เลขที่นั่งสอบ ☐

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

ข้อสอบปลายภาคการศึกษาที่ 1/2552

วันจันทร์ที่ 5 ตุลาคม 2552 เวลา 9.00 -12.00 น.

วิชา CPE 100 Computer Programming for Engineers. น.ศ. วศ.สิ่งแวดล้อม ปีที่ 1 กลุ่มที่ 1, 21

<u>คำสั่ง</u>

1. ข้อสอบมีทั้งสิ้น 5 ข้อ ข้อละ 7 คะแนน รวม 35 คะแนน จำนวน 7 แผ่น(รวมแผ่นนี้ และใบแนบ)
2. ให้ทำข้อสอบทุกข้อลงในตัวข้อสอบที่เว้นช่องไว้ให้
3. ไม่อนุญาตให้นำเครื่องคำนวณ และเอกสารใดๆเข้าห้องสอบ
4. เขียนชื่อ และ รหัสประจำตัว ลงในกระดาษคำตอบทุกแผ่น (และแผ่นนี้)

.....

(อ.พิพัฒน์ ศุภศิริสันต์)

ผู้ออกข้อสอบ (9082)

ข้อสอบนี้ได้ผ่านการประเมินจากภาควิชาวิศวกรรมคอมพิวเตอร์แล้ว

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

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

1. จงสร้างฟังก์ชัน Add_Data เพื่อเพิ่มข้อมูลลงในอาร์เรย์ของตัวเลขจำนวนจริงที่กำหนดโดยพารามิเตอร์ชื่อ data[ ] ซึ่งมีจำนวนทั้งสิ้น count ตัว ในตำแหน่งต่อจากข้อมูลตัวสุดท้าย โดยให้มีการวนรอบตั้งคำถามเหมือนที่กำหนดในตัวอย่าง แล้วให้ผู้ใช้ป้อนข้อมูลเพิ่มได้เรื่อยๆ โดยอัตโนมัติ และต้องมีการป้องกันความผิดพลาดจากการป้อนข้อมูลผิด (เช่น การป้อนข้อมูลเป็นตัวอักษร) จนกว่าผู้ใช้จะป้อนค่าเป็น -999 โปรแกรมจึงจะหลุดออกจากฟังก์ชันนี้

ตัวอย่าง ลักษณะคำถามที่ต้องการ กรณีที่ count = 11 และโปรแกรมกำลังต้องการให้ป้อนข้อมูลตัวที่ 12

```
Enter data #12 (Enter -999 to End) :
```

ตัวอย่างฟังก์ชัน main() เมื่อมีการเรียกใช้

```
void  main()
{   double  data[1000];
    int   count ;
          .....
          if (select == '1')  Add_Data ( data, &count ) ;
          .....
}
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. จงเติมคำสั่งฟังก์ชันชื่อ Average_Range เพื่อพิมพ์ค่าเฉลี่ยของข้อมูลในตัวแปรชื่อ data[ ] ซึ่งมีจำนวน count ตัว โดยมีเงื่อนไขให้หาค่าเฉลี่ยเฉพาะตัวที่มีค่าอยู่ในระหว่างช่วง min ถึง max ที่กำหนดโดยพารามิเตอร์เท่านั้น

คำแนะนำ ก่อนที่จะหาผลรวมของ data [ ] จะต้องมีการเปรียบเทียบ ว่าข้อมูลอยู่ในช่วง min ถึง max ที่ต้องการหรือไม่ ถ้าใช่ จะต้องมีการนับจำนวนข้อมูล และทำการหาผลรวม เมื่อวนรอบเสร็จแล้วจึงนำผลรวม และจำนวนข้อมูลที่หาได้ ไปหาค่าเฉลี่ย แล้วแสดงผล min, max และ ค่าเฉลี่ยที่คำนวนได้

ตัวอย่างการแสดงผล เมื่อต้องการหาค่าเฉลี่ย ของข้อมูลในช่วง min = 50 และ max = 70 และสมมุติว่าคำนวนค่าเฉลี่ยได้คำตอบเป็น 62.24

Average data in range **50 - 70** is **62.64**

```
void   Average_Range ( double  data[ ] ,int count , double min, double max )
{  .....
   .....
   .....
   .....
   .....
   .....
   .....
   .....
   .....
   .....
   .....
   .....
}
```

3. จงสร้างฟังก์ชัน Transpose_Matrix เพื่อหาค่า Transpose ของ MATRIX ซึ่งเป็นอาร์เรย์ 2 มิติ ของตัวเลขจำนวนจริง พร้อมขนาด row และ col ที่ส่งมาให้ทางพารามิเตอร์ ชุดแรก แล้วให้ส่งคำตอบกลับคืน โดยใช้พารามิเตอร์ชุดที่ 2 มีตัวอย่างการเรียกใช้งานดังนี้

```
void main()
{  double  MA[10][10], MB[10][10], MC[10][10] ;
   int  rowA, colA, rowB, colB, rowC, colC;
            .....
            Transpose_Matrix (MA , rowA , colA , MC , &rowC, &colC);
            .....
}
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

