

## อาร์เรย์

ในการเขียนโปรแกรมมีจำเป็นบ่อยครั้งที่ต้องมีการประมวลผลข้อมูลหลายๆ ตัว ที่มีลักษณะข้อมูลชนิดเดียวกัน เช่น ในระบบฐานข้อมูลของนักศึกษา จะมีการเก็บชื่อนักศึกษาตามจำนวนนักศึกษาในแต่ละห้องเรียน พบว่าชื่อของนักศึกษา เป็นชนิดข้อมูลเดียวกันคือเป็นชนิดตัวอักษร หรือในระบบของทะเบียนราษฎร มีการกำหนดชื่อเจ้าบ้านที่อยู่ วันเกิด วันที่ทำบัตร วันที่บัตรสิ้นอายุ ในแต่ละประเด็นที่ทำการเก็บ จะมีการเก็บจำนวนสมาชิกตามจำนวนที่ต้องการได้ โดยข้อมูลที่ป้อนให้กับตัวแปรของแต่ละประเด็นเป็นข้อมูลชนิดเดียวกันเสมอ ในการจัดเก็บข้อมูลควรเก็บข้อมูลที่เป็นชนิดเดียวกันโดยกำหนดให้มีชื่อเดียวแต่มีการเก็บข้อมูลได้หลายค่า เรียกตัวแปรชุดที่กำหนดเป็นลำดับข้อมูลนี้ว่า อาร์เรย์ (Alexander, A. ,Online)

สมาชิกของอาร์เรย์ (คือข้อมูลแต่ละตัวของตัวแปรนั้นๆ) ในการอ้างตำแหน่งของอาร์เรย์ หรืออินเด็กซ์ จะมีลักษณะการกำหนดโดยการเขียนชื่ออาร์เรย์ตามด้วยซับสคริปต์ (subscript) เป็นเลขบอกที่อยู่ที่ต้องการติดต่อกับอาร์เรย์ ในการติดต่อกับอาร์เรย์ หมายความว่า การกำหนดค่าให้อาร์เรย์ หรือเป็นการกำหนดค่าข้อมูลอาร์เรย์ที่ต้องการดึงออกมาใช้งาน กล่าวในรายละเอียดต่อไป ซึ่งอาร์เรย์สามารถแบ่งตามมิติของการเก็บข้อมูลได้ 3 ประเภท (Steven และ Lutfar, 2006) (Brian W. K., Online) ดังนี้

## อาร์เรย์ 1 มิติ (One Dimension Array)

อาร์เรย์ 1 มิติ เป็นตัวแปรชุด ที่มีเลขอ้างอิงตำแหน่งของตัวแปรเพียงตำแหน่งเดียว การนิยามอาร์เรย์ เหมือนกับการนิยามตัวแปรทั่วไป โดยการนิยามอาร์เรย์ต้องระบุขนาดอาร์เรย์ หลังชื่ออาร์เรย์ ภายในเครื่องหมายวงเล็บ []

### รูปแบบ

ชนิดของอาร์เรย์ ชื่ออาร์เรย์ [ขนาด];

ขนาด n จำนวน : มีความสัมพันธ์กับเลขอ้างอิงตำแหน่งเป็น n-1

### ตัวอย่างเช่น

```
int a[10];
```

หมายถึง การประกาศอาร์เรย์ชื่อว่า 'a' เป็นชนิดจำนวนเต็ม (integer) มีขนาดของจำนวนสมาชิกเท่ากับ 10 หน่วย นั่นคือจะมีอาร์เรย์ 10 ตัว สามารถแสดงเป็นลักษณะการจัดเก็บดังรูปที่ 6.1

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
ค่า	ค่า	ค่า	ค่า	ค่า	ค่า	ค่า	ค่า	ค่า	ค่า
ข้อมูล	ข้อมูล	ข้อมูล	ข้อมูล	ข้อมูล	ข้อมูล	ข้อมูล	ข้อมูล	ข้อมูล	ข้อมูล
(int)	(int)	(int)	(int)	(int)	(int)	(int)	(int)	(int)	(int)

