

### หน่วยที่ 3 ตัวแปรชนิดอาร์เรย์และสตริง

ตัวแปรประเภทหนึ่งที่ใช้ชื่อตัวแปรชื่อเดียว แต่สามารถเก็บข้อมูลเป็นกลุ่มได้ เรียกว่า ตัวแปรแบบอาร์เรย์ ถ้าหากประกาศตัวแปรประเภทนี้ขึ้นมาผู้ใช้สามารถเก็บข้อมูลหลาย ๆ ค่าติด ๆ กันได้ และสามารถเรียกข้อมูลแต่ละค่าขึ้นมาได้ ตัวแปรแบบอาร์เรย์นี้มีทั้งแบบหนึ่งมิติและหลายมิติ สำหรับตัวแปรแบบสตริงก็คือตัวแปรที่นำตัวอักขระมาต่อกันเป็นอาร์เรย์ประเภทหนึ่ง

ประเภทของข้อมูลที่เราเรียนมาเป็นข้อมูลแบบข้อมูลเดี่ยว โดยตัวแปรหนึ่งตัวสามารถเก็บข้อมูลได้หนึ่งตัว ถ้าหากต้องการเก็บข้อมูลหลายตัวจะต้องประกาศตัวแปรหลายตัว ซึ่งจะไม่สะดวกในการเก็บข้อมูลจำนวนมาก ในภาษาซีถ้าหากต้องการเก็บข้อมูลเป็นกลุ่มจะต้องใช้ตัวแปรประเภทอาร์เรย์ (Array) โดยอาร์เรย์จะทำหน้าที่จองเนื้อที่ในหน่วยความจำตามขนาดที่ระบุ และแบ่งหน่วยความจำนั้นออกเป็นช่อง ๆ ถ้าหากต้องการใช้หน่วยความจำตัวใดก็สามารถอ้างอิงหน่วยความจำนั้นแล้วดึงมาใช้ได้

1. ข้อมูลชนิดอาร์เรย์
2. ตัวแปรอาร์เรย์ 1 มิติ
3. ตัวแปรอาร์เรย์ 2 มิติและหลายมิติ
4. ข้อมูลชนิดสตริง

#### 1. ข้อมูลชนิดอาร์เรย์

อาร์เรย์ประกอบด้วยข้อมูลหลาย ๆ ตัวรวมกันเป็นกลุ่ม ข้อมูลแต่ละตัวในกลุ่มนั้นจะเรียกว่า **อีลีเมนต์ (Element)** หรือ **เซลล์ (Cell)** ในการอ้างอิงถึงข้อมูลในแต่ละเซลล์จะใช้ **อินเด็กซ์ (Index)** เป็นตัวชี้ตัวอย่างเช่น ถ้ามีข้อมูลอยู่กลุ่มหนึ่งซึ่งเป็นคะแนนของนักเรียน 8 คน สามารถเก็บข้อมูลได้ดังนี้

|   | X[0] | X[1] | X[2] | X[3] | X[4] | X[5] | X[6] | X[7] | ← Index |
|---|------|------|------|------|------|------|------|------|---------|
| X | 18   | 20   | 35   | 84   | 21   | 45   | 65   | 71   |         |

Array Length = 8

First Index = 0

Last Index = 7

จากข้อมูลข้างต้นสามารถบอกได้ว่า ข้อมูลอยู่ในอาร์เรย์ชื่อ X ในแต่ละเซลล์จะเก็บตัวเลขจำนวนเต็ม ส่วนตัวเลขที่อยู่ในเครื่องหมาย Square brackets ([ ]) เรียกว่า **อินเด็กซ์ (Index)** ซึ่งจะต้องเป็นข้อมูลประเภทจำนวนเต็มเท่านั้น ถ้าหากต้องการติดต่อกับเซลล์ใดก็ให้ใช้อินเด็กซ์เป็นตัวชี้ ถ้าหากเราอ้างตัวแปรอาร์เรย์ X[3] หมายความว่า เป็นการติดต่อกับอาร์เรย์ชื่อ X อ้างอิงถึงข้อมูลในเซลล์อินเด็กซ์ที่ 3

ตัวอย่าง

|             |                                                                                       |
|-------------|---------------------------------------------------------------------------------------|
| X[2]        | อ้างอิงถึงข้อมูลในเซลล์อินเด็กซ์ที่ 2 มีค่าเท่ากับ 35                                 |
| X[2] + X[3] | อ้างอิงถึงข้อมูลในเซลล์อินเด็กซ์ที่ 2 บวกกับเซลล์ที่ 3 จะได้ 35 + 84 มีค่าเท่ากับ 119 |
| X[1+3]      | อ้างอิงถึงข้อมูลในเซลล์อินเด็กซ์ที่ 4 มีค่าเท่ากับ 21                                 |
| X[5] + 1    | อ้างอิงถึงข้อมูลในเซลล์อินเด็กซ์ที่ 5 มาบวกด้วย 1 มีค่าเท่ากับ 46                     |

### การอ้างอิงถึงข้อมูลในตัวแปรอาร์เรย์หนึ่งมิติ

ตัวแปรอาร์เรย์เป็นตัวแปรที่เก็บข้อมูลเป็นชุด ดังนั้นภายในตัวแปรอาร์เรย์จึงประกอบไปด้วยข้อมูลหลายค่า การจะเข้าถึงข้อมูลตัวใดตัวหนึ่งในตัวแปรอาร์เรย์จึงต้องมีวิธีการเฉพาะ นั่นก็คือ การระบุ**อินเด็กซ์ (Index)** หรือตัวเลขแสดงตำแหน่งของข้อมูลภายในเครื่องหมาย Square brackets ([ ]) ต่อท้ายชื่อตัวแปรอาร์เรย์ ยกตัวอย่างเช่น `n[4]` , `x[0]` , `hightemp[1]` , `frequency[6]` เป็นต้น

โดยวิธีการระบุอินเด็กซ์ (Index) ของตัวแปรอาร์เรย์ก็คือ ตำแหน่งของข้อมูลที่ต้องการลบด้วยหนึ่ง อย่างเช่น ถ้าต้องการเข้าถึงข้อมูลตำแหน่งที่ 7 ของตัวแปรอาร์เรย์ชื่อ `num` เราต้องระบุอินเด็กซ์ (Index) เป็น 6 โดยเขียนเป็น `num[6]` เนื่องจากอินเด็กซ์ (Index) ของตัวแปรอาร์เรย์จะเริ่มที่ 0 ซึ่งข้อมูลตำแหน่งที่หนึ่งในตัวแปรอาร์เรย์ทุกตัวจะมีอินเด็กซ์ (index) เป็น 0

ตัวอย่างการอ้างอิงถึงข้อมูลในตัวแปรอาร์เรย์

|                         |                                                                                          |
|-------------------------|------------------------------------------------------------------------------------------|
| <code>num[2]</code>     | หมายถึง อ้างอิงถึงข้อมูลในเซลล์อินเด็กซ์ที่ 2 ของตัวแปรอาร์เรย์ชื่อ <code>num</code>     |
| <code>score[9]</code>   | หมายถึง อ้างอิงถึงข้อมูลในเซลล์อินเด็กซ์ที่ 9 ของตัวแปรอาร์เรย์ชื่อ <code>score</code>   |
| <code>temp[7]</code>    | หมายถึง อ้างอิงถึงข้อมูลในเซลล์อินเด็กซ์ที่ 7 ของตัวแปรอาร์เรย์ชื่อ <code>temp</code>    |
| <code>letter[8]</code>  | หมายถึง อ้างอิงถึงข้อมูลในเซลล์อินเด็กซ์ที่ 8 ของตัวแปรอาร์เรย์ชื่อ <code>letter</code>  |
| <code>total[0]</code>   | หมายถึง อ้างอิงถึงข้อมูลในเซลล์อินเด็กซ์ที่ 0 ของตัวแปรอาร์เรย์ชื่อ <code>total</code>   |
| <code>salary[10]</code> | หมายถึง อ้างอิงถึงข้อมูลในเซลล์อินเด็กซ์ที่ 10 ของตัวแปรอาร์เรย์ชื่อ <code>salary</code> |

#### NOTE :

1. ข้อมูลในอาร์เรย์ทุกเซลล์จะเป็นข้อมูลประเภทเดียวกัน
2. อินเด็กซ์แรกจะเริ่มต้นด้วย 0
3. การอ้างอิงอินเด็กซ์สามารถกระทำทางคณิตศาสตร์ได้

## 2. อาร์เรย์ 1 มิติ หรือ แถวลำดับ (One-Dimensional Array)

ลักษณะของตัวแปรอาร์เรย์แบบ 1 มิติ จะเก็บข้อมูลต่อเนื่องกันไปเป็นแถวลำดับ การประกาศตัวแปรจะใช้ชื่อเพียงชื่อเดียวตามด้วยเครื่องหมาย Square brackets ( [ ] ) คร่อมค่าตัวเลขที่บอกจำนวนของข้อมูลที่ต้องการ โดยมีรูปแบบดังนี้

รูปแบบ

```
type array_name[size] ;
```

หรือ

```
type array_name[size] = { value_list } ;
```

คำอธิบาย

**type** คือ ชนิดของข้อมูล เช่น int, float, char เป็นต้น

**array\_name** คือ ชื่อตัวแปรอาร์เรย์

**size** คือ จำนวนเซลล์ข้อมูลในอาร์เรย์ หรือ ขนาดของแถวลำดับ

**value\_list** คือ การกำหนดค่าเริ่มต้นให้กับข้อมูลในอาร์เรย์ ถ้ามีข้อมูลหลายตัวจะต้องแยกกันโดยใช้เครื่องหมายคอมม่า ( , ) คั่นไว้

**ตัวอย่างการประกาศตัวแปรอาร์เรย์ 1 มิติ (แบบไม่มีค่าเริ่มต้น)** ค่าของข้อมูลทุกเซลล์จะมีค่าเป็นศูนย์

```
int n[5] ;
```

```
int num[10] ;
```

```
int score[8] , salary[20] ;
```

```
float j[3] ;
```

```
float hightemp[31] , frequency[12] ;
```

**การเก็บข้อมูลลงในอีลีเมนต์ (Element) หรือ เซลล์ (Cell)**

```
int n[5] ;
```

Index → n[0]   n[1]   n[2]   n[3]   n[4]

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| n | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|

```
int num[10] ;
```

Index → num[0]   num[1]   num[2]   num[3]   num[4]   num[5]   num[6]   num[7]   num[8]   num[9]

|     |   |   |   |   |   |   |   |   |   |
|-----|---|---|---|---|---|---|---|---|---|
| num | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|-----|---|---|---|---|---|---|---|---|---|

```
int score[8] ;
```

Index → score[0]   score[1]   score[2]   score[3]   score[4]   score[5]   score[6]   score[7]

|       |   |   |   |   |   |   |   |
|-------|---|---|---|---|---|---|---|
| score | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|-------|---|---|---|---|---|---|---|

**ตัวอย่างการประกาศตัวแปรอาร์เรย์ 1 มิติ (แบบมีค่าเริ่มต้น)** สำหรับการกำหนดค่าของข้อมูลเพื่อนำไปเก็บในตัวแปรอาร์เรย์สามารถทำได้ 2 รูปแบบ โดยแบบแรกเป็นการกำหนดค่าของข้อมูลทุกตำแหน่งในตัวแปรอาร์เรย์พร้อมกันทีเดียว ซึ่งข้อมูลทั้งหมดจะต้องเขียนภายในเครื่องหมาย { } และคั่นระหว่างข้อมูลแต่ละตัวด้วยเครื่องหมายคอมม่า ( , ) ตัวอย่างเช่น

```
int    n[5] = { 0 , 9 , 12 , 5 , 36 } ;
float  hightemp[8] = { 35.5 , 36.1 , 29.8 } ;
int    frequency[8] = { 0 } ;
int    x[ ] = { 101 , 126 , 114 , 118 , 111 , 123 } ;
```

### การเก็บข้อมูลลงในอีลีเมนต์ (Element) หรือ เซลล์ (Cell)

```
int    n[5] = { 0 , 9 , 12 , 5 , 36 } ;
```

Index → n[0]   n[1]   n[2]   n[3]   n[4]

|   |   |   |    |   |    |
|---|---|---|----|---|----|
| n | 0 | 9 | 12 | 5 | 36 |
|---|---|---|----|---|----|

```
float  hightemp[8] = { 35.5 , 36.1 , 29.8 } ;
```

Index → hightemp[0]   hightemp[1]   hightemp[2]   hightemp[3]   hightemp[4]   hightemp[5]   hightemp[6]   hightemp[7]

|          |      |      |      |          |          |          |          |          |
|----------|------|------|------|----------|----------|----------|----------|----------|
| hightemp | 35.5 | 36.1 | 29.8 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
|----------|------|------|------|----------|----------|----------|----------|----------|

```
int    frequency[8] = { 0 } ;
```

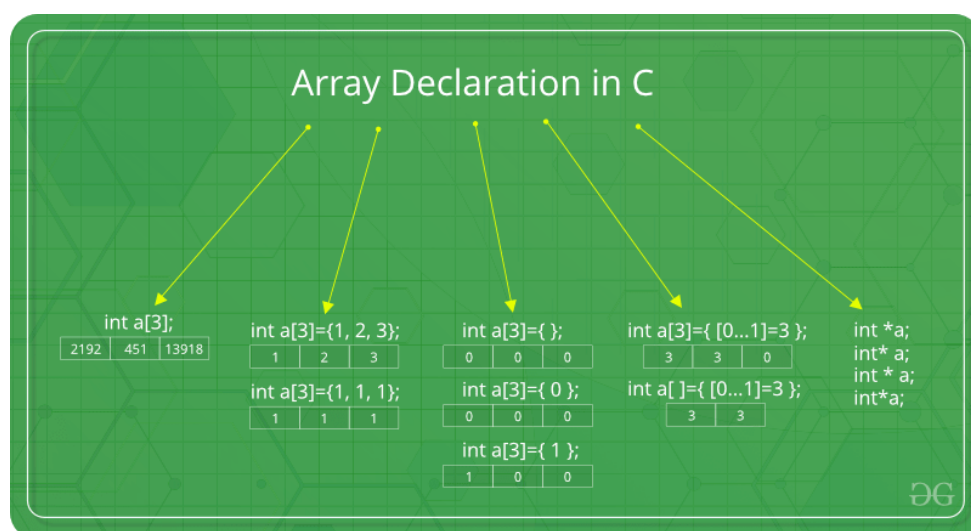
Index → frequency[0]   frequency[1]   frequency[2]   frequency[3]   frequency[4]   frequency[5]   frequency[6]   frequency[7]

|           |   |   |   |   |   |   |   |   |
|-----------|---|---|---|---|---|---|---|---|
| frequency | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|-----------|---|---|---|---|---|---|---|---|

```
int    x[ ] = { 101 , 126 , 114 , 118 , 111 , 123 } ;
```

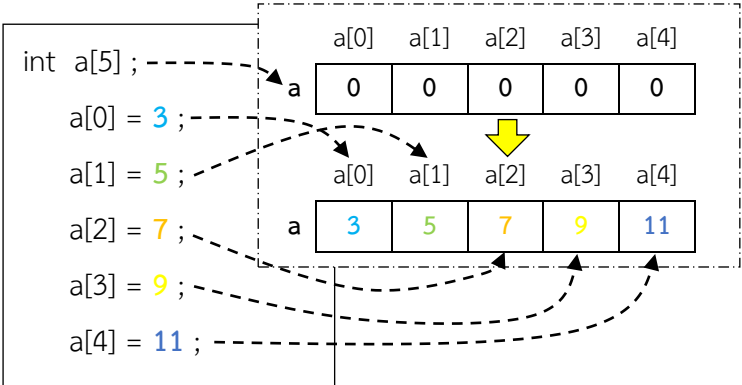
Index → x[0]   x[1]   x[2]   x[3]   x[4]   x[5]

|   |     |     |     |     |     |     |
|---|-----|-----|-----|-----|-----|-----|
| x | 101 | 126 | 114 | 118 | 111 | 123 |
|---|-----|-----|-----|-----|-----|-----|

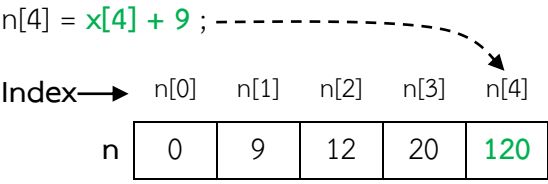
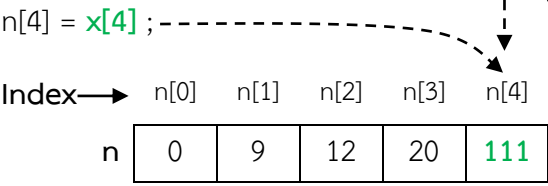
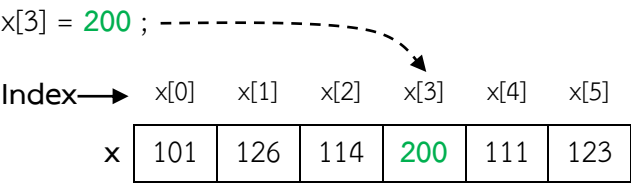
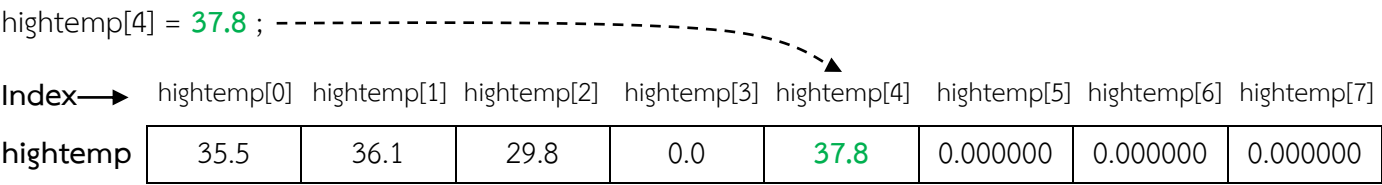
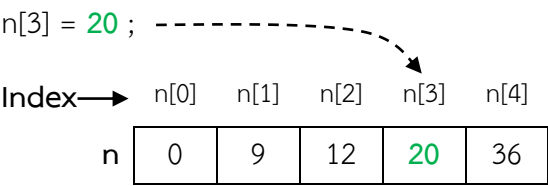