4. กำหนดให้ ฟังก์ชันที่เกี่ยวกับ MATRIX มีการสร้างไว้แล้ว และมีส่วนหัวในการเรียกใช้ดังนี้

- void Load_MATRIX (char filename[ ],double M[ ][10], int *row, int *col);
  ทำหน้าที่อ่านไฟล์ (filename) เก็บลงในเมตริก (M , row, col)
- void Save_MATRIX (char filename[ ],double M[ ][10], int row, int col);
  ทำหน้าที่นำข้อมูลเมตริก (M, row, col) เขียนลงในไฟล์ (filename)
- void Multiply_MATRIX (double M1[ ][10], int row, int col , double M2[ ][10], int row, int col , double M3[ ][10], int *row, int *col);
  ทำหน้าที่คูณเมตริกที่กำหนด M3 = M1 x M2
- void Sub_MATRIX (double M1[ ][10], int row, int col , double M2[ ][10], int row, int col , double M3[ ][10], int *row, int *col);
  ทำหน้าที่ลบเมตริกที่กำหนด M3 = M1 - M2
- void Transpose_MATRIX (double M1[ ][10], int row, int col , double M2[ ][10], int *row, int *col);
  ทำหน้าที่หาทรานโพสของเมตริก M2 = Transpose (M1)

คำสั่ง จงเขียนโปรแกรม เพื่อ อ่านเมตริกที่เก็บอยู่ในไฟล์ "MA.TXT" มาคูณกับ "MB.TXT" แล้วนำมาลบออกจาก Transpose ของเมตริกที่เก็บอยู่ในไฟล์ "MC.TXT" ผลลัพธ์ที่ได้ให้สร้างไว้ในไฟล์ชื่อ "ANS.TXT" ตามสูตร ไฟล์คำตอบ = Transpose(ไฟล์MC) - (ไฟล์MA x ไฟล์MB)
โดยกำหนดให้ใช้ตัวแปรเฉพาะเท่าที่มีอยู่เท่านั้น

```
void main ()
{ double MA[10][10], MB[10][10], MC[10][10] ;

  int rowA, colA, rowB, colB, rowC, colC;

  .....

  .....

  .....

  .....

  .....

  .....

  .....

  .....

  .....

}
```

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

5. กำหนดให้ ข้อมูลที่ใช้ในการคำนวณรูปร่าง ถูกเก็บในไฟล์ "STUDENT.TXT" ในแต่ละบรรทัด จะใช้แทนข้อมูลสำหรับ 1 คน ซึ่งจะประกอบด้วยข้อมูลย่อย 6 ฟิลด์ คือ รหัสประจำตัว ชื่อ นามสกุล อายุ น้ำหนัก และส่วนสูง ตามลำดับ โดยข้อมูลแต่ละตัวจะใช้เว้นวรรคเป็นตัวคั่นระหว่างข้อมูล ดังตัวอย่าง

51292302 นายกฤษฎา มัจฉาธิคุณ 18 65.2 170.5

ให้กำหนดขนาดความกว้างสูงสุดของ ชื่อ นามสกุล เท่าไรก็ได้แต่ให้สมเหตุผล

จงออกแบบโครงสร้างชื่อ student_info และสร้างฟังก์ชันชื่อ Load_Information เพื่อทำหน้าที่อ่านข้อมูลทั้งหมดจากไฟล์ดังกล่าว โดยการเติมโปรแกรมให้สมบูรณ์

```
struct student_info { .....
                      .....
                      .....
                      .....
                      .....
                      ..... } ;
void Load_Information (struct student_info data [ ] , int *count)
{ .....
  .....
  .....
  .....
  .....
  .....
  .....
  .....
  .....
  .....
  .....
  .....
  }
```

ARRAY(Statistics)

- ARRAY 1 มิติใช้เมื่อต้องการเก็บข้อมูลจำนวนมากไว้ในตัวแปร
- การจองตัวแปร ARRAY ให้กำหนดขนาดสูงสุดที่โปรแกรมสามารถทำงานได้ เช่น
  double data[200] ;
- ต้องกำหนดตัวแปร เพื่อระบุขอบเขตที่ใช้งานจริง และตัวนับเพื่อใช้วนรอบ
  int count , i ;
- ใช้การวนรอบเพื่อกระทำกับข้อมูลทีละตัวจนครบทุกตัว เช่น
  for (i=1 ; i<= count ; i++)
  { จะมองเห็นข้อมูลรอบละตัว คือ data[i]
  ให้ประมวลผลข้อมูล data[i] ตามที่กำหนด }
- การกำหนดค่าพารามิเตอร์ที่เป็นอาร์เรย์ในฟังก์ชัน ให้ละเว้บขนาดข้อมูลในวงเล็บ และใช้ได้ทั้ง input , output
  void Read_Data (double data[], int *count);
  void Process_Data (double data[], int count);
- การเรียกใช้ฟังก์ชันที่ต้องส่งค่าอาร์เรย์ทั้งตัว ให้ใช้ชื่อตัวแปรอาร์เรย์โดยตรง ไม่ต้องกำหนดขนาด
  Read_Data (data, &count);
  Process_Data (data, count);

