บทที่ 8 อาร์เรย์และโครงสร้าง

สาระการเรียนรู้

- ตัวแปรอาร์เรย์
- ตัวแปรโครงสร้าง

สมรรถนะการเรียนรู้

1. ขอกความหมายของตัวแปรอาร์เรย์ได้
2. ประกาศตัวแปรอาร์เรย์ 1 มิติได้
3. ประกาศตัวแปรอาร์เรย์ 2 มิติได้
4. เขียนโปรแกรมเข้าถึงตัวแปรอาร์เรย์ได้
5. ขอกความหมายของตัวแปรโครงสร้างได้
6. ประกาศตัวแปรโครงสร้างได้
7. กำหนดค่าเริ่มต้นให้กับตัวแปรโครงสร้างได้
8. เขียนโปรแกรมเข้าถึงตัวแปรโครงสร้างได้
9. ป้อนและทดสอบโปรแกรมตามตัวอย่างได้

แผนผังความคิด (Mind Mapping) ของหน่วยการเรียนรู้

อาร์เรย์ (Array)

ตัวแปรชุด หรือ อาร์เรย์ (Array) เป็นโครงสร้างข้อมูลพื้นฐานและใช้งานมากที่สุด ทำหน้าที่ในการจองเนื้อที่ในหน่วยความจำ โดยใช้ชื่อตัวแปรเดียวอ้างถึงข้อมูลโดยใช้ตัวชี้หรือดัชนี (Index)

ในภาษาซี ค่าของตัวชี้ของตัวแปรชุดตัวแรกจะมีค่าเป็น 0 ตัวต่อไปจะมีค่าเป็น 1,2,... จนถึง n-1

อาร์เรย์ 1 มิติ (One Dimension Array)

ใช้ในการจัดเก็บข้อมูลโดยตัวชี้ตำแหน่งอ้างอิงแถว (Row)

รูปแบบการประกาศตัวแปรอาร์เรย์ 1 มิติ

type var_name [row];

เช่น

1. int arr[5];

   เป็นการประกาศตัวแปรอาร์เรย์ชื่อ arr จองไว้ 5 แถว คือ

   arr[0]

   arr[1]

   arr[2]

   arr[3]

   arr[4]

2. char fname[30][40];

   เป็นการประกาศตัวแปรข้อความ (String) ที่มีความยาว 29 อักขระแบบอาร์เรย์ชื่อ fname จองไว้ 40 แถว ซึ่งมีลักษณะเป็นอาร์เรย์ 2 มิติของตัวแปรอักขระที่มี 30 คอลัมน์ 40 แถว นั่นเอง

อาร์เรย์ 2 มิติ (Two Dimension Array)

ใช้ในการจัดเก็บข้อมูลในลักษณะตาราง โดยตัวชี้ตำแหน่งอ้างอิงแถว (Row) และคอลัมน์ (Column)

รูปแบบการประกาศตัวแปรอาร์เรย์ 1 มิติ

type var_name [row][column];

เช่น

1. int arr[5][3];

เป็นการประกาศตัวแปรอาร์เรย์ชื่อ arr จองไว้ 5 แถว 3 คอลัมน์ คือ

arr[0][0], arr[0][1], arr[0][2]

arr[1][0], arr[1][1], arr[1][2]

arr[2][0], arr[2][1], arr[2][2]

arr[3][0], arr[3][1], arr[3][2]

arr[4][0], arr[4][1], arr[4][2]

อาร์เรย์ 3 มิติ (Three Dimension Array)

มีลักษณะการประกาศในทำนองเดียวกัน แต่เพิ่มความลึก หรือเรียกว่าจำนวนแผ่น แต่ในเบื้องต้นนี้จะกล่าวเพียงอาร์เรย์ 2 มิติ ส่วนอาร์เรย์หลายมิติโปรแกรมโดยทั่วไปมักไม่ค่อยมีใช้กัน จะมีก็จะเกี่ยวข้องกับการประมวลผลภาพหรือด้านกราฟิก 3 มิติ หรือ 4 มิติ

รูปแบบการประกาศตัวแปรอาร์เรย์ หลาย มิติ

```
type var_name [row][column][dept]...[...];
```

