

แผนการสอนประจำบทที่ 9

โครงสร้างและยูเนียน

หัวข้อสำคัญ

1. ความแตกต่างระหว่างข้อมูลอาร์เรย์และโครงสร้างชนิดของอาร์เรย์
2. ชนิดข้อมูลแบบโครงสร้าง (Structure) และการจองพื้นที่ในหน่วยความจำ
3. ชนิดข้อมูลแบบยูเนียน (Unions) และการจองพื้นที่ในหน่วยความจำ
4. การใช้งานตัวแปรชนิดโครงสร้าง
5. อาร์เรย์ของโครงสร้าง

วัตถุประสงค์เชิงพฤติกรรม

1. ผู้เรียนสามารถอธิบายชนิดข้อมูลแบบโครงสร้างได้
2. ผู้เรียนสามารถบอกความแตกต่างระหว่างข้อมูลอาร์เรย์และโครงสร้างชนิดของอาร์เรย์ได้
3. ผู้เรียนสามารถเขียนโปรแกรมจองพื้นที่หน่วยความจำและใช้งานตัวแปรชนิดยูเนียนได้

วิธีการสอนและกิจกรรมการเรียนรู้การสอน

1. การบรรยาย
2. การทำแบบฝึกหัด

สื่อที่ใช้ประกอบการสอน

1. เอกสารประกอบการสอน
2. เครื่องคอมพิวเตอร์

การวัดและประเมินผล

1. สังเกตจากความสนใจของผู้เรียน
2. ประเมินจากการตอบคำถามของผู้เรียนและกิจกรรมในชั้นเรียน
3. การทำแบบฝึกหัดท้ายบท

บทที่ 9

โครงสร้างและยูเนียน

9.1. ความแตกต่างระหว่างชนิดข้อมูลอาร์เรย์และโครงสร้าง

อาร์เรย์ (array) เก็บข้อมูลจำนวนมากๆได้ เข้าถึงข้อมูลแต่ละตัวได้โดยง่าย ข้อจำกัด : ข้อมูลเหล่านั้นต้องเป็นชนิดเดียวกันเท่านั้น

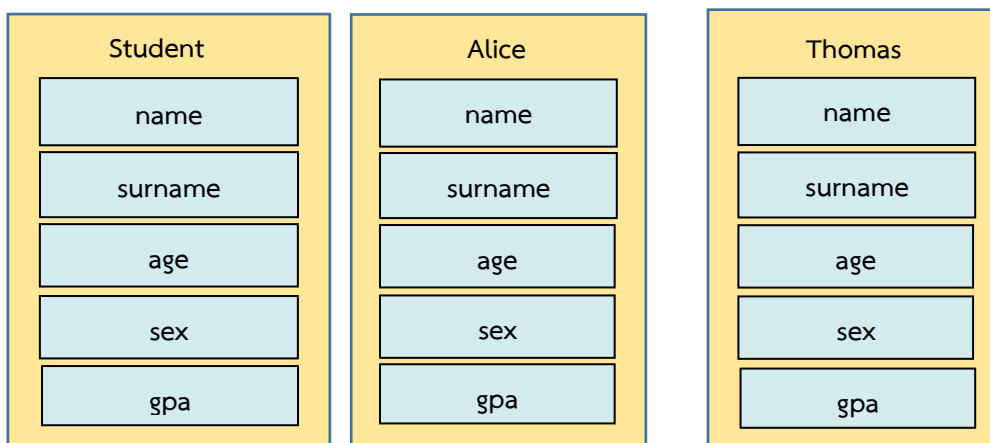
```
int point[5]
```

ข้อมูลแบบโครงสร้าง (Structure) ข้อมูลแต่ละตัวในกลุ่มสามารถมีชนิดต่างกันได้ เช่น ข้อมูลผลการเรียนของนิสิตแต่ละคน เทียบเท่ากับระเบียน (Record) ในภาษาอื่น

```
struct student {  
    char stdName[20];  
    int age;  
    float grade;  
}
```

9.2 ชนิดข้อมูลแบบโครงสร้าง (Structure)