ส่วนการกำหนดค่าของข้อมูลเพื่อนำไปเก็บในตัวแปรอาร์เรย์รูปแบบที่สอง ก็คือ การกำหนดค่า (Assignment) เพื่อเก็บลงในตำแหน่งใดตำแหน่งหนึ่ง โดยการระบุอินเด็กซ์ (Index) ของตำแหน่งที่ต้องการ ซึ่งแบบนี้จะเป็นการกำหนดค่าของข้อมูลที่ละค่า หรือครบทุกค่า ตัวอย่างเช่น

```
n[3] = 20 ;  
hightemp[4] = 37.8 ;  
x[3] = 200 ;  
n[4] = x[4] ;  
n[4] = x[4] + 9 ;
```



การเก็บข้อมูลลงในอีลีเมนต์ (Element) หรือ เซลล์ (Cell)



## การประกาศตัวแปรปกติ และตัวแปรอาร์เรย์

ตัวอย่าง : การประกาศตัวแปรสำหรับเก็บเกรดเฉลี่ยของนักเรียน 40 คน

### การประกาศตัวแปรแบบปกติ

```
float Gpax1, Gpax2, Gpax3, Gpax4, Gpax5, ..... , Gpax40 ;
```

### การประกาศตัวแปรอาร์เรย์

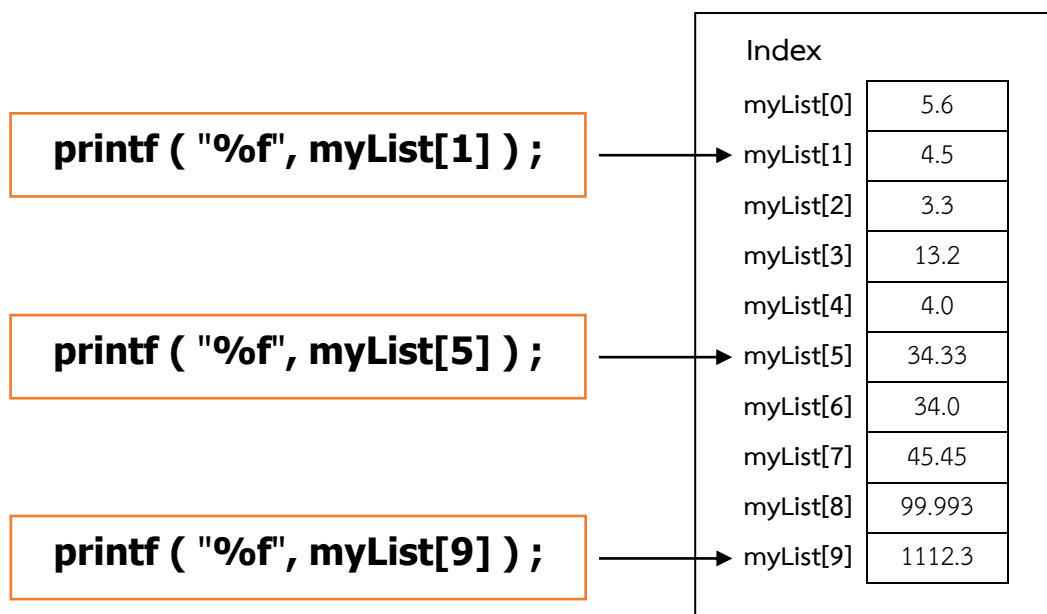
```
float Gpax[40] ;
```

### การนำอาร์เรย์ไปใช้กับคำสั่ง printf () และ scanf ()

```
int n[5] ;
scanf ("%d", &n[0]); // รับค่าจำนวนเต็มนำข้อมูลไปเก็บไว้ในเซลล์อินเด็กซ์ที่ 0 ของตัวแปรอาร์เรย์ชื่อ n
scanf ("%d", &n[1]); // รับค่าจำนวนเต็มนำข้อมูลไปเก็บไว้ในเซลล์อินเด็กซ์ที่ 1 ของตัวแปรอาร์เรย์ชื่อ n
scanf ("%d", &n[2]); // รับค่าจำนวนเต็มนำข้อมูลไปเก็บไว้ในเซลล์อินเด็กซ์ที่ 2 ของตัวแปรอาร์เรย์ชื่อ n
scanf ("%d", &n[3]); // รับค่าจำนวนเต็มนำข้อมูลไปเก็บไว้ในเซลล์อินเด็กซ์ที่ 3 ของตัวแปรอาร์เรย์ชื่อ n
scanf ("%d", &n[4]); // รับค่าจำนวนเต็มนำข้อมูลไปเก็บไว้ในเซลล์อินเด็กซ์ที่ 4 ของตัวแปรอาร์เรย์ชื่อ n

printf ("%d", n[0]); // แสดงผลข้อมูลจำนวนเต็มที่เก็บไว้ในเซลล์อินเด็กซ์ที่ 0 ของตัวแปรอาร์เรย์ชื่อ n
printf ("%d", n[1]); // แสดงผลข้อมูลจำนวนเต็มที่เก็บไว้ในเซลล์อินเด็กซ์ที่ 1 ของตัวแปรอาร์เรย์ชื่อ n
printf ("%d", n[4]); // แสดงผลข้อมูลจำนวนเต็มที่เก็บไว้ในเซลล์อินเด็กซ์ที่ 4 ของตัวแปรอาร์เรย์ชื่อ n
```

```
float myList[10] = { 5.6, 4.5, 3.3, 13.2, 4.0, 34.33, 34.0, 45.45, 99.993, 1112.3 } ;
```



## ตัวอย่างโปรแกรมตัวแปรอาร์เรย์ 1 มิติ

Ex1. โปรแกรมแสดงผลค่าข้อมูลที่เก็บในตัวแปรอาร์เรย์ 1 มิติ

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int i;
```

```
    int n[10] = { 5, 12, 1, 8, 10, 20, 16, 10, 5, 12 };
```

```
    printf ("Data items in order.\n");
```

```
    for ( i=0; i<10; i++ )
```

```
        printf ("%d\t", n[i]);
```

```
    return 0;
```

```
}
```

|   | n[0] | n[1] | n[2] | n[3] | n[4] | n[5] | n[6] | n[7] | n[8] | n[9] |
|---|------|------|------|------|------|------|------|------|------|------|
| n | 5    | 12   | 1    | 8    | 10   | 20   | 16   | 10   | 5    | 12   |

ผลลัพธ์

E:\สำหรับ\_hd\_5-10-2558\เนื้อหา C (รวม 2)\Code\_program\unit3\03-01.exe

Data items in order.

5      12      1      8      10      20      16      10      5      12

Process exited after 0.1655 seconds with return value 0

Press any key to continue . . .

Ex2. โปรแกรมรับข้อมูลตัวเลขจำนวนเต็ม 5 จำนวน นำไปเก็บในตัวแปรอาร์เรย์ 1 มิติ แล้วแสดงผลทางจอภาพ

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int i;
```

```
    int n[5];
```

```
    for ( i=0; i<5; i++ )
```

```
    {
```

```
        printf ("Input number %d = ", i+1);
```

```
        scanf ("%d", &n[i]);
```

```
    }
```

```
    printf ("\nData items in order.\n");
```

```
    for ( i=0; i<5; i++ )
```

```
        printf ("%d\t", n[i]);
```

```
    return 0;
```

```
}
```

|   | n[0] | n[1] | n[2] | n[3] | n[4] |
|---|------|------|------|------|------|
| n | 0    | 0    | 0    | 0    | 0    |



|   | n[0] | n[1] | n[2] | n[3] | n[4] |
|---|------|------|------|------|------|
| n | 5    | 10   | 15   | 20   | 25   |

ผลลัพธ์

E:\สำหรับ\_hd\_5-10-2558\เนื้อหา C (รวม 2)\Code\_program\unit3\03-02.exe

Input number 1 = 5

Input number 2 = 10

Input number 3 = 15

Input number 4 = 20

Input number 5 = 25

Data items in order.

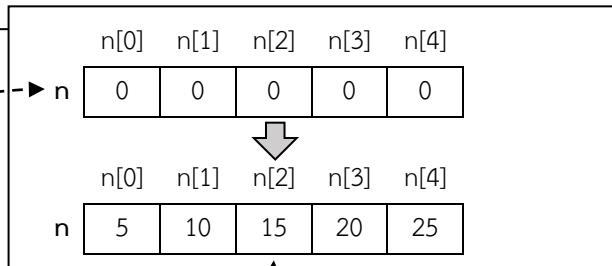
5      10      15      20      25

Process exited after 4.342 seconds with

Press any key to continue . . .

Ex3-1. โปรแกรมรับข้อมูลตัวเลขจำนวนเต็ม 5 จำนวน นำไปเก็บในตัวแปรอาร์เรย์ 1 มิติ ให้คำนวณหาผลรวมทั้งหมดและค่าเฉลี่ยแสดงผลออกทางจอภาพ

```
#include <stdio.h>
int main()
{
    int i, sum=0, n[5];
    float avg;
    for ( i=0; i<5; i++ )
    {
        printf ("Input number %d = ", i+1);
        scanf ("%d", &n[i]);
    }
    for ( i=0; i<5; i++ )
        sum = sum + n[i];
    avg = sum/5.0;
    printf ("\nThe sum is %d", sum);
    printf ("\nThe average is %.2f", avg);
    return 0;
}
```



#### ผลลัพธ์

```
E:\ทำงาน_hd_5-10-2558\เนื้อหา C (พจน 2)\Code_program\unit3\06-03-1.exe
Input number 1 = 5
Input number 2 = 10
Input number 3 = 15
Input number 4 = 20
Input number 5 = 25

The sum is 75
The average is 15.00
-----
Process exited after 3.85 seconds
Press any key to continue . . .
```

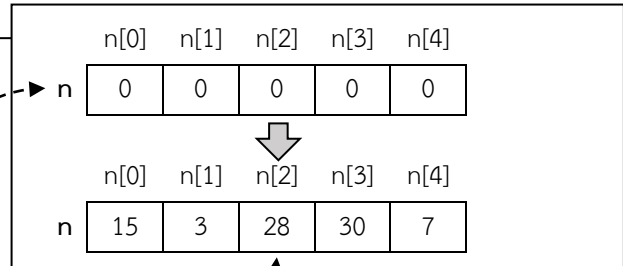
Ex3-2. หรือเขียนเป็นชุดคำสั่งได้อีกแบบ ดังนี้

```
#include <stdio.h>
int main()
{
    int i, sum=0, n[5];
    float avg;
    for ( i=0; i<5; i++ )
    {
        printf ("Input number %d = ", i+1);
        scanf ("%d", &n[i]);
        sum = sum + n[i];
    }
    avg = sum/5.0;
    printf ("\nThe sum is %d", sum);
    printf ("\nThe average is %.2f", avg);
    return 0;
}
```



Ex4-1. โปรแกรมรับข้อมูลตัวเลขจำนวนเต็มบวก 5 จำนวน นำไปเก็บในตัวแปรอาร์เรย์ 1 มิติ แล้วให้แสดงผลค่าสูงสุดและค่าต่ำสุดออกทางจอภาพ (ค่าสูงสุดไม่เกิน 500)

```
#include <stdio.h>
int main()
{
    int i, max=0, min=500, n[5];
    for ( i=0; i<5; i++ )
    {
        printf ("Input number %d = ", i+1);
        scanf ("%d", &n[i]);
    }
    for ( i=0; i<5; i++ )
    {
        if (min > n[i])
            min = n[i];
        if (max < n[i])
            max = n[i];
    }
    printf ("\nMinimum is %d", min);
    printf ("\nMaximum is %d", max);
    return 0;
}
```



#### ผลลัพธ์

```
E:\ทำงาน_hd_5-10-2558\เนื้อหา C (พจนม 2)\Code_program\unit3\03-04-1.exe
Input number 1 = 15
Input number 2 = 3
Input number 3 = 28
Input number 4 = 30
Input number 5 = 7

Minimum is 3
Maximum is 30

-----
Process exited after 20.94 seconds
Press any key to continue . . .
```

Ex4-2. หรือเขียนเป็นชุดคำสั่งได้อีกแบบ ดังนี้

```
#include <stdio.h>
int main()
{
    int i, max=0, min=500, n[5];
    for ( i=0; i<5; i++ )
    {
        printf ("Input number %d = ", i+1);
        scanf ("%d", &n[i]);
        if (min > n[i])
            min = n[i];
        if (max < n[i])
            max = n[i];
    }
    printf ("\nMinimum is %d", min);
    printf ("\nMaximum is %d", max);
    return 0;
}
```

Ex5. โปรแกรมแสดงการกำหนดค่าเริ่มต้นให้กับสมาชิกของตัวแปร 1 มิติแล้วนำไปสร้างฮิสโตแกรมจากข้อมูลที่ให้

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
    int i,j;
```

```
    int n[10] = { 5, 12, 1, 8, 10, 20, 16, 10, 5, 12 };
```

```
    printf ("Element \tValue \tHistogram\n");
```

```
    for ( i=0; i<10; i++ )
```

```
    {
```

```
        printf ("n[%d] \t%d \t",i ,n[i]);
```

```
        for ( j=1; j<=n[i]; j++ )
```

```
            printf ("*");
```

```
        printf ("\n");
```

```
    }
```

```
    return 0;
```

```
}
```

|   | n[0] | n[1] | n[2] | n[3] | n[4] | n[5] | n[6] | n[7] | n[8] | n[9] |
|---|------|------|------|------|------|------|------|------|------|------|
| n | 5    | 12   | 1    | 8    | 10   | 20   | 16   | 10   | 5    | 12   |

ผลลัพธ์

```
E:\สำรอง_hd_5-10-2558\เนื้อหา C (เทอม 2)\Code_program\unit3\03-05.exe

Element      Value      Histogram
n[0]         5          *****
n[1]        12          *****
n[2]         1           *
n[3]         8          *****
n[4]        10          *****
n[5]        20          *****
n[6]        16          *****
n[7]        10          *****
n[8]         5          *****
n[9]        12          *****

-----
Process exited after 0.0788 seconds with return value 0
Press any key to continue . . .
```

## การสลับค่าข้อมูล

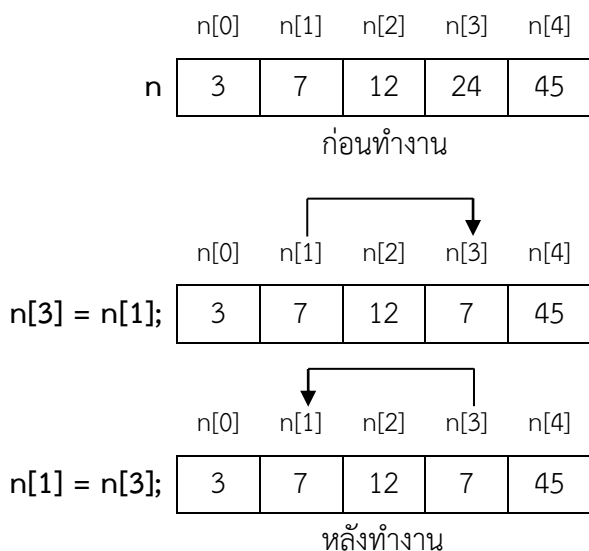
การเขียนโปรแกรมสลับข้อมูล มักถูกนำไปใช้ในการพัฒนาโปรแกรมต่าง ๆ มากมาย ในหัวข้อนี้จะลองทดลองเขียนโปรแกรมสลับค่าข้อมูลในเซลล์ของอาร์เรย์กัน

พิจารณาจากตัวแปรอาร์เรย์ต่อไปนี้ หากต้องการสลับข้อมูลในเซลล์  $n[3]$  กับเซลล์  $n[1]$  ถ้าหากเขียนชุดคำสั่งเป็น

```
n[3] = n[1];    //ย้ายค่าเซลล์ที่ 1 ไปใส่เซลล์ที่ 3
n[1] = n[3];    //ย้ายค่าเซลล์ที่ 3 ไปใส่เซลล์ที่ 1
```

ถ้าหากเขียนชุดคำสั่งในลักษณะนี้ การทำงานจะผิดพลาดเนื่องจากหลังการทำงานเซลล์ที่ 3 และเซลล์ที่ 1 จะมีค่าเท่ากัน ตัวอย่าง เช่น

