

บทที่ 6

อาร์เรย์ (Array)

◆ ความหมายของข้อมูลชนิดอาร์เรย์(array)

อาร์เรย์ (array) เป็นข้อมูลชนิดมีโครงสร้าง (structure) หมายถึงเป็นข้อมูลชุดหนึ่งที่มีจำนวนสมาชิกข้อมูลแน่นอนและต้องเป็นชนิดเดียวกันทั้งหมด แต่ละรายการของข้อมูลในชุดนั้น ๆ เรียกว่า สมาชิกของอาร์เรย์ (element of array) สมาชิกแต่ละตัวมีตัวชี้ตำแหน่งที่อยู่ เรียกว่า อินเด็กซ์ (index) เพื่อใช้ในการอ้างอิงถึงข้อมูลแต่ละสมาชิก โดยอินเด็กซ์เป็นข้อมูลชนิดที่มีลำดับ เช่น integer

ตัวอย่างข้อมูลชนิดอาร์เรย์ 1 มิติ เช่น คะแนนสอบของนักศึกษา 10 คน เป็นตัวเลขประเภท int กำหนดเป็นโครงสร้างของอาร์เรย์ดังนี้

| นักศึกษาคนที่ (index) | |
|-----------------------|---|
| | [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] |
| คะแนนที่ได้ | 20 25 15 30 28 16 23 24 30 26 |

ข้อมูลในอาร์เรย์

ความหมายของข้อมูลในอาร์เรย์โดยใช้ index เป็นตัวชี้ข้อมูลในแต่ละช่อง เป็นดังนี้

อาร์เรย์ช่องที่ [0] มีค่าคะแนนนักศึกษา คนที่ 1 เก็บอยู่ คือ 20

อาร์เรย์ช่องที่ [1] มีค่าคะแนนนักศึกษา คนที่ 2 เก็บอยู่ คือ 25

อาร์เรย์ช่องที่ [2] มีค่าคะแนนนักศึกษา คนที่ 3 เก็บอยู่ คือ 15

อาร์เรย์ช่องที่ [3] มีค่าคะแนนนักศึกษา คนที่ 4 เก็บอยู่ คือ 30

อาร์เรย์ช่องที่ [4] มีค่าคะแนนนักศึกษา คนที่ 4 เก็บอยู่ คือ 28

อาร์เรย์ช่องที่ [5] มีค่าคะแนนนักศึกษา คนที่ 4 เก็บอยู่ คือ 16

อาร์เรย์ช่องที่ [6] มีค่าคะแนนนักศึกษา คนที่ 4 เก็บอยู่ คือ 23

อาร์เรย์ช่องที่ [7] มีค่าคะแนนนักศึกษา คนที่ 4 เก็บอยู่ คือ 24

อาร์เรย์ช่องที่ [8] มีค่าคะแนนนักศึกษา คนที่ 4 เก็บอยู่ คือ 30

อาร์เรย์ช่องที่ [9] มีค่าคะแนนนักศึกษา คนที่ 10 เก็บอยู่ คือ 26

ถ้าคะแนนที่เก็บเป็นข้อมูลชนิด int จะจองพื้นที่หน่วยความจำในการเก็บข้อมูล 2 byte *

10 ช่อง เท่ากับ 20 byte

หมายเหตุ การจองพื้นที่หน่วยความจำของอาร์เรย์จะจองพื้นที่ในหน่วยความจำต่อเนื่องกันไป ตั้งแต่สมาชิกเริ่มต้นตัวที่ 0 จนถึงตัวสุดท้าย

◆ ข้อมูลอาร์เรย์ชนิด 1 มิติ

ข้อมูลชนิดอาร์เรย์ 1 มิติ หมายถึง อาร์เรย์ที่มีอินเด็กซ์ในการอ้างอิงถึงแต่ละสมาชิกเพียง 1 ตัว

1. การกำหนดข้อมูลชนิดอาร์เรย์ 1 มิติ ในโปรแกรม มีรูปแบบดังนี้

```
Array_Type Array_name[amount of element];
```

โดยที่ **Array_Type** คือ ชนิดข้อมูล (data type) ในอาร์เรย์หรือประเภทอาร์เรย์ เช่น int, char, float