รูปที่ 6.1 แสดงการเก็บอาร์เรย์ชนิดจำนวนเต็ม 1 มิติ

หรือ `char c[5];`

**หมายถึง** การประกาศอาร์เรย์ชื่อตัวแปรว่า 'c' เป็นชนิดอักขระ (character) มีขนาดของจำนวนสมาชิกเท่ากับ 5 หน่วย นั่นคือจะมีอาร์เรย์ 5 ตัว สามารถแสดงเป็นลักษณะการจัดเก็บรูปที่ 6.2

c[0]	c[1]	c[2]	c[3]	c[4]
ค่าข้อมูล (char)	ค่าข้อมูล (char)	ค่าข้อมูล (char)	ค่าข้อมูล (char)	ค่าข้อมูล (char)

รูปที่ 6.2 แสดงการเก็บอาร์เรย์ชนิดตัวอักขระชนิด 1 มิติ

การนิยามอาร์เรย์ โดยกำหนดค่าเริ่มต้น สามารถกระทำได้ด้วยการให้ค่าของสมาชิกอยู่ในเครื่องหมายวงเล็บปีกกา {} ใช้เครื่องหมายจุดภาค (comma) แยกค่าของสมาชิกแต่ละค่า ดังตัวอย่าง

```
int digit1[5] = {1,2,3,4,5} ;
char txt1[5] = {'a', 'b', 'c', 'd', 'e'};
float z1[5] = {0, 0.10, 0.05, 1.5, 500.55 };
```

```
int digit2[10] = {1,2,3,4,5} ;
char txt2[10] = {'a', 'b', 'c', 'd', 'e'};
float z2[10] = {0, 0.10, 0.05, 1.5, 500.55 };
```

สามารถเขียนการจัดเก็บข้อมูลของแต่ละอาร์เรย์ได้ดังรูปที่ 6.3 – 6.8 ตามลำดับ

digit1[0]	digit1[1]	digit1[2]	digit1[3]	digit1[4]
1	2	3	4	5

รูปที่ 6.3 แสดงการเก็บอาร์เรย์ int digit1[5]

txt1[0]	txt1[1]	txt1[2]	txt1[3]	txt1[4]
a	B	C	d	e

รูปที่ 6.4 แสดงการเก็บอาร์เรย์ char txt1[5]

z1[0]	z1[1]	z1[2]	z1[3]	z1[4]
0	0.10	0.05	1.5	500.55

รูปที่ 6.5 แสดงการเก็บอาร์เรย์ float z1[5]

digit2[0]	digit2[1]	digit2[2]	digit2[3]	digit2[4]	digit2[5]	digit2[6]	digit2[7]	digit2[8]	digit2[9]
1	2	3	4	5	0	0	0	0	0

รูปที่ 6.6 แสดงการเก็บอาร์เรย์ int digit2[10]

txt2[0]	txt2[1]	txt2[2]	txt2[3]	txt2[4]	txt2[5]	txt2[6]	txt2[7]	txt2[8]	txt2[9]
A	b	c	d	E	null*	null*	null*	null*	null*

null\* หมายถึง ว่างเปล่า ไม่มีข้อมูล (แตกต่างจากค่าศูนย์ (0) มีค่าทางตัวอักษรของเลขฐาน 16)

รูปที่ 6.7 แสดงการเก็บอาร์เรย์ char txt2[10]

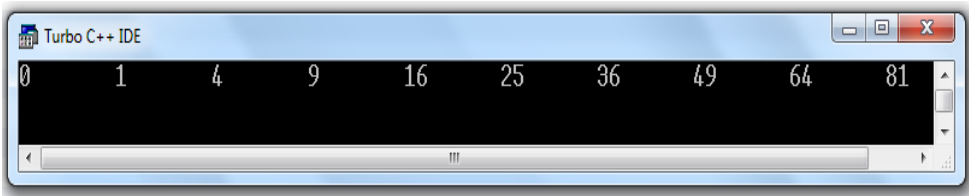
z2[0]	z2[1]	z2[2]	z2[3]	z2[4]	z2[5]	z2[6]	z2[7]	z2[8]	z2[9]
0	0.10	0.05	1.5	500.55	0	0	0	0	0

รูปที่ 6.8 แสดงการเก็บอาร์เรย์ float z2[10]