```
int n[5] = { 3, 7, 12, 24, 45 };
n[3] = n[1];    //ย้ายค่าเซลล์ที่ 1 ไปใส่เซลล์ที่ 3
n[1] = n[3];    //ย้ายค่าเซลล์ที่ 3 ไปใส่เซลล์ที่ 1
```

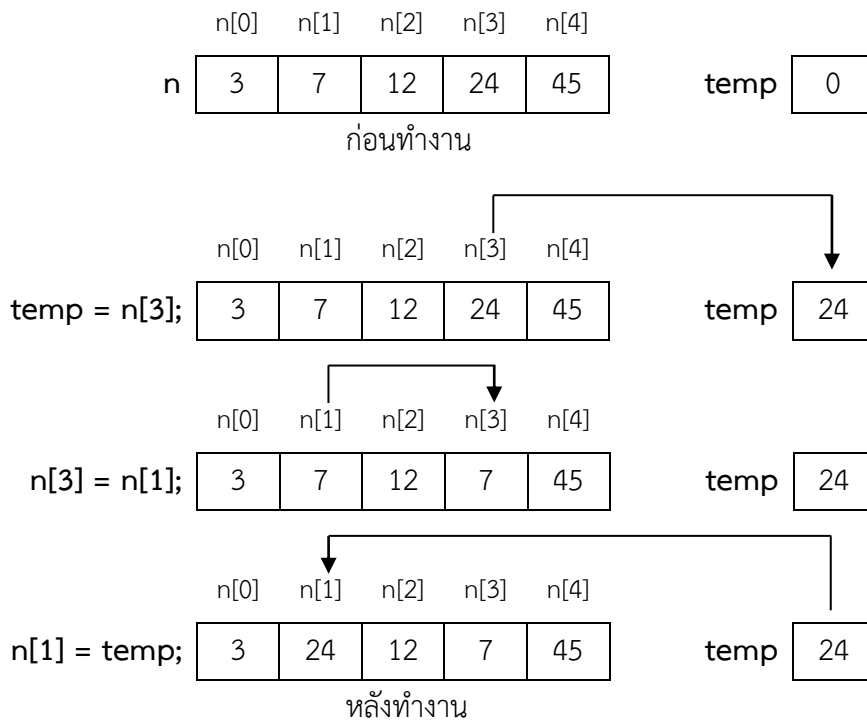


ในการเขียนโปรแกรมที่ถูกต้อง จะต้องสร้างตัวแปรสำรองขึ้นมาอีกหนึ่งตัวเพื่อเป็น**ที่พักข้อมูล** โดยเอาข้อมูลในเซลล์ที่ 3 ไปพักไว้ก่อน จากนั้นนำเซลล์ที่ 1 ไปใส่ในเซลล์ที่ 3 แล้วเอาข้อมูลที่พักไว้มาใส่ในเซลล์ที่ 1 แทน ถ้าหากให้ตัวแปรสำรองมีชื่อว่า temp จะเขียนชุดคำสั่งได้ดังนี้

```
temp = n[3];    //ย้ายค่าเซลล์ที่ 3 ไปไว้ตัวแปรสำรองชื่อ temp
n[3] = n[1];    //ย้ายค่าเซลล์ที่ 1 ไปใส่เซลล์ที่ 3
n[1] = temp;    //ย้ายค่าในตัวแปรสำรองชื่อ temp ไปใส่เซลล์ที่ 1
```

ตัวอย่างการสลับข้อมูลที่ถูกต้อง

```
int n[5] = { 3, 7, 12, 24, 45 };
int temp;
temp = n[3];           //ย้ายค่าเซลล์ที่ 3 ไปไว้ตัวแปรสำรองชื่อ temp
n[3] = n[1];           //ย้ายค่าเซลล์ที่ 1 ไปใส่เซลล์ที่ 3
n[1] = temp;           //ย้ายค่าตัวแปรสำรองชื่อ temp ไปใส่เซลล์ที่ 1
```



### การเรียงลำดับข้อมูล

ถ้าหากมีข้อมูลแบบลำดับหลายค่า เราสามารถนำข้อมูลนั้นมาเรียงกันได้ (Sorting data) การเรียงข้อมูลนี้เป็นเรื่องสำคัญในการเขียนโปรแกรมประยุกต์ เช่น การค้นหาเบอร์โทรศัพท์โดยเรียงข้อมูลตามชื่อตัวอักษร เป็นต้น โปรแกรมในตัวอย่างที่ Ex6 เป็นตัวอย่างการเรียงข้อมูลอย่างง่าย โดยนำข้อมูลในอาร์เรย์ `a` จำนวน 10 เซลล์มาเรียงกัน โดยใช้เทคนิคที่เรียกว่า bubble sort หรือ sinking sort เทคนิคนี้จะนิยมใช้กับข้อมูลจำนวนน้อย ๆ โดยจะนำข้อมูลทั้งหมดมาทดสอบว่าข้อมูลที่อยู่ติดกัน ข้อมูลใดมีค่ามากหรือน้อยกว่ากัน แล้วทำการสลับค่า ดังตัวอย่างในโปรแกรมที่ Ex6

การทำงานของโปรแกรมเริ่มแรกจะทำการดูโดยจะเปรียบเทียบข้อมูล `a[0]` กับ `a[1]` จากนั้นเปรียบเทียบ `a[1]` กับ `a[2]`, `a[2]` กับ `a[3]` ตามลำดับไปจนถึง `a[9]` ซึ่งเป็นเซลล์สุดท้าย ในการอ่านข้อมูลในอาร์เรย์ขึ้นมาสองเซลล์นั้นจะนำข้อมูลที่มีค่าน้อยกว่ามาเก็บในเซลล์ที่มีอินเด็กซ์ต่ำกว่า โดยทดสอบว่าค่าใน `a[i]` มากกว่าค่าใน `a[i+1]` จริงหรือไม่ ถ้าจริงให้สลับค่าทำชุดคำสั่ง ดังนี้

```
temp = a[i];
a[i] = a[i+1];
a[i+1] = temp;
```

เมื่อเปรียบเทียบข้อมูลไปจนถึง  $a[9]$  แล้วจะต้องทำซ้ำอีก จะเห็นว่าการเรียงข้อมูลด้วยวิธีนี้จะเป็นการวนรอบซ้ำซ้ำซ้อน หรือการทำลูปซ้อนลูป ถ้าหากมีข้อมูลจำนวนมากจะทำให้เวลาในการทำงานหรือประมวลผลมากขึ้นไปด้วย

Ex6. โปรแกรมเรียงลำดับข้อมูลจากน้อยไปหามาก

```
#include <stdio.h>
#define SIZE 10
main()
{
    int a[SIZE] = { 2, 6, 4, 8, 10, 12, 89, 68, 45, 37 };
    int i, pass, temp;
    printf ("Data items in original order.\n");
    for ( i=0; i<=SIZE-1; i++ )
        printf ("%4d", a[i]);
    for ( pass=0; pass<=SIZE-2; pass++ )
    {
        for ( i=0; i<=SIZE-pass-2; i++ )
        {
            if (a[i] > a[i+1])
            {
                temp = a[i];
                a[i] = a[i+1];
                a[i+1] = temp;
            }
        }
    }
    printf ("\n\nData items in ascending order.\n");
    for ( i=0; i<=SIZE-1; i++ )
        printf ("%4d", a[i]);
    printf ("\n");
    return 0;
}
```

|   | a[0] | a[1] | a[2] | a[3] | a[4] | a[5] | a[6] | a[7] | a[8] | a[9] |
|---|------|------|------|------|------|------|------|------|------|------|
| a | 2    | 6    | 4    | 8    | 10   | 12   | 89   | 68   | 45   | 37   |

|   | a[0] | a[1] | a[2] | a[3] | a[4] | a[5] | a[6] | a[7] | a[8] | a[9] |
|---|------|------|------|------|------|------|------|------|------|------|
| a | 2    | 4    | 6    | 8    | 10   | 12   | 37   | 45   | 68   | 89   |

ผลลัพธ์

E:\โครงงาน\_hd\_5-10-2558\เนื้อหา C (เทอม 2)\Code\_program\unit3\03-06.exe

```
Data items in original order.
  2  6  4  8 10 12 89 68 45 37
```

```
Data items in ascending order.
  2  4  6  8 10 12 37 45 68 89
```

```
-----
Process exited after 0.09303 seconds with return value 0
Press any key to continue . . .
```

### 3. ตัวแปรอาร์เรย์ 2 มิติ (Two-Dimensional Array) และหลายมิติ

ลักษณะของตัวแปรอาร์เรย์แบบ 2 มิติ จะเก็บข้อมูล 2 ทิศทางในแนวแถว (Row) และคอลัมน์ (Column) หรือด้านกว้างและยาว ซึ่งการนำตัวแปรอาร์เรย์ 2 มิติมาใช้ เหมาะสำหรับการเก็บข้อมูลบางประเภทที่นำไปเก็บในตัวแปรอาร์เรย์แบบมิติเดียวจะทำงานไม่สะดวก เช่น การเก็บคะแนนของนักเรียน 4 คน แต่ละคนมีคะแนน 4 วิชา เป็นต้น ซึ่งจะง่ายกว่าหากเรานำมาเก็บโดยจัดข้อมูลในแนวแถวและคอลัมน์ในรูปแบบของอาร์เรย์ 2 มิติ ดังรูป

| Score        |         | คอลัมน์ที่<br>(Column) | 0      | 1      | 2      | 3      |
|--------------|---------|------------------------|--------|--------|--------|--------|
| แถวที่ (Row) |         |                        | วิชา A | วิชา B | วิชา C | วิชา D |
| 0            | คนที่ 1 |                        | 56     | 42     | 65     | 52     |
| 1            | คนที่ 2 |                        | 42     | 89     | 24     | 18     |
| 2            | คนที่ 3 |                        | 12     | 23     | 78     | 36     |
| 3            | คนที่ 4 |                        | 76     | 85     | 96     | 52     |

ตัวแปรอาร์เรย์ 2 มิติ ชื่อ Score สำหรับเก็บคะแนนของนักเรียน มีจำนวนสมาชิกทั้งหมด 16 จำนวน (4x4)

#### การอ้างอิงถึงข้อมูลในตัวแปรอาร์เรย์สองมิติ

ตัวแปรอาร์เรย์สองมิติเป็นตัวแปรที่เก็บข้อมูลเป็นชุด ดังนั้นภายในตัวแปรอาร์เรย์จึงประกอบไปด้วยข้อมูลหลายค่า การจะเข้าถึงข้อมูลตัวใดตัวหนึ่งในตัวแปรอาร์เรย์สองมิติจึงต้องมีวิธีการเฉพาะ นั่นก็คือ การระบุ**อินเด็กซ์ (Index)** หรือตัวเลขแสดงตำแหน่งแถว (Row) และคอลัมน์ (Column) ของข้อมูลภายในเครื่องหมาย Square brackets ([ ]) ต่อท้ายชื่อตัวแปรอาร์เรย์สองมิติ ยกตัวอย่างเช่น `n[4][2]` , `x[0][1]` , `hightemp[1][3]` , `frequency[6][0]` เป็นต้น

โดยวิธีการระบุอินเด็กซ์ (Index) ของตัวแปรอาร์เรย์สองมิติ **ตำแหน่งแรก**ของตัวเลขจะแสดงตำแหน่ง**แถว (Row)** และ**ตำแหน่งที่สอง**ของตัวเลขจะแสดงตำแหน่ง**คอลัมน์ (Column)** ของข้อมูล ในตัวแปรอาร์เรย์สองมิติทุกตัวจะมีอินเด็กซ์ (index) เริ่มต้นเป็น 0

#### ตัวอย่างการอ้างอิงถึงข้อมูลในตัวแปรอาร์เรย์สองมิติ

`num[1][5]` ; หมายถึง อ้างอิงถึงข้อมูลในเซลล์ที่อยู่ในแถวที่ 1 และคอลัมน์ที่ 5 ของตัวแปรอาร์เรย์สองมิติชื่อ `num`  
`score[0][3]` ; หมายถึง อ้างอิงถึงข้อมูลในเซลล์ที่อยู่ในแถวที่ 0 และคอลัมน์ที่ 3 ของตัวแปรอาร์เรย์สองมิติชื่อ `score`  
`temp[4][1]` ; หมายถึง อ้างอิงถึงข้อมูลในเซลล์ที่อยู่ในแถวที่ 4 และคอลัมน์ที่ 1 ของตัวแปรอาร์เรย์สองมิติชื่อ `temp`

| Score        |         | คอลัมน์ที่<br>(Column) | 0      | 1      | 2      | 3      |
|--------------|---------|------------------------|--------|--------|--------|--------|
| แถวที่ (Row) |         |                        | วิชา A | วิชา B | วิชา C | วิชา D |
| 0            | คนที่ 1 |                        | [0][0] | [0][1] | [0][2] | [0][3] |
| 1            | คนที่ 2 |                        | [1][0] | [1][1] | [1][2] | [1][3] |
| 2            | คนที่ 3 |                        | [2][0] | [2][1] | [2][2] | [2][3] |
| 3            | คนที่ 4 |                        | [3][0] | [3][1] | [3][2] | [3][3] |

การอ้างอิงถึงตัวแปรอาร์เรย์ 2 มิติ ชื่อ Score สำหรับเก็บคะแนนของนักเรียน มีจำนวนสมาชิกทั้งหมด 16 จำนวน

## การประกาศตัวแปรอาร์เรย์ 2 มิติ

การประกาศตัวแปรอาร์เรย์แบบ 2 มิติ จะใช้ดัชนี 2 ตัว เพื่อระบุจำนวนสมาชิกในแต่ละแถวและคอลัมน์ การประกาศตัวแปรจะใช้ชื่อเพียงชื่อเดียวตามด้วยเครื่องหมาย Square brackets ( [ ] ) คร่อมค่าตัวเลขสองจำนวนที่บอกจำนวนแถวและคอลัมน์ของข้อมูลที่ต้องการ โดยมีรูปแบบดังนี้

### รูปแบบ

```
type array_name[Row] [Column] ;
```

หรือ

```
type array_name[Row] [Column] = { value_list } ;
```

### คำอธิบาย

**type** คือ ชนิดของข้อมูล เช่น int, float, char เป็นต้น

**array\_name** คือ ชื่อตัวแปรอาร์เรย์สองมิติหรือหลายมิติ

**Row** คือ จำนวนแถวของข้อมูลในอาร์เรย์ เริ่มต้นแถวที่ 0

**Column** คือ จำนวนคอลัมน์ของข้อมูลในอาร์เรย์ เริ่มต้นคอลัมน์ที่ 0

**value\_list** คือ การกำหนดค่าเริ่มต้นให้กับข้อมูลในอาร์เรย์สองมิติ ถ้ามีข้อมูลหลายตัวจะต้องแยกกัน

โดยใช้เครื่องหมายคอมม่า ( , ) คั่นไว้

### ตัวอย่างการประกาศตัวแปรอาร์เรย์ 2 มิติ (แบบไม่มีค่าเริ่มต้น)

```
int n[5][4] ;
```

```
int num[10][3] ;
```

```
int score[5][4] , salary[10][5] ;
```

```
float j[3][3] ;
```

```
float hightemp[7][31] , frequency[5][12] ;
```

### การเก็บข้อมูลลงในอีลีเมนต์ (Element) หรือ เซลล์ (Cell)

```
int n[5][4] ;
```

|       |   | Column → |          |          |          |
|-------|---|----------|----------|----------|----------|
|       |   | 0        | 1        | 2        | 3        |
| Row ↓ | 0 | 0 [0][0] | 0 [0][1] | 0 [0][2] | 0 [0][3] |
|       | 1 | 0 [1][0] | 0 [1][1] | 0 [1][2] | 0 [1][3] |
|       | 2 | 0 [2][0] | 0 [2][1] | 0 [2][2] | 0 [2][3] |
|       | 3 | 0 [3][0] | 0 [3][1] | 0 [3][2] | 0 [3][3] |
|       | 4 | 0 [4][0] | 0 [4][1] | 0 [4][2] | 0 [4][3] |

**ตัวอย่างการประกาศตัวแปรอาร์เรย์ 2 มิติ (แบบมีค่าเริ่มต้น)** สำหรับการกำหนดค่าของข้อมูลเพื่อนำไปเก็บในตัวแปรอาร์เรย์สองมิติสามารถทำได้ 2 รูปแบบ โดยแบบแรกเป็นการกำหนดค่าของข้อมูลทุกตำแหน่งในตัวแปรอาร์เรย์พร้อมกันทีเดียว ซึ่งข้อมูลทั้งหมดจะต้องเขียนภายในเครื่องหมาย { } และคั่นระหว่างข้อมูลแต่ละตัวด้วยเครื่องหมายคอมม่า ( , )

ตัวอย่าง เช่น

```
int    n[3][3] = { 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 } ;
int    num [2][3] = {
                        1 , 2 , 3 ,
                        4 , 5 , 6
                    } ;
int    b[2][2] = { { 1 , 2 } , { 3 , 4 } } ;
int    x[2][2] = { { 1 } , { 3 , 4 } } ;
int    x[ ][3] = { { 101 , 126 , 114 } , { 118 , 111 , 123 } } ;
```

#### การเก็บข้อมูลลงในอีลีเมนต์ (Element) หรือ เซลล์ (Cell)

```
int  n[3][3] = { 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 } ;
```

|   | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | 2 | 3 |
| 1 | 4 | 5 | 6 |
| 2 | 7 | 8 | 9 |

```
int  num [2][3] = {
```

```
    1 , 2 , 3 ,
    4 , 5 , 6
};
```

|   | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | 2 | 3 |
| 1 | 4 | 5 | 6 |