Array_name คือ ชื่อของตัวแปรอาร์เรย์

[amount of element] คือ จำนวนสมาชิกหรือรายการของอาร์เรย์

ตัวอย่างข้อมูลคะแนนนักศึกษา จำนวน 10 คน เช่น

| นักศึกษาคนที่ (index) | | [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] |
|-----------------------|--|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| คะแนนที่ได้ | | 20 | 25 | 15 | 30 | 28 | 16 | 23 | 24 | 30 | 26 |

กำหนดตัวแปรอาร์เรย์เก็บคะแนนเป็นชนิด int และจองจำนวนสมาชิกไว้ 10 รายการ ดังนี้

```
int score[10];
```

2. การอ้างอิงสมาชิกและข้อมูลในอาร์เรย์ การอ้างอิงถึงแต่ละสมาชิกของอาร์เรย์ทำได้โดยการเขียนชื่อของอาร์เรย์และระบบอินเด็กซ์ของสมาชิกไว้ใน `[]` โดยอินเด็กซ์เริ่มตั้ง 0 ถึง n-1 (n คือจำนวนสมาชิกของอาร์เรย์ทั้งหมด) เช่น

score[0] มีข้อมูล 20 เก็บอยู่

score[1] มีข้อมูล 25 เก็บอยู่

score[5] มีข้อมูล 16 เก็บอยู่

score[9] มีข้อมูล 26 เก็บอยู่

3. การกำหนดค่าให้อาร์เรย์ การกำหนดค่าให้อาร์เรย์ 1 มิติ มีวิธีการดังนี้

3.1 การกำหนดค่าคงที่ให้กับอาร์เรย์ เช่น

```
score[0] = 20;
```

```
score[1] = 25;
```

```
score[2] = 15;
```

```
score[9] = 26;
```

สามารถกำหนดค่าของอาร์เรย์ เป็น constant โดยระบุจำนวนสมาชิกหรือไม่ระบุก็ได้ดังนี้

```
const int score[10] = { 20,25,30,26,24,18,17,25,24,22};
```

```
const int score[] = { 20,25,30,26,24,18,17,25,24,22};
```

3.2 การกำหนดค่าให้อาร์เรย์จากตัวแปร เช่น

```
x = 20; y = 26;
```

```
score[0] = x;
```

```
score[9] = y;
```

4. การรับข้อมูลทางคีย์บอร์ดเก็บในอาร์เรย์โดยใช้คำสั่งลูป เพื่อความสะดวกและความยืดหยุ่นในโปรแกรม เราสามารถรับค่าคงที่เข้าไปเก็บไว้ในอาร์เรย์แต่ละสมาชิกโดยใช้ลูป for, while , do เช่น

```
➡ for( no=0 ; no<=9; ++no)
    _____ cin>>score[no]; //enter from keyboard to array and no is index
```

5. การแสดงผลข้อมูลอาร์เรย์ สามารถแสดงผลข้อมูลในอาร์เรย์แต่ละสมาชิกได้ โดยการอ้างชื่อและอินเด็กซ์ โดยใช้ลูป for , while ,do เพื่อแสดงผลข้อมูลในอาร์เรย์ทั้งหมดได้ เช่น

```
➡ for(no=0;no<=9;++no)
    _____ cout<<score[no]<<endl; //display data all element of array and no is index
```

- ตัวอย่างโปรแกรม *array1.cpp* แสดงการรับค่าคะแนนนักศึกษา 10 คน เก็บไว้ในอาร์เรย์คำนวณหาผลรวมคะแนนทั้งหมดและคำนวณหาค่าเฉลี่ยของคะแนน พร้อมแสดงผลข้อมูลจากอาร์เรย์ทั้งหมด

```
/*Program : array1.cpp
```

```
Process : input and display data for array 1 dimension */
```

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
void main() //begin program
```

```
{ int no,score[10];
```

```
float avg=0,sum=0;
```

```
clrscr();
```

```
//input integer constant to array
```

```
cout<< "Please enter score of student 10 persons\n\n";
```

```
for(no=0;no<=9;++no)
```

```
{ cout<< "Std.##"<<no+1<<" : ";
```

```
cin>>score[no];
```

```
}
```

```

cout<< "O.K. you enter complete...press any key to display\n";
getch();clrscr();