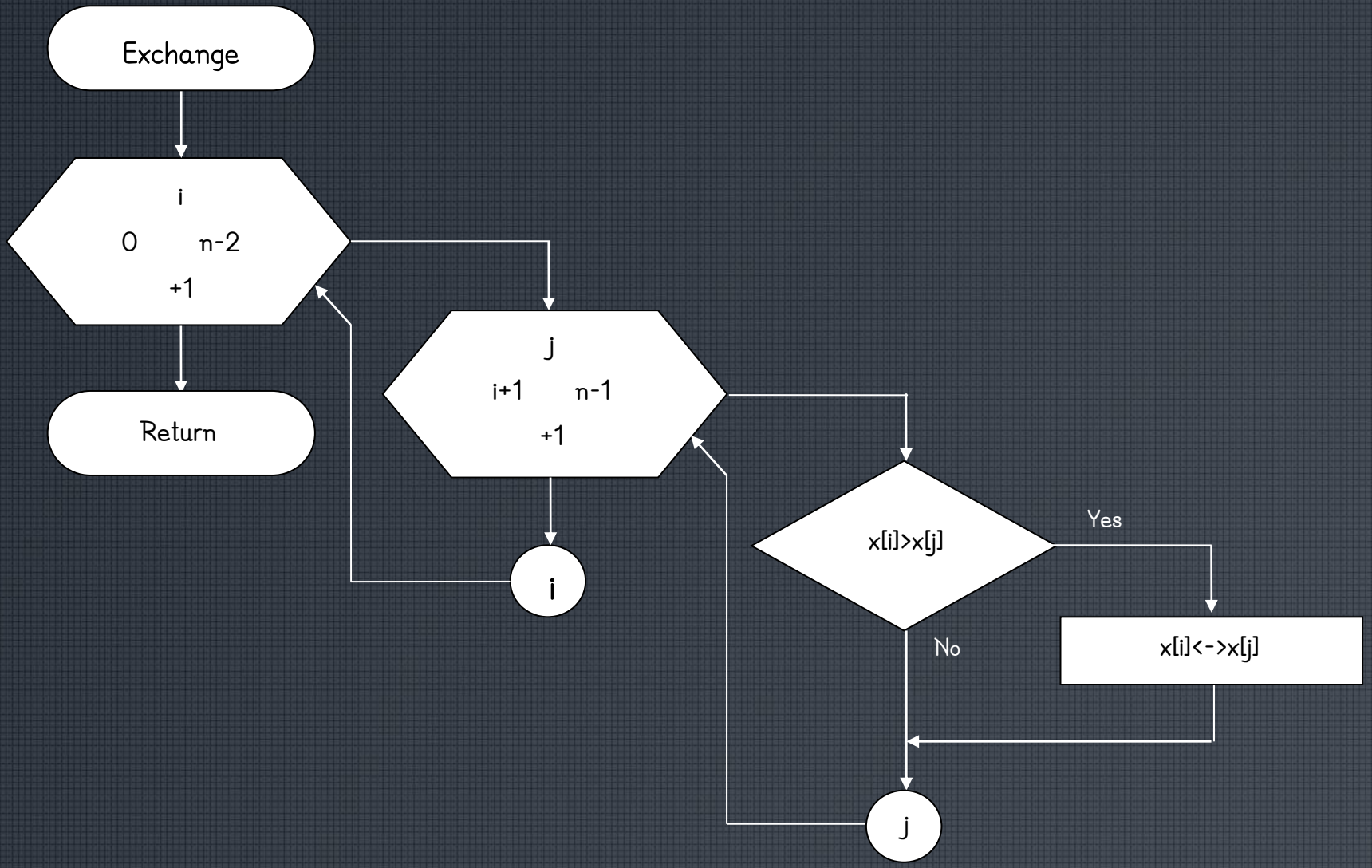
การหาค่าสูงสุด ค่าต่ำสุด ค่าผลรวมและค่าเฉลี่ยของชุดข้อมูล