ตัวอย่างการข้อมูลที่สามารถใช้การประกาศตัวแปรแบบโครงสร้าง ดังภาพ 9.1



ภาพ 9.1 ตัวอย่างการข้อมูลที่สามารถใช้การประกาศตัวแปรโครงสร้าง

9.3 ชนิดข้อมูลแบบโครงสร้าง (Structure)

โครงสร้าง (Structure) คือ ข้อมูลที่นำเอาข้อมูลที่มีประเภทข้อมูลชนิดพื้นฐานที่แตกต่างกันแต่มีความสัมพันธ์กันมารวมกันเป็นโครงสร้างข้อมูลชนิดใหม่ เช่น การเก็บข้อมูลนิสิต (student) จะต้องประกอบไปด้วยข้อมูลรหัสนิสิต, ชื่อ (name), นามสกุล (surname), สาขาวิชา (major), คณะ (faculty) ตัวแปรโครงสร้างและการประกาศชนิดข้อมูลของสมาชิกภายในโครงสร้างแสดงดังภาพ 9.2

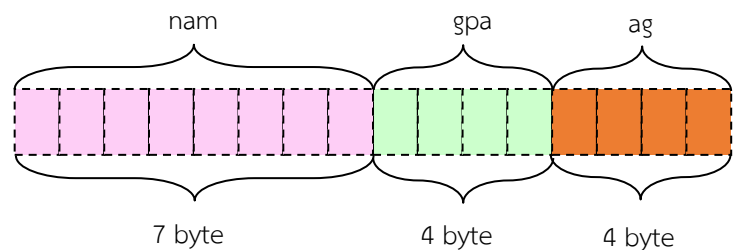
student0	student1	student2
char name[20];	char name[20];	char name[20];
char surname[30]	char surname[30]	char surname[30]
float gpa;	float gpa;	float gpa;
int age;	int age;	int age;

ภาพ 9.2 ตัวอย่างการตัวแปรโครงสร้างและชนิดข้อมูลของสมาชิก

9.4 การจองพื้นที่ในหน่วยความจำ

ตัวแปรชนิดโครงสร้างจะจองขนาดพื้นที่ในหน่วยความจำขึ้นอยู่กับชนิดข้อมูลของสมาชิกภายในตัวแปรสร้าง ซึ่งอ้างอิงจากโปรแกรม DEV C version 5.11 มีข้อมูลชนิดจำนวนเต็มและจำนวนจริงจองพื้นที่ 4 ไบต์ และตัวแปรชนิดอักขระจองพื้นที่ 1 ไบต์ จากการประกาศตัวแปรชนิดโครงสร้างชื่อ new_student คอมไพเลอร์จะจองพื้นที่ในหน่วยความจำแสดงดังภาพ 9.3

```
struct new_student{
    char name[7];
    float gpa;
    int age;
}
```



ภาพ 9.3 จำลองการจองพื้นที่ในหน่วยความจำของตัวแปร new_student

9.5 การใช้งานตัวแปรชนิดโครงสร้าง

9.5.1 การกำหนดโครงสร้างตัวแปรชนิดโครงสร้าง

ก่อนจะสร้างตัวแปรชนิดโครงสร้างมาใช้งานได้นั้น ต้องกำหนดโครงสร้างข้อมูลสำหรับตัวแปรนั้นก่อน ดังนี้

รูปแบบการประกาศตัวแปร Pointer	ตัวอย่าง
รูปแบบที่ 1 struct ชื่อของโครงสร้างที่จะสร้าง { ชนิดข้อมูลพื้นฐาน ชื่อตัวแปร; ชนิดข้อมูลพื้นฐาน ชื่อตัวแปร; ... ชนิดข้อมูลพื้นฐาน ชื่อตัวแปร; };	<pre>struct student { int stu_id; char stu_name[20]; char stu_surname[20]; char stu_major[20]; char stu_faculty[20]; };</pre>
รูปแบบที่ 2 typedef struct { ชนิดข้อมูลพื้นฐาน ชื่อตัวแปร; ชนิดข้อมูลพื้นฐาน ชื่อตัวแปร; ... ชนิดข้อมูลพื้นฐาน ชื่อตัวแปร; } ชื่อของโครงสร้างที่จะสร้าง;	<pre>typedef struct { int stu_id; char stu_name[20]; char stu_surname[20]; char stu_major[20]; char stu_faculty[20]; } STUDENT;</pre>