//display value of each element of array
cout<< "Display your enter score :\n";
for(no=0;no<=9;++no)
{ cout<< "Std.#"<<no+1<<" : "<<score[no];
  cout<<endl;
  sum+=score[no]; //calculate summation
}
avg=sum/10.0; //calculate average of score
cout<<"Summation of score = "<<sum<<endl;
cout<<"Average of score = "<<avg<<endl<<"\a';
getch();
} //end program

```

- ตัวอย่างโปรแกรม *arr_func.cpp* แสดงการรับค่าคะแนนนักศึกษา 10 คน เก็บไว้ในอาร์เรย์
คำนวณหาผลรวมคะแนนทั้งหมดและคำนวณหาค่าเฉลี่ยของคะแนน พร้อมแสดงผลข้อมูล
จากอาร์เรย์ทั้งหมด แต่เขียนแยกเป็นฟังก์ชันทำหน้าที่แยกกัน

```

/*Program : arr_func.cpp
Process : define input and display data for array 1 dimension */
#include <iostream.h>
#include <conio.h>
//declaration prototype function
void input();
void display();
//declaration external variable
int score[10];
float avg=0,sum=0;

void main() //begin main program
{
  clrscr();
  input(); //call function
  display(); //call function
  avg=sum/10.0; //calculate average of score
  cout<<"Summation of score = "<<sum<<endl;
  cout<<"Average of score = "<<avg<<endl<<"\a';
  getch();
} //end main program

void input() //input() function for enter integer constant to array

```

```

{ int no;
  cout<< "Please enter score of student 10 persons\n\n";
  for(no=0;no<=9;++no)
  { cout<< "Std.#"<<no+1<<" : ";
    cin>>score[no];
  }
  cout<< "O.K. you enter complete...press any key to display\n";
  getch();clrscr();
}

```

void display() //display() function for show value of each element of array

```

{ int no;
  cout<< "Display your enter score : \n";
  for(no=0;no<=9;++no)
  { cout<< "Std.#"<<no+1<<" : "<<score[no];
    cout<<endl;
    sum+=score[no]; //calculate summation
  }
}

```

◆ อาร์เรย์ 2 มิติ

อาร์เรย์ 2 มิติ (two-dimension array) หมายถึง อาร์เรย์ที่มีตัวอินเด็กซ์สำหรับการอ้างอิงข้อมูลจำนวน 2 ชุด คือ อินเด็กซ์แนวแถว(row) และแนวคอลัมน์ (column)

ลักษณะของข้อมูลชนิดที่เป็นอาร์เรย์ 2 มิติ เช่น คะแนนสอบของนักศึกษาจำนวน 10 คน ทั้งหมด 3 วิชา ดังตาราง

| นักศึกษาคนที่ | แนวคอลัมน์ | | |
|---------------|------------|------------|-------------|
| | คณิตศาสตร์ | ภาษาอังกฤษ | วิทยาศาสตร์ |
| 1 | 25 | 30 | 35 |
| 2 | 28 | 25 | 32 |
| 3 | 30 | 29 | 33 |
| 4 | 29 | 30 | 35 |
| 5 | 35 | 20 | 37 |
| 6 | 26 | 31 | 30 |
| 7 | 20 | 26 | 30 |
| 8 | 26 | 25 | 24 |
| 9 | 29 | 30 | 35 |
| 10 | 26 | 32 | 25 |

จากตาราง 2 มิติข้างบน ถ้าต้องการเก็บคะแนนสอบของนักศึกษาดังกล่าว สามารถใช้ตัวแปรอาร์เรย์ชนิด 2 มิติเพื่อเก็บข้อมูลชุดนี้ได้ สังเกตจากตารางดังกล่าว (เฉพาะที่เป็นข้อมูลตัวเลขคะแนน) ประกอบด้วย 10 แถว และ 3 คอลัมน์ ตัวอย่าง การอ่านคะแนนจากตาราง แถวที่ 5 คอลัมน์ที่ 3 คือ 37 เป็นคะแนนของนักศึกษาคนที่ 5 วิชาที่ 3 คือ วิชาวิทยาศาสตร์

1. การกำหนดอาร์เรย์ 2 มิติ การประกาศตัวแปรประเภทอาร์เรย์ 2 มิติ มีรูปแบบดังนี้

