

อาร์เรย์หลายมิติ (Multi Dimension Array)

Principle of Programming

บุญชู จิตนุพงศ์

วิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ

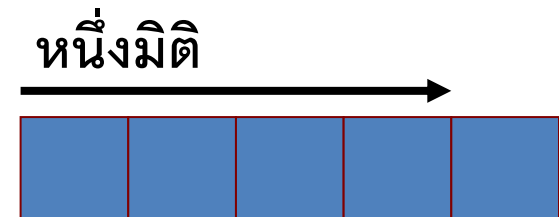
คณะทรัพยากรธรรมชาติและสิ่งแวดล้อม ม.เกษตรศาสตร์ วิทยาเขตศรีราชา

Agenda

- อาร์เรย์?
- อาร์เรย์หลายมิติ (Multi-Dimension Array)?
- อาร์เรย์ **2** มิติ
- การอ้างอิงถึงสมาชิกในอาร์เรย์

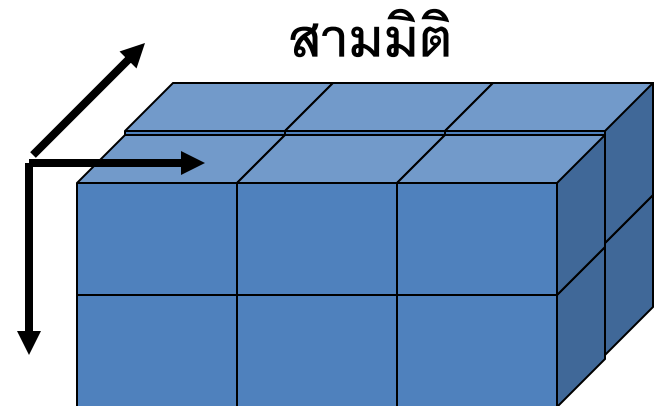
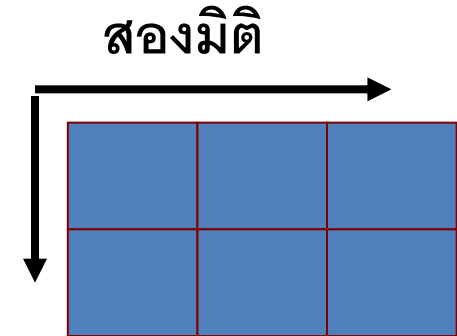
อาร์เรย์ (Revise)

- อาร์เรย์ คือ ชุดของตัวแปรที่มีชื่อตัวแปร และชนิดตัวแปรเดียวกัน มักใช้กับตัวแปรชนิดเดียวกันหลายๆ ตัว ที่มีการทำงานเหมือนกัน เข้ามาแก้ไขปัญหาของการเขียนโปรแกรมที่มีความจำเป็นต้องใช้ตัวแปรประเภทเดียวกันจำนวนมากได้ โดยข้อมูลแต่ละตัวของอาร์เรย์จะเรียกว่า อีลีเมนต์ (**Element**) และข้อมูลแต่ละอีลีเมนต์จะมีหมายเลขเพื่อใช้ในการอ้างอิงถึงเรียกตัวเลขนี้ว่า เลขดัชนี (**Index**)
- อาร์เรย์ **1 มิติ** คือ ตัวแปรชุดของตัวแปรประเภทพื้นฐาน



อาร์เรย์หลายมิติ (Multi-Dimension Array)

- ก็คือ “ตัวแปรชุดของตัวแปรชุด”
- อาร์เรย์ **2** มิติ ก็คือ “อาร์เรย์ของอาร์เรย์”
 - เข้าถึงสมาชิกด้วย ตัวเลขดัชนี (index) 2 ตัว
- อาร์เรย์ **3** มิติ ก็คือ “อาร์เรย์ของอาร์เรย์ของอาร์เรย์”
 - เข้าถึงสมาชิกด้วย ตัวเลขดัชนี (index) 3 ตัว



อาร์เรย์ 2 มิติ

- อาร์เรย์ 2 มิติ เป็นตัวแปรชุดที่มีการจัดการข้อมูล **Row** (แถว) , **Column** (หลัก) ซึ่งอยู่ในรูปแบบตาราง ที่มีแสดงตำแหน่ง 2 ตัว
- อาร์เรย์ 2 มิติ คือ **array of array** กล่าวคือ **array 2 มิติ** เป็น **array** ของ **array 1 มิติ** นั่นเอง

มิติที่ 1 (แถว)

	index	0	1	2	3	4
มิติที่ 2 (หลัก)	0					
	1					

การประกาศตัวแปรอาร์เรย์ 2 มิติ

- แบบที่ 1 แบบระบุขนาดไม่กำหนดค่าเริ่มต้น

```
data_type array_name[row_size][column_size];
```

- ตัวอย่าง

```
int score[2][10];  
char id[2][10];
```

สร้างตัวแปรที่มี 2 แถว 10 หลัก
สำหรับเก็บตัวเลขจำนวนเต็ม

การประกาศตัวแปรอาร์เรย์ 2 มิติ (ต่อ)