```
int  b[2][2] = { { 1 , 2 } , { 3 , 4 } } ;
```

|   | 0 | 1 |
|---|---|---|
| 0 | 1 | 2 |
| 1 | 3 | 4 |



```
int x[2][2] = { { 1 }, { 3, 4 } };
```

|   | 0 | 1 |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 3 | 4 |

\* หากกำหนดค่าเริ่มต้นของแต่ละแถวไม่ครบทุกหลัก เซลล์ของอาร์เรย์ในหลักที่ไม่ถูกกำหนดจะเก็บค่า 0

```
int x[ ][3] = { { 101, 126, 114 }, { 118, 111, 123 } };
```

|   | 0   | 1   | 2   |
|---|-----|-----|-----|
| 0 | 101 | 126 | 114 |
| 1 | 118 | 111 | 123 |

\* หากไม่กำหนดขนาดของแถวหรือคอลัมน์ โปรแกรมจะคำนวณขนาดของแถวหรือคอลัมน์ให้อัตโนมัติโดยดูจากข้อมูลที่กำหนดมาให้

ส่วนการกำหนดค่าของข้อมูลเพื่อนำไปเก็บในตัวแปรอาร์เรย์สองมิติรูปแบบที่สอง ก็คือ การกำหนดค่า (Assignment) เพื่อเก็บลงในตำแหน่งใดตำแหน่งหนึ่ง โดยการระบุอินเด็กซ์ (Index) ของตำแหน่งที่ต้องการ ซึ่งแบบนี้จะเป็นการกำหนดค่าของข้อมูลที่ละค่า ตัวอย่างเช่น

```
int num[3][3];
num[0][0] = 7;
num[0][1] = 8;
num[0][2] = 9;
```

|   |   |   |
|---|---|---|
| 7 | 8 | 9 |
| 4 | 5 | 6 |
| 1 | 2 | 3 |

```
num[1][0] = 4;
num[1][1] = 5;
num[1][2] = 6;

num[2][0] = 1;
num[2][1] = 2;
num[2][2] = 3;
```

| num | 0        | 1        | 2        |
|-----|----------|----------|----------|
| 0   | 7 [0][0] | 8 [0][1] | 9 [0][2] |
| 1   | 4 [1][0] | 5 [1][1] | 6 [1][2] |
| 2   | 1 [2][0] | 2 [2][1] | 3 [2][2] |

การนำอาร์เรย์สองมิติไปใช้กับคำสั่ง printf () และ scanf ()

```
int    n[2][3] ;
scanf ("%d", &n[0][1]);    // รับค่าจำนวนเต็มนำไปเก็บไว้ในเซลล์แถวที่ 0 คอลัมน์ที่ 1 ของตัวแปรอาร์เรย์ชื่อ n
scanf ("%d", &n[1][0]);    // รับค่าจำนวนเต็มนำไปเก็บไว้ในเซลล์แถวที่ 1 คอลัมน์ที่ 0 ของตัวแปรอาร์เรย์ชื่อ n
scanf ("%d", &n[1][2]);    // รับค่าจำนวนเต็มนำไปเก็บไว้ในเซลล์แถวที่ 1 คอลัมน์ที่ 2 ของตัวแปรอาร์เรย์ชื่อ n

printf ("%d", n[0][1]);    // แสดงผลข้อมูลจำนวนเต็มที่เก็บไว้ในเซลล์แถวที่ 0 คอลัมน์ที่ 1 ของตัวแปรอาร์เรย์ชื่อ n
printf ("%d", n[1][0]);    // แสดงผลข้อมูลจำนวนเต็มที่เก็บไว้ในเซลล์แถวที่ 1 คอลัมน์ที่ 0 ของตัวแปรอาร์เรย์ชื่อ n
printf ("%d", n[1][2]);    // แสดงผลข้อมูลจำนวนเต็มที่เก็บไว้ในเซลล์แถวที่ 1 คอลัมน์ที่ 2 ของตัวแปรอาร์เรย์ชื่อ n
```

```
int  num[3][3] = {
    { 7 , 8 , 9 },
    { 4 , 5 , 6 },
    { 1 , 2 , 3 }
};
```

|   |   |   |
|---|---|---|
| 7 | 8 | 9 |
| 4 | 5 | 6 |
| 1 | 2 | 3 |

```
printf ( "%d", num[0][1] ) ;
```

```
printf ( "%d", num[1][2] ) ;
```

```
printf ( "%d", num[2][0] ) ;
```

| num | 0        | 1        | 2        |
|-----|----------|----------|----------|
| 0   | 7 [0][0] | 8 [0][1] | 9 [0][2] |
| 1   | 4 [1][0] | 5 [1][1] | 6 [1][2] |
| 2   | 1 [2][0] | 2 [2][1] | 3 [2][2] |

## ตัวอย่างโปรแกรมตัวแปรอาร์เรย์ 2 มิติ

Ex7. โปรแกรมแสดงผลค่าข้อมูลที่เก็บในตัวแปรอาร์เรย์ 2 มิติ

```
#include <stdio.h>
int main()
{
    int i, j;
    int arr[3][3] = { { 7, 8, 9 },
                      { 4, 5, 6 },
                      { 1, 2, 3 } };
    printf ("Data items in 2 dimensional array.\n");
    for ( i=0; i<3; i++ )
    {
        for ( j=0; j<3; j++ )
        {
            printf ("%d ", arr[i][j]);
        }
        printf ("\n");
    }
    return 0;
}
```

| arr | 0        | 1        | 2        |
|-----|----------|----------|----------|
| 0   | 7 [0][0] | 8 [0][1] | 9 [0][2] |
| 1   | 4 [1][0] | 5 [1][1] | 6 [1][2] |
| 2   | 1 [2][0] | 2 [2][1] | 3 [2][2] |

### ผลลัพธ์

```
E:\สอน_hd_5-10-2558\เนื้อหา C (พจน 2)\Code_program\unit3\03-07.exe
Data items in 2 dimensional array.
7 8 9
4 5 6
1 2 3

-----
Process exited after 0.2673 seconds
Press any key to continue . . .
```

Ex8. โปรแกรมรับข้อมูลตัวเลขจำนวนเต็ม ขนาด  $3 \times 3$  จำนวน นำไปเก็บในตัวแปรอาร์เรย์ 2 มิติ แล้วแสดงผล

ทางจอภาพ

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int i, j;
```

```
    int arr[3][3];
```

```
    for ( i=0; i<3; i++ )
```

```
    {
```

```
        for ( j=0; j<3; j++ )
```

```
        {
```

```
            printf ("Input number in [%d][%d] = ", i, j);
```

```
            scanf ("%d", &arr[i][j]);
```

```
        }
```

```
    }
```

```
    printf ("\nData items in 2 dimensional array.\n");
```

```
    for ( i=0; i<3; i++ )
```

```
    {
```

```
        for ( j=0; j<3; j++ )
```

```
        {
```

```
            printf ("%d ", arr[i][j]);
```

```
        }
```

```
    printf ("\n");
```

```
    }
```

```
    return 0;
```

```
}
```

| arr | 0 | 1 | 2 |   | arr | 0 | 1 | 2 |
|-----|---|---|---|---|-----|---|---|---|
| 0   | 0 | 0 | 0 | → | 0   | 1 | 3 | 7 |
| 1   | 0 | 0 | 0 |   | 1   | 2 | 1 | 5 |
| 2   | 0 | 0 | 0 |   | 2   | 8 | 4 | 1 |

ผลลัพธ์

```
E:\ศรชช_5-10-2558\เนื้อหา C (รวม 2)\Code_program\unit3\03-08.exe
Input number in [0][0] = 1
Input number in [0][1] = 3
Input number in [0][2] = 7
Input number in [1][0] = 2
Input number in [1][1] = 1
Input number in [1][2] = 5
Input number in [2][0] = 8
Input number in [2][1] = 4
Input number in [2][2] = 1

Data items in 2 dimensional array.
1 3 7
2 1 5
8 4 1
```

Ex9. โปรแกรมรับข้อมูลตัวเลขจำนวนเต็ม ตามขนาด  $m \times n$  ที่กำหนด นำไปเก็บในตัวแปรอาร์เรย์ 2 มิติ แล้วให้แสดงผลทรานสโพสของเมทริกซ์ (Transpose Matrix) ของข้อมูลที่รับเข้าไปออกทางจอภาพ

ตัวอย่าง เช่น  $A = \begin{bmatrix} 1 & 2 & -1 \\ 0 & 3 & 5 \end{bmatrix}_{2 \times 3}$   $A^t = \begin{bmatrix} 1 & 0 \\ 2 & 3 \\ -1 & 5 \end{bmatrix}_{3 \times 2}$

ทรานสโพสของเมทริกซ์ A เกิดจากการเปลี่ยนจากแถวเป็นหลักของเมทริกซ์ A

สัญลักษณ์  $A^t$  แทน ทรานสโพสของเมทริกซ์ A

```
#include <stdio.h>
int main()
{
    int m, n, i, j ;
    printf ("Input Row : ");
    scanf ("%d", &m);
    printf ("Input Column : ");
    scanf ("%d", &n);
    int arr[m][n];
    for ( i=0; i<m; i++ )
    {
        for ( j=0; j<n; j++ )
        {
            printf ("Input number in [%d][%d] = ", i+1, j+1);
            scanf ("%d", &arr[i][j]);
        }
    }
    printf ("\nData items in 2 dimensional array. Size %d x %d.\n", m, n);
    for ( i=0; i<m; i++ )
    {
        for ( j=0; j<n; j++ )
        {
            printf ("%3d", arr[i][j]);
        }
        printf ("\n");
    }
    printf ("\nData items in Transpose Matrix. Size %d x %d.\n", n, m);
    for ( i=0; i<n; i++ )
    {
        for ( j=0; j<m; j++ )
        {
            printf ("%3d", arr[j][i]);
        }
        printf ("\n");
    }
    return 0;
}
```

| arr | 0 | 1 | 2 |
|-----|---|---|---|
| 0   | 0 | 0 | 0 |
| 1   | 0 | 0 | 0 |

→

| arr | 0 | 1 | 2  |
|-----|---|---|----|
| 0   | 1 | 2 | -1 |
| 1   | 0 | 3 | 5  |

#### ผลลัพธ์

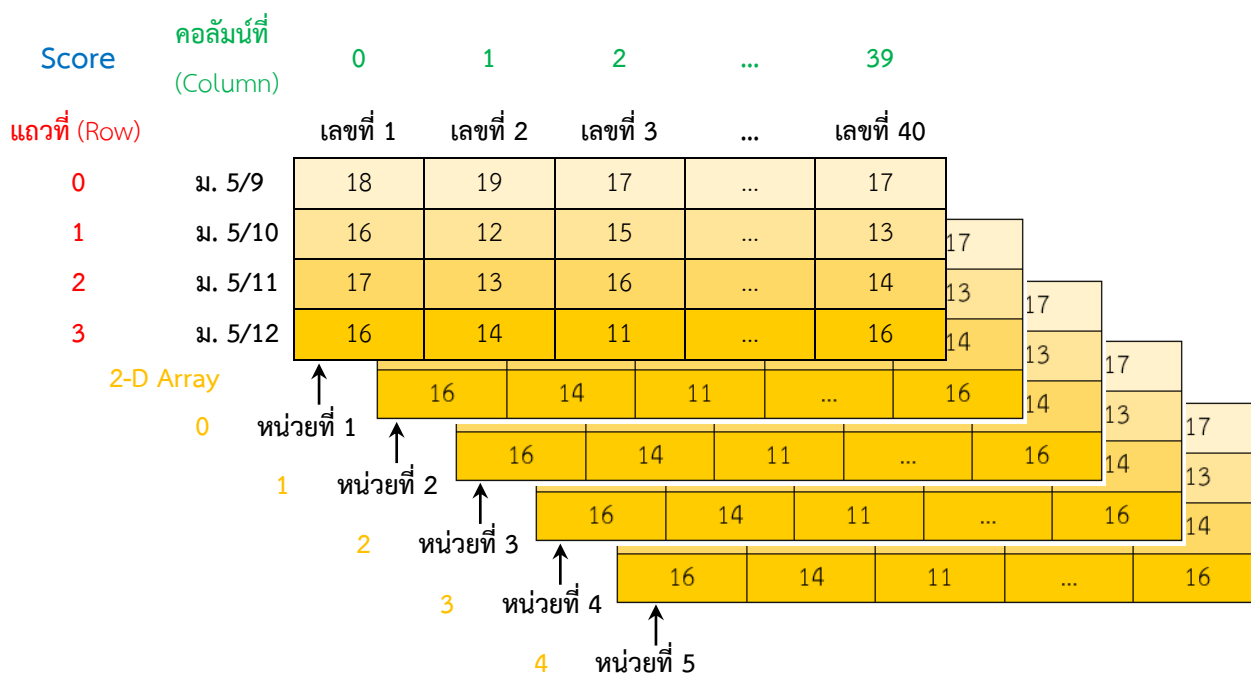
```
E:\hdu4_hdu5-10-2558\down C (visual C)\Code_program\unit3\03-09.exe
Input Row : 2
Input Column : 3
Input number in [1][1] = 1
Input number in [1][2] = 2
Input number in [1][3] = -1
Input number in [2][1] = 0
Input number in [2][2] = 3
Input number in [2][3] = 5

Data items in 2 dimensional array. Size 2 x 3.
 1  2 -1
 0  3  5

Data items in Transpose Matrix. Size 3 x 2.
 1  0
 2  3
-1  5
```

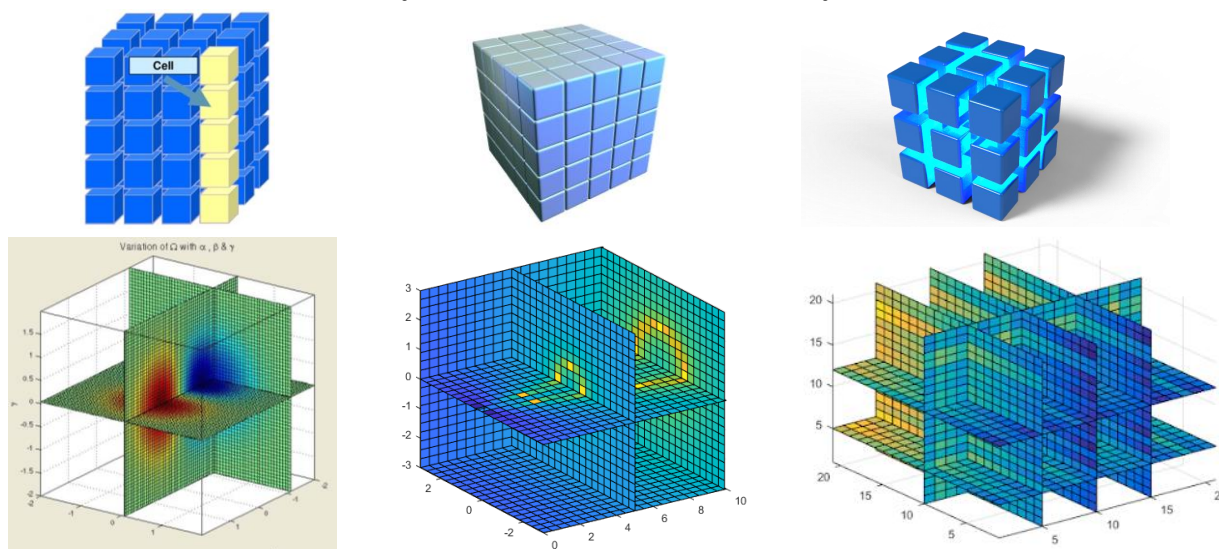
## อาร์เรย์หลายมิติ (Multi-Dimensional Array)

ลักษณะของตัวแปรอาร์เรย์หลายมิติ จะเก็บข้อมูลมากกว่า 2 ทิศทาง เช่น ในแนวแถว (Row) คอลัมน์ (Column) และความลึก (Depth) ซึ่งการนำตัวแปรอาร์เรย์หลายมิติมาใช้ เหมาะสำหรับการเก็บข้อมูลบางประเภทที่นำไปเก็บในตัวแปรอาร์เรย์แบบมิติเดียวหรือสองมิติจะทำงานไม่สะดวก ตัวอย่างการใช้อาร์เรย์แบบสามมิติ เช่น ต้องการเก็บคะแนนการสอบของนักเรียน 4 ห้องเรียน แต่ละห้องมีนักเรียนจำนวน 40 คน และต้องการเก็บคะแนนรายหน่วย 5 หน่วย ดังรูป



ตัวแปรอาร์เรย์ 3 มิติ ชื่อ Score สำหรับเก็บคะแนนของนักเรียน มีจำนวนสมาชิกทั้งหมด 800 จำนวน (4x40x5)

มาตรฐาน ANSI C กำหนดให้อาร์เรย์มีจำนวนมิติได้ไม่เกิน 12 มิติ อย่างไรก็ตาม การใช้ตัวแปรอาร์เรย์หลายมิติ (Multi-Dimensional Array) ที่มีจำนวนมิติมากกว่า 3 มิติ จะทำให้โปรแกรมมีความซับซ้อนมากยิ่งขึ้น การประกาศตัวแปรอาร์เรย์หลายมิติ จะใช้ดัชนีตามจำนวนข้อมูลที่ต้องการเก็บ เพื่อระบุจำนวนสมาชิกในแต่ละแถว คอลัมน์ และความลึก การประกาศตัวแปรจะใช้ชื่อเพียงชื่อเดียวตามด้วยเครื่องหมาย Square brackets ( [ ] ) คร่อมค่าตัวเลขตามจำนวนที่บอกจำนวนแถว คอลัมน์ และความลึกของข้อมูลที่ต้องการไปเรื่อย ๆ ซึ่งตัวอย่างของรูปแบบอาร์เรย์หลายมิติ ดังนี้