```
Array_Type Array_name[row][column];
```

โดยที่ **Array_Type** คือ ชนิดข้อมูลในอาร์เรย์หรือประเภทอาร์เรย์ 2 มิติ เช่น char , int , float ,double

Array_name คือ ชื่อของตัวแปรอาร์เรย์ 2 มิติ

[row] คือ จำนวนสมาชิกหรือรายการของอาร์เรย์ในแนวแถว

[column] คือ จำนวนสมาชิกหรือรายการของอาร์เรย์ในแนวคอลัมน์

การคำนวณจำนวนรายการหรือสมาชิกของอาร์เรย์ 2 มิติ คือ **row*column**

2. การอ้างอิงถึงสมาชิกอาร์เรย์ 2 มิติ มีวิธีการอ้างอิงโดยการเขียนชื่ออาร์เรย์และอินเด็กซ์ของสมาชิกนั้น เช่น

score[3][2] หมายถึงสมาชิกอาร์เรย์ตำแหน่งแถวที่ 3 คอลัมน์ที่ 2

score[0][3] หมายถึงสมาชิกอาร์เรย์ตำแหน่งแถวที่ 0 คอลัมน์ที่ 3

3. การกำหนดค่าให้อาร์เรย์ 2 มิติ มีวิธีการกำหนดค่าเช่นเดียวกับอาร์เรย์ 1 มิติ คือกำหนดค่าคงที่ให้กับอาร์เรย์ เช่น

```
score[3][2] = 25;
```

```
score[2][1] = 30;
```

```
x = 35;
```

```
score[3][3] = x;
```

ถ้าต้องการกำหนดค่าอาร์เรย์ 2 มิติเป็นลักษณะ constant กำหนดได้เช่นเดียวกับอาร์เรย์

1 มิติ ดังนี้

```
const int score[5][3] = { {25,30,22},
                           {30,25,20},
                           {25,24,18},
                           {22,21,24},
                           {20,18,17}
                           };
```

หรือไม่ต้องระบุจำนวนแถวของอาร์เรย์ ดังนี้

```
const int score[][3] = { {25,30,22},
                          {30,25,20},
                          {25,24,18},
                          {22,21,24},
                          {20,18,17}
                        };
```

โดยที่มิติแรก คือ จำนวนแถวไม่ต้องระบุก็ได้ C++ compiler จะทราบเองจากการตรวจสอบเซตของค่าคงที่ที่กำหนดในลักษณะ const

เพื่อความสะดวกและความยืดหยุ่นในโปรแกรม การกำหนดค่าให้อาร์เรย์โดยการรับค่าจากการกรอกทางคีย์บอร์ด จะใช้ลูป for, while หรือ do จำนวน 2 ลูปซ้อนกันเพื่อรับข้อมูลไปเก็บไว้ในอาร์เรย์ สามารถกำหนดลูปได้ 2 ลักษณะ

- กรอกในแนวแถวก่อน เช่น จากข้อมูลตัวอย่าง จะกรอกคะแนนของนักศึกษาคนที่ 1 ให้ครบทั้ง 3 วิชาก่อน แล้วจึงเริ่มกรอกคะแนนของนักศึกษาคนที่ 2 ให้ครบทุกวิชา ทำเช่นนี้เรื่อย ๆ จนถึงคะแนนคนที่ 10 วิชาที่ 3 ดังตัวอย่างการใช้ลูป for

```

    for(std_no=0;std_no<=9;++std_no) //ลูปนอก
    {
        for(subj_no=0;subj_no<=2;++subj_no) //ลูปใน
            cin>>score[std_no][subj_no];
    }
```

std_no คือ ตัวแปรเก็บค่าตัวชี้ตำแหน่งแถวหรือนักศึกษาคนที่ 0 - 9

subj_no คือ ตัวแปรเก็บค่าตัวชี้ตำแหน่งคอลัมน์หรือวิชาที่ 0-2

- กรอกในแนวคอลัมน์ก่อน เช่น จากข้อมูลตัวอย่าง จะกรอกคะแนนนักศึกษาของวิชาที่ 1 ให้ครบทั้ง 10 คนก่อน จากนั้นจึงเริ่มกรอกวิชาที่ 2 ให้ครบก่อน แล้วจึงเริ่มกรอกคะแนนวิชาที่ 3 จนกว่าจะกรอกคะแนนวิชาที่ 3 ของนักศึกษาคนที่ 10 ซึ่งเป็นคนสุดท้าย ดังตัวอย่างการใช้ลูป for