9.5.2 การสร้างตัวแปรชนิดโครงสร้าง

เมื่อกำหนดโครงสร้างข้อมูลให้กับตัวแปรตามต้องการแล้ว ต่อไปคือขั้นตอนการสร้างตัวแปรจากโครงสร้างที่ได้กำหนดไว้ โดยมีรูปแบบดังนี้

กำหนดโครงสร้างในรูปแบบที่ 1 (**struct**)

struct ชนิดของข้อมูลชนิดโครงสร้าง ชื่อตัวแปร;

เช่น

struct student stu1, stu2;

กำหนดโครงสร้างในรูปแบบที่ 2 (**typedef struct**)

ชนิดของข้อมูลชนิดโครงสร้าง ชื่อตัวแปร;

เช่น

STUDENT stu1, stu2;

9.5.3 การกำหนดค่าเริ่มต้นให้กับตัวแปรชนิดโครงสร้าง

การกำหนดค่าเริ่มต้นให้ตัวแปรชนิดโครงสร้างมีรูปแบบคล้ายกับการการกำหนดค่าเริ่มต้นกับอาร์เรย์กำหนดค่าไว้ภายใน { } และแยกค่าของแต่ละสมาชิกด้วยเครื่องหมาย , (comma) เช่น

กำหนดโครงสร้างในรูปแบบที่ 1 (struct)

```
struct student stu = {101, "Alice", "Wonderland", "CS", "Science"};
```

กำหนดโครงสร้างในรูปแบบที่ 2 (typedef struct)

```
STUDENT stu = {101, "Thomas", "Edison", "IT", "Science"};
```

9.5.4 การอ้างถึงสมาชิกในตัวแปรชนิดโครงสร้าง

การอ้างถึงตัวแปรหรือสมาชิกแต่ละตัวในตัวแปรประเภท struct สามารถทำได้ด้วยเครื่องหมาย dot (.) ตามด้วยชื่อตัวแปรหรือสมาชิก โดยมีรูปแบบดังนี้

รูปแบบการอ้างถึงสมาชิกในตัวแปรชนิดโครงสร้าง	ตัวอย่าง
ชื่อตัวแปรชนิดโครงสร้าง.ชื่อตัวแปรสมาชิก	stu1.stu_id stu1.stu_surname

9.5.5 การกำหนดข้อมูลให้กับตัวแปรชนิดโครงสร้าง

การกำหนดค่าต้องกำหนดให้ถูกต้องตามชนิดข้อมูลของสมาชิก และใช้คำสั่ง printf เพื่อแสดงผลข้อมูลค่าของสมาชิกของตัวแปรชนิดโครงสร้าง

ตัวอย่างการตัวแปรชนิดโครงสร้าง

โปรแกรม	ผลลัพธ์
<pre> 1. #include <stdio.h> 2. main() 3. { 4. typedef struct { 5. int stu_id; 6. char stu_name[20]; 7. char stu_surname[20]; 8. }STUDENT; 9. STUDENT stu = {101, "Jirawan", "Charoensuk"}; 10. printf("---- Student Information ----\n"); 11. printf("Student ID: %d\n", stu.stu_id); 12. printf("Name: %s\n", stu.stu_name); 13. printf("Surname: %s\n", stu.stu_surname); 14. }</pre>	

9.5.6 การรับข้อมูลและแสดงผลข้อมูลของตัวแปรชนิดโครงสร้าง

ใช้คำสั่ง `scanf` และ `printf` เพื่อรับค่าและแสดงผลเหมือนตัวแปรทั่วไป และต้องกำหนดให้รูปแบบข้อมูลให้สอดคล้องกับชนิดข้อมูลของสมาชิกแต่ละค่า

