

# แบบฝึกหัด เรื่อง Structure

คะแนนรวม 10/10 ?

ขอให้นิสิตทำแบบฝึกหัดนี้ด้วยตนเอง จะได้เข้าใจเนื้อหาได้ถ่องแท้มากยิ่งขึ้น ---โดยนิสิตสามารถส่งคำตอบก็ครั้งก็ได้ตามต้องการ เพื่อแก้ไขข้อที่ผิดให้ถูกต้อง

เมื่อ submit แล้ว จะมีปุ่มให้ดูคะแนน (View score) ให้นิสิตกดเข้าไปดูคะแนน จะมีการแจ้งว่า นิสิตทำข้อใดถูกต้องและข้อใดผิด และจะมีอีเมลคำตอบส่งไปให้นิสิต เพื่อให้นิสิตสามารถแก้ไขคำตอบ (Edit your response) ได้ก็ครั้งก็ได้ตามต้องการ --- แต่ขอให้นิสิตทำแบบฝึกหัดด้วยตนเองนะคะ

ระบบบันทึกอีเมลของผู้ตอบ (thitichot.k@ku.th) ไว้แล้วเมื่อส่งแบบฟอร์มนี้

0 จาก 0 คะแนน

ชื่อ-สกุล \*

ฐิติโชติ กิ่งไพบูลย์

รหัสนิสิต \*

6610402001

structure เบื้องต้น

4 จาก 4 คะแนน



```
struct complex_number {  
    float real;  
    float imaginary;  
};
```

☐ Option 1

```
struct person {  
    char name[20];  
    int age;  
} steve_jobs, elon_musk;
```

☐ Option 2

```
struct {  
    int student_id;  
    char student_name[30];  
};
```

☒ Option 3 ✓

```
typedef struct {  
    char account_name[30];  
    float balance;  
} BankAccount;
```

☐ Option 4

#### ข้อเสนอแนะ

ตัวเลือก 3 นี้ไม่ถูกต้อง เนื่องจากประกาศ struct โดยไม่ใช้ typedef ซึ่งจะต้องมี tag ด้วย (tag เป็นเหมือนชื่อของ struct)

ส่วนตัวเลือกอื่น ๆ struct มี tag ครบถ้วน (ตัวเลือก 1, 2) หรือถ้าไม่มี tag ต้องมีการใช้ typedef ในการประกาศ เพื่อกำหนดประเภทข้อมูลใหม่ (ตัวเลือก 4)

ในตัวเลือก 2 เป็นการประกาศ/นิยาม struct ที่ถูกต้อง พร้อมกับประกาศตัวแปร struct ไปพร้อมกัน ส่วนตัวเลือกอื่น ๆ เป็นการประกาศ/นิยาม struct แต่ยังไม่ได้ประกาศตัวแปร

- ✓ การประกาศตัวแปร t1 ที่เป็น structure สามเหลี่ยมในข้อใดไม่ถูกต้อง (แม้ว่าจะคอม 1/1 ไฟล์ผ่านก็ตาม)

```
struct triangle {  
    int base;  
    int height;  
};  
  
struct triangle t1;
```

☐ Option 1

```
struct triangle {  
    int base;  
    int height;  
} t1;
```

☐ Option 2

```
typedef struct triangle {  
    int base;  
    int height;  
} Triangle;  
  
Triangle t1;
```

☐ Option 3

```
typedef struct triangle {  
    int base;  
    int height;  
} t1;
```

☒ Option 4



#### ข้อเสนอนะ

การประกาศในตัวเลือก 4 นี้ เป็นการประกาศ structure triangle ที่มี typedef ชื่อ t1 แต่ไม่ใช้การประกาศตัวแปร t1 ที่มีประเภทข้อมูลเป็น structure โดยหากรันการประกาศนี้ จะคอมไฟล์ผ่าน แต่ไม่มีการประกาศตัวแปร จึงไม่ตรงกับข้อกำหนดของโจทย์

ในตัวเลือก 1 เป็นการประกาศ structure แยกจากการประกาศตัวแปร t1 โดยการประกาศ type ของ t1 จะยาวหน่อย ใช้ 2 คำ คือ struct triangle