```

    for(subj_no=0;subj_no<=2;++subj_no) //ลูปนอก
    {
        for(std_no=0;std_no<=9;++std_no) //ลูปใน
            cin>>score[std_no][subj_no];
    }
```

//subj_no คือ ตัวแปรเก็บค่าของคอลัมน์หรือวิชาที่ 0-2

//std_no คือ ตัวแปรเก็บค่าของแถวหรือนักศึกษาคนที่ 0 - 9

4. การแสดงผลอาร์เรย์ 2 มิติ การแสดงผลข้อมูลแต่ละสมาชิกในอาร์เรย์ 2 มิติ มีวิธีการใช้ลูป 2 ลูปซ้อนกัน เหมือนกับการกำหนดค่าเช่นเดียวกัน เพียงแต่ใช้ cout แทน cin เพื่อแสดงค่าตัวแปร จากตัวอย่างคะแนนสอบนักศึกษา 10 คน 3 วิชา สามารถเขียนลูป for เพื่อแสดงผลได้ ดังนี้

- กรณีแสดงค่าแนวแถวก่อน

```

for(std_no=0;std_no<=9;++std_no) //ลูปนอก
    for(subj_no=0;subj_no<=2;++subj_no) //ลูปใน
        cout<<score[std_no][subj_no];

```

- กรณีแสดงค่าแนวคอลัมน์ก่อน

```

for(subj_no=0;subj_no<=2;++subj_no) //ลูปนอก
    for(std_no=0;std_no<=9;++std_no) //ลูปใน
        cin<<score[std_no][subj_no];
//subj_no คือ ตัวแปรเก็บค่าของคอลัมน์หรือวิชาที่ 0-2
//std_no คือ ตัวแปรเก็บค่าของแถวหรือนักศึกษาคนที่ 0 - 9

```

- ตัวอย่างโปรแกรม *arr_2dim.cpp* แสดงการรับค่าและแสดงค่าอาร์เรย์ 2 มิติ คะแนนสอบของนักศึกษา 10 คน เป็นจำนวนเต็ม วิชาที่สอบ 3 วิชา โดยการกรอกข้อมูลเป็นรายคน ได้ดังนี้

/*Program : arr_2dim.cpp

Process : define input and display data for array 2 dimension */

#include <iostream.h>

#include <conio.h>

//declaration prototype function

void input();

void display();

//declaration external variable

int score[10][3];

void main() //begin main program

{

clrscr();

input();

display();

cout<< "\n\nO.K. press any key to exit\n\n";

getch();

} //end main program


```

void input() //input() function for enter integer constant to array
{
    int std_no,subj_no,col,row;
    cout<< "Please enter score of student 10 persons\n\n";
    gotoxy(5,3);cout<<"Std #   Math   English   Science";
    col=15;row =5; //variable for function gotoxy()
    for(std_no=0;std_no<=9;++std_no)
    {
        gotoxy(7,row); //skip cursor to enter data position
        cout<<std_no+1;
        for(subj_no=0;subj_no<=2;++subj_no)
        {
            gotoxy(col,row);
            cin>>score[std_no][subj_no];
            col+=15; //increase column position
        }
        row++; //increase row position
        col=15; //set column at old position in new row (next student)
    }
    cout<< "O.K. you enter complete...press any key to display\n\n";
    getch();clrscr();
}

```

