

แผนการสอนประจำบทที่ 6

อาร์เรย์

หัวข้อสำคัญ

1. ความหมายของอาร์เรย์
2. ชนิดของอาร์เรย์
3. การประกาศตัวแปรอาร์เรย์
4. การอ้างอิงค่าจากอาร์เรย์
5. อาร์เรย์และฟังก์ชัน

วัตถุประสงค์เชิงพฤติกรรม

1. ผู้เรียนสามารถอธิบายความหมายของอาร์เรย์ได้
2. ผู้เรียนสามารถบอกชนิดของอาร์เรย์ได้
3. ผู้เรียนสามารถเขียนโปรแกรมประกาศตัวแปรอาร์เรย์และการอ้างอิงค่าจากอาร์เรย์ได้
4. ผู้เรียนสามารถเขียนโปรแกรมอาร์เรย์โดยการประยุกต์ใช้กับฟังก์ชันได้

วิธีการสอนและกิจกรรมการเรียนการสอน

1. การบรรยาย
2. การทำแบบฝึกหัด

สื่อที่ใช้ประกอบการสอน

1. เอกสารประกอบการสอน
2. เครื่องคอมพิวเตอร์
3. เครื่องฉายภาพนิ่ง

การวัดและประเมินผล

1. สังเกตจากความสนใจของผู้เรียน
2. ประเมินจากการตอบคำถามของผู้เรียนและกิจกรรมในชั้นเรียน
3. การทำแบบฝึกหัดท้ายบท

บทที่ 6

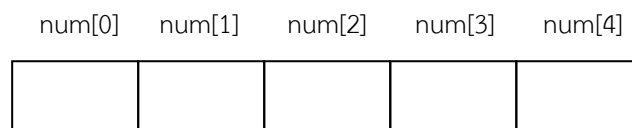
อาร์เรย์

6.1 ความหมายของอาร์เรย์

แถวลำดับ (array) คือ ข้อมูลที่ประกอบไปด้วยรายการชนิดข้อมูลเดียวกันจำนวนหนึ่ง สามารถกำหนดขนาดของอาร์เรย์ได้แต่ละรายการเรียกว่า เซลล์ (cell) และแต่ละเซลล์จะมี อินเด็กซ์ (index) สำหรับใช้อ้างอิงข้อมูล

ปัญหา : ถ้าเราต้องการเก็บค่าคะแนนนักเรียน 5 คน และค่าเฉลี่ยคะแนน

จากปัญหาดังกล่าวจึงเกิดข้อมูลประเภท แถวลำดับ (array) แถวลำดับเป็นโครงสร้างทางคอมพิวเตอร์ที่พบได้มากที่สุด โดยภาพจำลองอาร์เรย์ชื่อ num ขนาดเท่ากับ 5 แสดงดังภาพ 6.1



ภาพ 6.1 ภาพจำลองอาร์เรย์ที่มีขนาดเท่ากับ 5

6.2. ชนิดและการประกาศตัวแปรของอาร์เรย์

แบ่งได้เป็น 2 ชนิด ได้แก่

- 1) อาร์เรย์ชนิด 1 มิติ (One-Dimensional)
- 2) อาร์เรย์ชนิดหลายมิติ (Multi-Dimensional)

6.2.1 อาร์เรย์ชนิด 1 มิติ (One-Dimensional)

- 1) การประกาศตัวแปรอาร์เรย์ชนิด 1 มิติ มี 2 แบบ ดังต่อไปนี้

ชนิดข้อมูล หมายถึง ชนิดข้อมูลที่อ้างอิงตามหลักเกณฑ์ของโครงสร้างภาษา C เช่น int, float, char

ชื่อตัวแปรอาร์เรย์ หมายถึง ชื่อของตัวแปรอาร์เรย์

n หมายถึง ขนาดของตัวแปรที่ต้องการสร้าง

1.1) การประกาศตัวแปรอาร์เรย์ชนิด 1 มิติ โดยกำหนดขนาดให้กับอาร์เรย์

รูปแบบการประกาศ ตัวแปรอาร์เรย์ 1 มิติ แบบที่ 1	ตัวอย่าง
ชนิดข้อมูล ชื่อตัวแปรอาร์เรย์[n];	<pre>#define SIZE 20 char score[10*10]; int num[SIZE+1]; float student [5]; student [0], student [1], student [2], student [3], student [4]</pre>

1.2) การประกาศตัวแปรอาร์เรย์ชนิด 1 มิติ พร้อมกับการกำหนดค่าเริ่มต้น

รูปแบบการประกาศ ตัวแปรอาร์เรย์ 1 มิติ แบบที่ 2 พร้อมกับการกำหนดค่าเริ่มต้น
ชนิดข้อมูล ชื่อตัวแปรอาร์เรย์[n] = {ข้อมูลที่1, ข้อมูลที่2, ..., ข้อมูลที่ n-1};
ตัวอย่าง
<pre>int number[5] = {0,1,2,3,4}; char grade[3] = {'A','B','C'} float student [5]={50,55.50,60,65.25,70};</pre>