ตัวอย่างการรับและแสดงผลสมาชิกในตัวแปรชนิดโครงสร้าง

โปรแกรม
<pre> 1. #include <stdio.h> 2. main() 3. { 4. typedef struct { 5. int stu_id; 6. char stu_name[20]; 7. char stu_surname[20]; 8. }STUDENT;</pre>

```

9.  STUDENT stu;
10. printf("Enter student ID: ");
11. scanf("%d", &stu.stu_id);
12. printf("Enter name: ");
13. scanf("%s", &stu.stu_name);
14. printf("Enter surname: ");
15. scanf("%s", &stu.stu_surname);
16. printf("ID: %d, Name: %s, Surname: %s", stu.stu_id, stu.stu_name, stu.stu_surname);
17. }

```

ผลลัพธ์

9.5.7 ตัวแปรชนิดโครงสร้างกับการใช้อาร์เรย์

การสร้างตัวแปรอาร์เรย์ของตัวแปรชนิดโครงสร้าง

ตัวอย่างการใช้งาน

```

typedef struct {
    int stu_id;
    char stu_name[20];
    char stu_surname[20];
    char stu_major[20];
    char stu_faculty[20];
} STUDENT;
STUDENT stu[10];

```

stu[0]

stu_id

stu_name

stu_surname

stu_major

stu_faculty

stu[9]

stu_id

stu_name

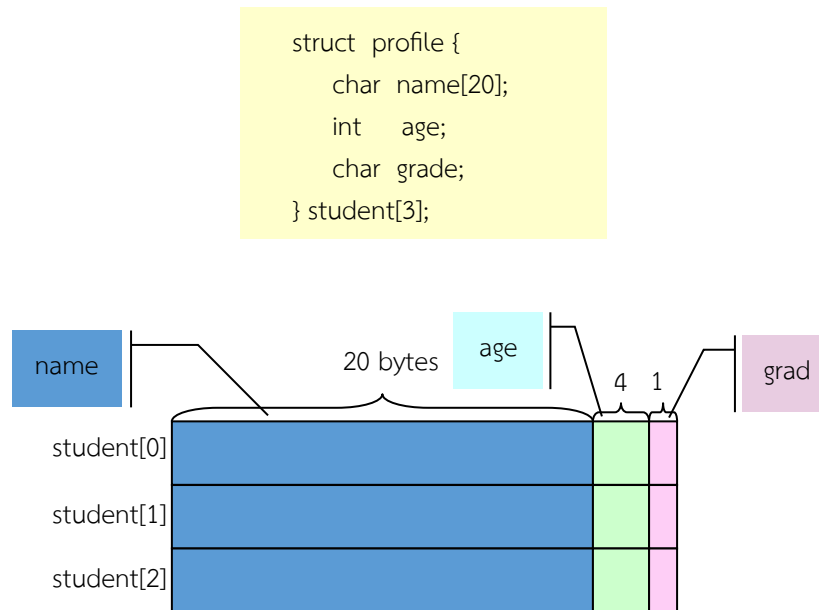
stu_surname

stu_major

stu_faculty

9.6 อาร์เรย์ของโครงสร้าง

ตัวแปรชนิดอาร์เรย์โครงสร้างจะจองขนาดพื้นที่ในหน่วยความจำขึ้นอยู่กับชนิดข้อมูลของสมาชิกภายในตัวแปรสร้างและจำนวนขนาดของอาร์เรย์ที่กำหนด จากการประกาศตัวแปรชนิดโครงสร้างชื่อ profile คอมไพเลอร์จะจองพื้นที่ในหน่วยความจำแสดงดังภาพ 9.4



ภาพที่ 9.4 จำลองการจองพื้นที่ในหน่วยความจำของตัวแปร profile

ตัวอย่างต่อไปนี้จะแสดงการใช้อาร์เรย์ของโครงสร้าง

```
#include <stdio.h>
main() {
    struct profile {
        char name[20];
        int age;
        char grade;
    };
    struct profile student[10];
}
```