## การประกาศตัวแปรอาร์เรย์หลายมิติ

การประกาศตัวแปรอาร์เรย์หลายมิติ จะใช้ดัชนีตามจำนวนข้อมูลที่ต้องการเก็บ เพื่อระบุจำนวนสมาชิกในแต่ละแถว คอลัมน์ และความลึก การประกาศตัวแปรจะใช้ชื่อเพียงชื่อเดียวตามด้วยเครื่องหมาย Square brackets ([ ]) คร่อมค่าตัวเลขตามจำนวนที่บอกจำนวนแถว คอลัมน์ และความลึกของข้อมูลที่ต้องการไปเรื่อย ๆ โดยมีรูปแบบดังนี้

### รูปแบบ

```
type array_name[size1][size2] ... [sizeN];
```

หรือ

```
type array_name[size1][size2] ... [sizeN] = { value_list };
```

### คำอธิบาย

**type** คือ ชนิดของข้อมูล เช่น int, float, char เป็นต้น

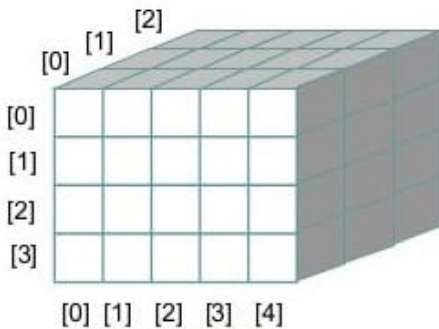
**array\_name** คือ ชื่อตัวแปรอาร์เรย์สองมิติหรือหลายมิติ

**size1, size2, ... sizeN** คือ จำนวนข้อมูลในอาร์เรย์มิติที่ 1, มิติที่ 2 จนถึง มิติที่ N

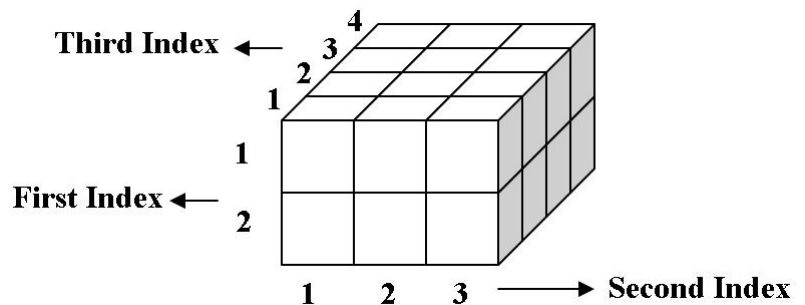
ค่าเริ่มต้นแต่ละมิติจะเริ่มที่ 0

**value\_list** คือ การกำหนดค่าเริ่มต้นให้กับข้อมูลในอาร์เรย์ ถ้ามีข้อมูลหลายตัวจะต้องแยกกันโดยใช้เครื่องหมายคอมม่า ( , ) คั่นไว้

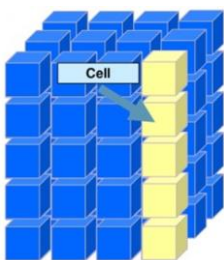
### ตัวอย่างรูปแบบอาร์เรย์ 3 มิติ



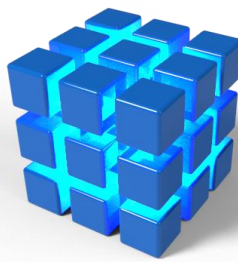
3-D Array, with size 4x5x3.



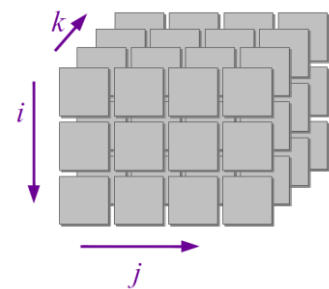
3-D Array, with size 2x3x4.



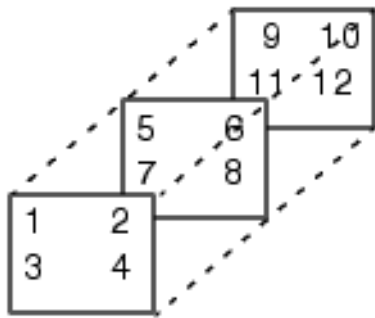
3-D Array, with size 5x4x3.



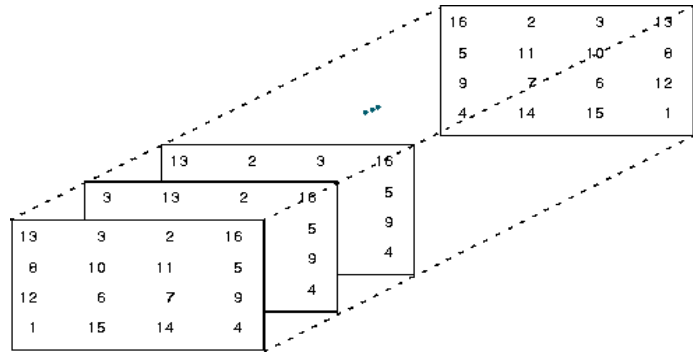
3-D Array, with size 3x3x3.



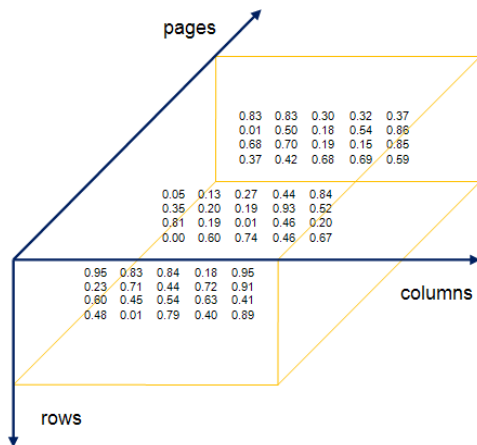
3-D Array, with size 3x4x4.



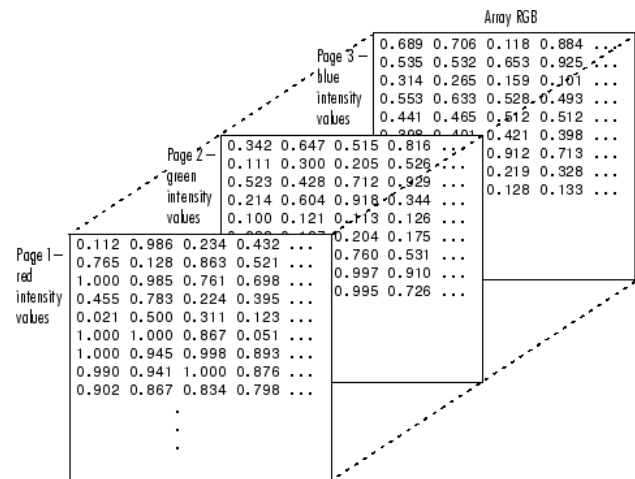
3-D Array, with size 2x2x3.



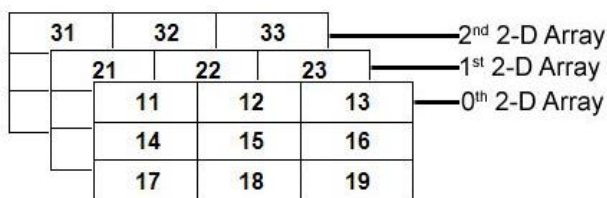
3-D Array, with size 4x4x[...]



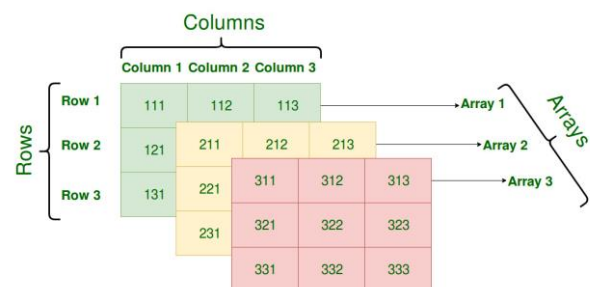
3-D Array, with size 4x5x3.



3-D Array, with size [...x...x3.



3-D Array, with size 3x3x3



3-D Array, with size 3x3x3

### การอ้างอิงถึงข้อมูลในตัวแปรอาร์เรย์สามมิติ

ตัวแปรอาร์เรย์สามมิติเป็นตัวแปรที่เก็บข้อมูลเป็นชุด ดังนั้นภายในตัวแปรอาร์เรย์จึงประกอบไปด้วยข้อมูลหลายค่า การจะเข้าถึงข้อมูลตัวใดตัวหนึ่งในตัวแปรอาร์เรย์สามมิติจึงต้องมีการเฉพาะ นั่นก็คือ การระบุ**บุนต์เก้ซ** (Index) หรือตัวเลขแสดงตำแหน่งแถว (Row) คอลัมน์ (Column) และความลึก (Depth) ของข้อมูลภายในเครื่องหมาย Square brackets ([ ]) ต่อท้ายชื่อตัวแปรอาร์เรย์สามมิติ ยกตัวอย่างเช่น `n[3][4][2]` , `x[2][3][5]` , `score[4][40][5]` เป็นต้น

โดยวิธีการระบุ**บุนต์เก้ซ** (Index) ของตัวแปรอาร์เรย์สามมิติ **ตำแหน่งแรก**ของตัวเลขจะแสดงตำแหน่งแถว (Row) **ตำแหน่งที่สอง**ของตัวเลขจะแสดงตำแหน่งคอลัมน์ (Column) และ**ตำแหน่งที่สาม**ของตัวเลขจะแสดงตำแหน่งความลึก (Depth) ของข้อมูล ในตัวแปรอาร์เรย์ทุกตัวจะมี**บุนต์เก้ซ** (index) เริ่มต้นเป็น 0

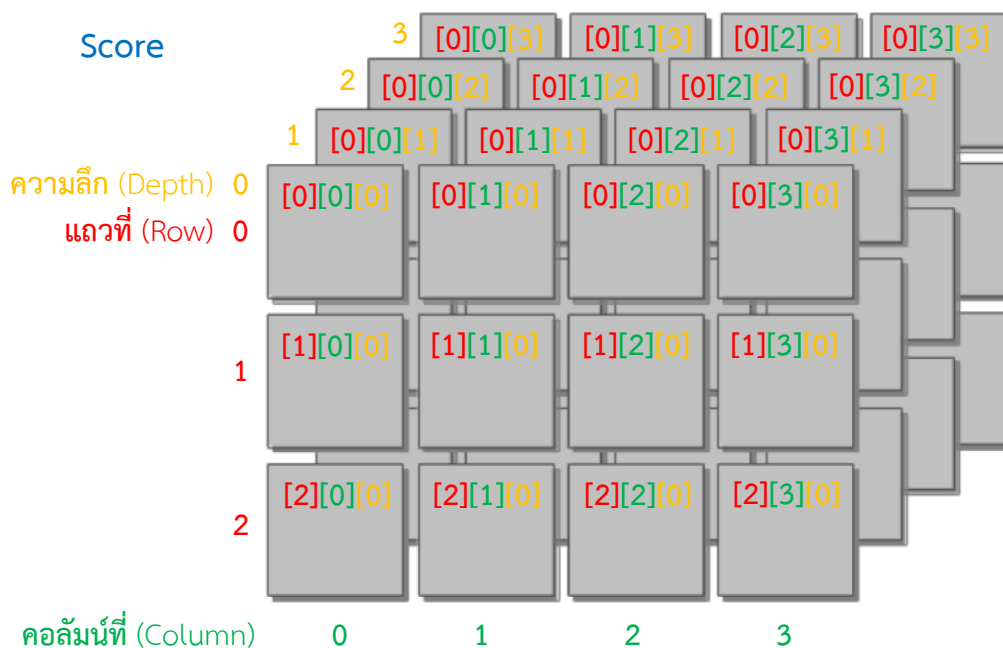


### ตัวอย่างการอ้างอิงถึงข้อมูลในตัวแปรอาร์เรย์สองมิติ

num[1][5][3] ; หมายถึง การอ้างอิงถึงข้อมูลในเซลล์ที่อยู่ในแถวที่ 1 คอลัมน์ที่ 5 และแนวลึกที่ 3 ของตัวแปรอาร์เรย์สามมิติชื่อ num

score[0][3][1] ; หมายถึง การอ้างอิงถึงข้อมูลในเซลล์ที่อยู่ในแถวที่ 0 คอลัมน์ที่ 3 และแนวลึกที่ 1 ของตัวแปรอาร์เรย์สามมิติชื่อ score

temp[4][1][2] ; หมายถึง การอ้างอิงถึงข้อมูลในเซลล์ที่อยู่ในแถวที่ 4 คอลัมน์ที่ 1 และแนวลึกที่ 2 ของตัวแปรอาร์เรย์สามมิติชื่อ temp



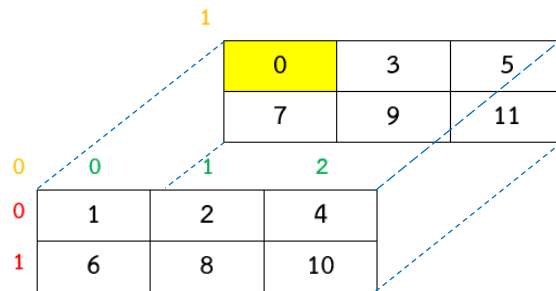
การอ้างอิงถึงตัวแปรอาร์เรย์ 3 มิติ ชื่อ Score สำหรับเก็บคะแนนของนักเรียน มีจำนวนสมาชิกทั้งหมด 48 จำนวน (ขนาด 3x4x4)

### ตัวอย่างการประกาศตัวแปรอาร์เรย์ 3 มิติ (แบบไม่มีค่าเริ่มต้น)

```
int    n[5][4][3] ;
int    num[2][10][3] ;
int    score[4][40][5] , salary[8][10][5] ;
float  j[3][3][3] ;
float  hightemp[5][31][2] , frequency[12][10][3] ;
```



```
int x[2][3][2] = {
    {{ 1 }, { 2, 3 }, { 4, 5 }},
    {{ 6, 7 }, { 8, 9 }, { 10, 11 }}
};
```



\* หากกำหนดค่าเริ่มต้นของแต่ละแถวไม่ครบทุกหลัก เซลล์ของอาร์เรย์ในหลักที่ไม่ถูกกำหนดจะเก็บค่า 0

\*\* หากไม่กำหนดขนาดของแถวหรือคอลัมน์ โปรแกรมจะคำนวณขนาดของแถวหรือคอลัมน์ให้อัตโนมัติโดยดูจากข้อมูลที่กำหนดมาให้

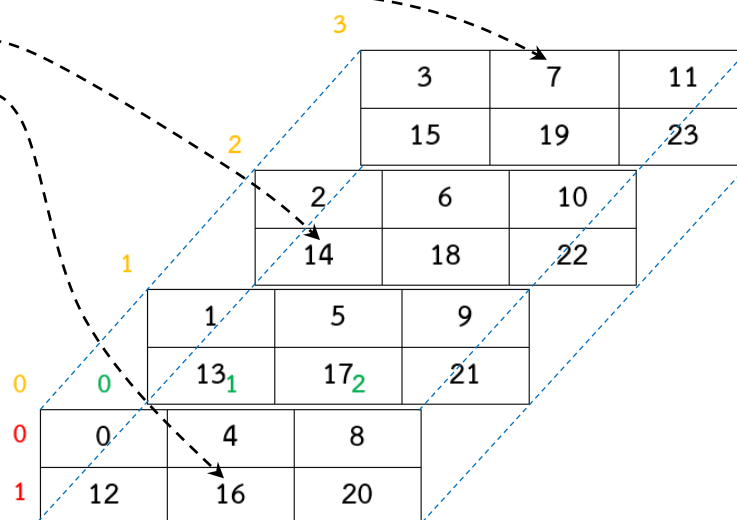
ส่วนการกำหนดค่าของข้อมูลเพื่อนำไปเก็บในตัวแปรอาร์เรย์สามมิติรูปแบบที่สอง ก็คือ การกำหนดค่า (Assignment) เพื่อเก็บลงในตำแหน่งใดตำแหน่งหนึ่ง โดยการระบุอินเด็กซ์ (Index) ของตำแหน่งที่ต้องการ ซึ่งแบบนี้จะเป็นการกำหนดค่าของข้อมูลที่ละค่า ตัวอย่างเช่น

```
int x[2][3][4];
```

```
x[0][1][3] = 7;
```

```
x[1][0][2] = 14;
```

```
x[1][1][0] = 16;
```



