



Lab 08:

C 113 Lab 8 read-only

01 กำหนดโครงสร้างข้อมูล struct

Recent submission on 19:37 28 Feb 2020 Result: PASSED [PPPPPP]

Current

Recent

All

01 - กำหนดโครงสร้างข้อมูล struct

02 - ค้นหาและใส่คะแนน

03 - หาค่า cross product ระหว่าง 2 เวกเตอร์

04 - คำนวณเกรด โดยใช้ struct และ array

05 - หาคะแนนสูงสุดโดยใช้ struct และ array

06 - หาค่าผลบวกและผลคูณระหว่างจำนวนเชิงซ้อน 2 จำนวน

กำหนดโครงสร้างข้อมูล struct

ให้กำหนดโครงสร้างข้อมูล struct ที่ใช้เก็บข้อมูลเกี่ยวกับโทรศัพท์มือถือ ซึ่งมีชื่อฟิลด์และรายละเอียดข้อมูลดังต่อไปนี้:

- name - ชื่อเจ้าของโทรศัพท์ ซึ่งมีชนิดข้อมูลเป็นอระเรย์ของตัวอักษรจำนวนไม่เกิน 40 ตัว
- brand - ยี่ห้อของโทรศัพท์ ซึ่งมีชนิดข้อมูลเป็นอระเรย์ของตัวอักษรจำนวนไม่เกิน 20 ตัว
- calls - จำนวนครั้งที่เคยโทรออก
- price - ราคาของโทรศัพท์ที่ยังไม่รวมภาษีมูลค่าเพิ่ม
- vat - ภาษีมูลค่าเพิ่ม

ให้กำหนดโครงสร้างนี้ในบริเวณที่เว้นไว้ให้

```
struct cellphone {
```

```
};
```

[hide line #]

- 01 - กำหนดโครงสร้างข้อมูล struct
- 02 - ค้นหาและใส่คะแนน
- 03 - หาค่า cross product ระหว่าง 2 เวกเตอร์
- 04 - คำนวนเกรด โดยใช้ struct และ array
- 05 - หาคะแนนสูงสุดโดยใช้ struct และ array
- 06 - หาค่าผลบวกและผลคูณระหว่างจำนวนเชิงซ้อน 2 จำนวน

CurrentRecentAll

ค้นหาและใส่คะแนน

ให้นักนิสิตสร้าง struct ของ studentRecord ซึ่งมี data member 2 ตัว คือ id ซึ่งเป็น string ขนาด 12 ตัวอักษรเก็บรหัสนิสิต และ score ซึ่งเป็นเลขจำนวนเต็มเก็บคะแนนของนิสิต

จากนั้นให้เขียนฟังก์ชันชื่อ init() ซึ่งจะใส่ข้อมูลของนิสิต 3 คน โดยใส่รหัสนิสิตเป็น “55100001”, “55100002”, และ “55100003” ตามลำดับ และใส่คะแนนของนิสิตเป็น 0 ไว้ก่อน

จากนั้นให้เขียนฟังก์ชันชื่อ enterScore() ซึ่งจะรับข้อมูลจากผู้ใช้มาสองค่า คือรหัสกับคะแนน (จาก standard input) เป็นจำนวน size บรรทัด คำนรหัสนิสิตและคะแนนด้วยช่องว่าง

ฟังก์ชันนี้จะต้องนำคะแนนไปใส่ใน struct ของรหัสนิสิตให้ถูกต้อง

ผู้ใช้สามารถใส่รหัสนิสิตแค่บางส่วนได้ เช่นถ้าจะกรอกคะแนนของนิสิตรหัส 55100001 นั้นผู้ใช้สามารถพิมพ์แค่ 01 หรือ 001 หรือ 0001 หรือ 00001 ตามด้วยคะแนน ก็สามารถกรอกคะแนนได้ กำหนดว่ารหัสบางส่วนที่กรอกเข้ามาจะไม่คลุมเครือ (นั่นคือไม่ไปตรงกับบันทึกของนิสิตมากกว่า 1 คน) แต่อาจจะไม่มีนิสิตที่มีรหัสนั้น