ตัวอย่าง โปรแกรมที่ 6.1

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int square[10], c;
    for(c = 0; c < 10; c++)
    {
        square[c] = c*c;
        printf("%d\t",square[c]);
    }
    getch();
}
```

## ผลการทำงานโปรแกรม



จากผลทำงานโปรแกรม เป็นการกำหนดค่าให้กับอาร์เรย์ square เริ่มจากค่าศูนย์ และเพิ่มค่าทีละหนึ่ง จบครบเงื่อนไขของคำสั่ง for ว่า  $c > 10$  โดยมีแสดงค่าออกสู่หน้าจอ แต่ละค่าทำการนำค่าของค่าตัวแปร  $c$  กับ  $c$  นำมาคูณกัน ได้ผลแต่ละค่าที่ได้ทำการคูณยกกำลังแล้วนำค่าแสดงโดยมีการ  $\backslash t$  เป็นการย่อหน้าระหว่างตัวอักษรด้วยคำสั่ง

### ตัวอย่าง โปรแกรมที่ 6.2

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int a[5], i;
    for(i = 0; i < 5 ; i++)
    {
        printf("Enter number ");
        scanf("%d",&a[i]);
    }
    for(i = 4; i >= 0; i--)
        printf("\n%d\t",a[i]);
    getch();
}
```

## ผลการทำงานโปรแกรม

```

Turbo C++ IDE
Enter number 11
Enter number 22
Enter number 33
; Enter number 44
Enter number 55

55
44
33
22
11

```

โปรแกรมการรับค่าจากแป้นพิมพ์ของผู้ใช้งานทำการป้อนค่า ด้วยตัวเลข 5 จำนวน ดังตัวอย่างผลการทำงานโปรแกรม ค่าตัวเลขที่ทำการป้อนคือ 11 22 33 44 และ 55 ตามลำดับ ไปเก็บไว้ที่อาร์เรย์  $a[i]$  โดยค่า  $i$  ทำหน้าที่เป็นตัวอินเด็กซ์ของอาร์เรย์ ขั้นตอนถัดมาของ ผลการทำงานของโปรแกรมจะทำการแสดงค่าที่ผู้ใช้งานได้ส่งเก็บไว้ที่อาร์เรย์  $a[i]$  ออกมาแสดง โดยแสดงค่าในลำดับท้ายที่ป้อนเข้าไปเก็บนำมาแสดงเป็นลำดับแรกจนครบ 5 จำนวน ตามที่ได้กำหนดให้กับอาร์เรย์  $a[i]$

## อาร์เรย์ 2 มิติ (Two Dimension Array)

อาร์เรย์ 2 มิติ เป็นตัวแปรชุดที่มีการจัดข้อมูลเป็นแถว (row) เป็นคอลัมน์ (column) ลักษณะการของข้อมูลแบบ เมตริก (matrix) ก็จะมีตัวเลขที่แสดงตำแหน่งอ้างอิงค่าใน อาร์เรย์ 2 ตัว โดยแรกจะเป็นตัวเลขที่แสดงตำแหน่งแถว ส่วนตัวที่ 2 จะเป็นตัวเลขที่แสดง ตำแหน่งคอลัมน์

### รูปแบบ

ชนิดของอาร์เรย์ ชื่ออาร์เรย์ [ตำแหน่งแถว] [ตำแหน่งคอลัมน์];

### ตัวอย่างเช่น

```
int var[2][3];
```

### หมายถึง

การประกาศตัวแปร var ให้เป็นตัวแปรแบบอาร์เรย์และเป็น ตัวแปรชนิดจำนวนเต็ม (integer) โดยมีลักษณะการจัดเก็บ ค่าข้อมูล รูปที่ 6.9

var[0][0]	var[0][1]	var[0][2]
ค่าข้อมูล	ค่าข้อมูล	ค่าข้อมูล
ค่าข้อมูล	ค่าข้อมูล	ค่าข้อมูล
var[1][0]	var[1][1]	var[2][2]

รูปที่ 6.9 ตัวแปรชุดแบบ 2 มิติ

การนิยามอาร์เรย์ 2 มิติ สามารถกระทำได้ด้วยการให้ค่าของสมาชิกอยู่ภายในเครื่องหมาย วงเล็บปีกกา {} ชุดแรกเป็นข้อมูลมิติแถว ใช้เครื่องหมายจุลภาค (,) แยกค่าของสมาชิกแต่ละค่า และแยกแถวของข้อมูล ดังตัวอย่าง



```
float mat [2] [5] = {{0.5, 1.5, 2.5, 3.5, 4.5}, {5.5, 6.5, 7.5, 8.5, 9.5}};
```