ข้อควรระวังในการประกาศตัวแปร Array

1) การประกาศตัวแปรอาร์เรย์แล้ว ไม่มีกำหนดค่าเริ่มต้นให้กับตัวแปรนั้นแล้ว ค่าที่อยู่ในตัวแปรจะเป็นค่าที่ค้างอยู่ในหน่วยความจำ

2) การประกาศตัวแปรอาร์เรย์และกำหนดค่าเริ่มต้นตั้งแต่ตอนประกาศตัวแปรแล้ว แต่กำหนดค่าไม่ครบ ในกรณีที่ชนิดข้อมูลของอาร์เรย์ตัวเลขทั้งจำนวนเต็มและจำนวนจริง ค่าที่เหลือจะถูกกำหนดเป็น 0 โดยอัตโนมัติ เช่น float price[5] = {50.5,2.25,10.0} ;

3) การประกาศตัวแปรอาร์เรย์และกำหนดค่าเริ่มต้นตั้งแต่ตอนประกาศตัวแปรแล้ว ไม่จำเป็นต้องใส่ขนาดของอาร์เรย์ก็ได้ เช่น float value[] = {1,2,3,4,5} ; หมายความว่า เป็นการกำหนดตัวแปรอาร์เรย์ของจำนวนจริงแบบ float ขนาด 5 จำนวน

หมายเหตุ การประกาศตัวแปรอาร์เรย์โดยไม่ใส่ขนาดของอาร์เรย์ไม่สามารถทำได้ ยกเว้นมีการกำหนดค่าเริ่มต้นให้กับตัวแปรอาร์เรย์เท่านั้น

4) การประกาศตัวแปรอาร์เรย์โดยไม่กำหนดขนาดของอาร์เรย์ และไม่กำหนดค่าเริ่มต้นให้กับตัวแปรอาร์เรย์ เช่น `int value[]`; ถือว่าเป็นประกาศผิด!!!

ตัวอย่างการประกาศตัวแปรอาร์เรย์ชนิด 1 มิติ เพื่อเก็บค่าคะแนนนิสิตจำนวน 5 คน

ตัวชี้ (index)	ตัวแปร score
0	สมาชิก <code>score[0]</code> = 90.00
1	สมาชิก <code>score[1]</code> = 80.00
2	สมาชิก <code>score[2]</code> = 60.00
3	สมาชิก <code>score[3]</code> = 50.00
4	สมาชิก <code>score[4]</code> = 40.00

ภาพ 6.2 ภาพจำลองการเก็บข้อมูลของอาร์เรย์

ประกาศตัวแปรอาร์เรย์ชนิด 1 มิติ ชื่อ `score` ให้มีชนิดข้อมูลเป็น `float` และจำนวนสมาชิก 5 รายการ ดังภาพ 6.2 โดยลำดับของสมาชิกเรียกว่า `index` เป็นตัวเลขที่อยู่ในเครื่องหมาย `[]` ประกอบด้วยตัวชี้ (`index`) 0, 1, 2, 3, และ 4 ตามลำดับ จากข้อมูลข้างต้นประกาศตัวแปรอาร์เรย์ดังกล่าวได้ดังนี้

```
float score[5];
```

2) การอ้างถึงแต่ละรายการสมาชิก

การประกาศขนาดของอาร์เรย์ จะประกอบด้วยรายการสมาชิกที่มีลำดับเป็นตัวกำหนด ลำดับดังกล่าวจะเรียกว่า `index` เป็นตัวเลขที่อยู่ในเครื่องหมาย `[]` และจะเริ่มต้นด้วย `index = 0` เสมอ เมื่อต้องการอ้างอิงรายการสมาชิกลำดับที่เท่าไร จะใช้งานโดย ชื่อตัวแปรอาร์เรย์ ตามด้วยเครื่องหมาย `[]` ที่ด้านในระบุลำดับของรายการสมาชิกที่ต้องการอ้างอิง เช่น `score[4]`, `number[3]`

ตัวอย่างการอ้างถึงแต่ละรายการสมาชิก

กำหนดค่าให้นักเรียนคนที่ 20 ได้คะแนน 75 ทำได้ดังนี้

```
score [19] = 75;
```

ตัวอย่างเพิ่มเติม

<code>student [1]</code>	หมายถึง	<code>student [1]</code>
<code>student [1+1]</code>	หมายถึง	<code>student [2]</code>
<code>student [4*4]</code>	หมายถึง	<code>student [16]</code>

3) การเขียนและรับข้อมูลแต่ละเซลล์

การแสดงค่าคะแนนของนักเรียนคนที่ 20 ให้มีเลขทศนิยม 2 ตำแหน่ง ทำได้ดังนี้

```
printf("%.2f",student[19]);
```

การรับค่าคะแนนของนักเรียนคนที่ 20 จากแป้นพิมพ์ทำได้ดังนี้

```
scanf("%f",&student[19]);
```

4) การใช้ loop กับ array

การใช้คำสั่งวนซ้ำ หรือ loop จะช่วยลดความยุ่งยากสำหรับการรับค่า แสดงค่า และกำหนดค่าตัวแปรประเภทอาร์เรย์