ในตัวเลือก 2 เป็นการประกาศ structure พร้อมกับการประกาศตัวแปร t1

ในตัวเลือก 3 เป็นการประกาศ structure และตั้งชื่อประเภทข้อมูลเป็น Triangle ก่อนนำชื่อประเภทข้อมูลมาประกาศตัวแปร



- ✓ พิจารณาการประกาศ structure สามเหลี่ยมด้านล่าง จงตอบว่าการกำหนดค่าให้ member ของตัวแปร t1 ที่เป็น structure ในข้อใดไม่ถูกต้อง 1/1

```
typedef struct triangle {  
    int base;  
    int height;  
} Triangle;
```

```
Triangle t1;  
t1.base = 10;  
t1.height = 15;
```

☐ Option 1

```
Triangle t1.base = 10;  
Triangle t1.height = 15;
```

☒ Option 2 ✓

```
struct triangle t1 = { 10, 15 };
```

☐ Option 3

```
Triangle t1;  
scanf("%d %d", &t1.base, &t1.height);
```

☐ Option 4

ข้อเสนอนะ

ตัวเลือก 2 ไม่ถูกต้อง

ตัวเลือก 1 ประกาศตัวแปร t1 แล้วใช้ dot operator ในการกำหนดค่า member

ตัวเลือก 3 ประกาศตัวแปร t1 กำหนดค่าเริ่มต้นให้ member ของ t1 เลย

ตัวเลือก 4 ประกาศตัวแปร t1 แล้วให้ผู้ใช้กำหนดค่า member ผ่านฟังก์ชัน scanf



```
1  #include <stdio.h>
2
3  struct complex_number {
4      float real;
5      float imaginary;
6  };
7
8  int main() {
9      struct complex_number n1 = { 0.8, 1.1 };
10     struct complex_number n2;
11     n2 = n1;
12
13     printf("n1 = %.1f + %.1fi\n", n1.real, n1.imaginary);
14     printf("n2 = %.1f + %.1fi\n", n2.real, n2.imaginary);
15
16     n1.real = 10;
17
18     printf("n1 = %.1f + %.1fi\n", n1.real, n1.imaginary);
19     printf("n2 = %.1f + %.1fi\n", n2.real, n2.imaginary);
20
21 }
```

```
n1 = 0.8 + 1.1i
n2 = 0.8 + 1.1i
n1 = 10.0 + 1.1i
n2 = 0.8 + 1.1i
```

- โค้ดบรรทัดที่ 11 เป็นการเชื่อมโยงตัวแปร n1 กับ n2 ด้วย structure อันเดียวกัน เมื่อพิมพ์ในบรรทัดที่ 13-14 ค่าของ n1 กับ n2 จึงเท่ากัน แต่เมื่อมีการเปลี่ยนค่า n1 ในบรรทัดที่ 16 ค่า n1 จึงเปลี่ยนไป ขณะที่ n2 ยังคงเดิม เพราะ n1, n2 เป็น structure เดียวกัน

```
n1 = 0.8 + 1.1i
n2 = 0.8 + 1.1i
n1 = 10.0 + 1.1i
n2 = 0.8 + 1.1i
```

- ◎ โค้ดบรรทัดที่ 11 เป็นการ copy ค่า ✓ จาก n1 มาใส่ n2 คือ เป็น structure คนละอันกัน โดยมีค่า member เหมือนกัน เมื่อพิมพ์ในบรรทัดที่ 13-14 ค่าของ n1 กับ n2 จึงเท่ากัน แต่เมื่อมีการเปลี่ยนค่า n1 ในบรรทัดที่ 16 ค่า n1 จึงเปลี่ยนไปขณะที่ n2 ยังคงเดิม เพราะ n1, n2 เป็นคนละ structure กัน

```

n1 = 0.8 + 1.1i
n2 = 0.8 + 1.1i
n1 = 10.0 + 1.1i
n2 = 10.0 + 1.1i

```

n1,n2 เป็น structure เดียวกัน

```

n1 = 0.8 + 1.1i
n2 = 0.8 + 1.1i
n1 = 10.0 + 1.1i
n2 = 10.0 + 1.1i

```

บรรทัด 16 ค่า n2 จึงเปลี่ยนตาม n1 เพราะ n1,n2 เป็นคนละ structure กัน

ข้อเสนอแนะ

ต้องอ่านคำอธิบายดี ๆ ค่ะ

ตัวเลือก 2 มีผลการทำงานถูกต้อง และอธิบายเหตุผลถูกต้อง ตัวเลือกอื่นเป็นตัวเลือกหลอก