นิสิตสามารถใช้ฟังก์ชันที่อยู่ใน string.h ช่วยในการเขียนโปรแกรมได้

ตัวอย่าง 1

01	50	
ID	Score	
55100001	50	
55100002	0	
55100003	0	

ตัวอย่าง 2

002	75	
ID	Score	
55100001	0	
55100002	75	
55100003	0	

หมายเหตุ: นิสิตสามารถเขียนฟังก์ชัน printRecords() ไว้ทดสอบเองได้ ตอนตรวจอาจารย์จะมีฟังก์ชัน printRecords() ของอาจารย์เอง

ดังนั้นนิสิตจึงไม่จำเป็นต้องระวังเรื่องรูปแบบการแสดงผลของ printRecords() แต่นิสิตต้องสร้าง struct ให้ถูกต้องตามข้อกำหนดไม่อย่างนั้นฟังก์ชัน printRecords() ของอาจารย์จะใช้ struct ของนิสิตไม่ได้

การตรวจจะทำเป็นส่วน คือจะตรวจฟังก์ชัน init() ก่อน ถ้าถูกต้องก็จะให้คะแนนส่วนหนึ่ง จากนั้นค่อยตรวจฟังก์ชัน enterScore() ดังนั้นนิสิตควรพยายามเขียนฟังก์ชัน init() ให้ได้เป็นอย่างน้อย

[hide line #]

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

typedef struct {
} studentRecord;

void init(studentRecord list[], int size) {

}

void enterScore(studentRecord list[], int size){

}

void printRecords(studentRecord list[], int size) {
    int i;

    printf("    ID          Score\n");
    for (i = 0; i < size; i++) {
        printf("%s          %d\n", list[i].id, list[i].score);
    }
}

int main() {
    int enter;
    studentRecord list[3];

    init(list, 3);
    enterScore(list, 3);
    printRecords(list, 3);
}
```

- 01 - กำหนดโครงสร้างข้อมูล struct ✔
- 02 - ค้นหาและใส่คะแนน ✔
- 03 - หาค่า cross product ระหว่าง 2 เวกเตอร์ ✔
- 04 - คำนวณเกรด โดยใช้ struct และ array ✔
- 05 - หาคะแนนสูงสุดโดยใช้ struct และ array ✔
- 06 - หาค่าผลบวกและผลคูณระหว่างจำนวนเชิงซ้อน 2 จำนวน ✔

03 หาค่า cross product ระหว่าง 2 เวกเตอร์

Recent submission on 20:12 28 Feb 2020 Result: PASSED [PPPPP]

CurrentRecentAll

ถ้ากำหนดให้เวกเตอร์ $\vec{u} = (u_x, u_y, u_z)$ และเวกเตอร์ $\vec{v} = (v_x, v_y, v_z)$, ค่า cross product ระหว่างเวกเตอร์ \vec{u} และเวกเตอร์ \vec{v} , ซึ่งสามารถเขียนอยู่ในรูปของ $\vec{u} \times \vec{v}$, จะมีค่าเป็นเวกเตอร์ซึ่งมีค่าเท่ากับ:

$$\vec{u} \times \vec{v} = (u_y v_z - u_z v_y, u_z v_x - u_x v_z, u_x v_y - u_y v_x)$$

จงเติมโปรแกรมข้างล่างให้สมบูรณ์เพื่อคำนวณหาค่า cross product ระหว่าง 2 เวกเตอร์ที่รับค่าเข้ามา

ตัวอย่างผลลัพธ์ที่ 1

Enter u: 1 2 3
Enter v: 2 3 4
u x v = -1.0 2.0 -1.0

ตัวอย่างผลลัพธ์ที่ 2

Enter u: 1.0 2.5 3.5
Enter v: 0.0 0.0 1.0
u x v = 2.5 -1.0 0.0