```

void display() //display() function for display score
{
    int std_no,subj_no,col,row;
    cout<< "*** Display score of student 10 persons 3 subjects ***\n\n";
    gotoxy(5,3);cout<<"Std #   Math   English   Science";
    col=15;row =5;
    for(std_no=0;std_no<=9;++std_no)
    {
        gotoxy(7,row);
        cout<<std_no+1;
        for(subj_no=0;subj_no<=2;++subj_no)
        {
            gotoxy(col,row);
            cout<<score[std_no][subj_no];
            col+=15;
        }
        row++; col=15;
    }
}

```

◆ อาร์เรย์ 3 มิติ

อาร์เรย์ 3 มิติ (3 dimension array) เป็นอาร์เรย์ที่มีอินเด็กซ์อ้างอิงถึงตำแหน่งข้อมูลจำนวน 3 ชุด คือ แถว, คอลัมน์ และแนวลึก ตัวอย่างข้อมูลที่สามารถนำมาเก็บในลักษณะของอาร์เรย์ 3 มิติได้ เช่น คะแนนสอบนักศึกษา จำนวน 5 คน(แถว) วิชาสอบทั้งหมด 3 วิชา(คอลัมน์) แต่ละวิชาสอบ 2 ครั้ง(ลึก) มีข้อมูลดังตารางต่อไปนี้

ข้อมูลดังตารางต่อไปนี้

แนวคอลัมน์ มีทั้งหมด 3 คอลัมน์

| นักศึกษาคนที่ | คะแนนวิชา | | | | | | |
|----------------------------|------------|------------|------------|------------|-------------|------------|----|
| | คณิตศาสตร์ | | ภาษาอังกฤษ | | วิทยาศาสตร์ | | |
| | ครั้งที่ 1 | ครั้งที่ 2 | ครั้งที่ 1 | ครั้งที่ 2 | ครั้งที่ 1 | ครั้งที่ 2 | |
| แนวแถว ทั้งหมด 5 แถว | 1 | 24 | 25 | 28 | 30 | 20 | 35 |
| | 2 | 26 | 28 | 24 | 25 | 30 | 32 |
| | 3 | 31 | 30 | 30 | 29 | 26 | 33 |
| | 4 | 25 | 29 | 25 | 30 | 30 | 35 |
| | 5 | 30 | 35 | 18 | 20 | 32 | 37 |

แต่ละคอลัมน์มีแนวลึกอีก 2 แนวลึก

จากข้อมูลคะแนนทั้งหมดในตาราง เปรียบเทียบกับโครงสร้างข้อมูลอาร์เรย์แบบ 3 มิติ ได้ดังนี้

แนวแถว แทน จำนวนนักศึกษา 5 คน

แนวคอลัมน์ แทน จำนวนวิชา 3 วิชา

แนวลึก แทน จำนวนครั้งที่สอบ 2 ครั้ง

1. การกำหนดข้อมูลอาร์เรย์ 3 มิติ มีรูปแบบการกำหนด ดังนี้

```
Array_Type Array_name[row][column][depth];
```

โดยที่ **Array_Type** คือ ชนิดข้อมูลในอาร์เรย์หรือประเภทของอาร์เรย์ 3 มิติ

Array_name คือ ชื่อของตัวแปรอาร์เรย์ 3 มิติ

[row] คือ จำนวนสมาชิกหรือรายการของอาร์เรย์ในแนวแถว

[column] คือ จำนวนสมาชิกหรือรายการของอาร์เรย์ในแนวคอลัมน์

[depth] คือ จำนวนสมาชิกหรือรายการในแนวลึก

จำนวนสมาชิกของอาร์เรย์ 3 มิติ เท่ากับ $row * column * depth$ เช่น `int score[5][3][2];`

จะมีจำนวนสมาชิกของอาร์เรย์ทั้งหมด $5 * 3 * 2 = 30$ สมาชิก

2. การอ้างถึงสมาชิกอาร์เรย์ 3 มิติ ต้องเขียนชื่ออาร์เรย์และตามด้วยอินเด็กซ์ทั้ง 3 ชุด เช่น
`score[0][1][0];` หมายถึง คะแนนสอบของนักศึกษาคนที่ 1 วิชาที่ 2 สอบครั้งที่ 1 เนื่องจากอินเด็กซ์เริ่มที่ 0

3. การกำหนดค่าให้กับอาร์เรย์ 3 มิติ มีวิธีการกำหนดเช่นเดียวกับอาร์เรย์ 2 มิติ เพียงแต่ต้องเพิ่มตำแหน่งอินเด็กซ์ชุดที่ 3 (แนวลึก) เช่น

```
score[0][1][1] = 25;
```

```
score[4][2][0] = 30;
```

การกำหนดเป็น constant ต้องกำหนดค่าคงที่ให้กับอาร์เรย์ให้ครบทุกช่อง ดังนี้