การนำอาร์เรย์สามมิติไปใช้กับคำสั่ง printf () และ scanf ()

```
int    n[2][3][4] ;
scanf ("%d", &n[0][1][3]);    // รับค่าจำนวนเต็มนำไปเก็บไว้ในเซลล์แถวที่ 0 คอลัมน์ที่ 1 และแนวลึกที่ 3
                                ของตัวแปรอาร์เรย์สามมิติชื่อ n

scanf ("%d", &n[1][0][2]);    // รับค่าจำนวนเต็มนำไปเก็บไว้ในเซลล์แถวที่ 1 คอลัมน์ที่ 0 และแนวลึกที่ 2
                                ของตัวแปรอาร์เรย์สามมิติชื่อ n

scanf ("%d", &n[1][2][0]);    // รับค่าจำนวนเต็มนำไปเก็บไว้ในเซลล์แถวที่ 1 คอลัมน์ที่ 2 และแนวลึกที่ 0
                                ของตัวแปรอาร์เรย์สามมิติชื่อ n

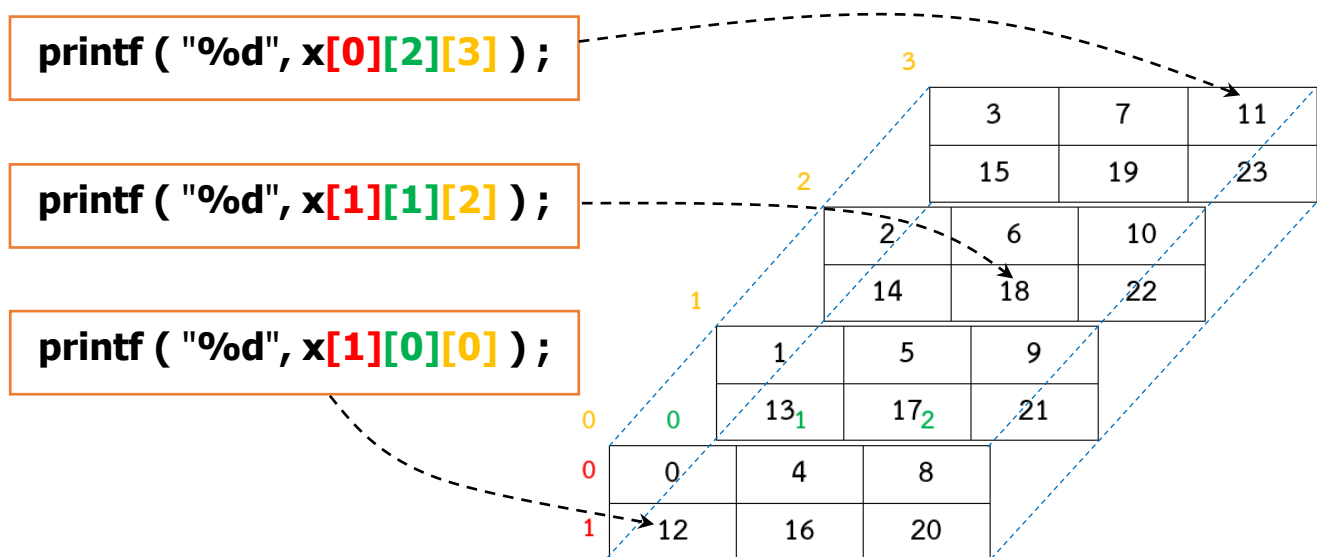
printf ("%d", n[0][0][3]);    // แสดงผลข้อมูลจำนวนเต็มที่เก็บไว้ในเซลล์แถวที่ 0 คอลัมน์ที่ 0 และแนวลึกที่ 3
                                ของตัวแปรอาร์เรย์สามมิติชื่อ n

printf ("%d", n[1][0][2]);    // แสดงผลข้อมูลจำนวนเต็มที่เก็บไว้ในเซลล์แถวที่ 1 คอลัมน์ที่ 0 และแนวลึกที่ 2
                                ของตัวแปรอาร์เรย์สามมิติชื่อ n

printf ("%d", n[1][2][3]);    // แสดงผลข้อมูลจำนวนเต็มที่เก็บไว้ในเซลล์แถวที่ 1 คอลัมน์ที่ 2 และแนวลึกที่ 3
                                ของตัวแปรอาร์เรย์สามมิติชื่อ n
```

เช่น

```
int  x[2][3][4] = {
    { { 0, 1, 2, 3 }, { 4, 5, 6, 7 }, { 8, 9, 10, 11 } },
    { { 12, 13, 14, 15 }, { 16, 17, 18, 19 }, { 20, 21, 22, 23 } }
};
```



## ตัวอย่างโปรแกรมตัวแปรอาร์เรย์หลายมิติ

Ex10. โปรแกรมแสดงผลค่าข้อมูลที่เก็บในตัวแปรอาร์เรย์ 3 มิติ

```
#include <stdio.h>
int main()
{
    int i, j, k;
    int x[2][3][2] = {
        { {0,1}, {2,3}, {4,5} },
        { {6,7}, {8,9}, {10,11} }
    };
    for ( i=0; i<2; i++ )
    {
        for ( j=0; j<3; j++ )
        {
            for ( k=0; k<2; k++ )
            {
                printf ("Element at x[%d][%d]%d] = %d\n", i, j, k, x[i][j][k]);
            }
        }
    }
    printf ("\nData items in 3 Dimensional array.\n");
    for ( i=0; i<2; i++ )
    {
        for ( j=0; j<2; j++ )
        {
            for ( k=0; k<3; k++ )
            {
                printf ("%3d ", x[j][k][i]);
            }
            printf ("\n");
        }
    }
    printf ("\n");
    return 0;
}
```

## ผลลัพธ์

E:\สอน\_hd\_5-10-2558\หน่วยที่ 2)\Code\_program\unit3\03-10.exe

```
Element at x[0][0]0] = 0
Element at x[0][0]1] = 1
Element at x[0][1]0] = 2
Element at x[0][1]1] = 3
Element at x[0][2]0] = 4
Element at x[0][2]1] = 5
Element at x[1][0]0] = 6
Element at x[1][0]1] = 7
Element at x[1][1]0] = 8
Element at x[1][1]1] = 9
Element at x[1][2]0] = 10
Element at x[1][2]1] = 11
```

Data items in 3 Dimensional array.

```
0  2  4
6  8 10

1  3  5
7  9 11
```

Ex11. โปรแกรมรับข้อมูลตัวเลขจำนวนเต็ม ขนาด  $2 \times 3 \times 2$  จำนวน นำไปเก็บในตัวแปรอาร์เรย์ 3 มิติ แล้วแสดงผลทางจอภาพ

```
#include <stdio.h>
int main()
{
    int i, j, k;
    int x[2][3][2];
    for ( i=0; i<2; i++ )
    {
        for ( j=0; j<3; j++ )
        {
            for ( k=0; k<2; k++ )
            {
                printf ("Element at x[%d][%d]%d] = ", i, j, k);
                scanf ("%d", &x[i][j][k]);
            }
        }
    }
    printf ("\nData items in 3 Dimensional array.\n");
    for ( i=0; i<2; i++ )
    {
        for ( j=0; j<2; j++ )
        {
            for ( k=0; k<3; k++ )
            {
                printf ("%3d ", x[j][k][i]);
            }
            printf ("\n");
        }
    }
    printf ("\n");
    return 0;
}
```

ผลลัพธ์

```
Element at x[0][0]0] = 0
Element at x[0][0]1] = 1
Element at x[0][1]0] = 2
Element at x[0][1]1] = 3
Element at x[0][2]0] = 4
Element at x[0][2]1] = 5
Element at x[1][0]0] = 6
Element at x[1][0]1] = 7
Element at x[1][1]0] = 8
Element at x[1][1]1] = 9
Element at x[1][2]0] = 10
Element at x[1][2]1] = 11

Data items in 3 Dimensional array.
  0  2  4
  6  8 10

  1  3  5
  7  9 11
```



ตัวอย่างการประกาศตัวแปรสตริงหรือสายอักขระ (แบบมีค่าเริ่มต้น) สำหรับการกำหนดค่าของข้อมูลเพื่อนำไปเก็บในตัวแปรสตริงหรือสายอักขระสามารถทำได้ 2 รูปแบบ โดยแบบแรกเป็นการกำหนดค่าของข้อมูลทุกตำแหน่งในตัวแปรอาร์เรย์พร้อมกันทีเดียว ซึ่งข้อมูลทั้งหมดจะต้องเขียนภายในเครื่องหมาย { } และคั่นระหว่างข้อมูลแต่ละตัวด้วยเครื่องหมายคอมม่า ( , ) ตัวอย่างเช่น

```
char str[10] = { 'C', 'a', 't', '\0' };
char text[ ] = { 'H', 'o', 'r', 's', 'e', '\0' };
char sport[ ] = "Football";
char zoo[ ] = "monkey";
```

การเก็บข้อมูลลงในอีลีเมนต์ (Element) หรือ เซลล์ (Cell)

```
char str[10] = { 'C', 'a', 't', '\0' };
```

| Index→ | str[0] | str[1] | str[2] | str[3] | str[4] | str[5] | str[6] | str[7] | str[8] | str[9] |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| srt    | C      | a      | t      | \0     |        |        |        |        |        |        |

```
char text[ ] = { 'H', 'o', 'r', 's', 'e', '\0' };
```

| Index→ | text[0] | text[1] | text[2] | text[3] | text[4] | text[5] |
|--------|---------|---------|---------|---------|---------|---------|
| text   | H       | o       | r       | s       | e       | \0      |

```
char sport[ ] = "Football";
```

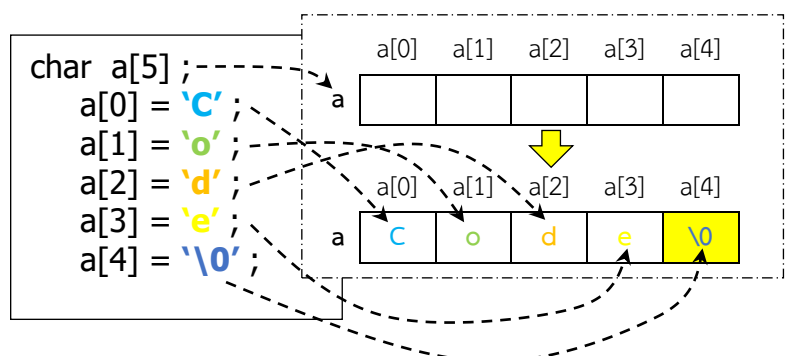
| Index→ | sport[0] | sport[1] | sport[2] | sport[3] | sport[4] | sport[5] | sport[6] | sport[7] | sport[8] |
|--------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| sport  | F        | o        | o        | t        | b        | a        | l        | l        | \0       |

```
char zoo[ ] = "monkey";
```

| Index→ | zoo[0] | zoo[1] | zoo[2] | zoo[3] | zoo[4] | zoo[5] | zoo[6] |
|--------|--------|--------|--------|--------|--------|--------|--------|
| zoo    | m      | o      | n      | k      | e      | y      | \0     |

ส่วนการกำหนดค่าของข้อมูลเพื่อนำไปเก็บในตัวแปรอาร์เรย์รูปแบบที่สอง ก็คือ การกำหนดค่า (Assignment) เพื่อเก็บลงในตำแหน่งใดตำแหน่งหนึ่ง โดยการระบุอินเด็กซ์ (Index) ของตำแหน่งที่ต้องการ ซึ่งแบบนี้จะเป็นการกำหนดค่าของข้อมูลที่ละค่า หรือครบทุกค่า ตัวอย่างเช่น

```
str[1] = 'A';
text[0] = 'h';
zoo[0] = 'M';
```





การเก็บข้อมูลลงในอีลีเมนต์ (Element) หรือ เซลล์ (Cell)

str[1] = 'A' ;

|         |        |        |        |        |        |        |        |        |        |        |
|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Index → | str[0] | str[1] | str[2] | str[3] | str[4] | str[5] | str[6] | str[7] | str[8] | str[9] |
| str     | C      | A      | t      | \0     |        |        |        |        |        |        |

text[0] = 'h' ;

|         |         |         |         |         |         |         |
|---------|---------|---------|---------|---------|---------|---------|
| Index → | text[0] | text[1] | text[2] | text[3] | text[4] | text[5] |
| text    | h       | o       | r       | s       | e       | \0      |

zoo[0] = 'M' ;

|         |        |        |        |        |        |        |        |
|---------|--------|--------|--------|--------|--------|--------|--------|
| Index → | zoo[0] | zoo[1] | zoo[2] | zoo[3] | zoo[4] | zoo[5] | zoo[6] |
| zoo     | M      | o      | n      | k      | e      | y      | \0     |

การนำข้อมูลสตริงหรือสายอักขระ ไปใช้กับคำสั่ง gets() scanf() และ printf()

สำหรับการรับข้อมูลสตริงทางคีย์บอร์ดนั้นจะใช้ฟังก์ชัน gets() และเมื่อฟังก์ชันนี้ถูกเรียกใช้คอมพิวเตอร์จะรอรับข้อมูลทางแป้นพิมพ์ จนกว่าผู้ใช้จะกดปุ่ม <Enter> ถึงจะเก็บข้อมูลเข้าไปในตัวแปรอาร์เรย์ที่เก็บสตริง

หรือ เราสามารถที่จะรับค่าข้อมูลสตริงจากคีย์บอร์ดแล้วเก็บในอาร์เรย์ของข้อมูลแบบสตริงหรือสายอักขระได้โดยใช้ scanf() และการใช้การกำหนดรหัสรูปแบบข้อมูลเป็น %s (ใช้สำหรับข้อมูลแบบสตริงหรือสายอักขระ) การกรอกข้อมูลข้อมูลแบบสตริงหรือสายอักขระ ความยาวของข้อมูลนั้นต้องไม่เกินขนาดของอาร์เรย์ที่ประกาศไว้

นอกจากนั้น การใช้คำสั่งในอาณัติที่รับค่าตัวแปรของคำสั่ง scanf() จะแตกต่างจากที่ผ่านมา คือ **จะไม่มีตัวดำเนินการ &** ที่ใช้แสดงตำแหน่งบนหน่วยความจำของตัวแปรกำกับอยู่หน้าตัวแปร เนื่องจากชื่อของตัวแปรแล้วลำดับในการประกาศตัวแปรแบบสตริงหรือสายอักขระ หมายถึง ตำแหน่งบนหน่วยความจำของสมาชิกตัวแรกของแถวลำดับ ดังนั้นจึงไม่ต้องใช้ตัวดำเนินการ & เพื่อบอกตำแหน่งบนหน่วยความจำของตัวแปร ดังตัวอย่าง

```
char name[20] ;
```

```
scanf ("%s", name) ; // รับค่าข้อมูลทางแป้นพิมพ์แบบสตริงแล้วนำไปเก็บในตัวแปรอาร์เรย์ชื่อ name
```

หรือ

```
gets (name) ; // รับค่าข้อมูลทางแป้นพิมพ์แบบสตริงแล้วนำไปเก็บในตัวแปรอาร์เรย์ชื่อ name
```

```
printf ("%s", name) ; // แสดงผลข้อมูลแบบสตริงที่เก็บไว้ในตัวแปรอาร์เรย์ชื่อ name ออกทางจอภาพ
```

หรือ

```
printf ("%c", name[0]) ; // แสดงผลข้อมูลหนึ่งตัวอักขระที่เก็บไว้ในเซลล์อินเด็กซ์ที่ 0 ของตัวแปรอาร์เรย์ชื่อ name
```

## ตัวอย่างโปรแกรมข้อมูลแบบสตริงหรือสายอักขระ

Ex12. โปรแกรมแสดงข้อมูลแบบสตริงหรือสายอักขระที่เก็บในตัวแปร (ใช้รหัสรูปแบบข้อมูล %s)

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    char str[10] = { 'N', 'a', 'n', 'g', 'r', 'o', 'n', 'g', '\0' };
```

```
    printf ("Data items in array.\n");
```

```
    printf ("%s", str);
```

```
    return 0;
```

```
}
```

|     | str[0] | str[1] | str[2] | str[3] | str[4] | str[5] | str[6] | str[7] | str[8] | str[9] |
|-----|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| str | N      | a      | n      | g      | r      | o      | n      | g      | \0     |        |

ผลลัพธ์

```
E:\สำรอง_hd_5-10-2558\เนื้อหา C (พจน 2)\Code_program\unit3\03-12.exe
Data items in array.
Nangrong
-----
Process exited after 0.08378 seconds
Press any key to continue . . .
```

Ex13. โปรแกรมแสดงข้อมูลแบบสตริงหรือสายอักขระที่เก็บในตัวแปร (ใช้รหัสรูปแบบข้อมูล %c)

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int i;
```

```
    char str[10] = { 'N', 'a', 'n', 'g', 'r', 'o', 'n', 'g', '\0' };
```

```
    printf ("Data items in array.\n");
```

```
    for ( i=0; i<10; i++ )
```

```
    {
```

```
        printf ("%c", str[i]);
```

```
    }
```

```
    return 0;
```

```
}
```

|     | str[0] | str[1] | str[2] | str[3] | str[4] | str[5] | str[6] | str[7] | str[8] | str[9] |
|-----|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| str | N      | a      | n      | g      | r      | o      | n      | g      | \0     |        |

ผลลัพธ์

```
E:\สำรอง_hd_5-10-2558\เนื้อหา C (พจน 2)\Code_program\unit3\03-13.exe
Data items in array.
Nangrong
-----
Process exited after 0.1535 seconds
Press any key to continue . . .
```