หมายถึง การกำหนดค่าให้กับตัวแปรอาร์เรย์แบบ 2 มิติให้เก็บค่าดังรูปที่ 6.10

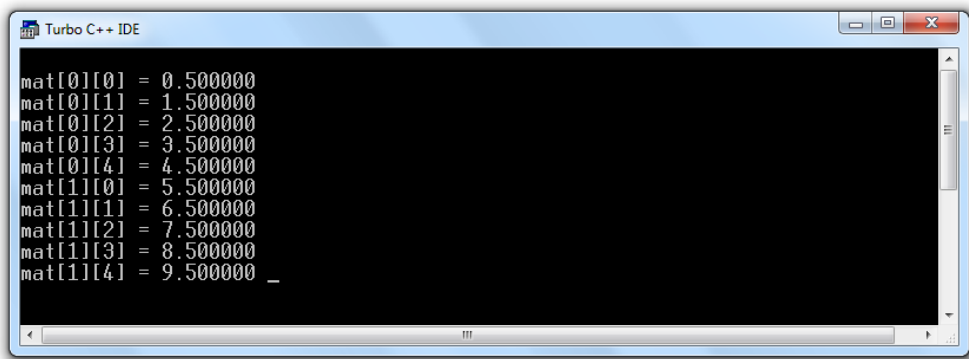
mat[0][0]	mat[0][1]	mat[0][2]	mat[0][3]	mat[0][4]
0.5	1.5	2.5	3.5	4.5
5.5	6.5	7.5	8.5	9.5
mat[1][0]	mat[1][1]	mat[1][2]	mat[1][3]	mat[1][4]

รูปที่ 6.10 การกำหนดค่าให้กับอาร์เรย์แบบ 2 มิติ

ตัวอย่าง โปรแกรม 6.3 โปรแกรมทำการกำหนดค่าให้อาร์เรย์ 2 มิติ และแสดงผลการเก็บข้อมูล 2 มิติ

```
#include<stdio.h>
#include<conio.h>
float mat[2][5] = {{0.5,1.5,2.5,3.5,4.5},{5.5,6.5,7.5,8.5,9.5}};
int i, j;
void main()
{
    clrscr();
    for(i=0;i<2;i++)
        {
            for(j=0;j<5;j++)
                { printf("\nmat[%d][%d] = %f",i,j,mat[i][j]); }
        }
    getch(); }
```

## ผลการทำงานโปรแกรม



```

Turbo C++ IDE
mat[0][0] = 0.500000
mat[0][1] = 1.500000
mat[0][2] = 2.500000
mat[0][3] = 3.500000
mat[0][4] = 4.500000
mat[1][0] = 5.500000
mat[1][1] = 6.500000
mat[1][2] = 7.500000
mat[1][3] = 8.500000
mat[1][4] = 9.500000 _
  
```

การทำงานของโปรแกรม มีการกำหนดค่าให้กับอาร์เรย์ 2 มิติ `mat[2][5]` มีขนาด 2 แถว 5 คอลัมน์ ส่วนในการแสดงค่าที่จัดเก็บไว้ในอาร์เรย์ ใช้คำสั่ง `for()` ที่ใช้ตัวแปร `i` เป็นตัวกำหนดมิติแถวมีจำนวน 2 แถวคือ แถว 0 กับแถว 1 ส่วนตัวแปร `j` เป็นตัวกำหนดมิติคอลัมน์ มีจำนวน 5 คอลัมน์คือ คอลัมน์ 0 ถึง คอลัมน์ 4 โดยค่าที่แสดงผลของข้อมูลอาร์เรย์ได้จาก `mat[i][j]`