```
const int score[5][3][2] = {
    { {25,24},{21,23},{24,23} },
    { {24,20},{29,30},{28,29} },
    { {27,32},{24,29},{29,30} },
    { {24,29},{24,30},{25,28} },
    { {24,35},{28,21},{28,23} }
};
```

เราสามารถใส่ลูป `for`, `while` หรือ `do` กำหนดค่าโดยการกรอกข้อมูลทางคีย์บอร์ดได้ โดยใช้คำสั่งลูปซ้อนกัน 3 ลูป ตัวอย่างการใช้ลูป `for` ดังนี้

```
for(std_no=0;std_no<=4;++std_no)
    for(subj_no=0;subj_no<=2;++subj_no)
        for(time_no=0;time_no<=1;++time_no)
            cin>>score[std_no][subj_no][time_no]; //input from
                                                    keyboard
```

4. การแสดงผลจากอาร์เรย์ 3 มิติ สามารถใช้ลูป `for`, `while` หรือ `do` ตัวอย่างเช่น การใช้ลูป `for` แสดงค่าในอาร์เรย์ `score[5][3][2]` เป็นดังนี้

```
for(std_no=0;std_no<=4;++std_no)
    for(subj_no=0;subj_no<=2;++subj_no)
        for(time_no=0;time_no<=1;++time_no)
            cout<<score[std_no][subj_no][time_no]; //display
```

◆ การส่งค่าอาร์เรย์ให้ฟังก์ชัน

ชื่อของอาร์เรย์สามารถใช้เป็นอาร์กิวเมนต์ส่งให้แก่ฟังก์ชันได้ ทำให้สามารถส่งอาร์เรย์ทั้งหมดไปให้แก่ฟังก์ชัน เพื่อนำค่าของอาร์เรย์ไปใช้ในฟังก์ชันได้ แต่วิธีการการส่งอาร์เรย์ไปให้ฟังก์ชันมีข้อแตกต่างจากการส่งตัวแปรธรรมดา คือในการส่งอาร์เรย์ไปให้ฟังก์ชันนั้นจะต้องเขียนเฉพาะชื่อของอาร์เรย์ไม่ต้องกำหนดตัวชี้หรือ index ส่วนในการกำหนดพารามิเตอร์รับค่าของอาร์เรย์นั้น ให้กำหนดตัวแปรชนิดอาร์เรย์รองรับไว้ โดยไม่ต้องระบุจำนวนสมาชิกในอาร์เรย์ ดังตัวอย่างต่อไปนี้

- ตัวอย่างโปรแกรม *Arr_Fun1.CPP* แสดงการส่งค่าของอาร์เรย์ทั้งหมดให้แก่ฟังก์ชัน เพื่อนำข้อมูลในอาร์เรย์ทั้งหมดไปแสดงผล คำนวณผลรวม และคำนวณค่าเฉลี่ย

```
/*Program:Arr_Fun1.Cpp
```

```
Process: display and calculate summation, average of all array
```

```
*/
```

```
#include <iostream.h>
```

```
#include <ctype.h>
```

```
#include <conio.h>
```

```
//prototype function
```

```
void ShowArray(int x[]);
```

```
float AvgArray(int x[]);
```

```
int SumArray(int x[]);
```

```
//global variable
```

```
int number[5]={10,20,30,40,50};
```

```
void main() //main function
```

```
{ clrscr();
```

```
ShowArray(number);
```

```
cout<<"\nSummation of All Array : "<<SumArray(number);
```

```
cout<<"\nAverage of All Array : "<<AvgArray(number);
```

```
getch();
```

```
}
```

ส่งอาร์กิวเมนต์ไปเฉพาะ
ชื่อของอาร์เรย์ number

ส่งอาร์กิวเมนต์ไปเฉพาะ
ชื่อของอาร์เรย์ number

ส่งอาร์กิวเมนต์ไปเฉพาะ
ชื่อของอาร์เรย์ number

void ShowArray(int x[]) //function display all array element