Ex14. โปรแกรมรับข้อมูลชื่อที่เป็นสตริง แล้วแสดงผลออกทางจอภาพที่ละเซลล์

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int i;
```

```
    char str[30];
```

```
    printf ("Input your name = ");
```

```
    gets(str);
```

```
    printf ("\nMy name is ");
```

```
    for ( i=0; str[i] ; i++ )
```

```
    {
```

```
        printf ("%c", str[i]);
```

```
    }
```

```
    return 0;
```

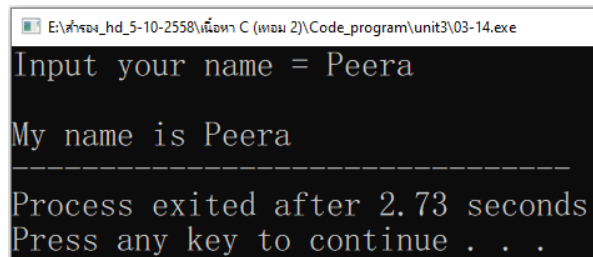
```
}
```

|     | str[0] | str[1] | str[2] | str[3] | str[4] | str[5] | str[6] | str[7] | ... | str[29] |
|-----|--------|--------|--------|--------|--------|--------|--------|--------|-----|---------|
| str |        |        |        |        |        |        |        |        |     |         |
|     | str[0] | str[1] | str[2] | str[3] | str[4] | str[5] | str[6] | str[7] | ... | str[29] |
| str | P      | e      | e      | r      | a      | \0     |        |        |     |         |

```
// scanf ("%s", str); or gets(str);
```

```
// str[i] or i<30
```

ผลลัพธ์



```
E:\สอน_hd_5-10-2558\เนื้อหา C (พอม 2)\Code_program\unit3\03-14.exe
Input your name = Peera
My name is Peera
-----
Process exited after 2.73 seconds
Press any key to continue . . .
```

จากโปรแกรมในตัวอย่างที่ EX14. จะเห็นว่าการรับข้อมูลสตริงเข้าไปนั้นก็คืออาร์เรย์ของตัวอักษร ซึ่งจะพบว่าเราสามารถนำอาร์เรย์มาพิมพ์ที่ละเซลล์ได้ และถ้าสังเกตคำสั่ง for ในโปรแกรมจะเห็นว่าในส่วนที่ใช้ตรวจสอบเงื่อนไขจะเป็น str[i] ก็เนื่องจาก str[i] ถ้ามีตัวอักษรอยู่จะถือว่ามีความเป็นจริง แต่ถ้า str[i] ไม่มีตัวอักษรอยู่จะถือว่ามีความเป็นเท็จ ดังนั้นค่า i จะนับไปเรื่อย ๆ เพิ่มขึ้นทีละหนึ่งไปจนถึงตำแหน่งที่ str[i] ไม่มีตัวอักษรเก็บอยู่

Ex15. โปรแกรมรับข้อความที่เป็นสตริงแล้วเข้ารหัสซีซาร์ (Caesar cipher) แล้วแสดงผลข้อความที่เข้ารหัสออกทางจอภาพ (ถ้ากรอก A จะเข้ารหัสเป็น D, ถ้ากรอก b จะเข้ารหัสเป็น e เป็นต้น)

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int i;
    char decode[30];
    printf ("Enter text to encrypt : ");
    gets(decode);
    system ("cls");
    printf ("unCrypt = %s\n", decode);
    printf ("Crypted = ");
    for (i=0; i<30; i++)
    {
        decode[i] = decode[i]+3;
        printf ("%c", decode[i]);
    }
    return 0;
}
```

|     |        |        |        |        |        |        |        |        |     |         |
|-----|--------|--------|--------|--------|--------|--------|--------|--------|-----|---------|
|     | str[0] | str[1] | str[2] | str[3] | str[4] | str[5] | str[6] | str[7] | ... | str[29] |
| str |        |        |        |        |        |        |        |        |     |         |

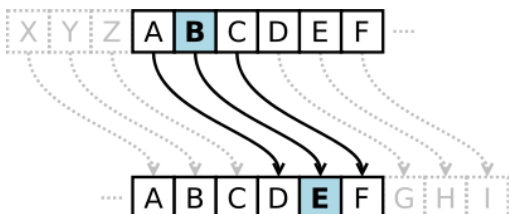
|     |        |        |        |        |        |        |        |        |     |         |
|-----|--------|--------|--------|--------|--------|--------|--------|--------|-----|---------|
|     | str[0] | str[1] | str[2] | str[3] | str[4] | str[5] | str[6] | str[7] | ... | str[29] |
| str | S      | c      | h      | o      | o      | l      | \0     |        |     |         |

ผลลัพธ์

```
unCrypt = School
Crypted = Vfkrro♥♥♥♥♥♥♥♥♥♥
Process exited after 4.252 seconds with return value 0
Press any key to continue . . .
```

`system ("cls");` เป็นคำสั่งที่ใช้ในการเคลียร์หน้าจอ (Clear screen) ซึ่งถ้าจะใช้งานคำสั่งล้างหน้าจอจะต้องเรียกใช้ฟังก์ชันซึ่งจะเก็บอยู่ใน “`stdlib.h`” ซึ่งจะต้องเรียกใช้โดยการเพิ่ม `#include <stdlib.h>`

**รหัสซีซาร์ (Caesar cipher)** ในทางด้านวิทยาการเข้ารหัสลับ หรือเป็นที่รู้จักกันในชื่ออื่นว่า shift cipher Caesar's code หรือ Caesar shift เป็นเทคนิคการเข้ารหัสที่ง่ายและแพร่หลายที่สุด โดยใช้หลักการแทนที่ตัวอักษร ซึ่งในแต่ละตัวอักษรที่อยู่ในข้อความจะถูกแทนที่ด้วยตัวอักษรที่อยู่ลำดับถัดไปตามจำนวนตัวอักษรที่แน่นอน อย่างเช่น แปลงตัวอักษรไป 3 ตัว อักษร "A" ก็จะถูกแทนที่ด้วยอักษร "D" การเข้ารหัสแบบดังกล่าวตั้งชื่อตามจูเลียส ซีซาร์ ซึ่งคิดค้นขึ้นเพื่อการติดต่อสื่อสารกับแม่ทัพนายกองของเขา



รูปแบบการเข้ารหัสแบบซีซาร์ใช้หลักการแทนที่ตัวอักษร จากตัวอย่างแสดงให้เห็นถึงการแทนที่ตัวอักษรด้วยตัวอักษรที่อยู่ถัดไป 3 ตัว ดังนั้นตัวอักษร "A" จะถูกแทนที่ด้วย "D" และ "B" จะถูกแทนที่ด้วย "E" ไปเรื่อย ๆ

## ฟังก์ชันเกี่ยวกับสตริง

ในการเขียนโปรแกรมด้วยภาษาซี มีฟังก์ชันเกี่ยวกับสตริงที่น่าสนใจอยู่หลายฟังก์ชัน โดยฟังก์ชันต่าง ๆ จะเก็บอยู่ใน “string.h” ฟังก์ชันดำเนินการกับสตริงที่น่าสนใจ มีดังนี้

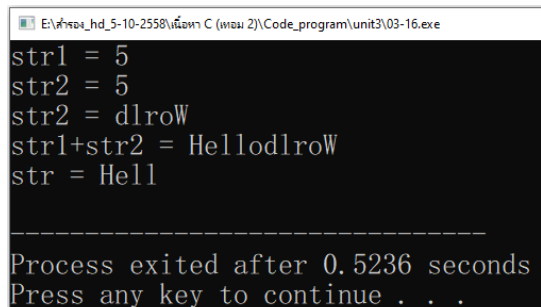
| คำสั่ง               | ใช้สำหรับ                                                          |
|----------------------|--------------------------------------------------------------------|
| strlen(str)          | หาความยาวของสตริงหรือสายอักขระ จะได้ตัวเลขจำนวนเต็มออกมา           |
| strlwr(str)          | เปลี่ยนสตริงให้เป็นตัวพิมพ์เล็ก                                    |
| strupr(str)          | เปลี่ยนสตริงให้เป็นตัวพิมพ์ใหญ่                                    |
| strrev(str)          | กลับตัวอักษรในสตริงจากซ้ายไปขวา                                    |
| strcpy(str1,str2)    | คัดลอกสตริง str2 ไปให้ตัวแปร str1                                  |
| strncpy(str1,str2,n) | คัดลอกสตริง str2 ไปให้ตัวแปร str1 เป็นจำนวน n ตัว                  |
| strcat(str1,str2)    | เชื่อมต่อสตริง str2 เข้ากับส่วนท้ายของสตริง str1                   |
| strcmp(str1,str2)    | เปรียบเทียบสตริงของตัวแปร str1 กับ str2 ถ้าเท่ากันจะคืนค่า 0 ออกมา |

เมื่อ str, str1 และ str2 เป็นตัวแปรสตริงหรือสายอักขระใด ๆ

Ex16. โปรแกรมแสดงการใช้ฟังก์ชันเกี่ยวกับสตริง

```
#include <stdio.h>
#include <string.h>
int main()
{
    char str1[12] = "Hello";
    char str2[12] = "World";
    char str3[12];
    printf ("str1 = %d\n", strlen(str1));
    printf ("str2 = %d\n", strlen(str2));
    printf ("str2 = %s\n", strrev(str2));
    printf ("str1+str2 = %s\n", strcat(str1,str2));
    strncpy(str3, str1, 4);
    printf ("str = %s\n", str3);
    return 0;
}
```

ผลลัพธ์



```
E:\ฟรอม_hd_5-10-2558\เนื้อหา C (เล่ม 2)\Code_program\unit3\03-16.exe
str1 = 5
str2 = 5
str2 = dlroW
str1+str2 = HellodlroW
str = Hell
-----
Process exited after 0.5236 seconds
Press any key to continue . . .
```

การสุ่ม/ เกมทายตัวเลขจากการสุ่ม `ran()`

การเข้ารหัส/ ทายบัตรเครดิต

### ใบงานหน่วยที่ 3

#### ตัวแปรชนิดอาร์เรย์และสตริง

โจทย์ปัญหาโดยใช้ตัวแปรอาร์เรย์ 1 มิติ หรือแถวลำดับ

1. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มบวกไปเก็บในอาร์เรย์ 5 จำนวน จากนั้นให้โปรแกรมหาค่าผลรวมและค่าเฉลี่ยของข้อมูลทั้งหมด

| Input                      | Output                      |
|----------------------------|-----------------------------|
| 20<br>15<br>30<br>12<br>11 | Sum = 88<br>Average = 17.6  |
| 45<br>17<br>22<br>13<br>37 | Sum = 134<br>Average = 26.8 |

2. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มบวกไปเก็บในอาร์เรย์ 5 จำนวน จากนั้นให้โปรแกรมหาค่าผลรวมและค่าเฉลี่ย ค่าสูงสุด ค่าต่ำสุด

| Input                      | Output                                              |
|----------------------------|-----------------------------------------------------|
| 20<br>15<br>30<br>12<br>11 | Sum = 88<br>Average = 17.6<br>Max = 30<br>Min = 11  |
| 45<br>17<br>22<br>13<br>37 | Sum = 134<br>Average = 26.8<br>Max = 45<br>Min = 13 |

3. จงเขียนโปรแกรมรับค่าจำนวนเต็มแล้วนำไปเก็บในอาร์เรย์หนึ่งมิติ และสัญลักษณ์ที่จะสร้างแผนภูมิภาพ จากนั้นให้แสดงผลเป็นกราฟในแนวนอน

| Input | Output          |
|-------|-----------------|
| 5 *   | Value Histogram |
| 10    | 10 *****        |
| 20    | 20 *****        |
| 16    | 16 *****        |
| 20    | 10 *****        |
| 12    | 12 *****        |
| 6 #   | Value Histogram |
| 5     | 5 #####         |
| 1     | 1 #             |
| 8     | 8 #####         |
| 12    | 12 #####        |
| 4     | 4 #####         |
| 20    | 20 #####        |

4. จงเขียนโปรแกรมเพื่อรับค่าจำนวนเต็ม 5 เก็บในอาร์เรย์หนึ่งมิติ แสดงจำนวนเต็มทีกรอกทั้งหมด และแสดงตำแหน่ง Index Number ของอาร์เรย์ที่ข้อมูลที่มีค่ามากที่สุดและน้อยที่สุด

| Input                      | Output            |
|----------------------------|-------------------|
| 20<br>15<br>30<br>12<br>11 | Hi = 2<br>Low = 4 |
| 45<br>17<br>22<br>13<br>37 | Hi = 0<br>Low = 3 |

5. ข้อมูลที่กำหนดให้ `int arr[10]={ 4, 3, 5, 2, 7, 9, 8, 1, 3, 12 }`; จงเขียนโปรแกรมสำหรับค้นหาและแสดงตำแหน่ง Index Number ของอาร์เรย์แรกที่พบข้อมูลที่เก็บอยู่ในอาร์เรย์

| Input      | Output |
|------------|--------|
| find ?: 7  | 7 : 4  |
| find ?: 12 | 12 : 9 |
| find ?: 3  | 3 : 1  |

6. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็ม 10 จำนวน แล้วจัดเรียงข้อมูล 10 จำนวนนี้จากน้อยไปหามาก (Ascending) แสดงผลทางจอภาพ และแสดงค่าน้อยที่สุด 2 ลำดับแรก

| Input                                                   | Output                                  |
|---------------------------------------------------------|-----------------------------------------|
| 11<br>21<br>35<br>8<br>42<br>55<br>31<br>28<br>17<br>32 | 8 11 17 21 28 31 32 35 42 55<br>8<br>11 |

7. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็ม 7 จำนวน แสดงค่ามากที่สุด 3 ลำดับแรก

| Input                                  | Output   |
|----------------------------------------|----------|
| 21<br>35<br>42<br>55<br>31<br>17<br>32 | 55 42 35 |



### โจทย์ปัญหาโดยใช้ตัวแปรอาร์เรย์ 2 มิติ

8. จงสร้าง Matrix ขนาด  $3 \times 4$  กรอกข้อมูลลงแต่ละช่อง และคำนวณหาผลรวมในแนวนอน และแนวคอลัมน์ของ Matrix นี้ ดังตัวอย่างผลลัพธ์ต่อไปนี้

|       | Column 1 | Column 2 | Column 3 | Column 4 | Total |
|-------|----------|----------|----------|----------|-------|
| Row 1 | 20       | 30       | 10       | 10       | 70    |
| Row 2 | 10       | 25       | 5        | 40       | 80    |
| Row 3 | 20       | 40       | 30       | 30       | 120   |
| Total | 50       | 95       | 45       | 80       | 270   |

9. วิชาแคลคูลัส 1 มีนักศึกษาอยู่  $N$  คน และมีการสอบย่อยทั้งหมด  $k$  ครั้ง โดยที่  $N \leq 1000$  และ  $k \leq 5$  อาจารย์ผู้สอนได้ทำการบันทึกคะแนนสอบของนักศึกษาทีละคน คือ นำคะแนนสอบทั้งหมด  $k$  ครั้งของนักศึกษาค้นแรกบันทึกลงไปในสมุดก่อน แล้วจึงบันทึกคะแนนนักเรียนคนถัดมาทีละคนในลักษณะเดียวกัน ทั้งนี้อาจารย์ผู้สอนต้องการหาด้วยว่า นักศึกษาที่ทำคะแนนได้สูงสุดนั้นได้คะแนนเท่าใด และมีการสอบย่อยกี่ครั้งที่นักศึกษาค้นดังกล่าวได้คะแนนสูงสุดในการสอบครั้งนั้น ๆ ด้วย ทั้งนี้สมมติให้ผู้ที่ทำคะแนนรวมสูงสุดมีเพียงคนเดียวเท่านั้น

#### ข้อมูลเข้า

- บรรทัดแรกเป็นเลขจำนวนเต็ม  $N$  และ  $k$  ตามลำดับ ข้อมูลทั้งสองคั่นด้วยช่องว่าง
- บรรทัดที่สองถึงบรรทัดที่  $N + 1$  เป็นคะแนนสอบของนักศึกษาแต่ละคน หนึ่งคนหนึ่งบรรทัด โดยคะแนนสอบเป็นเลขจำนวนเต็มเรียงจากครั้งที่หนึ่งถึงครั้งที่  $k$  ตามลำดับ คั่นด้วยช่องว่าง

#### ผลลัพธ์

- บรรทัดแรกเป็นเลขจำนวนเต็มแสดงคะแนนรวมสูงสุด
- บรรทัดที่สองคือจำนวนการสอบที่นักศึกษาค้นที่ทำคะแนนรวมดีที่สุดได้คะแนนสูงสุด

| Input                                   | Output  |
|-----------------------------------------|---------|
| 4 3<br>5 7 8<br>7 3 9<br>8 6 9<br>1 2 3 | 23<br>2 |
| 3 4<br>5 6 8 8<br>9 1 2 9<br>1 7 8 4    | 27<br>1 |

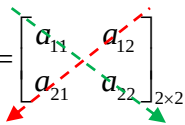
10. จงเขียนโปรแกรมรับตัวเลขจำนวนเต็ม ตามขนาด  $2 \times 2$  ที่กำหนด นำไปเก็บในตัวแปรอาร์เรย์ 2 มิติ แล้วให้แสดงผลดีเทอร์มิแนนต์ของเมทริกซ์ (Determinant) ของข้อมูลที่รับเข้าไปออกทางจอภาพ