```
#include <stdio.h>

struct vector {
    
};

int main()
{
    

    return 0;
}
```

[hide line #]

01 - กำหนดโครงสร้างข้อมูล struct

02 - ค้นหาและใส่คะแนน

03 - หาค่า cross product ระหว่าง 2 เวกเตอร์

04 - คำนวนเกรด โดยใช้ struct และ array

05 - หาคะแนนสูงสุดโดยใช้ struct และ array

06 - หาค่าผลบวกและผลคูณระหว่างจำนวนเชิงซ้อน 2 จำนวน

Current

Recent

All

คำนวณเกรดโดยเก็บข้อมูลใน struct

จงเขียน โปรแกรมคำนวณเกรด โดยให้นิยาม struct ที่ใช้เก็บคะแนนสอบของนิสิต และสร้าง array ของ struct นี้ขึ้นมาโดยมีขนาด 5 elements รับข้อมูลเข้าจากผู้ใช้ โดยจะมีข้อมูล ID ของนิสิต คะแนนสอบมิดเทอมและไฟนอล จากนั้นให้เรียกฟังก์ชัน calculateGrade ซึ่งจะคำนวณเกรดให้โดยมีหลักการดังนี้

- ถ้าคะแนนเท่ากับ 80 คะแนนหรือมากกว่า นิสิตจะได้เกรด A
- ถ้าคะแนนเท่ากับ 70 ถึง 79 คะแนนจะได้เกรด B
- ถ้าคะแนนเท่ากับ 60 ถึง 69 คะแนนจะได้เกรด C
- ถ้าคะแนนเท่ากับ 50 ถึง 59 คะแนนจะได้เกรด D
- ถ้าคะแนนต่ำกว่านั้นจะได้เกรด F

จากนั้นให้พิมพ์ ID และเกรดของนิสิตที่ละคนเรียงตามลำดับข้อมูลที่นำเข้า

ตัวอย่างผลลัพธ์ที่ 1

```
Enter Student ID: 5610456784
Enter Student Midterm: 26
Enter Student Final: 37
Enter Student ID: 5610469821
Enter Student Midterm: 40
Enter Student Final: 37
Enter Student ID: 5610457893
Enter Student Midterm: 10
Enter Student Final: 6
Enter Student ID: 5610475947
Enter Student Midterm: 26
Enter Student Final: 35
Enter Student ID: 5610438273
Enter Student Midterm: 30
Enter Student Final: 25
Student ID 5610456784 receives grade C.
Student ID 5610469821 receives grade B.
Student ID 5610457893 receives grade F.
Student ID 5610475947 receives grade C.
Student ID 5610438273 receives grade D.
```

- 01 - กำหนดโครงสร้างข้อมูล struct
- 02 - ค้นหาและใส่คะแนน
- 03 - หาค่า cross product ระหว่าง 2 เวกเตอร์
- 04 - คำนวนเกรด โดยใช้ struct และ array
- 05 - หาคะแนนสูงสุดโดยใช้ struct และ array
- 06 - หาค่าผลบวกและผลคูณระหว่างจำนวนเชิงซ้อน 2 จำนวน

05 หาคะแนนสูงสุดโดยใช้ struct และ array

Recent submission on 11:52 04 Mar 2020 Result: PASSED [PPP]

CurrentRecentAll

หาค่าสูงสุดของคะแนนโดยเก็บข้อมูลใน struct

จงเขียนโปรแกรมหาค่าสูงสุด โดยให้นิยาม data type ใหม่ เป็นประเภท struct และสร้าง array ของ struct นี้ขึ้นมาโดยมีขนาด 5 elements รับข้อมูลเข้าจากผู้ใช้ โดยจะมีข้อมูล ID ของนิสิต คะแนนสอบมิตเทอมและไฟนอล จากนั้นให้เรียกฟังก์ชัน findTop ซึ่งจะหา record ที่มีคะแนนมิตเทอมสูงสุด และ record ที่มีคะแนนไฟนอลสูงสุด

จากนั้นให้พิมพ์ ID และคะแนนของนิสิตที่ได้คะแนนสูงสุดทั้งสอง ถ้านิสิตสองคนมีคะแนนเท่ากัน ให้ย้ดนิสิตคนที่ผู้ใช้ใส่ข้อมูลก่อนว่าเป็นผู้ที่มีคะแนนสูงสุด

ตัวอย่างผลลัพธ์ที่ 1

Enter Student ID: 501456784
Enter Student Midterm: 26
Enter Student Final: 37
Enter Student ID: 501769821
Enter Student Midterm: 40
Enter Student Final: 37
Enter Student ID: 501567893
Enter Student Midterm: 10
Enter Student Final: 6
Enter Student ID: 501875947
Enter Student Midterm: 26
Enter Student Final: 35
Enter Student ID: 519838273
Enter Student Midterm: 30
Enter Student Final: 25
Top Score for Midterm:
Student ID 501769821 with score 40
Top Score for Final:
Student ID 501456784 with score 37

[hide line #]