ในการเขียนโปรแกรมรับค่าข้อมูลจากแป้นพิมพ์หรืออ่านมาจากแฟ้มข้อมูล เพื่อนำมาเก็บไว้ในตัวแปรอาร์เรย์ มี 2 กรณีคือ กรณีที่ทราบจำนวนข้อมูล ในการเขียนโปรแกรมจะใช้โครงสร้างของคำสั่งวนรอบหรือลูป (Loop) ไม่ว่าจะเป็นคำสั่ง do-while, while หรือคำสั่ง For ส่วนอีกกรณีคือไม่ทราบจำนวนที่แน่นอนของจำนวนข้อมูล จะใช้เทคนิคการอ่านข้อมูลเข้ามาที่ละตัวเพื่อมาเปรียบเทียบกับข้อมูลที่กำหนดขึ้น ซึ่งข้อมูลที่กำหนดขึ้นนี้ จะเป็นข้อมูลที่ไม่มีอยู่ในข้อมูลชุดนั้น เช่น ถ้าอ่านคะแนนที่มีค่าระหว่าง 0-100 เข้ามา เราอาจกำหนดข้อมูลที่ใช้เปรียบเทียบเป็น 999 หรือ -999 ซึ่งจะเป็นการตรวจเพื่อให้สิ้นสุดการรับข้อมูล

การจัดเรียงข้อมูล (Sorting)

การประมวลผลข้อมูลที่เป็นกระบวนการหลักอย่างหนึ่งในโปรแกรมต่าง ๆ ที่เกี่ยวข้องกับการจัดการข้อมูล ก็คือการจัดเรียงข้อมูล (Sorting) ข้อมูลที่จะนำมาจัดเรียงจะถูกจัดเก็บในแบบของตัวแปรอาร์เรย์ หรือตัวแปรโครงสร้างแบบอาร์เรย์ ส่วนวิธีการจัดเรียงข้อมูลนั้น มีมากมายหลายวิธี ได้แก่ แบบ bubble, cocktail, exchange, selection, insertion, shell, merge, heap, quick เป็นต้น ถึงแม้ว่าวิธีการจัดเรียงข้อมูลที่เร็วที่สุด ซึ่งมีความซับซ้อนทำความเข้าใจได้ยาก ก็อาจไม่เหมาะกับงานที่มีข้อมูลน้อย ๆ เพราะการบำรุงรักษาโปรแกรมต้องสามารถกระทำได้ อาจไม่ใช่บุคคล คนเดียวกับผู้เขียนโปรแกรมคนแรกของงานชิ้นนั้น ดังนั้นอาจเลือกใช้วิธีที่คนส่วนใหญ่ควรจะรู้จักและสามารถพัฒนาหรือปรับปรุงโปรแกรมในภายหลังได้ แต่เป็นวิธีการที่มีเสถียรภาพและความถูกต้องแม่นยำในการประมวลผล