ตัวอย่างการใช้คำสั่งวนซ้ำกับอาร์เรย์ 1 มิติ

กำหนดให้อาร์เรย์ชื่อ score เป็นอาร์เรย์ 1 มิติ ขนาดเท่ากับ 5 มีข้อมูลดังต่อไปนี้ แสดงดัง

ภาพ 6.3

ตัวชี้ (index)	ตัวแปร	ค่าคะแนนที่รับเข้า
0	score[0]	90.00
1	score[1]	80.00
2	score[2]	70.00
3	score[3]	60.00
4	score[4]	50.00

ภาพ 6.3 ข้อมูลของอาร์เรย์ score

คำสั่งวนซ้ำ for เหมาะที่ใช้ในการใช้งานกับอาร์เรย์ เนื่องจากทราบจำนวนสมาชิกที่ต้องการอ้างอิง โดยคำสั่ง for ด้านล่างจะกำหนดให้ตัวแปร i เป็นตัวแปรที่ใช้วนรับค่าของอาร์เรย์ score ซึ่งค่า i เริ่มต้นที่ 0-4 ประกอบด้วยค่า 0,1,2,3 และ 4 ตามลำดับ จำนวน 5 รายการ

```
for(i=0;i<=4,i++)
    scanf("%f",&score[i]);
```

ตัวอย่างการรับค่าตัวแปรอาร์เรย์โดยใช้ for

โปรแกรม	ผลลัพธ์
<pre> 1. #include<stdio.h> 2. float student[5], sum = 0, avg = 0; int i; 3. main() 4. { 5. for(i=0;i<=4;i++){ 6. printf("Enter score of student %d : ",i+1); 7. scanf("%f",&student[i]); 8. sum = sum + student[i]; 9. } 10. avg = sum/5; 11. printf("The average of student score = %.2f",avg); 12. }</pre>	Input ตามภาพ 6.3

การใช้ตัวแปรอาร์เรย์ในการรับ-แสดงค่า โดยใช้ for

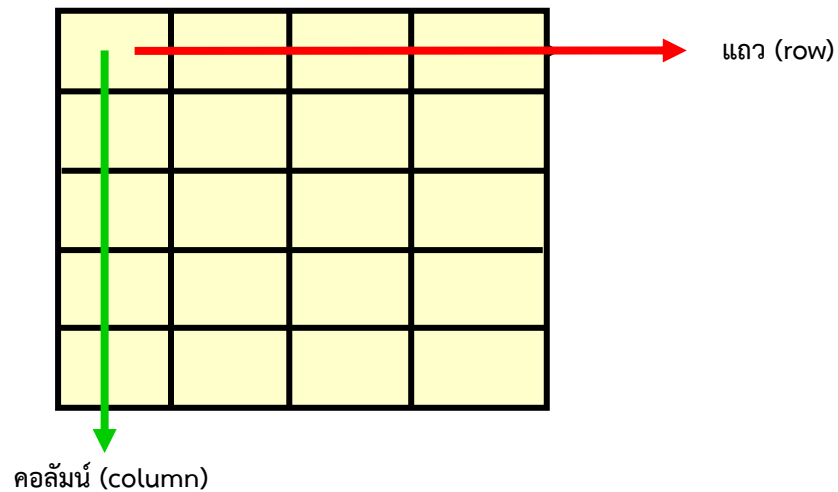
โปรแกรม	ผลลัพธ์
<pre> 1. #include<stdio.h> 2. float student[5]; int i; 3. main() 4. { 5. for(i=0;i<=4;i++){ 6. printf("Enter score of student %d : ",i+1); 7. scanf("%f",&student[i]); 8. printf("score[%d+1] = %.2f\n",i,student[i]); 9. } 10. }</pre>	

6.2.2 อาร์เรย์ชนิดหลายมิติ (Multi-Dimensional)

1) อาร์เรย์ 2 มิติ (2-dimensional Arrays)

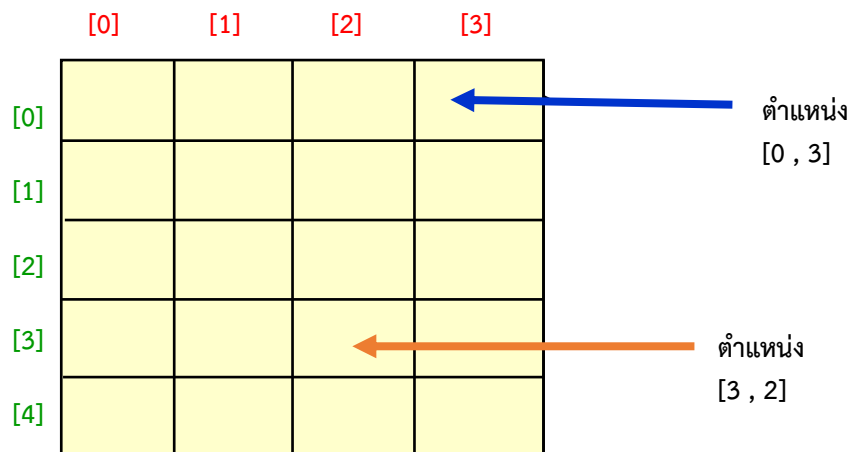
ข้อมูลในอาร์เรย์ 2 มิติจะเป็นลักษณะของตาราง ประกอบด้วยแถวและคอลัมน์ แสดงดังภาพ