9.6.1 การกำหนดค่าเริ่มต้นให้ข้อมูลอาร์เรย์ของโครงสร้าง

รูปแบบการกำหนดค่าเริ่มต้นให้ข้อมูลอาร์เรย์ของโครงสร้าง
<pre>struct ชื่อโครงสร้าง ตัวแปร[จำนวนสมาชิก] = { { รายการสมาชิกตัวที่ 1}, { รายการสมาชิกตัวที่ 2}, ... { รายการสมาชิกตัวที่ n} };</pre>
ตัวอย่าง
<pre>struct student stu[100] = { {"60302010211", "Alice", 20, 15, 30, 'C'}, {"60302010212", "Thomas", 25, 18, 40, 'A'}, {"60302010213", "Zani", 12, 15, 25, 'D'} };</pre>

9.6.2 ตัวแปรชนิดโครงสร้างกับการใช้อาร์เรย์

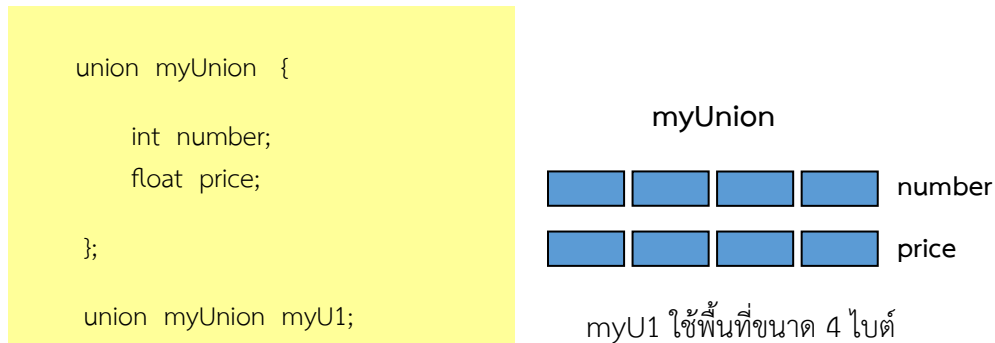
โปรแกรม
<pre>1. #include <stdio.h> 2. main() 3. { 4. typedef struct { 5. int stu_id; 6. char stu_name[20]; 7. char stu_surname[20]; 8. }STUDENT; 9. STUDENT stu[3]; 10. int i;</pre>

```
11.   for(i=0; i<=2;i++){
12.       printf("Enter student ID: ");
13.       scanf("%d", &stu[i].stu_id);
14.       printf("Enter name: ");
15.       scanf("%s", &stu[i].stu_name);
16.       printf("Enter surname: ");
17.       scanf("%s", &stu[i].stu_surname);
18.       printf("Added !\n\n");
19.   }
20.   for(i=0; i<=2;i++){
21.       printf("%d. ID: %d\tName: %s\tSurname: %s\n", i+1, stu[i].stu_id,
22.           stu[i].stu_name, stu[i].stu_surname);
23.   }
```

ผลลัพธ์

9.7 ยูเนียน (Unions)

