

**การทดลองที่ 4-2****เรื่อง อาร์เรย์ Array (ตอนที่ 2 อาร์เรย์สองมิติ และหลายมิติ)****วัตถุประสงค์**

เพื่อให้นักศึกษาเข้าใจและสามารถใช้งาน อาร์เรย์แบบ 2 มิติ และหลายมิติได้

บทนำ**ตัวแปรอาร์เรย์แบบ 2 มิติ (2-dimensional array)**

ตัวแปรอาร์เรย์แบบ 2 มิติจะเป็นตัวแปรที่มีการอ้างอิงถึงค่าข้อมูลโดยใช้ค่าเลขดัชนี 2 ค่าซึ่งประกอบไปด้วยค่าดัชนีที่ใช้ในการอ้างอิงในแนวแถว (row) และค่าดัชนีที่ใช้อ้างอิงในแนวคอลัมน์ (column) ตัวอย่างเช่น

8	16	9	52
3	15	27	6
14	25	2	10

เราเรียกอาร์เรย์ในตัวอย่างนี้ว่าอาร์เรย์ 2 มิติของจำนวนเต็ม ซึ่งประกอบด้วย 3 แถวและ 4 คอลัมน์ การประกาศตัวแปรอาร์เรย์แบบ 2 มิติจะเหมือนกับการประกาศตัวแปรอาร์เรย์แบบ 1 มิติเพียงแค่เพิ่มเติมการกำหนดขนาด ซึ่งจะต้องระบุทั้งในแนวแถวและคอลัมน์ เช่น

```
int val[3][4];
```

เป็นการประกาศตัวแปรอาร์เรย์แบบ 2 มิติชื่อ val เพื่อเก็บข้อมูลชนิดจำนวนเต็ม ซึ่งจะจองหน่วยความจำเท่ากับ $4 \text{ bytes} * 3 * 4 = 48 \text{ bytes}$

```
double prices[10][50];
```

เป็นการประกาศตัวแปรอาร์เรย์แบบ 2 มิติชื่อ prices เพื่อเก็บข้อมูลชนิดเลขทศนิยม ซึ่งจะจองหน่วยความจำเท่ากับ $8 \text{ bytes} * 10 * 50 = 4000 \text{ bytes}$

```
char code[6][10];
```

เป็นการประกาศตัวแปรอาร์เรย์แบบ 2 มิติชื่อ code เพื่อเก็บข้อมูลชนิดตัวอักษร ซึ่งจะจองหน่วยความจำเท่ากับ $1 \text{ bytes} * 6 * 10 = 60 \text{ bytes}$

อาร์เรย์หลายมิติ (multi-dimensional array)

อาร์เรย์หลายมิติ มีวิธีการประกาศและการใช้งานคล้ายกับอาร์เรย์ทั่วไป เพียงแต่มีจำนวนดัชนีในการอ้างอิงข้อมูลตามจำนวนมิติที่ประกาศไว้ เช่น

```
char code[6][10][20];
```

```
int ds[5][4][2][4];
```



การทดลอง อาร์เรย์ 2 มิติ

- ศึกษาการประกาศตัวแปรประเภทอาร์เรย์ การให้ค่าเริ่มต้น และการรับและแสดงค่าสำหรับอาร์เรย์ 2 มิติ
 - ศึกษาการทำงานของโปรแกรมที่ 5 ดังต่อไปนี้

โปรแกรมที่ 1 (labArray2Dex1.c)

```

1. //labArray2Dex15.c
2. #include<stdio.h>
3. int main()
4. {   int i, j, scores[3][5] = { {84, 71, 96, 65, 79},
5.                               {90, 55, 83, 68, 96},
6.                               {61, 77, 82, 94, 59}};
7.     float sum;
8.     for (i=0; i<3; i++)
9.     {   for (j=0, sum=0.0; j<5; j++)
10.        sum = sum+scores[i][j];
11.        printf("Sum of row %d = %.2f \n",i+1, sum);
12.    }
13.    return 0;
14. }
```

Checkpoint 1

- จงปรับปรุงโปรแกรม เพื่อให้สามารถรับค่าข้อมูลจำนวนเต็มไปเก็บไว้ในอาร์เรย์ scores โดยใช้คำสั่ง *scanf* แทนการกำหนดค่าเริ่มต้น โดยให้ผลของโปรแกรมเหมือนเดิม
- เพิ่มการคำนวณ *หาค่าเฉลี่ยรวม* ของทุกจำนวนในอาร์เรย์ และแสดงผลลัพธ์ ก่อนจบโปรแกรม

เมตริกซ์ Matrix

- ใช้อาร์เรย์สองมิติ เพื่อจัดเก็บเมตริกซ์ และ คำนวณต่างๆเกี่ยวกับเมตริกซ์
 - ศึกษาการทำงานของโปรแกรมต่อไปนี้