A เป็นเมทริกซ์ขนาด  $n \times n$  ดีเทอร์มิแนนต์ของ A เขียนแทนด้วย  $\det(A)$

ตัวอย่าง เช่น  $A = \begin{bmatrix} 3 & 1 \\ 2 & 4 \end{bmatrix}_{2 \times 2}$

$$\begin{aligned}\det(A) &= (3)(4) - (1)(2) \\ &= 12 - 2 \\ &= 10\end{aligned}$$

วิธีคิด  $\det(A)$



$$\det(A) = a_{11}a_{22} - a_{12}a_{21}$$

| Input       | Output |
|-------------|--------|
| 3 2<br>4 5  | 7      |
| 8 4<br>-2 3 | 32     |

11. จงเขียนโปรแกรมรับตัวเลขจำนวนเต็ม ตามขนาดเมทริกซ์จัตุรัส (Square Matrix) ที่มีจำนวนแถวและหลักเท่ากัน นำไปเก็บในตัวแปรอาร์เรย์  $n \times n$  มิติ แล้วให้แสดงผลว่าเป็นเมทริกซ์เอกลักษณ์ (Identity Matrix) ของข้อมูลที่รับเข้าไปออกทางจอภาพ

A เป็นเมทริกซ์ขนาด  $n \times n$  ดีเทอร์มิแนนต์ของ A เขียนแทนด้วย  $\det(A)$

ตัวอย่าง เช่น  $A = \begin{bmatrix} 3 & 1 \\ 2 & 4 \end{bmatrix}_{2 \times 2}$

$$\begin{aligned}\det(A) &= (3)(4) - (1)(2) \\ &= 12 - 2 \\ &= 10\end{aligned}$$

12. จงเขียนโปรแกรมรับตัวเลขจำนวนเต็ม ตามขนาด  $3 \times 3$  ที่กำหนด นำไปเก็บในตัวแปรอาร์เรย์ 2 มิติ แล้วให้แสดงผลดีเทอร์มิแนนต์ของเมทริกซ์ (Determinant) ของข้อมูลที่ได้รับเข้าไปออกทางจอภาพ

A เป็นเมทริกซ์ขนาด  $n \times n$  ดีเทอร์มิแนนต์ของ A เขียนแทนด้วย  $\det(A)$

ตัวอย่าง เช่น  $A = \begin{bmatrix} 1 & -1 & 2 \\ 2 & 1 & 1 \\ 1 & 3 & 2 \end{bmatrix}_{3 \times 3}$

$$\begin{aligned} \det(A) &= (1)(1)(2) + (-1)(1)(1) + (2)(2)(3) - (2)(1)(1) - (1)(1)(3) - (-1)(2)(2) \\ &= 2 - 1 + 12 - 2 - 3 + 4 \\ &= 12 \end{aligned}$$

วิธีคิด  $\det(A)$

$$\det(A) = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

$$\det(A) = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{31}a_{22}a_{13} - a_{32}a_{23}a_{11} - a_{33}a_{21}a_{12}$$

| Input                    | Output |
|--------------------------|--------|
| 1 -1 2<br>2 1 1<br>1 3 2 | 12     |
| 1 0 2<br>-2 2 1<br>1 4 3 | -18    |

การพัฒนาโจทย์สำหรับประยุกต์ใช้อาเรย์ 2 มิติ

13. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มของอาเรย์  $3 \times 4$  แล้วแสดงค่าทั้งหมดออกมาในแถวเดียว

| Input                               | Output                        |
|-------------------------------------|-------------------------------|
| 1 0 12 -1<br>7 -3 2 5<br>-5 -2 2 -9 | 1 0 12 -1 7 -3 2 5 -5 -2 2 -9 |
| 8 0 2 -2<br>-2 4 1 7<br>1 4 3 0     | 8 0 2 -2 -2 4 1 7 1 4 3 0     |

14. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มของอาเรย์  $3 \times 4$  แล้วให้เรียงลำดับข้อมูลโดยอ้างอิงจากคอลัมน์แรก เรียงจากน้อยไปมาก และให้ข้อมูลแต่ละแถวขยับตามทั้งแถว

| Input                               | Output                              |
|-------------------------------------|-------------------------------------|
| 1 0 12 -1<br>7 -3 2 5<br>-5 -2 2 -9 | -5 -2 2 -9<br>1 0 12 -1<br>7 -3 2 5 |
| 8 0 2 -2<br>-2 4 1 7<br>1 4 3 0     | -2 4 1 7<br>1 4 3 0<br>8 0 2 -2     |

15. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มของอาเรย์  $3 \times 4$  แล้วให้เรียงลำดับข้อมูลแต่ละแถวของอาเรย์จากน้อยไปมากก่อน จึงเรียงลำดับข้อมูลแต่ละแถวโดยอ้างอิงข้อมูลจากคอลัมน์แรกให้เรียงจากน้อยไปมาก และให้ข้อมูลแต่ละแถวขยับตามทั้งแถว

| Input                               | Output                              |
|-------------------------------------|-------------------------------------|
| 1 0 12 -1<br>7 -3 2 5<br>-5 -2 2 -9 | -9 -5 -2 2<br>-3 2 5 7<br>-1 0 1 12 |
| 8 0 2 -2<br>-2 4 1 7<br>1 4 3 0     | -2 4 1 7<br>1 4 3 0<br>8 0 2 -2     |

16. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มของอาเรย์  $3 \times 3$  จงเขียนโปรแกรมตรวจสอบว่าอาเรย์นี้เป็น Magic Square หรือไม่ (ผลบวกของแนวตั้ง แนวนอน และแนวทแยงเท่ากันทั้งหมด)

| Input                   | Output            |
|-------------------------|-------------------|
| 8 1 6<br>3 5 7<br>4 9 2 | Magic Square.     |
| 4 9 2<br>3 5 7<br>8 1 6 | Magic Square.     |
| 7 8 2<br>3 5 4<br>9 1 6 | Not Magic Square. |

17. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มของอาร์เรย์  $3 \times 3$  จงเขียนโปรแกรมตรวจสอบว่าอาร์เรย์นี้เป็นเมทริกซ์สามเหลี่ยมบนหรือไม่ (เมทริกซ์จัตุรัสที่สมาชิกใต้เส้นทแยงมุมหลักเป็นศูนย์ทั้งหมด)

| Input                   | Output                       |
|-------------------------|------------------------------|
| 2 1 4<br>0 6 7<br>0 0 2 | Upper triangular Matrix.     |
| 4 9 2<br>0 0 1<br>0 1 6 | Not Upper triangular Matrix. |
| 7 8 2<br>0 0 4<br>0 0 6 | Upper triangular Matrix.     |

18. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มของอาร์เรย์  $3 \times 3$  จงเขียนโปรแกรมตรวจสอบว่าอาร์เรย์นี้เป็นเมทริกซ์สามเหลี่ยมล่างหรือไม่ (เมทริกซ์จัตุรัสที่สมาชิกบนเส้นทแยงมุมหลักเป็นศูนย์ทั้งหมด)

| Input                   | Output                       |
|-------------------------|------------------------------|
| 1 0 0<br>0 6 7<br>1 0 2 | Not lower triangular Matrix. |
| 2 0 0<br>1 0 0<br>0 1 7 | Lower triangular Matrix.     |
| 5 0 0<br>3 1 0<br>0 0 6 | Lower triangular Matrix.     |

19. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มของอาร์เรย์ A และอาร์เรย์ B เป็นเมทริกซ์ขนาด  $3 \times 3$  จงเขียนโปรแกรมหาค่าเมทริกซ์  $A \times B$

| Input                                                        | Output |
|--------------------------------------------------------------|--------|
| A<br>2 1 4<br>0 6 7<br>0 0 2<br>B<br>8 3 3<br>0 2 9<br>0 0 1 |        |
| A<br>4 9 2<br>5 2 7<br>3 1 6<br>B<br>7 8 9<br>2 6 4<br>4 3 6 |        |

20. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มของอาร์เรย์ A ขนาด  $3 \times 4$  จงเขียนโปรแกรมบวกค่าของเมทริกซ์ทุกตัวเพิ่มอีก 5 โดยจำกัดขอบเขตบนของสมาชิกทุกตัวให้ไม่เกิน 10

| Input   | Output     |
|---------|------------|
| 2 1 4 3 | 7 6 9 8    |
| 0 6 7 5 | 5 10 10 10 |
| 0 0 2 4 | 5 5 7 9    |
| 7 8 2 0 | 10 10 7 5  |
| 3 5 4 6 | 8 10 9 10  |
| 9 1 6 8 | 10 6 10 10 |

21. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มของอาร์เรย์ A ขนาด  $3 \times 3$  ที่มีค่าอยู่ระหว่าง 0 – 10 จงนับจำนวนของค่าตัวเลขแต่ละตัวในอาร์เรย์ 2 มิติแล้วแสดงผล

| Input | Output |
|-------|--------|
| 2 1 4 | 0 3    |
| 0 6 6 | 1 1    |
| 0 0 2 | 2 2    |
|       | 4 1    |
|       | 6 2    |
| 7 8 1 | 1 2    |
| 8 5 7 | 5 1    |
| 1 8 8 | 7 2    |
|       | 8 4    |

22. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มของอาร์เรย์ A ขนาด  $3 \times 3$  ที่มีค่าอยู่ระหว่าง 0 – 255 ค่าใดที่มีค่าน้อยกว่า 128 ให้เปลี่ยนค่าเป็น 0 ถ้าไม่น้อยกว่า 128 ให้เปลี่ยนค่าเป็น 255

| Input       | Output      |
|-------------|-------------|
| 128 150 0   | 255 255 0   |
| 33 50 43    | 0 0 0       |
| 200 255 200 | 255 255 255 |
| 52 10 108   | 0 0 0       |
| 134 5 94    | 255 0 0     |
| 120 215 190 | 0 255 255   |

- การประมวลผลภาพดิจิทัล (ภาพขาว – ดำ)



23. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มของอาเรย์ A ขนาด  $3 \times 3$  ที่มีค่าอยู่ระหว่าง 0 – 255 จงหาส่วนเบี่ยงเบนมาตรฐานของค่าในอาเรย์นี้

| Input                                | Output |
|--------------------------------------|--------|
| 128 150 0<br>33 50 43<br>200 255 200 |        |
| 52 10 108<br>134 5 94<br>120 215 190 |        |

$$\sqrt{\frac{1}{MN} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (I_{ij} - \bar{I})^2}$$

- หาค่าส่วนเบี่ยงเบนมาตรฐาน เพื่อนำไปใช้ในการประมวลผลภาพ

24. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มของอาเรย์ A เป็นเมทริกซ์ขนาด  $5 \times 5$  ที่มีค่าแค่เลข 0 และเลข 1 โดยเลข 0 เป็นพื้นหลัง และเลข 1 เป็นวัตถุ จงหาช่องทั้งหมดที่เป็นเส้นขอบของวัตถุ

| Input                                                         | Output |
|---------------------------------------------------------------|--------|
| 1 1 1 1 1<br>0 1 1 1 1<br>0 0 0 1 1<br>0 0 1 1 1<br>0 0 1 1 0 | 6      |
| 1 1 0 0 0<br>0 1 1 0 0<br>1 0 0 1 1<br>1 0 1 1 0<br>0 1 1 0 0 | 10     |

- การประมวลผลภาพดิจิทัล (การหาเส้นขอบของรูปภาพ)

25. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มของอาเรย์ A เป็นเมทริกซ์ขนาด  $5 \times 5$  เปลี่ยนค่าเลข 2 ทั้งหมดให้เป็นเลข 8

| Input                                                         | Output                                                        |
|---------------------------------------------------------------|---------------------------------------------------------------|
| 1 5 2 2 2<br>8 2 6 2 2<br>0 5 2 2 2<br>2 3 6 2 2<br>2 4 7 7 0 | 1 5 8 8 8<br>8 8 6 8 8<br>0 5 8 8 8<br>8 3 6 8 8<br>8 4 7 7 0 |
| 6 5 2 2 3<br>5 2 6 9 2<br>7 5 2 2 6<br>2 3 6 2 2<br>4 2 2 7 0 | 6 5 8 8 3<br>5 8 6 9 8<br>7 5 8 8 6<br>8 3 6 8 8<br>4 8 8 7 0 |

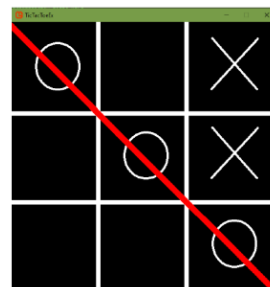
26. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มของอาเรย์ A เป็นเมทริกซ์ขนาด  $5 \times 5$  เริ่มจากตำแหน่งบนขวา จงเปลี่ยนค่าเลข 2 ที่ติดกันในทิศบน ล่าง ซ้าย และขวาทั้งหมดให้เป็นเลข 8

| Input                                                         | Output |
|---------------------------------------------------------------|--------|
| 1 5 2 2 2<br>8 2 6 2 2<br>0 5 2 2 2<br>2 3 6 2 2<br>2 4 7 7 0 |        |
| 6 5 2 2 3<br>5 2 6 9 2<br>7 5 2 2 6<br>2 3 6 2 2<br>4 2 2 7 0 |        |

### โจทย์ฝึกสร้างเกม

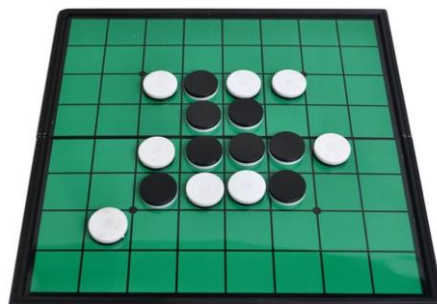
27. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มของอาเรย์ A เป็นเมทริกซ์ขนาด  $3 \times 3$  กำหนดให้สร้างเกม Tic-Tac-Toe (โอเอ็กซ์) โดยให้เลข 0 (ศูนย์) แทนช่องว่าง เลข 1 แทน O (โอ) และ -1 แทน X (เอ็กซ์) เกมจะจบเมื่อฝ่ายใดฝ่ายหนึ่งวางหมากได้เรียงกัน 3 ตัวเป็นเส้นตรง หรือไม่มีช่องว่างเหลือให้ลง

| Input                       | Output    |
|-----------------------------|-----------|
| 1 0 -1<br>0 1 -1<br>0 0 1   | O Winner. |
| -1 1 0<br>0 1 0<br>-1 -1 -1 | X Winner. |



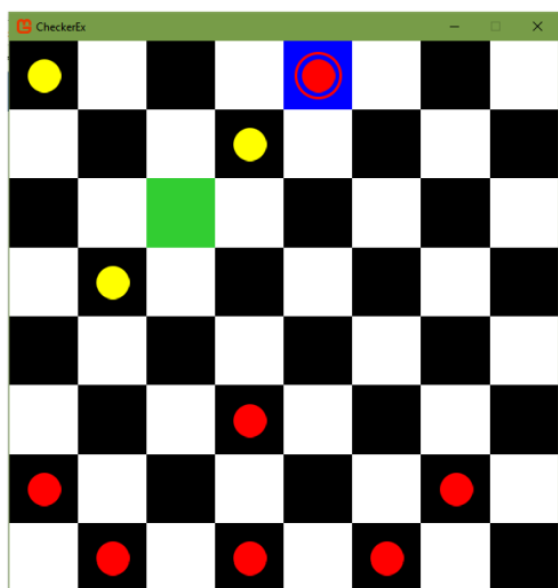
28. กำหนดให้อาเรย์ A เป็นอาเรย์ 2 มิติที่มีขนาด  $8 \times 8$  กำหนดให้สร้างเกม Othello โดยให้เลข 0 (ศูนย์) แทนช่องว่าง เลข 1 แทน หมากสีดำ และ -1 แทน หมากสีขาว เกมจะจบเมื่อไม่มีช่องว่างเหลือให้ลง และฝ่ายที่มีจำนวนหมากมากกว่าจะชนะ

GoodEasy





## 29. หมากฮอส...???



โจทย์ปัญหาโดยใช้ตัวแปรอาร์เรย์ 3 มิติ

30. ในการสอบของนักเรียน จำนวน 10 คน มีคะแนนสอบ 3 วิชา ดังตาราง จงเขียนโปรแกรมแสดงผลนักเรียนคน  
ที่ได้คะแนนรวมทั้ง 3 วิชาสูงสุด และได้คะแนนรวมสูงที่สุดเท่าใด ทั้งนี้สมมติให้ผู้ที่ทำคะแนนรวมสูงสุดมีเพียงคนเดียวเท่านั้น

| นักเรียนคนที่ | คะแนนสอบ              |                        |                       |
|---------------|-----------------------|------------------------|-----------------------|
|               | คณิตศาสตร์ (50 คะแนน) | วิทยาศาสตร์ (50 คะแนน) | ภาษาอังกฤษ (50 คะแนน) |
| 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                    |

31. จากข้อมูลในตารางต่อไปนี้ จงเขียนโปรแกรมเพื่อกรอกคะแนนของนักเรียน จำนวน 5 คน จำนวน 3 รายวิชา แต่ละวิชามีคะแนนการสอบ 2 ครั้ง และให้หาคะแนนเฉลี่ยของการสอบแต่ละครั้ง ทั้ง 3 รายวิชา แล้วแสดงรายงานทางจอภาพดังนี้ (เส้นตารางไม่ต้องแสดง)

| นักเรียนคนที่ | คะแนนสอบ   |            |             |            |            |            |
|---------------|------------|------------|-------------|------------|------------|------------|
|               | คณิตศาสตร์ |            | วิทยาศาสตร์ |            | ภาษาอังกฤษ |            |
|               | 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      |

โจทย์ปัญหาโดยใช้ตัวแปรข้อมูลชนิดสตริง

Coming Soon.