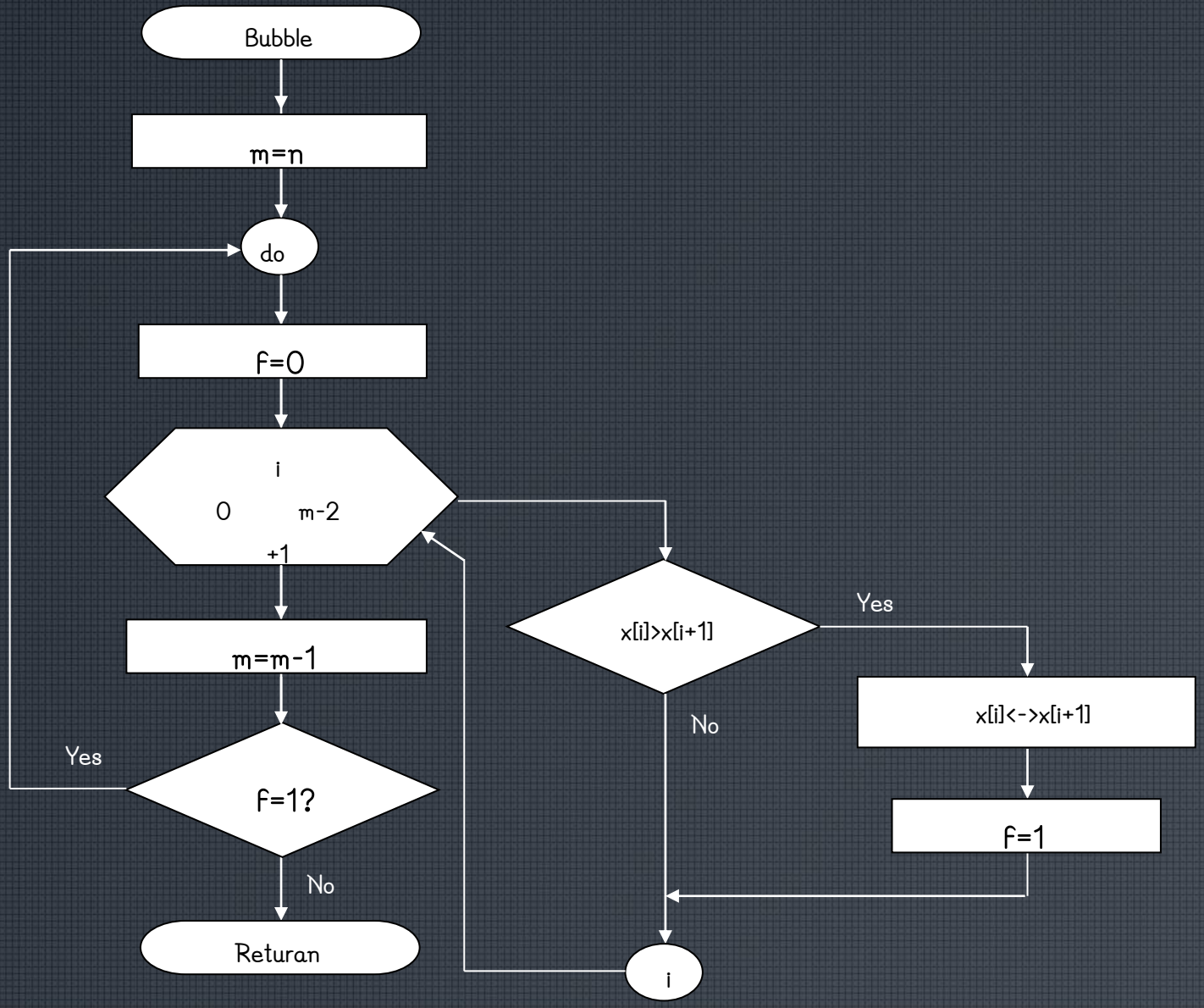
การจัดเรียงข้อมูลแบบ Exchange Sort



Exchange Sort

```
int x[1000],n;
// exchange Sort
void exchange()
{
        int i,j;
        for (i=0; i<=n-2; i++)
                for (j=i+1; j<=n-1; j++)
                        if (x[i]>x[j])
                                swap(i,j);
}
void swap(int a, int b)
{
        if (a != b)
        {
                int t=x[a]; x[a]=x[b]; x[b]=t;
        }
}
```