### อาร์เรย์หลายมิติ (Multi Dimension Array)

อาร์เรย์หลายมิติ เช่น อาร์เรย์หลายมิติ มีวิธีการการนิยามคล้ายกับอาร์เรย์หนึ่งมิติ ที่ใช้วงเล็บก้ามปูหนึ่งคู่สำหรับข้อมูลหนึ่งชุด ถ้าอาร์เรย์สองมิติต้องใช้วงเล็บก้ามปูสองชุด อาร์เรย์สามมิติใช้วงเล็บก้ามปูสามชุดเช่นกันในการกำหนดค่าให้กับอาร์เรย์ โดยการกำหนดลำดับของวงเล็บก้ามปูทั้งสามคือ วงเล็บก้ามปูตัวแรก กำหนดให้เป็นมิติตาราง วงเล็บก้ามปูตัวที่สอง กำหนดให้เป็นมิติแถว และวงเล็บก้ามปูตัวที่สาม กำหนดให้เป็นมิติคอลัมน์ เช่น `Arr[3][10][5]` หมายความว่า อาร์เรย์ชื่อ `Arr` ประกอบด้วย 3 มิติ มีจำนวน 3 ตาราง 10 แถว 5 คอลัมน์ ตัวอย่างการประกาศอาร์เรย์ 3 มิติเช่น

```

int arr[3][2][4] = {      {          //ตารางที่ 1
                          {1, 2, 3, 4},
                          {5, 6, 7, 8}
                        },
                      {          //ตารางที่ 2
                          {21, 22, 23, 24},
                          {25, 26, 27, 28,}
                        },
                      {          //ตารางที่ 3
                          {321, 322, 323, 324},
                          {325, 326, 327, 328}
                        },
                      };

```

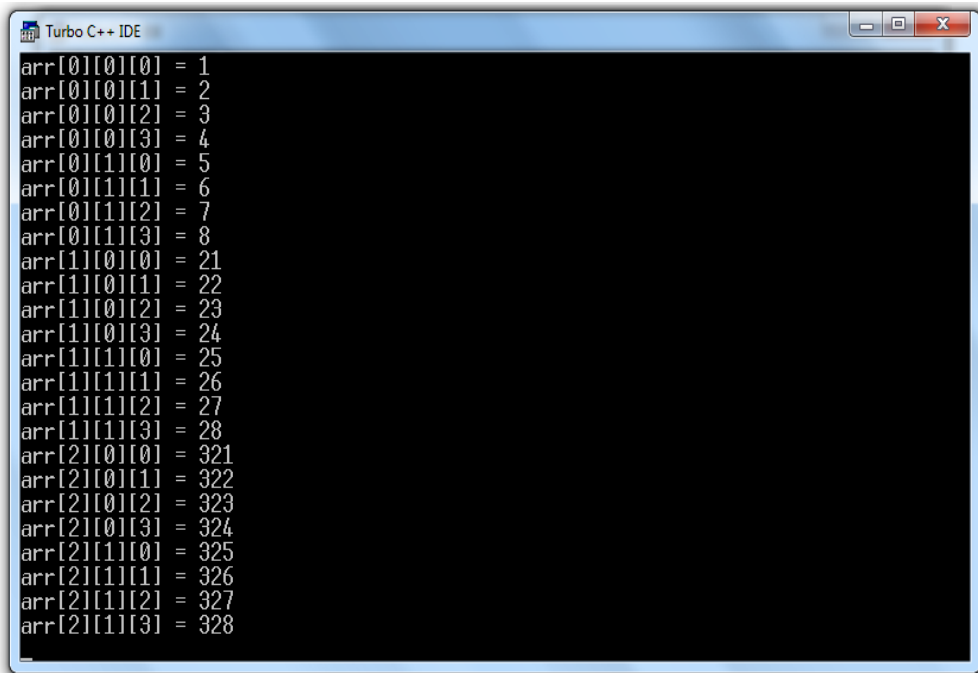
จากการประกาศอาร์เรย์ arr ขนาด 3 มิติ สำหรับเก็บสมาชิกจำนวน 3 ตาราง โดยตารางที่ 1 ประกอบด้วยสมาชิกแถวที่ 1 มี 4 คอลัมน์คือ {1, 2, 3, 4} สมาชิกแถวที่ 2 มี 4 คอลัมน์คือ {5, 6, 7, 8} ตารางที่ 2 ประกอบด้วยแถวที่ 1 มี 4 คอลัมน์คือ {21, 22, 23, 24} สมาชิกแถวที่ 2 มี 4 คอลัมน์คือ {25, 26, 27, 28} และตารางที่ 3 ประกอบด้วยแถวที่ 1 มี 4 คอลัมน์คือ {321, 322, 323, 324} สมาชิกแถวที่ 2 มี 4 คอลัมน์คือ {325, 326, 327, 328}