6.4



ภาพ 6.4 ลักษณะการอ้างอิงถึงข้อมูลอาร์เรย์ 2 มิติ

การอ้างอิงถึงข้อมูลอาร์เรย์ 2 มิติ แสดงดังภาพ 6.5-6.7



ภาพ 6.5 ตัวอย่างการอ้างอิงถึงข้อมูลอาร์เรย์ 2 มิติ ขนาด 4 x 3

	col [0]	col [1]
row [0]	A[0,0]	A[0,1]
row [1]	A[1,0]	A[1,1]
row [2]	A[2,0]	A[2,1]

ภาพ 6.6 การอ้างอิงถึงข้อมูลอาร์เรย์ 2 มิติ

ขนาด 3x2

	col [0]	col [1]	col [2]
row [0]	A[0,0]	A[0,1]	A[0,2]
row [1]	A[1,0]	A[1,1]	A[1,2]
row [2]	A[2,0]	A[2,1]	A[2,2]
row [3]	A[3,0]	A[3,1]	A[3,2]

ภาพ 6.7 การอ้างอิงถึงข้อมูลอาร์เรย์ 2 มิติ

ขนาด 4x3

ตัวอย่างการประยุกต์ใช้งานตัวแปรอาร์เรย์ 2 มิติ

การเก็บข้อมูลคะแนนเก็บ 4 ครั้ง ครั้งละ 10 คะแนนของนิสิต 3 คน แสดงดังภาพ 6.8

	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4
นาย A	5.6	8.5	2.5	4.5
นาย B	6.0	7.25	5.0	5.0
น.ส. C	6.0	7.2	8.0	4.0

ภาพ 6.8 ตัวอย่างการเก็บข้อมูลของอาร์เรย์ 2 มิติ

2) การประกาศตัวแปรอาร์เรย์ 2 มิติ

ชนิดข้อมูล หมายถึง ชนิดข้อมูลที่อ้างอิงตามหลักเกณฑ์ของโครงสร้างภาษา C เช่น int, float, char

ชื่อตัวแปรอาร์เรย์ หมายถึง ชื่อของตัวแปรอาร์เรย์

m หมายถึง ขนาดของแถว (row) ที่ต้องการสร้าง

n หมายถึง ขนาดของคอลัมน์ (column) ที่ต้องการสร้าง

รูปแบบการประกาศ ตัวแปรอาร์เรย์ 2 มิติ
ชนิดข้อมูล ชื่อตัวแปรอาร์เรย์ [row _m] [column _n];
ตัวอย่าง
<pre> #define SIZE_M 20 #define SIZE_N 20 char score[10][5]; float value[2*2][1*3]; int num[SIZE_M][SIZE_N]; int student[3][4]; /* มีจำนวนรายการเซลล์ของอาร์เรย์ ดังต่อไปนี้ student[0][0], student[0][1], student[0][2], student[0][3] student[1][0], student[1][1], student[1][2], student[1][3] student[2][0], student[2][1], student[2][2], student[2][3] */ </pre>

3) การกำหนดค่าข้อมูลเริ่มต้นให้กับตัวแปร Array 2 มิติ

เป็นการกำหนดค่าข้อมูลเริ่มต้นให้กับตัวแปร array เช่น `int number[2][3] = {{1,2,3},{4,5,6}};` เป็นคำสั่งกำหนดค่าในอาร์เรย์จำนวน 2 แถว 3 คอลัมน์ ได้ดังภาพ 6.9

	col [0]	col [1]	col [2]
row [0]	1	2	3
row [1]	4	5	6

ภาพ 6.9 : ตัวอย่างการกำหนดค่าในอาร์เรย์จำนวน 2 แถว 3 คอลัมน์

`int number[3][2]={1,2},{3,4},{5,6}};` เป็นคำสั่งกำหนดค่าในอาร์เรย์จำนวน 3 แถว 2 คอลัมน์ได้ดังภาพ 6.10

	col [0]	col [2]
row [0]	1	2
row [1]	3	4
row [2]	5	6

ภาพ 6.10 : ตัวอย่างการกำหนดค่าในอาร์เรย์จำนวน 3 แถว 2 คอลัมน์

`int number[][5]= {{1, 2, 3, 4, 5}, {2, 4, 6, 8, 10}, {1, 3, 5, 7, 9}};` การกำหนดค่าเริ่มต้นจะทำพร้อมกับการประกาศ Array เท่านั้น โดยไม่จำเป็นต้องกำหนดขนาดจำนวนแถวของ array ก็ได้ เนื่องจากเมื่อมีการกำหนดค่าเริ่มต้น จะมีการจองพื้นที่ Array ให้เท่ากับข้อมูลที่ใช้ในการกำหนดค่าเริ่มต้นนั้น จากตัวอย่าง ในที่นี้จะจอง array 2 มิติขนาด 3 แถว 5 คอลัมน์ เครื่องหมาย { } จะเป็นส่วนของการกำหนดค่าในแต่ละแถว ดังรูปด้านล่าง

	col [0]	col [1]	col [2]	col [3]	col [4]
row [0]	1	2	3	4	5
row [1]	2	4	6	8	10
row [2]	1	3	5	7	9