```
{
    int i;
    for(i=0;i<=4;i++)
        cout<<"#"<<i+1<<" : "<<x[i]<<endl;
}
```

พารามิเตอร์รับค่า ไม่ต้องระบุจำนวนสมาชิกของอาร์เรย์

int SumArray(int x[]) //function calculate summation of all array

```
{
    int i,sum=0;
    for(i=0;i<=4;i++)
        sum=sum+x[i];
    return sum;
}
```

พารามิเตอร์รับค่า ไม่ต้องระบุจำนวนสมาชิกของอาร์เรย์

float AvgArray(int x[])//function calculate average of all array

```
{
    int i,sum=0;
    for(i=0;i<=4;i++)
        sum=sum+x[i];
    return sum/5;
}
```

พารามิเตอร์รับค่า ไม่ต้องระบุจำนวนสมาชิกของอาร์เรย์

◆ แบบฝึกหัดท้ายบท

1. ให้เขียนโปรแกรมเพื่อรับค่าจำนวนเต็ม 10 จำนวนทางคีย์บอร์ด แสดงค่าจำนวนเต็มที่กรอกทั้งหมด พร้อมกับหาค่าข้อมูลทีมากที่สุดและข้อมูลน้อยที่สุดของข้อมูลชุดนี้ พร้อมบอกตำแหน่งของอาร์เรย์ที่พบข้อมูลทีมากที่สุดและทีน้อยที่สุดด้วย แสดงผลทางจอภาพ
2. จงสร้าง Matrix ขนาด 3x4 กรอกข้อมูลลงในแต่ละช่อง และคำนวณหาผลรวมในแนวแถว และแนวคอลัมน์ของ Matrix นี้ ดังตัวอย่างผลลัพธ์ต่อไปนี้

| | col1 | col2 | col3 | col4 | Total |
|-------|------|------|------|------|-------|
| row1 | 20 | 30 | 10 | 10 | 70 |
| row2 | 10 | 25 | 5 | 40 | 80 |
| row3 | 20 | 40 | 30 | 30 | 120 |
| Total | 50 | 95 | 45 | 80 | 270 |

กรอกข้อมูล
ทางคีย์บอร์ด

3. จากข้อมูลในตารางต่อไปนี้ ให้เขียนโปรแกรมเพื่อกรอกคะแนน นักศึกษา จำนวน 5 คน จำนวน 3 รายวิชา แต่ละวิชามีคะแนนสอบ 2 ครั้ง และให้หาคะแนนเฉลี่ยของการสอบแต่ละครั้ง ทั้ง 3 รายวิชา แล้วแสดงรายงานทางจอภาพ ดังนี้ (เส้นตารางไม่ต้องสร้าง)

| นักศึกษาคนที่ | คะแนนวิชา | | | | | |
|---------------|------------|------------|------------|------------|-------------|------------|
| | คณิตศาสตร์ | | ภาษาอังกฤษ | | วิทยาศาสตร์ | |
| | ครั้งที่ 1 | ครั้งที่ 2 | ครั้งที่ 1 | ครั้งที่ 2 | ครั้งที่ 1 | ครั้งที่ 2 |
| 1 | 24 | 25 | 28 | 30 | 20 | 35 |
| 2 | 26 | 28 | 24 | 25 | 30 | 32 |
| 3 | 31 | 30 | 30 | 29 | 26 | 33 |
| 4 | 25 | 29 | 25 | 30 | 30 | 35 |
| 5 | 30 | 35 | 18 | 20 | 32 | 37 |
| รวม | 999 | 999 | 999 | 999 | 999 | 999 |
| เฉลี่ย | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 |

4. ให้เขียนโปรแกรมรับข้อมูลตัวเลขจำนวนเต็ม 10 จำนวน และสร้างฟังก์ชันในการจัดเรียงข้อมูล 10 จำนวนนี้ คือ Ascending() เรียงข้อมูลจากน้อยไปมาก ฟังก์ชัน Descending() เรียงข้อมูลจากมากไปน้อย
5. ให้เขียนโปรแกรมรับข้อมูลตัวอักษร 5 ตัว และสร้างฟังก์ชันในการจัดเรียงข้อมูล 5 อักษรนี้ คือฟังก์ชัน Ascending() เรียงข้อมูลจากน้อยไปมาก ฟังก์ชัน Descending() เรียงข้อมูลจากมากไปน้อย