- ✓ พิจารณาการประกาศ structure บัญชีธนาคารต่อไปนี้ จงตอบว่า การเขียนฟังก์ชันที่ 1/1  
รับ structure เป็นพารามิเตอร์ และการเรียกใช้และส่ง structure ไปฟังก์ชันในข้อใด  
ถูกต้อง

```
typedef struct account {  
    int number;    // เลขที่บัญชีธนาคาร  
    float balance; // ยอดคงเหลือในบัญชี  
} BankAccount;
```

```
void show_account(struct account act) {  
    printf("บัญชี %d มียอดคงเหลือ %.2f\n",  
          act.number, act.balance);  
}  
  
int main() {  
    struct account my_act = {1, 100.0};  
    show_account(my_act);  
}
```

- ☐ ประกาศพารามิเตอร์เป็น struct  
account และส่งตัวแปรที่เป็น  
struct account เข้าไปในฟังก์ชัน

```
void show_account(BankAccount act) {  
    printf("บัญชี %d มียอดคงเหลือ %.2f\n",  
          act.number, act.balance);  
}  
  
int main() {  
    BankAccount my_act = {1, 100.0};  
    show_account(my_act);  
}
```

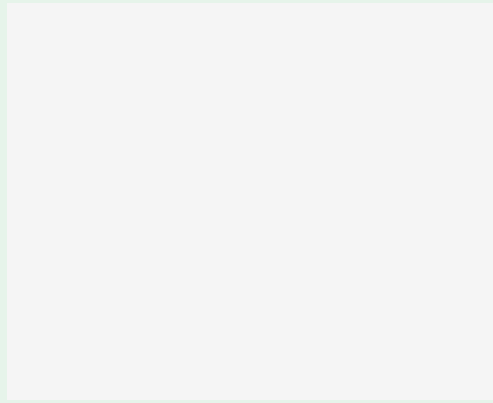
- ☐ ประกาศพารามิเตอร์เป็น  
BankAccount และส่งตัวแปรที่  
เป็น BankAccount เข้าไปใน  
ฟังก์ชัน

```
void show_account(struct account act) {  
    printf("บัญชี %d มียอดคงเหลือ %.2f\n",  
          act.number, act.balance);  
}  
  
int main() {  
    BankAccount my_act = {1, 100.0};  
    show_account(my_act);  
}
```

- ☐ ประกาศพารามิเตอร์เป็น struct  
account แต่ส่งตัวแปรที่เป็น  
BankAccount เข้าไปในฟังก์ชัน

```
void show_account(BankAccount act) {  
    printf("บัญชี %d มียอดคงเหลือ %.2f\n",  
          act.number, act.balance);  
}  
  
int main() {  
    struct account my_act = {1, 100.0};  
    show_account(my_act);  
}
```

- ☐ ประกาศพารามิเตอร์เป็น  
BankAccount แต่ส่งตัวแปรที่เป็น  
struct account เข้าไปในฟังก์ชัน



ถูกทุกข้อ



#### ข้อเสนอแนะ

เราสามารถรับ structure เข้าไปในฟังก์ชันได้ โดยประกาศ type ของพารามิเตอร์ให้เป็น structure โดยใช้ได้ 2 รูปแบบ คือ แบบไม่ใช้ typedef และแบบใช้ typedef

การส่ง structure เข้าไปในฟังก์ชันได้ ต้องส่งตัวแปรที่เป็น structure ประเภทนั้น โดยประกาศตัวแปรได้ 2 รูปแบบ คือ แบบไม่ใช้ typedef และแบบใช้ typedef





- ✓ พิจารณาโปรแกรมต่อไปนี้ ซึ่งประกาศ structure บัญชีธนาคาร และมีฟังก์ชันการฝากเงิน ข้อใดตอบผลการรันและเหตุผลที่ถูกต้อง 1/1

```
1  #include <stdio.h>
2
3  typedef struct account {
4      int number;    // เลขที่บัญชีธนาคาร
5      float balance; // ยอดคงเหลือในบัญชี
6  } BankAccount;
7
8  // ฟังก์ชันการฝากเงินในบัญชีธนาคาร
9  // โดยจำนวนเงินฝากอยู่ในพารามิเตอร์ amount
10 void deposit(struct account act, float amount) {
11     act.balance = act.balance + amount;
12 }
13
14 int main() {
15
16     BankAccount my_act = {1, 100.0};
17     deposit(my_act, 50);
18
19     printf("%.2f\n", my_act.balance);
20 }
```