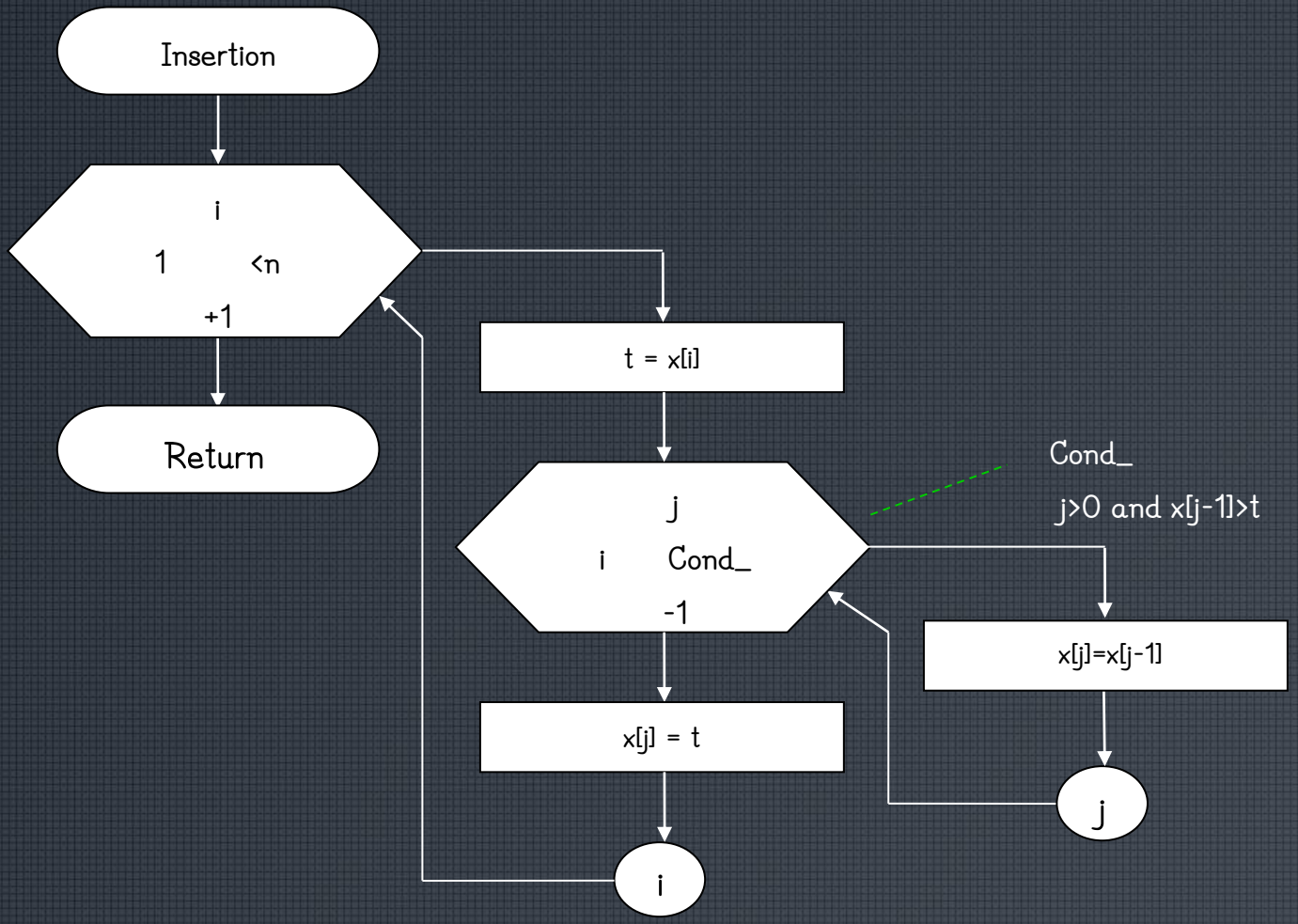
การจัดเรียงข้อมูลแบบ Bubble Sort



Bubble Sort

```
int x[1000],n;
// Bubble Sort
void bubble()
{
        int f,i,m,n;
        m = n;
        do {
                f=0;
                for (i=0; i<=m-2; i++)
                        if (x[i] > x[i+1]) {
                                swap(i,i+1); f=1;
                        }
                m--;
        } while (f==1);
}
```

การจัดเรียงข้อมูลแบบ Insertion Sort



Insertion Sort

```
int x[1000],n;

// insertion Sort
void insertion()
        int i, j, t;
        for (i = 1; i < n; i++)
        {
                t = x[i];
                for (j = i; j > 0 && x[j-1] > t; j--)
                {
                        x[j] = x[j-1];
                }
                x[j] = t;
        }
}
```

ตัวแปรโครงสร้าง

โครงสร้างหรือสตรักเจอร์ เป็นข้อมูลโครงสร้างที่ประกอบด้วยสมาชิก (Member) หลายตัว โดยแต่ละตัวอาจมีชนิดของข้อมูลแตกต่างกัน

การประกาศโครงสร้าง

ในการเขียนโปรแกรมใช้งานตัวแปรโครงสร้าง จะต้องมีการนิยามโครงสร้างและประกาศตัวแปรโครงสร้างก่อน จึงจะสามารถใช้ตัวแปรโครงสร้างนั้นได้

รูปแบบการประกาศโครงสร้าง

```
struct <structure_name> {
        type var1;
        type var2;
        type varN;
} [<var1_structure>][,<var2_structure>,...];
```