ภาพ 6.11 : ตัวอย่างการกำหนดค่าในอาร์เรย์จำนวน 3 แถว 5 คอลัมน์

ตัวอย่างการเขียนโปรแกรมแบบอาร์เรย์ 2 มิติ โดยไม่มีการกำหนดค่าเริ่มต้น

โปรแกรม	ผลลัพธ์
<pre> 1. #include<stdio.h> 2. main() 3. { 4. int number[3][4]; 5. int row,column; 6. for (row=0; row<=2;row++) 7. { 8. for (column=0; column<=3; column++){ 9. number[row][column] = 0; 10. printf("%d\t",number[row][column]); 11. } 12. printf("\n"); 13. } 14. }</pre>	

ตัวอย่างการเขียนโปรแกรมแบบอาร์เรย์ 2 มิติ โดยมีการกำหนดค่าเริ่มต้น

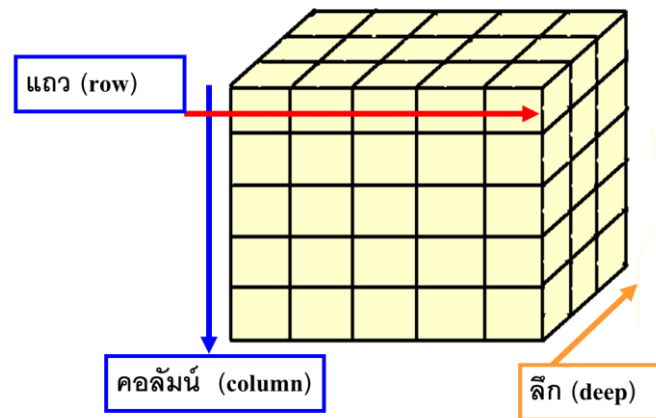
โปรแกรม	ผลลัพธ์
<pre> 1. #include<stdio.h> 2. main() 3. { 4. int number[3][4] = { {2,4,6,8},{1,3,5,7},{1,2,3,4} }; 5. int row,column; 6. for (row=0; row<=2;row++) 7. { 8. for (column=0; column<=3;column ++){ 9. { 10. printf ("%d\t",number[row][column]); 11. } 12. printf ("\n"); 13. } 14. }</pre>	

4) อาร์เรย์ 3 มิติ (3-dimensional Arrays)

ข้อมูลในอาร์เรย์ 3 มิติจะมีลักษณะเป็นลูกบาศก์ประกอบด้วยแถว คอลัมน์ และสูง แสดงดัง

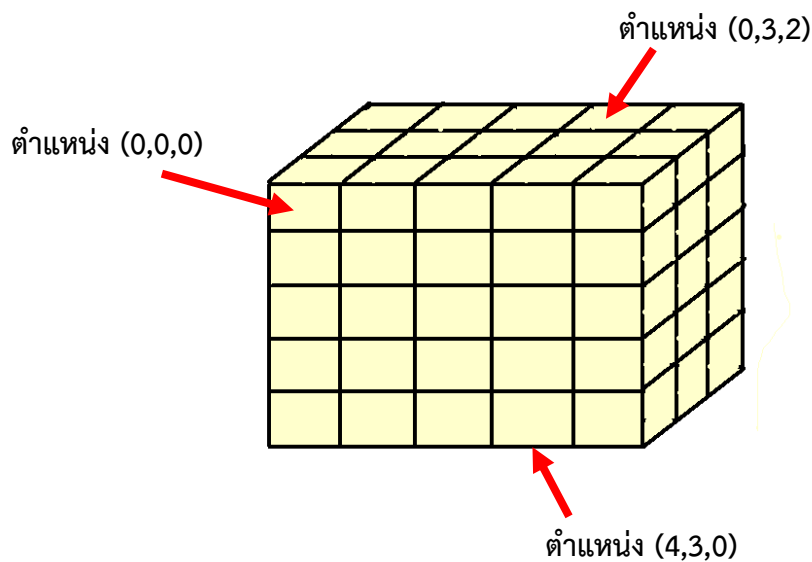
ภาพ 6.12

การอ้างอิงข้อมูลในอาร์เรย์ 3 มิติ



ภาพ 6.12 ภาพแบบจำลองอาร์เรย์ 3 มิติ

การอ้างอิงข้อมูลในอาร์เรย์ 3 มิติ จะอ้างอิงคล้ายกับอาร์เรย์ 2 มิติ เพียงเพิ่มมาอีก 1 ค่าคือ ความลึก ซึ่งจะได้ลักษณะเหมือนพิกัดที่อ้างอิง 1 ตำแหน่ง (แถว, คอลัมน์, ลึก) ภาพจำลองอาร์เรย์ 3 มิติขนาด 5x5x3 แสดงดังภาพ 6.13