- แบบที่ 2 แบบระบุขนาดและกำหนดค่าเริ่มต้น

```
data_type array_name[row][col] = {  
    {value_11, value_12, ... , value_1col},  
    {value_21, value_22, ... , value_2col},  
    .  
    .  
    .  
    {value_row1, value_row2, ... , value_rowcol}  
};
```

- ตัวอย่าง

```
int score[2][3] = {{1, 2, 3},{4,5,6}};
```

การประกาศตัวแปรอาร์เรย์ 2 มิติ (ต่อ)

- แบบที่ 3 ไม่ระบุขนาดแต่กำหนดค่าเริ่มต้น

```
data_type array_name[][col] = {  
    {value_11, value_12, ... , value_1col},  
    {value_21, value_22, ... , value_2col},  
    .  
    .  
    .  
    {value_row1, value_row2, ... , value_rowcol}  
};
```

- ตัวอย่าง

```
int score[][3] = {{1, 2, 3},{4,5,6}};
```


การประกาศตัวแปรอาร์เรย์ 2 มิติ (ต่อ)

- `int score[2][3] = {11, 12, 13, 21, 22, 23};`
- `int score[2][3] = {{11, 12, 13}, {21, 22, 23}};`
- `int score[][3] = {{11, 12, 13}, {21, 22, 23}};`
- การประกาศและกำหนดค่าเริ่มต้นตัวแปรทั้ง 3 แบบนี้ ให้ผลลัพธ์เหมือนกัน คือ

score

0	11	12	13
1	21	22	23
	0	1	2

- หมายเหตุ อาร์เรย์ขนาด 2 มิติขึ้นไป จะไม่ระบุขนาดได้เฉพาะส่วนมิติที่ 1 เท่านั้น ส่วนมิติอื่นๆ ต้องมีการระบุขนาดด้วยทุกครั้ง

การอ้างอิงถึงสมาชิกในอาร์เรย์

- อ้างอิงถึง คือ เข้าถึงเพื่อกำหนดค่าให้กับสมาชิก หรือเข้าถึงเพื่ออ่านค่าของสมาชิก
- รูปแบบ

```
array_name[row_index][column_index]
```

- index มีชนิดข้อมูลเป็นจำนวนเต็ม
- index ตัวแรกมีค่าเป็น 0 เสมอ
- index อาจอยู่ในรูปค่าคงที่, ตัวแปร หรือ นิพจน์ที่ให้ค่าเป็นจำนวนเต็มก็ได้
- หากอาร์เรย์มีขนาด N ในแต่ละมิติ (row และ column) จะมี index ตัวแรกคือ 0 และ index ตัวสุดท้ายคือ N-1 (ในแต่ละมิติ)

การอ้างอิงถึงสมาชิกในอาร์เรย์ (ต่อ)

- ตัวอย่าง

```
int a[2][5] = {{2,5,1,6,8},{7,9,3,4,0}};  
int i = 4;  
printf("%d %d %d",a[0][0],a[i][2],a[1][i-3]);
```

		a				
0	1	2	5	1	6	8
7		9	3	4	0	
		0	1	2	3	4

ผลลัพธ์

2 8 9

a[0][0]	// อ้างถึงสมาชิกตัวแรก (index ที่ 0,0)
a[0][i]	// อ้างถึงสมาชิกตัวที่ 5 (index ที่ 0,4)
a[1][i-3]	// อ้างถึงสมาชิกตัวที่ 7 (index ที่ 1,1)

การอ้างอิงถึงสมาชิกในอาร์เรย์ (ต่อ)

```
int  row, col, x=11, score[2][5];  
  
for(row=0; row<2; row++) {  
    for(col=0; col<5; col++) {  
        score[row][col] = x+col;  
    }  
    x=21;  
}
```

score					
0	11	12	13	14	15
1	21	22	23	24	25
	0	1	2	3	4

การอ้างอิงถึงสมาชิกในอาร์เรย์ (ต่อ)

- คำสั่งวนซ้ำเพื่อแสดงค่าที่เก็บอยู่ในสมาชิกแต่ละตัวของอาร์เรย์

```
for (row=0; row<2; row++) {  
    for (col=0; col<5; col++) {  
        printf("%d, ", score[row][col]);  
    }  
    printf("\n");  
}
```

ตัวอย่างการประยุกต์ใช้อาร์เรย์ 2 มิติ

- เขียนโปรแกรมเพื่อรับค่าข้อมูลการวัดอุณหภูมิร่างกายของคนไข้รายหนึ่งเป็นเวลา **7** วัน โดยแต่ละวันจะทำการวัดอุณหภูมิ **4** ครั้งในเวลาที่ตรงกันคือ **6.00 น. / 12.00 น. / 18.00 น. / 24.00 น.**
- ให้แสดงผลลัพธ์การคำนวณอุณหภูมิเฉลี่ยที่วัดได้ในแต่ละวันทั้ง **7** วัน และอุณหภูมิเฉลี่ยโดยรวมทั้งหมด

ตัวอย่างการประยุกต์ใช้อาร์เรย์ 2 มิติ (ต่อ)

