d%d", &row, &col); /*อ่านจำนวน row ,col จากไฟล์*/
for (i=1; i<=row; i++) /*กำหนดจำนวน row ที่จะอ่าน*/
for (j=1; j<=col; j++) /*กำหนดจำนวน col ที่จะอ่าน*/
{ fscanf(fp, "%d", &M[i][j]); }
```
 - ถ้าต้องการเขียนข้อมูลลงไฟล์ให้ใช้คำสั่ง fprintf(fp, "...", ..., ...);
 - ใช้คำสั่งปิดไฟล์เมื่อเลิกใช้งาน

```
fclose(fp);
```

STRUCTURE

- ชนิดข้อมูลโครงสร้าง ใช้เมื่อตัวแปร แต่ละตัวประกอบด้วยคุณสมบัติ(ตัวแปร) ย่อยๆ หลายตัวมารวมกัน มักใช้ร่วมกับอาร์เรย์ 1 มิติ เช่น
 - โครงสร้างข้อมูลสมุดโทรศัพท์ ประกอบด้วย ชื่อ(ตัวอักษร) เบอร์โทร (ตัวเลข)
- สร้างโครงสร้างข้อมูลใหม่ ไว้ในส่วนที่กีดจาก #include
- ใช้คำสั่ง struct ชื่อโครงสร้าง {ชื่อตัวแปร(ฟิลด์)ที่นำมารวมกัน..... }

```
struct phonebook { char name[20];
long int tel; };
```
- จองตัวแปรอาร์เรย์ของโครงสร้างสำหรับเก็บข้อมูล และจำนวนข้อมูลที่เก็บจริงในฟังก์ชัน main

```
struct phonebook phone[100];
int count;
```
- การส่งค่าตัวแปรแบบโครงสร้างเป็นพารามิเตอร์ ใช้เช่นเดียวกับอาร์เรย์

```
void Process_Data(struct phonebook phone[], int count)
เมื่อต้องการเรียกใช้ Process_Data( phone, count);
```
- การใช้งานตัวแปรย่อยในโครงสร้าง ให้ใช้ชื่อตัวแปร.ชื่อฟิลด์
 - ตัวแปร x จะใช้ได้เป็น x.name, x.tel
 - for (i=1; i<=count; i++)
{ ตัวแปร phone[i] จะใช้ได้เป็น phone[i].name และ phone[i].tel }