ภาพ 6.13 ภาพจำลองอาร์เรย์ 3 มิติขนาด 5x5x3

การประกาศตัวแปรอาร์เรย์ 3 มิติ

ชนิดข้อมูล หมายถึง ชนิดข้อมูลที่อ้างอิงตามหลักเกณฑ์ของโครงสร้างภาษา C เช่น int, float, char

ชื่อตัวแปรอาร์เรย์ หมายถึง ชื่อของตัวแปรอาร์เรย์

m หมายถึง ขนาดของแถว (Row) ที่ต้องการสร้าง

n หมายถึง ขนาดของคอลัมน์ (Column) ที่ต้องการสร้าง

o หมายถึง ขนาดของความลึก (Deep) ที่ต้องการสร้าง

รูปแบบการประกาศ ตัวแปรอาร์เรย์ 3 มิติ
ชนิดข้อมูล ชื่อตัวแปรอาร์เรย์ [row _m] [column _n][deep _o];
ตัวอย่าง
<pre>#define SIZE_row 20 #define SIZE_col 10 #define SIZE_deep 5 char score[10][5][2]; float value[2*2][1*3][2*2]; int num[SIZE_row][SIZE_col] [SIZE_deep];</pre>

5) อาร์เรย์หลายมิติ (Multi-dimensional Arrays)

การประกาศตัวแปรอาร์เรย์หลายมิติ

ชนิดข้อมูล หมายถึง ชนิดข้อมูลที่อ้างอิงตามหลักเกณฑ์ของโครงสร้างภาษา C เช่น int, float, char

ชื่อตัวแปรอาร์เรย์ หมายถึง ชื่อของตัวแปรอาร์เรย์

index หมายถึง ขนาดของ Array ที่ต้องการสร้างตามลำดับ 1..n

รูปแบบการประกาศ ตัวแปรอาร์เรย์ 3 มิติ
ชนิดข้อมูล ชื่อตัวแปรอาร์เรย์ $[index_1][index_2]...[index_n];$
ตัวอย่าง
<pre>char score[10][5][2][3]; float value[1*2][2*2][3*2][4*2][5*2];</pre>

6.3 อาร์เรย์และฟังก์ชัน

6.3.1 วิธีการส่งผ่านค่าของอาร์เรย์และฟังก์ชัน

1) ตัวอย่างการส่งผ่านค่าโดยตรง (Function call by value)

โปรแกรม	ผลลัพธ์
<pre>1. #include <stdio.h> 2. display (int num) 3. { 4. printf ("%d", num); 5. } 6. main () 7. { 8. int i,value[3] = { 2, 4, 6 }; 9. for (i=0;i<=2;i++){ 10. printf ("Number %d = ",i); 11. display (value[i]); 12. printf ("\n"); 13. } 14. }</pre>	

2) ตัวอย่างการส่งผ่านค่าโดยอ้างอิงที่อยู่ (Function call by reference)

โปรแกรม	ผลลัพธ์
<pre> 1. #include <stdio.h> 2. display(int *num) 3. { 4. printf("Value = %d \n", *num); 5. } 6. main () 7. { 8. int val[] = {1,2,3,4,5}; 9. int i; 10. for (i=0; i<5; i++) display (&val[i]); 11. }</pre>	

6.3.2 วิธีการรับและส่งค่าไปทำงานที่ฟังก์ชัน

1) ตัวอย่างการส่งค่าอาร์เรย์ 1 มิติ แบบมีการกำหนดค่า index ไปที่ฟังก์ชัน

โปรแกรม	ผลลัพธ์
<pre> 1. #include <stdio.h> 2. void display (int num) 3. { 4. printf ("%d", num); 5. } 6. main () 7. { 8. int i,value[3] = { 2, 4, 6 }; 9. for (i=0;i<=2;i++){ 10. printf ("Number %d = ",i); 11. display (value[i]); 12. printf ("\n"); 13. } 14. }</pre>	

2) ตัวอย่างการส่งค่าอาร์เรย์ 1 มิติ ไปที่ฟังก์ชัน (ส่งค่าทั้งหมดของตัวแปร)

โปรแกรม	ผลลัพธ์
<pre>1. #include <stdio.h> 2. float average (float num[]); 3. main () 4. { 5. float avg, value[] = { 10,20,30,40,50, }; 6. avg = average(value); 7. printf ("Average of value = %.2f", avg); 8. } 9. float average (float num[]) 10. { 11. int i; 12. float avg, sum = 0; 13. for (i = 0; i <=5; ++i) sum += num[i]; 14. avg = (sum / 6); 15. return avg; 16. }</pre>	

3) ตัวอย่างการส่งค่าอาร์เรย์ 2 มิติ ไปที่ฟังก์ชัน (call by value)