ตัวอย่างโครงสร้าง

```
struc person {
        char id[13];
        char fname[30];
        int  age;
        char address[40];
        char tel[10];
} cust1, cust2, emps[30];
```

โครงสร้างข้างต้นนี้ ประกอบด้วยสมาชิก (Member) 5 สมาชิก ได้แก่ id, fname, age, address, tel หลังจากนิยามโครงสร้างแล้ว ยังได้ประกาศตัวแปรโครงสร้าง 2 ตัวเป็นแบบทั่วไป คือ cust1 และ cust2 ส่วน emps เป็นตัวแปรโครงสร้างที่เป็นตัวแปรชุดหรือตัวแปรอาร์เรย์ 1 มิติ

การประกาศตัวแปรโครงสร้าง

หลังจากได้มีการนิยามโครงสร้างไว้แล้ว เราสามารถประกาศตัวแปรโครงสร้างได้ดังนี้

รูปแบบการประกาศตัวแปรโครงสร้าง

```
struc <structure_name> var_struct [,var_struct,...];
```

เช่น

```
struc person cust3, cust4;
```

หรือประกาศเป็นตัวแปรชุด

```
struc person boss[5];
```

ตัวแปรโครงสร้างเป็นสมาชิกหนึ่งในตัวแปรโครงสร้าง

```
struc date {
        int  dd;
        int  mm;
        int  yy;
}
struct employee {
        char id_code[13];
        char emp_name[30];
        struct date birthday;
} emp[20];
```

การเข้าถึงสมาชิกของตัวแปรแบบโครงสร้าง

การเข้าถึงทำได้โดยการอ้างถึงชื่อตัวแปรโครงสร้างคั่นด้วยจุด(.) แล้วตามด้วยชื่อสมาชิกที่ต้องการ

รูปแบบ

var_struct.member

เช่น

การกำหนดค่าให้กับสมาชิก

```
emp[0].salary = 56000.00f;
```

การรับค่าจากแป้นพิมพ์

```
scanf("%f",&emp[0].salary);
```

การแสดงผลทางจอภาพ

```
printf("Salary = %10.2f",emp[0].salary);
```

บิตฟิลด์ (Bit Field)

บิตฟิลด์เป็นตัวแปรโครงสร้างประเภทหนึ่ง ซึ่งสามารถกำหนดขนาดหน่วยความจำเป็นจำนวนบิตให้กับสมาชิกแต่ละตัว ทำให้ประหยัดหน่วยความจำที่สุด

รูปแบบ

Type var_name : size

Size มีหน่วยเป็นบิต

ตัวอย่างโครงสร้างแบบบิตฟิลด์หรับเก็บลักษณะของไพ่ (Card) สามารถกำหนดได้ดังนี้

```
struct bitcard {
        unsigned face : 4;  // 2^4 = 16, (ไพ่มี 13 หน้า)
        unsigned suit  : 2;  // 2^2 = 4, (ไพ่มี 4 แบบ)
        unsigned color : 1; // 2^1 = 2, (ไพ่มี 2 สี)
}
```

ยูเนียน (Unions)

เป็นข้อมูลโครงสร้างอีกแบบหนึ่ง แต่มีข้อแตกต่างกันกับตัวแปรโครงสร้าง คือสมาชิกแต่ละตัวของตัวแปรโครงสร้างจะใช้หน่วยความจำคนละตัวกัน ส่วนยูเนียนสมาชิกแต่ละตัวจะใช้หน่วยความจำร่วมกัน การนิยามและการใช้งานคล้ายกับตัวแปรโครงสร้าง เพียงเปลี่ยนคีย์เวิร์ดจาก struct มาเป็น union เท่านั้น

ข้อมูลชนิด Enumerations

ในภาษาซี เราสามารถสร้างตัวแปรประเภทใหม่ขึ้นมาเก็บข้อมูลแบบเรียงลำดับได้ ข้อมูลประเภทนี้เรียกว่า enumeration

รูปแบบ

```
enum new_type {data set} var [,var,...];
```

เช่น

```
enum months {Jan, Feb, Mar, Apr, May,
    Jun, Jul, Aug, Sep, Oct, Nov, Dec};
```

จบการนำเสนอ

บทที่ 8
อาร์เรย์และโครงสร้าง