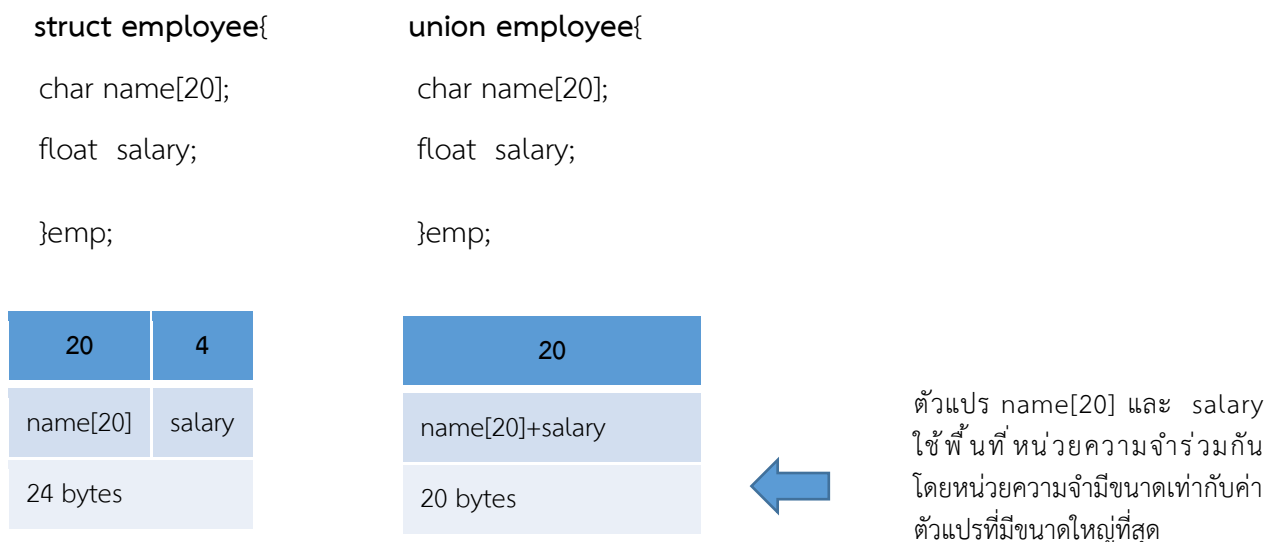
คือ การกำหนดรูปแบบชนิดข้อมูลขึ้นมาใหม่ คล้ายกับข้อมูลแบบโครงสร้าง เพียงแต่ตัวแปรย่อยภายในจะ
ใช้พื้นที่ หน่วยความจำร่วมกัน ดังนั้นถ้ากลุ่มตัวแปรย่อยในยูเนียนแล้ว Compiler จะจัดสรรเนื้อที่ในหน่วยความจำ
ให้เท่ากับข้อมูล (ตัวแปร) ที่มีขนาดใหญ่ที่สุด ขนาดของตัวแปรแบบยูเนียน จะเท่ากับขนาดของตัวแปรย่อยที่ใหญ่
ที่สุด ยูเนียนถูกใช้เมื่อต้องการเก็บข้อมูลเพียงค่าใดค่าหนึ่ง และไม่ต้องการประกาศตัวแปรเพิ่ม



ภาพที่ 9.5 จำลองการจองพื้นที่ในหน่วยความจำของตัวแปรชนิดอาร์เรย์ของยูเนียน

ความแตกต่างระหว่างโครงสร้างและยูเนียน

ตัวแปรชนิดอาร์เรย์โครงสร้างและยูเนียน จากการประกาศตัวแปรชนิดโครงสร้างและยูเนียนชื่อ employee คอม
ไพเลอร์จะจองพื้นที่ในหน่วยความจำแสดงดังภาพ 9.6



ภาพที่ 9.6 จำลองการจองพื้นที่ในหน่วยความจำของตัวแปรโครงสร้างและยูเนียน

ตัวอย่างการประกาศและการทำงานของตัวแปรโครงสร้างและยูเนียน

โปรแกรม
<pre>1. #include <stdio.h> 2. union u_employee 3. { 4. char name[20]; 5. float salary; 6. int workerNo; 7. } u_emp; 8. struct s_employee 9. { 10. char name[20]; 11. float salary; 12. int workerNo; 13. } s_emp; 14. main() 15. { 16. printf("size of union = %d\n", sizeof(u_emp)); 17. printf("size of structure = %d", sizeof(s_emp)); 18. }</pre>
ผลลัพธ์

ณ เวลานั้นๆ สามารถใช้ค่าในตัวแปรแบบยูเนียนได้เพียงตัวเดียวเท่านั้นค่าที่คงอยู่ คือ ค่าของตัวแปรสมาชิกที่ถูกใช้ครั้งล่าสุด เพราะตัวแปรสมาชิกเหล่านั้นจะใช้พื้นที่หน่วยความจำร่วมกัน

ตัวอย่างที่ 1 การประกาศและการทำงานของยูเนียน

โปรแกรม	ผลลัพธ์
<pre> 1. #include <stdio.h> 2. main() 3. { 4. union myunion { 5. float i; 6. float x; 7. }; 8. union myunion myU1; 9. myU1.i = 5; 10. printf("%.2f\n", myU1.i); 11. myU1.x = 3.5; 12. printf("%.2f\n", myU1.x); 13. printf("%.2f\n", myU1.i); 14. printf("%.2f\n", myU1.x); 15. 16. }</pre>	