โปรแกรม	ผลลัพธ์
<pre> 1. #include <stdio.h> 2. #define ROWS 3 3. #define COLS 4 4. printMatrix (int matrix[ROWS][COLS]) 5. { 6. int i, j; 7. printf ("Elements in matrix: \n"); 8. for (i = 0; i < ROWS; i++) 9. { 10. for (j = 0; j < COLS; j++) 11. { 12. printf ("%d ", matrix[i][j]); 13. } 14. printf ("\n"); 15. } 16. } 17. main () 18. { 19. int data[ROWS][COLS] = { 20. {1, 2, 3,4}, 21. {5, 6, 7,8}, 22. {9, 10 ,11 , 11} 23. }; 24. printMatrix (data); 25. }</pre>	

4) ตัวอย่างการส่งค่าอาร์เรย์ 2 มิติ ไปที่ฟังก์ชัน (call by reference)

โปรแกรม	ผลลัพธ์
<pre> 1. #include <stdio.h> 2. #define ROWS 2 3. #define COLS 3 4. void printMatrix (int *data, int row, int col) 5. { 6. int i, j; 7. printf ("Elements in 2D matrix: \n"); 8. for (i = 0; i < row; i++) 9. { 10. for (j = 0; j < col; j++) 11. { 12. printf("%d ", *((data+i*col) + j)); 13. } 14. printf ("\n"); 15. } 16. } 17. main() 18. { 19. int data [ROWS][COLS] = { 20. {1,2,3},{4,5,6} 21. }; 22. printMatrix ((int *)data, ROWS, COLS); 23. }</pre>	

5) ตัวอย่างการส่งค่าอาร์เรย์ 1 มิติ จากฟังก์ชันกลับมา

โปรแกรม	ผลลัพธ์
<pre> 1. #include <stdio.h> 2. int * displayArray() 3. { 4. int static num[] = {1, 2, 3, 4, 5}; 5. int i; 6. printf ("The result of function: \n"); 7. for (i = 0; i < 5; i++) 8. { 9. printf ("Value[%d] = %d \n",i+1,num[i]); 10. } 11. return num; 12. } 13. main () 14. { 15. int i; 16. int * num; 17. num = displayArray(); 18. printf ("\nThe result of main: \n"); 19. for (i = 0; i < 5; i++) 20. { 21. printf ("Value[%d] = %d \n",i+1,num[i]); 22. } 23. }</pre>	

6.3.3. การส่งค่าอาร์เรย์ 2 มิติ จากฟังก์ชันกลับมา

ปรับปรุงโปรแกรมจาก <https://www.programming-techniques.com/2011/08/returning-two-dimensional-array-from-a-function-in-c.html>

ตัวอย่างการส่งค่าอาร์เรย์ 2 มิติ จากฟังก์ชันกลับมา

โปรแกรม	ผลลัพธ์
<pre> 1. #include <stdio.h> 2. #include <stdlib.h> 3. int **matrix_sum (int matrix1[][3], int matrix2[][3]){ 4. int i, j; 5. int **matrix3; 6. matrix3 = malloc (sizeof(int*) * 3); 7. 8. for (i = 0; i < 3; i++) { 9. matrix3[i] = malloc (sizeof(int*) * 3); 10. } 11. 12. for (i = 0; i < 3; i++){ 13. for(j = 0; j < 3; j++){ 14. matrix3[i][j] = matrix1[i][j] + matrix2[i][j]; 15. } 16. } 17. return matrix3; 18. } 19. main(){ 20. int mat1[3][3], mat2[3][3]; 21. int **sumMatrix; 22. int i,j; 23. printf ("Enter the matrix1: \n"); 24. for (i = 0; i < 3; i++){ 25. for (j = 0; j < 3; j++){ 26. scanf ("%d",&mat1[i][j]); </pre>	

โปรแกรม	ผลลัพธ์
<pre> 27. } 28. } 29. printf("Enter the matrix2: \n"); 30. for(i = 0; i < 3; i++){ 31. for(j = 0; j < 3; j++){ 32. scanf("%d",&mat2[i][j]); 33. } 34. } 35. sumMatrix = matrix_sum(mat1,mat2); 36. printf("The sum of the matrix is: \n"); 37. for(i = 0; i < 3; i++){ 38. for(j = 0; j < 3; j++){ 39. printf("%d",sumMatrix[i][j]); 40. printf("\t"); 41. } 42. printf("\n"); 43. } 44. //free the memory 45. for(i = 0; i < 3; i++) { 46. free(sumMatrix[i]); 47. } 48. free(sumMatrix); 49. }</pre>	

คำถามท้ายบทที่ 6

1. จงอธิบายความหมายของการประกาศตัวแปร

1.1.char score[20] ;

1.2.int book[5][2] ;

1.3.float value[10][3][1];

1.4.char answer[2] = {'T','F'};

1.5.int data[5] = {10,20,30,40,50};