- การวิเคราะห์
 - ต้องรับค่า **(Input)** อุณหภูมิทั้งหมด 7 วัน วันละ 4 เวลา
 - ใช้ array 2 มิติ ขนาด 7x4
 - Temp[7][4]

	0	1	2	3	เวลา
0					
1					
2					
3					
4					
5					
6					
วัน					

ตัวอย่างการประยุกต์ใช้อาร์เรย์ 2 มิติ (ต่อ)

- Process

- รับค่าอุณหภูมิ
- คำนวณอุณหภูมิเฉลี่ยในแต่ละวัน
- คำนวณอุณหภูมิเฉลี่ยรวม

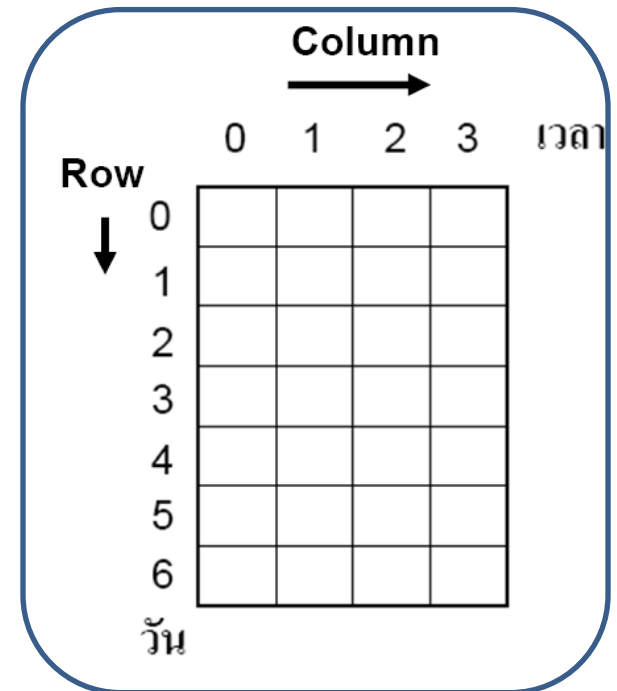
- Output

- แสดงผลค่าเฉลี่ยแต่ละวัน
- แสดงผลค่าเฉลี่ยรวม



ตัวอย่างการประยุกต์ใช้อาร์เรย์ 2 มิติ (ต่อ)

- ขั้นตอนการรับค่า ใช้ `for` ซ้อน `for`
 - `for` ชั้นแรกวนตั้งแต่วันที่ **1** ถึง **7** (กำหนดจำนวนวัน)
 - `for` ชั้นที่สองวนตั้งแต่วันที่ **1** ถึง **4** ในแต่ละวัน (กำหนดจำนวนช่วงเวลา)



ตัวอย่างการประยุกต์ใช้อาร์เรย์ 2 มิติ (ต่อ)

- หาค่าเฉลี่ยในแต่ละวัน
 - ในแต่ละ **Row** ซึ่งแทนข้อมูลของหนึ่งวันให้นำอุณหภูมิทุก **Column** มารวมกัน แล้วหารด้วย **4** จะได้อุณหภูมิเฉลี่ยของวันนั้นๆ
 - ทำเช่นเดียวกันทุกๆ **Row** จะได้ครบทั้ง **7** วัน
- หาค่าเฉลี่ยทั้ง **7** วัน
 - นำข้อมูลทุก **Row** และทุก **Column** มารวมกัน
 - นำไปหารด้วย **28** จะได้อุณหภูมิเฉลี่ยรวม หรือ
 - นำข้อมูลค่าอุณหภูมิเฉลี่ยในแต่ละวันมารวมกัน
 - นำไปหารด้วย **7** จะได้อุณหภูมิเฉลี่ยรวม

```

# include <stdio.h>
void main ( ) {
    int i,j;
    float temp[7][4];
    float sumDay, ansDay;
    float sumAll=0, ansAll;
    for (i = 0; i < 7; i++){          // i ควบคุม row
        sumDay=0;
        for(j=0;j<4;j++){            // j ควบคุม column
            printf("Day %d period %d Temp=:",i+1,j+1);
            scanf("%f",&data[i][j]);
            sumDay=sumDay+temp[i][j];
        }
        ansDay=sumDay/4.0;
        printf("Average of day %d is %.2f",i+1,ansDay);
        sumAll=sumAll+sumDay;
        printf("=====\n");
    }
    ansAll=sumAll/28.0;
    printf("Average All is %f",sumAll);
}

```

อาร์เรย์ 3 มิติ

```
int  a[2][2][3] = {  
    { {1, 2, 3} , {4, 5, 6} },  
    { {7, 8, 9} , {10, 11, 12} }  
};
```

a[0][0][0] = 1	a[0][0][1] = 2	a[0][0][2] = 3
a[0][1][0] = 4	a[0][1][1] = 5	a[0][1][2] = 6
a[1][0][0] = 7	a[1][0][1] = 8	a[1][0][2] = 9
a[1][1][0] = 10	a[1][1][1] = 11	a[1][1][2] = 12