- ☒ 100.00 เนื่องจากการส่ง struct ไปฟังก์ชันเป็นการส่งแบบ pass by value คือ เป็นการ copy ค่า structure จาก my\_act ไปที่พารามิเตอร์ act ในฟังก์ชัน ทำให้ค่าใน my\_act ไม่เปลี่ยนแปลงเมื่อเรียกฟังก์ชัน ✓
- ☐ 100.00 เนื่องจากการส่ง struct ไปฟังก์ชันเป็นการส่งแบบ pass by reference คือ การส่งที่อยู่ของ struct my\_act ไปที่ฟังก์ชัน ทำให้ my\_act และ act เป็น structure อันเดียวกัน เมื่อค่า act เปลี่ยน ค่า my\_act จึงเปลี่ยนตาม
- ☐ 150.00 เนื่องจากการส่ง struct ไปฟังก์ชันเป็นการส่งแบบ pass by reference คือ การส่งที่อยู่ของ struct my\_act ไปที่ฟังก์ชัน ทำให้ my\_act และ act เป็น structure อันเดียวกัน เมื่อค่า act เปลี่ยน ค่า my\_act จึงเปลี่ยนตาม
- ☐ 150.00 เนื่องจากการส่ง struct ไปฟังก์ชันเป็นการส่งแบบ pass by value คือ เป็นการ copy ค่า structure จาก my\_act ไปที่พารามิเตอร์ act ในฟังก์ชัน ทำให้ค่าใน my\_act ไม่เปลี่ยนแปลงเมื่อเรียกฟังก์ชัน

ข้อเสนอแนะ

ต้องอ่านคำอธิบายดี ๆ ค่ะ

ตัวเลือก 1 มีผลการทำงานถูกต้อง และอธิบายเหตุผลถูกต้อง ตัวเลือกอื่นเป็นตัวเลือกหลอก



- ✓ พิจารณาโปรแกรมต่อไปนี้ ซึ่งประกาศ structure person โดยในฟังก์ชัน main() มีการเรียกใช้ฟังก์ชัน sizeof() ให้พิมพ์ขนาดของ structure person ข้อใดตอบผลการรันและเหตุผลที่ถูกต้อง

```
struct person {  
    int id;  
    char name[15];  
    float height;  
    double weight;  
};  
  
int main() {  
    int size = sizeof(struct person);  
    printf("%d\n", size);  
}
```

- ☐ 27 เนื่องจากจำนวน byte ของแต่ละ member บวกกันเท่ากับ 27
- ☐ 31 เนื่องจากจำนวน byte ของแต่ละ member บวกกันเท่ากับ 31
- ☒ 32 แม้ว่าจำนวน byte ของแต่ละ member บวกกันเท่ากับ 31 แต่มีการเพิ่ม internal padding ให้มีจำนวนครบเต็ม word ของเครื่องคอมพิวเตอร์ โดย 1 word อาจเป็น 4 หรือ 8 ตามแต่ละเครื่อง ✓
- ☐ 32 ต้องปัดขึ้นไปจำนวน 1 byte เสมอ เนื่องจากต้องเก็บชื่อ struct ด้วย
- ☐ 35 เนื่องจากจำนวน byte ของแต่ละ member บวกกันเท่ากับ 35

- ✓ พิจารณาโค้ดต่อไปนี้ซึ่งประกาศ structure ซ้อนภายใน structure คือ ใน structure 1/1 customer (ลูกค้า) มี structure account (บัญชีธนาคาร) ซ่อนอยู่ภายใน ข้อใดกำหนดค่าให้ member ต่าง ๆ ของ structure เหล่านี้ไม่ถูกต้อง

```
struct account {  
    int number;  
    float balance;  
};  
  
struct customer {  
    int national_id;  
    struct account account;  
};
```

```
struct customer mark = {  
    99102, { 1, 100.00 }  
};
```