2. จงเขียนโปรแกรมเพื่อรับค่าเมตริกซ์จำนวน 2 เมตริกซ์ โดยรับขนาดและข้อมูลเมตริกซ์เป็นจำนวนเต็มผ่านทางแป้นคีย์บอร์ด และแสดงผลลัพธ์ของการบวกและลบของ 2 เมตริกซ์ดังกล่าว โดยใช้อาร์เรย์ 2 มิติ หมายเหตุ เมตริกซ์ต้องมีขนาดไม่เกิน 20*20 และมีหน้าจอการแสดงผลดังต่อไปนี้

```
Please input number of row: 2
Please input number of column: 3

*****Input MatrixA*****
Please input MatrixA[1][1]: 1
Please input MatrixA[1][2]: 2
Please input MatrixA[1][3]: 3
Please input MatrixA[2][1]: 4
Please input MatrixA[2][2]: 5
Please input MatrixA[2][3]: 6

*****MatrixA*****
1      2      3
4      5      6

*****Input MatrixB*****
Please input MatrixB[1][1]: 7
Please input MatrixB[1][2]: 8
Please input MatrixB[1][3]: 9
Please input MatrixB[2][1]: 10
Please input MatrixB[2][2]: 11
Please input MatrixB[2][3]: 12

*****MatrixB*****
7      8      9
10     11     12

*****Result*****
8      10     12
14     16     18
```

3. จงเขียนโปรแกรมเหมือนข้อ 2 โดยใช้มีการอาร์เรย์ 2 มิติ โดยกำหนดให้มีการเรียกใช้ 3 ฟังก์ชัน ดังนี้
- 1) ฟังก์ชันรับค่าข้อมูลทั้ง 2 เมตริกซ์
 - 2) ฟังก์ชันบวก
 - 3) ฟังก์ชันลบ
4. จงเขียนโปรแกรมในการรับค่าจำนวนจริงทางแป้นคีย์บอร์ดแล้วนำไปเก็บใน ตัวแปรชุดแบบ 2 มิติ มีขนาด 4 แถว 2 คอลัมน์ (4x2) แสดงผลค่าในสมาชิกของ ตัวแปรชุดทุกตัวทางจอภาพ

	[col 0]	[col 1]
[row 0]		
[row 1]		
[row 2]		
[row 3]		

5. จงเขียนโปรแกรมในการรับค่าคะแนนเก็บวิชา Computer Programming จำนวน 3 ครั้ง ครั้งละ 10 คะแนน ของนิสิตจำนวน 5 คน โดยกำหนดให้
- รับค่าคะแนนทางแป้นคีย์บอร์ด
 - หาค่าผลรวมคะแนนเก็บของแต่ละคน
 - หาค่าคะแนนเก็บที่มากที่สุดในแต่ละครั้งว่า พร้อมทั้งระบุว่ามีนิสิตคนใดที่ได้คะแนนดังกล่าว

ครั้งที่/คนที่	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	รวม
1	5.00	9.00	6.00	20.00
2	4.00	7.50	10.00	21.50
3	7.00	10.00	3.00	20.00
4	9.00	3.00	4.50	16.50
5	8.00	1.50	9.50	19.00

6. จงเขียนโปรแกรมเหมือนข้อ 5 โดยใช้มีการอาร์เรย์ 2 มิติ โดยกำหนดให้มีการ 3 ฟังก์ชันในการรับค่าคะแนนเก็บ

7. จงเขียนโปรแกรมเพื่อคำนวณหาค่า determinant ของ matrix

$$\det = (4)(3)(8) + (2)(7)(5) + (1)(-1)(6) - (5)(3)(1) - (6)(7)(4) - (8)(-1)(2)$$

8. จงเขียนโปรแกรมเปรียบเทียบเงินเดือนของพนักงาน 10 คน โดยใช้ array กำหนดให้ชนิดชนิดข้อมูลของข้อมูลเงินเดือนพนักงานเป็น float โดยแสดงผลดังต่อไปนี้

1. ลำดับของพนักงานที่ได้เงินเดือนมากที่สุด
2. เงินเดือนที่มากที่สุด

Result

No. Person : 9

Salary : 30000.00

9. จงเขียนโปรแกรมหาผลรวมเงินเดือน 12 ปีของพนักงาน 1 คนโดยใช้ array กำหนดให้ชนิดชนิดข้อมูลของข้อมูลเงินเดือนพนักงานเป็น Real โดยแสดงผลดังต่อไปนี้

1. แสดงลำดับที่ของเดือนที่พนักงานได้เงินเดือนมากกว่า 15000 บาท
2. ผลรวมของเงินเดือนพนักงานจำนวน 12 เดือน

Result

No. Month : 1

No. Month : 12

Total Salary : 3000000

10. จงเขียนคำนวณค่าอาหารของร้านอาหารแห่งหนึ่งที่มีโต๊ะอาหาร 10 โต๊ะโดยใช้ array กำหนดให้ราคาอาหารเป็น float โดยแสดงผลดังต่อไปนี้

1. ลำดับของโต๊ะอาหารที่มีค่าอาหารมากที่สุด
2. เงินค่าอาหารที่มากที่สุด

Result

No. Table : 1

Money : 3000