## ตัวอย่างโปรแกรม 6.4 โปรแกรมทำการกำหนดค่าให้อาร์เรย์ 3 มิติ และแสดงผลการเก็บข้อมูล 3 มิติ

```
#include<stdio.h>
#include<conio.h>
int arr[3][2][4] = { { {1,2,3,4},{5,6,7,8} },
                    { {21,22,23,24},{25,26,27,28} },
                    { {321,322,323,324},{325,326,327,328} }
                    };

int m,n,k;
void main()
{   clrscr();
    for(m=0;m<3;++m)
    {
        for(n=0;n<2;++n)
        {
            for(k=0;k<4;++k)
            printf("arr[%d][%d][%d] = %d\n ",m,n,k,arr[m][n][k]);
        }
    }
    getch();
}
```

## ผลการทำงานโปรแกรม



```

arr[0][0][0] = 1
arr[0][0][1] = 2
arr[0][0][2] = 3
arr[0][0][3] = 4
arr[0][1][0] = 5
arr[0][1][1] = 6
arr[0][1][2] = 7
arr[0][1][3] = 8
arr[1][0][0] = 21
arr[1][0][1] = 22
arr[1][0][2] = 23
arr[1][0][3] = 24
arr[1][1][0] = 25
arr[1][1][1] = 26
arr[1][1][2] = 27
arr[1][1][3] = 28
arr[2][0][0] = 321
arr[2][0][1] = 322
arr[2][0][2] = 323
arr[2][0][3] = 324
arr[2][1][0] = 325
arr[2][1][1] = 326
arr[2][1][2] = 327
arr[2][1][3] = 328

```

จากการศึกษาอาร์เรย์ ที่ขนาดต่างๆ ทั้งอาร์เรย์ขนาด 1 มิติ อาร์เรย์ 2 มิติ และอาร์เรย์ 3 มิติ พบว่าจำนวนของวงเล็บก้ามปูที่ใช้ในการกำหนดขนาดของอาร์เรย์ จากการประกาศอาร์เรย์ `arr[3][2][4]` ขนาดของวงเล็บก้ามปู ขวามือสุดคือ อาร์เรย์ `arr[3]` เป็นอาร์เรย์ที่ใช้ในการกำหนดคอลัมน์ จะถูกเรียกหรือทำการหมุนข้อมูลเร็วสุดและถูกเรียกถี่ที่สุด สังเกตจากผลการทำงานโปรแกรมด้านบน และจากตัวอย่างโปรแกรม 6.4 ในส่วนของ การเรียกใช้ฟังก์ชัน `for()` ชั้นในสุดก่อนคือ `for(k=0;k<4;++k)` ส่วนมิติอาร์เรย์ถัดมา คือ อาร์เรย์ `arr[2]` เป็นอาร์เรย์ที่ใช้ในการกำหนดแถว จะที่ถูกเรียกใช้บ่อยรองลงมา ด้วยการเรียกฟังก์ชัน `for(n=0;n<2;++n)` ส่วนมิติที่กำหนดให้เป็น ตารางของอาร์เรย์ ถูกเรียกใช้น้อยที่สุด คือ อาร์เรย์ `arr[3]` เพราะเป็นเรียกฟังก์ชัน `for(m=0;m<3;++m)` เป็นลูปวงนอกสุดของโปรแกรม

## ข้อสังเกต

ในการกำหนดค่าข้อมูลให้กับอาร์เรย์ทุกขนาดมิติ จำนวนของค่าข้อมูลสามารถ มีจำนวนน้อยกว่าจำนวนของขนาดในแต่ละมิติได้ โดยค่าที่ไม่ได้กำหนดให้กับอาร์เรย์ จะถูกกำหนดค่าให้มีค่าเป็น 0 (ศูนย์) ทันที เช่น

```
int a[4] = {10,20,30}; // โปรแกรมสามารถทำการแปลคำสั่งได้
```

ค่าข้อมูลของอาร์เรย์  $a[ ]$  จะมีสมาชิกดังนี้  $a[0]= 10$  ค่า  $a[1]= 20$  ค่า  $a[2]= 30$  ส่วนมีค่า  $a[3]= 0$  เช่น

```
int a[4] = {10,20,30,40,50};
```