ตัวอย่างที่ 2 การประกาศและการทำงานของยูเนียน

โปรแกรม	ผลลัพธ์
<pre>1. #include <stdio.h> 2. union u_employee 3. { 4. char name[20]; 5. float salary; 6. int workerNo; 7. } u_emp; 8. main() 9. { 10. printf("Enter name:"); 11. scanf("%s", &u_emp.name); 12. printf("Enter salary:"); 13. scanf("%f", &u_emp.salary); 14. printf("\n*****\n"); 15. printf("Name :%s\n", u_emp.name); 16. printf("Salary: %.2f", u_emp.salary); 17. }</pre>	

คำถามท้ายบทที่ 9

1. จงบอกความแตกต่างระหว่างชนิดข้อมูลอาร์เรย์และชนิดข้อมูลโครงสร้าง
2. จงอธิบายความแตกต่างระหว่างชนิดข้อมูลโครงสร้างและยูเนียน
3. จงอธิบายความหมายของการประกาศตัวแปรดังต่อไปนี้

```
1. struct hotel
2. {
3.     char room[20];
4.     float price;
5.     int type;
6. } Ahotel[20];
```

4. จงประกาศตัวแปรชนิดโครงสร้างเพื่อเก็บข้อมูลค่าน้ำ ค่าไฟ ค่าส่วนกลาง ของห้องพักจำนวน 20 ห้อง
5. จงเขียนโปรแกรมรับข้อมูลหนังสือจำนวน 3 เล่ม โดยมีข้อกำหนดดังนี้
 - 5.1) ข้อมูลหนังสือ 1 เล่มประกอบด้วย ชื่อหนังสือ, ชื่อผู้แต่ง, ราคา
 - 5.2) ให้ประกาศตัวแปรหนังสือเป็นข้อมูลชนิดโครงสร้าง
 - 5.3) สรุปและแสดงผลข้อมูลหนังสือแต่ละเล่ม
 - 5.4) คำนวณและแสดงชื่อหนังสือที่ราคาแพงที่สุด
6. จากการประกาศตัวแปรโครงสร้างด้านล่าง จงตอบคำถามดังต่อไปนี้

```
1. struct test
2. {
3.     int val[3];
4.     char arr[8];
5.     float salary;
6. };
```

- 6.1) struct test มีการจองพื้นที่ในหน่วยความจำทั้งหมดกี่ bytes
- 6.2) จงเขียนคำสั่งเพื่อแสดงขนาดของการจองพื้นที่ในหน่วยจำของตัวแปร struct test
- 6.3) ถ้าเปลี่ยนชนิดข้อมูลจาก struct เป็น union ตัวแปร test มีการจองพื้นที่ในหน่วยความจำทั้งหมดกี่ bytes

7. จงหาผลลัพธ์จากโปรแกรมดังต่อไปนี้

โปรแกรม	ผลลัพธ์(แสดงผลพร้อมบรรทัดบนหน้าจอคอมพิวเตอร์)
<pre> 1. # include <stdio.h> 2. # include <string.h> 3. main() 4. { 5. struct student 6. { 7. int code; 8. char name[20]; 9. float salary 10. }; 11. struct student string1 = { 1,"steve", 100000 }; 12. struct student string2 = string1; 13. printf("%s", string2.name); 14. }</pre>	<pre> 1. 2. 3. กำหนดให้เปลี่ยนชนิดข้อมูลของ student เป็น union 1. 2. 3.</pre>

8. จงหาผลลัพธ์จากโปรแกรมดังต่อไปนี้

โปรแกรม	ผลลัพธ์(แสดงผลพร้อมบรรทัดบนหน้าจอคอมพิวเตอร์)
<pre> 1. #include<stdio.h> 2. struct course 3. { 4. int courseno; 5. char coursename[25]; 6. }; 7. main() 8. { 9. struct course c[] = { {102, "C"}, 10. {103, "C++"}, 11. {104, "Java"}, 12. {105, "Php"} 13. }; 14. printf("%d\n", c[1].courseno); 15. printf("%s\n", (*(c+2)).coursename); 16. }</pre>	<pre> 1. 2. 3. กำหนดให้เปลี่ยนชนิดข้อมูลของ course เป็น union 4. 5. 6.</pre>