```
#include <stdio.h>

#define ARRAY_SIZE 5
struct StudentRecord {
    char id[12];
    int midterm;
    int final;
    char grade;
}

void findTop(StudentRecord pRec[], int size, StudentRecord **topMid, StudentRecord **topFin) {

}

int main() {

    StudentRecord students[ARRAY_SIZE], *pTopMid, *pTopFin;
    int gradeCount[5] = {0};
    int i;

    for (i=0; i < ARRAY_SIZE; i++) {
        printf("Enter Student ID: ");
        scanf("%s", students[i].id);
        printf("Enter Student Midterm: ");
        scanf("%d", &students[i].midterm);
        printf("Enter Student Final: ");
        scanf("%d", &students[i].final);

    }

    findTop(students, ARRAY_SIZE, &pTopMid, &pTopFin);

    //print grade
    printf("Top Score for Midterm:\n");
    printf("Student ID %s with score %d\n", students[pTopMid->id], students[pTopMid->midterm]);
    printf("Top Score for Final:\n");
    printf("Student ID %s with score %d\n", students[pTopFin->id], students[pTopFin->final]);

    return 0;
}
```

Icons from Sweetie.

- 01 - กำหนดโครงสร้างข้อมูล struct
- 02 - ค้นหาและใส่คะแนน
- 03 - หาค่า cross product ระหว่าง 2 เวกเตอร์
- 04 - คำนวณเกรด โดยใช้ struct และ array
- 05 - หาคะแนนสูงสุดโดยใช้ struct และ array
- 06 - หาค่าผลบวกและผลคูณระหว่างจำนวนเชิงซ้อน 2 จำนวน

Current

Recent

All

ถ้ากำหนดให้ C เป็นจำนวนเชิงซ้อนแล้ว C สามารถถูกเขียนให้อยู่ในรูปของ $C = a + bi$ โดยที่ $i^2 = -1$ และหากกำหนดให้ $D = c + di$ แล้ว

- ผลบวกระหว่าง C และ D จะมีค่าเท่ากับ $(a + c) + (b + d)i$
- ผลคูณระหว่าง C และ D จะมีค่าเท่ากับ $(ac - bd) + (ad + bc)i$

จงเติมโปรแกรมข้างล่างให้สมบูรณ์เพื่อคำนวณหาค่าผลบวก (add) และผลคูณ (multiply) ของค่าจำนวนเชิงซ้อน 2 จำนวนที่รับค่าเข้ามา

ตัวอย่างผลลัพธ์ที่ 1 (Input: $C = 1.0 + 2.0i$ และ $D = 2 + 3i$)

Enter C: **1.0 2.0**

Enter D: **2 3**

C + D = 3.0 + 5.0i

C x D = -4.0 + 7.0i

ตัวอย่างผลลัพธ์ที่ 2 (Input: $C = 2i$ และ $D = 3i$)

Enter C: **0 2**

Enter D: **0 3**

C + D = 5.0i

C x D = -6.0

ตัวอย่างผลลัพธ์ที่ 3 (Input: $C = 1 + 2i$ และ $D = -3i$)

Enter C: **1 2**

Enter D: **0 -3**

C + D = 1.0 - 1.0i

C x D = 6.0 - 3.0i

#include <stdio.h>

struct complex {

};

add_complex()

{

}

multiply_complex()

{

}

void print_complex()

{

}

int main()

{

printf("Enter C: ");

scanf("%f %f", &C. , &C.);

E = add_complex(C, D);

F = multiply_complex(C, D);

print_complex("C + D = ", E);

print_complex("C x D = ", F);

return 0;

}

[hide line #]

Icons from Sweetie.