เมื่อทำการสั่งให้แปลโปรแกรมจะเกิดการผิดพลาด (error) เพราะมีการกำหนดค่าข้อมูลเกิน คือ 50 ไม่สามารถกำหนดให้เป็นสมาชิกในตำแหน่งใดได้ ด้วย  $a[4]$  มีการกำหนดให้มีสมาชิกสูงสุดได้ 4 จำนวน และจำนวนสุดท้ายคือ  $a[3] = 40$  นั่นเอง

## สรุป

ในการเก็บข้อมูลในชีวิตประจำวัน รวมทั้งในระบบหน่วยความจำของคอมพิวเตอร์ มักจะทำการเก็บข้อมูลชนิดเดียวกัน ให้อยู่ในชุดหรือกลุ่มเดียวกัน เพื่อความสะดวกในการค้นหา นำมาใช้งานในภายหลัง การเก็บข้อมูลในระบบหน่วยความจำของคอมพิวเตอร์ เป็นการเก็บข้อมูล ที่เป็นอาร์เรย์ คือเป็นการนำข้อมูลชนิดเดียวกันมาเก็บเป็นลักษณะการวางซ้อน หรือเรียงต่อกันเป็นลำดับติดกันเป็นชุด ถ้าเป็นการเก็บเป็นชุดเดียวกันในหนึ่งแถว หรือหนึ่งคอลัมน์ เราเรียกว่า อาร์เรย์ 1 มิติ ส่วนอาร์เรย์ 2 มิติเป็นการนำทั้งข้อมูลในลักษณะเป็นแถว และเป็นคอลัมน์ มาจัดเก็บเป็นลักษณะตาราง โดยกำหนดให้เป็นข้อมูลชุดเดียวกันหรือตัวแปรเดียวกัน ส่วนอาร์เรย์ 3 มิติเป็นการกำหนดให้ใน ตัวแปรอาร์เรย์ 1 ตัวแปร สามารถเก็บข้อมูลได้มากกว่าหนึ่งตาราง เป็นต้น

## แบบฝึกหัดท้ายบทที่ 6

### ตอนที่ 1 จงเติมคำหรือข้อความในช่องว่างต่อไปนี้ให้ถูกต้อง

1. ให้ผู้อ่านเขียนความแตกต่างระหว่างอาร์เรย์กับตัวแปรทั่วไปอย่างไร  
.....
2. การกำหนดตำแหน่งของสมาชิกอาร์เรย์มีวิธีการดำเนินการอย่างไร เขียนตัวอย่างมา  
หนึ่งตัวอย่าง  
.....
3. ให้ผู้อ่านกำหนดค่าเริ่มต้นอาร์เรย์ขนาด 1 มิติโดยมีจำนวน 15 สมาชิก  
.....
4. ให้ผู้อ่านกำหนดค่าเริ่มต้นขนาด 2 มิติโดยมีจำนวน 5 สมาชิกและ 6 สมาชิก  
ตามลำดับ  
.....
5. ให้ผู้อ่านกำหนดค่าเริ่มต้นขนาด 3 มิติโดยมีจำนวนสมาชิกตามผู้อ่านกำหนด  
.....
6. ให้ผู้อ่านอธิบายของอาร์เรย์ `char a[10];`  
.....
7. ให้ผู้อ่านอธิบายของอาร์เรย์ `int b[5][10];`  
.....
8. ให้ผู้อ่านอธิบายของอาร์เรย์ `double score[4][8][ ];`  
.....
9. ให้ผู้อ่านอธิบายของอาร์เรย์ `int score[4][5][4];`  
.....
10. ให้ผู้อ่านอธิบายของอาร์เรย์ `int score[0][0][0];`  
.....