MATRIX

- MATRIX กับ ARRAY 2 มิติ
- การจองตัวแปร matrix ให้จองเป็นอาร์เรย์ 2 มิติของตัวเลข เช่น
  int Ma[10][10] ;
- ต้องกำหนดตัวแปร row และ col เพื่อระบุขอบเขตที่ใช้งานจริง
  int rowa, cola ;
- ใช้การวนรอบซ้อน 2 ชั้น เพื่อกระทำกับข้อมูลทีละตัวจนครบ เช่น
  for (i=1; i<= rowa; i++)
  { รอบการประมวลผลของ i ทีละแถว
  for (j = 1; j<= cola ; j++)
  {รอบของการประมวลผล j ทีละหลัก จนครบทุกหลัก
  ประมวลผลข้อมูล Ma[i][j] }
  }
- การกำหนดค่าพารามิเตอร์ที่เป็นอาร์เรย์ มากกว่า 1 มิติในฟังก์ชัน ให้ละเว้นขนาดข้อมูลในวงเล็บช่องแรก และใช้ได้ทั้ง input , output
  void Process_Matrix (int M[][10], int row, int col)
- การเรียกใช้ฟังก์ชันที่ต้องส่งค่าอาร์เรย์ ให้ใช้ชื่อตัวแปรอาร์เรย์ โดยไม่ต้องกำหนดขนาด
  Process_Matrix (Ma, rowa, cola);

Text File

- ภาษาซี ถ้าต้องการใช้ TEXT File ต้องทำตามขั้นตอนดังนี้
  - จองตัวแปรสำหรับใช้เป็นไฟล์
    - FILE *fp ; /* fp คือชื่อตัวแปรไฟล์ที่ต้องการนำไปใช้ */
  - คำสั่งกำหนดชื่อไฟล์ในดิสก์ แล้วเปิดไฟล์
    - fp = fopen("filename", "[r,w]") ;
    /* filename คือชื่อไฟล์ที่อยู่ในดิสก์ที่ต้องการใช้งาน */
    /* "r" เมื่อต้องการอ่านไฟล์ที่มีอยู่แล้ว */
    /* "w" เมื่อต้องการสร้างสร้างไฟล์ใหม่ */
  - ใช้คำสั่ง fprintf(fp,"...",...) และ fscanf(fp,"...", ...) แทนการอ่านและเขียนลงไฟล์
    - วนรอบในเขียนข้อมูลที่อยู่ในอาร์เรย์ เก็บลงไปในไฟล์
      for (i=1; i<=count; i++) /* วนรอบอาร์เรย์ตั้งแต่ตัวแรกถึงตัวสุดท้าย */
      { นำข้อมูลที่อยู่ในอาร์เรย์ตำแหน่งที่ i เขียนลงในไฟล์
      fprinf(fp, "....", ...) ; }
  - ใช้วนรอบในการอ่านข้อมูลจากไฟล์ เก็บไว้ในอาร์เรย์
    while (fscanf(fp, "....", ...) == 1) /*อ่านไปเรื่อยๆ ขณะที่ยังอ่านได้*/
    { เพิ่มค่าตัวนับ
    นำข้อมูลที่อ่านได้ไปเก็บในอาร์เรย์ที่กำหนดโดยตัวนับ }
  - คำสั่งปิดไฟล์
    - fclose(fp);

STRUCTURE

- ชนิดข้อมูลโครงสร้าง ใช้เมื่อตัวแปร แต่ละตัวประกอบด้วยคุณสมบัติ(ตัวแปร) ย่อยๆ หลายๆตัวมารวมกัน มักใช้ร่วมกับอาร์เรย์ 1 มิติ เช่น
  - โครงสร้างข้อมูลสมุดโทรศัพท์ ประกอบด้วย ชื่อ เบอร์โทร
- กำหนดชื่อโครงสร้าง ไว้ในส่วนที่ถัดจาก #include
- struct ชื่อโครงสร้าง { .....ชื่อตัวแปร(ฟิลด์)ที่นำมารวมกัน..... }
  - struct phonebook { char name[20] ;
    long int tel ; } ;
- โครงสร้างใหม่ที่เกิดขึ้นชื่อว่า struct phonebook สามารถนำไปใช้ได้ในทุกฟังก์ชัน
- จองตัวแปรสำหรับเก็บข้อมูลโครงสร้าง ในฟังก์ชัน main
  - struct phonebook x, phone[100];
    จองตัวแปร x (1 ตัว) และ phone(ARRAY 100 ตัว)
- จองตัวแปรสำหรับนับ และเก็บจำนวนข้อมูลที่มีอยู่จริง
  - int i, count ; ใช้ count เป็นตัวนับข้อมูลที่มีอยู่จริง และ i เป็นตัววนรอบทำงาน
- การส่งค่าตัวแปรแบบโครงสร้างเป็นพารามิเตอร์ ใช้เช่นเดียวกับอาร์เรย์
  - void Process_Data(struct phonebook phone[], int count)
- การเรียกใช้งานตัวแปรแบบโครงสร้าง ให้ใช้ ชื่อตัวแปร.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}