9. จงเขียนโปรแกรมคิดค่าสาธารณูปโภคห้องพักของหอพักแห่งหนึ่งจำนวน 3 ห้อง โดยมีข้อกำหนดดังนี้
กำหนดให้ประกาศตัวแปรค่าใช้จ่ายหอพักเป็นข้อมูลชนิดโครงสร้าง

ค่าสาธารณูปโภค ประกอบด้วย

1. ค่าไฟฟ้า คิดค่าบริการหน่วยละ 15 บาท
2. ค่าน้ำ คิดค่าบริการหน่วยละ 30 บาท
3. ค่าขยะ คิดค่าบริการเดือนละ 100 บาท
4. ค่าบริการส่วนกลาง คิดค่าบริการเดือนละ 150 บาท

สรุปและแสดงผลค่าสาธารณูปโภคของแต่ละห้อง คำนวณและแสดงผลหมายเลขห้องที่ใช้ไฟฟ้าและน้ำมากที่สุด

```
Room number 1
Enter number of electric: 10
Enter number of water: 1
Added !

Room number 2
Enter number of electric: 1
Enter number of water: 10
Added !

Room number 3
Enter number of electric: 2
Enter number of water: 2
Added !
```

```
Room number 1
Number of Electric = 10 Cost of Electric = 150.00
Number of Water = 1 Cost of Water = 30.00
Cost of waste = 100.00
Cost of service = 150.00
Cost of Room number 1 = 430.00
*****
Room number 2
Number of Electric = 1 Cost of Electric = 15.00
Number of Water = 10 Cost of Water = 300.00
Cost of waste = 100.00
Cost of service = 150.00
Cost of Room number 2 = 565.00
*****
Room number 3
Number of Electric = 2 Cost of Electric = 30.00
Number of Water = 2 Cost of Water = 60.00
Cost of waste = 100.00
Cost of service = 150.00
Cost of Room number 3 = 340.00
*****

High electric is Room number1
High water is Room number2_
```

10. จงหาผลลัพธ์จากโปรแกรมดังต่อไปนี้

โปรแกรม	ผลลัพธ์(แสดงผลพร้อมบรรทัดบนหน้าจอคอมพิวเตอร์)
<pre> 17. #include <stdio.h> 18. struct Distance { 19. int km; 20. int meter; 21. } d1, d2, result; 22. int divmeter; 23. main() { 24. printf("Enter 1st distance\n"); 25. printf("Enter kilometer: "); 26. scanf("%d", &d1.km); 27. printf("Enter meter: "); 28. scanf("%d", &d1.meter); 29. 30. printf("Enter 2st distance\n"); 31. printf("Enter kilometer: "); 32. scanf("%d", &d2.km); 33. printf("Enter meter: "); 34. scanf("%d", &d2.meter); 35. 36. divmeter = ((d1.km - d2.km)*100) + (d1.meter- d2.meter); 37. result.meter = divmeter%100; 38. result.km = divmeter/100; 39. 40. printf("Sum of distances = %d km and %d meter", result.km, result.meter); 41. }</pre>	<p>ผลลัพธ์ของโปรแกรม</p> <p><u>กำหนดให้รับค่าของตัวแปร</u></p> <p>d1.km เป็น 5 และ d1.meter เป็น 0 d2.km เป็น 3 และ d2.meter เป็น 0</p> <p>7.</p> <p>8.</p> <p>9.</p> <p>10.</p> <p>11.</p> <p>12.</p> <p>13.</p> <p>14.</p> <p><u>กำหนดให้รับค่าของตัวแปร</u></p> <p>d1.km เป็น 5 และ d1.meter เป็น 0 d2.km เป็น 3 และ d2.meter เป็น 30</p> <p>1.</p> <p>2.</p> <p>3.</p> <p>4.</p> <p>5.</p> <p>6.</p> <p>7.</p> <p>8.</p>