- ☐ กำหนดค่า member ผ่านการใช้ {} เพื่อกำหนดค่าเริ่มต้น

```
struct customer bill;  
bill.national_id = 82375;  
bill.account.number = 2;  
bill.account.balance = 800.00;
```

- ☐ กำหนดค่า member ทีละตัว โดยกำหนดค่าบัญชีผ่าน dot operator จากตัวแปร bill.account

```
struct customer donald;  
donald.national_id = 14513;  
donald.account = { 3, 500.00 };
```

- ☒ กำหนดค่า member ทีละตัว โดยกำหนดค่าบัญชีแบบค่าเริ่มต้น {} ✓

```
struct account julia_act = { 4, 200.00 };  
struct customer julia;  
julia.national_id = 26681;  
julia.account = julia_act;
```

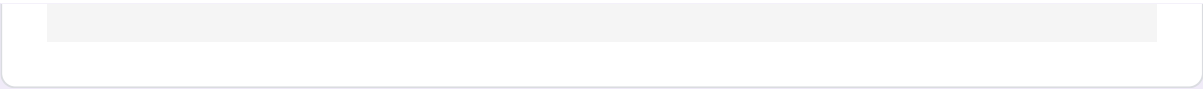
- ☐ ประกาศตัวแปรบัญชีธนาคาร ก่อนนำไปใช้ในการกำหนดค่า customer

ข้อเสนอนะ

การกำหนดค่าเริ่มต้นแบบ {} ต้องทำตั้งแต่ตอนประกาศตัวแปร มากำหนดทีหลังไม่ได้

ข้ออื่น ๆ กำหนดค่า member ได้ถูกต้อง นิสิตสามารถเลือกกำหนดได้ตามความเหมาะสม





- ✓ พิจารณาการประกาศ structure ต่อไปนี้ ข้อใดประกาศอาเรย์ของ structure และ กำหนดค่าให้ structure ในอาเรย์ไม่ถูกต้อง 1/1

```
struct account {  
    int number;  
    float balance;  
};
```

```
struct account act_list[3] = {  
    { 1, 100.00 },  
    { 2, 200.00 },  
    { 3, 300.00 }  
};
```

☐ Option 1

```
struct account act_list[3];  
act_list[0].number = 1;  
act_list[0].balance = 100.00;  
act_list[1].number = 2;  
act_list[1].balance = 200.00;  
act_list[2].number = 3;  
act_list[2].balance = 300.00;
```

☐ Option 2

```
struct account a1 = { 1, 100.0 };  
struct account a2 = { 2, 200.0 };  
struct account a3 = { 3, 300.0 };  
struct account act_list[3] = { a1, a2, a3 };
```

☐ Option 3

```
struct account a1 = { 1, 100.0 };  
struct account a2 = { 2, 200.0 };  
struct account a3 = { 3, 300.0 };  
struct account act_list[3];  
act_list[0] = a1;  
act_list[1] = a2;  
act_list[2] = a3;
```

☐ Option 4

```
struct account act_list[3] = {  
    act_list[0], act_list[1], act_list[2]  
};
```

☒ Option 5



ข้อเสนอแนะ

ตัวเลือก 5 นี้คอมไพล์ไม่ผ่าน และยังไม่ได้กำหนดใด ๆ เลย

ตัวเลือก 1 - 4 เป็นการประกาศและกำหนดค่าถูกต้อง ซึ่งเราทำได้หลากหลายรูปแบบ ลองเขียนโปรแกรมตามดูนะคะ

✓ จากการประกาศอาเรย์ของ structure ในข้อก่อนหน้านี้ ข้อใดเรียกใช้ตัวแปร balance ภายใน structure ที่อยู่ในอาเรย์ที่ index 1 ได้ถูกต้อง

1/1

```
printf("%f\n", act_list.balance);
```

☐ Option 1

```
printf("%f\n", act_list[1]->balance);
```

☐ Option 2

```
printf("%f\n", balance);
```

☐ Option 3

```
printf("%f\n", act_list[1].balance);
```

☒ Option 4



ข้อเสนอแนะ

เราต้องใช้[] ในการระบุ index ของอาเรย์ จากนั้น จึงใช้ dot operator ในการเข้าถึง member ของ structure ภายในอาเรย์

แบบฟอร์มนี้ถูกสร้างขึ้นภายใน มหาวิทยาลัยเกษตรศาสตร์

Google ฟอร์ม