## ตอนที่ 2 จงทำเครื่องหมายกากบาท (x) ทับหน้าข้อที่ถูกต้องที่สุด

โดย อาร์เรย์ คือ `int x[5] = {10, 20, 30, 'a', 'A'};` ใช้ตอบข้อที่ 4-7

และ `int y[2][5] = [{1, 2, 3, 4, 5}, {10, 20, 30, 40, 50}];` ใช้ตอบข้อที่ 8-10

- คำสั่งใดเขียนผิดหลักการกำหนดตัวแปรชนิด array
 

ก. <code>int a[];</code>	ข. <code>int a[12];</code>
ค. <code>int a[2,3];</code>	ง. <code>int a[]={1,2,3};</code>
- คำสั่งใดเขียนผิดหลักการกำหนดตัวแปรชนิด array
 

ก. <code>int b[ ][ ];</code>	ข. <code>int b[12][ ];</code>
ค. <code>int b[2],[3];</code>	ง. <code>char b[ ]={'1','2'};</code>
- คำสั่งใดเขียนผิดหลักการกำหนดตัวแปรชนิด array
 

ก. <code>char c[ ][ ];</code>	ข. <code>char c[12][ ];</code>
ค. <code>char c[2][3]={1,2};</code>	ง. <code>char c[ ]={'1','2'};</code>
- อยากทราบว่าข้อมูลของ `x[1]` เป็นเท่าใด
 

ก. 10	ข. 30
ค. 97	ง. 41
- อยากทราบว่าข้อมูลของ `x[2]` เป็นเท่าใด
 

ก. 10	ข. 30
ค. 97	ง. 41
- อยากทราบว่าข้อมูลของ `x[3]` เป็นเท่าใด
 

ก. 10	ข. 30
ค. 97	ง. 41
- อยากทราบว่าข้อมูลของ `x[4]` เป็นเท่าใด
 

ก. 10	ข. 30
ค. 97	ง. 41





2. ให้ผู้อ่านคิดและเขียนผลการทำงานของโปรแกรม ดังนี้

```
#include<stdio.h>

void main()
{
    int n, m=0;
    int z[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};
    for(n=0; n<5;n++)
        if(n %2 == 0) m += z[n];
    printf(“%d”,m);
}
```

3. ให้ผู้อ่านคิดและเขียนผลการทำงานของโปรแกรม ดังนี้

```
#include<stdio.h>

void main()
{
    int n, m=0;
    int z[5] = {5, 4, 3, 2, 1};
    for(n=0; n<5;n++)
        if(z[n] %2 == 1) m += z[n];
    printf(“%d”,m);
}
```

4. ให้ผู้อ่านคิดและเขียนผลการทำงานของโปรแกรม ดังนี้

```
#include<stdio.h>

void main()
{
    int n, m=0;
    int z[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};
    for(n=0; n<5;n++)
        if(n %2 == 1) m += z[n];
    printf(“%d”,m);
}
```

5. ให้ผู้อ่านคิดและเขียนผลการทำงานของ โปรแกรม ดังนี้

```
#include<stdio.h>
#define R = 3
#define C = 4
int z[R][C] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10,11 12};
void main()
{
    int n, m, q =999;
    for (n=0; n<R;n++)
        for(m =0 ;m <C; m++)
            if(z[n][m] <q == 0) q += z[n][m];
    printf(“%d”,q);
}
```

## เอกสารอ้างอิง

- ศรัณย์ อินทโกสุม (2539). ทฤษฎีและตัวอย่างโจทย์การเขียนโปรแกรมด้วยภาษาซี  
กรุงเทพฯ : แมคกรอฮิล อินเทอร์เน็ต เนชั่นแนล เอ็นเตอร์ไพรส์, ینگค์.
- ชันวา ศรีประโมง (2539). การเขียนโปรแกรมภาษาซีสำหรับวิศวกรรม. พิมพ์ครั้งที่  
ที่ 4. กรุงเทพฯ : มหาวิทยาลัยเทคโนโลยีมหานคร.
- วิจักขณ์ ศรีสังจะเลิศวาจา และคุณฎี ประเสริฐฐิติพงษ์ ออนไลน์ :  
[www.satit.su.ac.th/soottin](http://www.satit.su.ac.th/soottin).
- Brian, W. K. Programming in C: A Tutorial Online :  
<http://www.lysator.liu.se/c/bwktutor.html>.
- Byron S. Gottfried, “Schaum ’s Theory and problems of programming with c”  
McGraw-Hill, Inc., 1990.
- Steven, H. & Lutfar, R. (2006). Art of Programming Contest: C Programming |  
Data Structure | Algorithms (ACM supported), 2nd Edition.