โปรแกรมที่ 2 (labMatrix.c)

```

//labMatrix.c
#include <stdio.h>
#define MAX 10
int main()
{   int MatA[MAX][MAX]={}, MatB[MAX][MAX]={}, MatC[MAX][MAX]={};
    int ra,rb,ca,cb,i,j;
    printf("Matrix A\n");
    printf("Number of rows = ");           scanf("%d",&ra);
    printf("Number of columns = ");        scanf("%d",&ca);
    // Input elements of Matrix A(ra x ca)
    for (i=0; i<ra; i++)
    {   for(j=0; j<ca; j++)
        {   printf("matA(%d,%d)=",i+1,j+1);
            scanf("%d",&matA[i][j]);
        }
    }
    // Show elements of Matrix A(ra x ca)
    for (i=0; i<ra; i++)
    {   for(j=0; j<ca; j++) printf("%5d",matA[i][j]);
        printf("\n");
    }
    printf("\nMatrix B\n");
    printf("Number of rows = ");           scanf("%d",&rb);
    printf("Number of columns = ");        scanf("%d",&cb);
```



```

// Input elements of Matrix B(rb x cb)
for (i=0; i<rb; i++)
{
    for(j=0; j<cb; j++)
    {
        printf("matB(%d,%d)=", i+1, j+1);
        scanf("%d", &matB[i][j]);
    }
}

// Show elements of Matrix B(rb x cb)
for (i=0; i<rb; i++)
{
    for(j=0; j<cb; j++) printf("%5d", matB[i][j]);
    printf("\n");
}

printf("Matrix C = A + B \n");
// Addition: Matrix C = A + B
for (i=0; i<ra; i++)
{
    for(j=0; j<ca; j++) matC[i][j]=matA[i][j]+matB[i][j];
}

// Show elements of Matrix C(ra x ca)
for (i=0; i<ra; i++)
{
    for(j=0; j<ca; j++) printf("%5d", matC[i][j]);
    printf("\n");
}
getch();
return 0;
}

```

- ทดลอง รันโปรแกรม ให้กำหนด ขนาดของแมตริกส์ และ ข้อมูลดังตัวอย่าง

$$A = \begin{bmatrix} 2 & 0 & 13 & 3 \\ 4 & 10 & 2 & 8 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 3 & 4 & -1 \\ 2 & 0 & -2 & -2 \end{bmatrix}$$

- แมตริกส์ C มีค่าเท่าไร? มีขนาดเท่าไร?

- ขนาดของ แมตริกส์ A และ B กำหนดโดยตัวแปรใดบ้าง?

- จากโปรแกรม เราสามารถกำหนดขนาดของแมตริกส์ ได้สูงสุดเท่าไร? จะแก้ไขได้อย่างไร?

Checkpoint 2 (ง่าย)

- จงแก้ไข เพิ่มเติมโปรแกรม เพื่อให้สามารถ ทำการคูณแต่ละสมาชิกของแมตริกส์ A และ B ในตำแหน่งที่ตรงกันได้ เช่น

$$A = \begin{bmatrix} 2 & 3 & 5 \end{bmatrix}, B = \begin{bmatrix} 4 & 0 & 8 \end{bmatrix}, C = A * B = \begin{bmatrix} 8 & 0 & 40 \end{bmatrix} \quad \text{หมายเหตุ } A, B \text{ และ } C \text{ มีขนาดเท่ากัน}$$

Checkpoint 3 (ยาก)

- จงแก้ไข เพิ่มเติมโปรแกรม เพื่อให้สามารถ ทำการคูณแมตริกส์ ได้

$$\text{แมตริกส์ } C(m \times p) = A(m \times n) \times B(n \times p)$$

แมตริกส์ ที่คูณกันได้ ตัวตั้ง (A) ต้องมีจำนวนหลัก เท่ากับจำนวนแถว ของตัวคูณ (B) และ แมตริกส์ผลคูณ (C) จะมีจำนวนแถว เท่ากับตัวตั้งและจำนวนหลักเท่ากับตัวคูณ เช่น

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \quad B = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \end{bmatrix}$$



$$C = \begin{bmatrix} a_{11} * b_{11} + a_{12} * b_{21} & a_{11} * b_{12} + a_{12} * b_{22} & a_{11} * b_{13} + a_{12} * b_{23} & a_{11} * b_{14} + a_{12} * b_{24} \\ a_{21} * b_{11} + a_{22} * b_{21} & a_{21} * b_{12} + a_{22} * b_{22} & a_{21} * b_{13} + a_{22} * b_{23} & a_{21} * b_{14} + a_{22} * b_{24} \\ a_{31} * b_{11} + a_{32} * b_{21} & a_{31} * b_{12} + a_{32} * b_{22} & a_{31} * b_{13} + a_{32} * b_{23} & a_{31} * b_{14} + a_{32} * b_{24} \end{bmatrix}$$

- ตัวอย่าง ผลการคูณเมตริกซ์

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 3 & 4 & -1 \\ 2 & 0 & -2 & -2 \end{bmatrix} \quad C = \begin{bmatrix} 5 & 3 & 0 & -5 \\ 11 & 9 & 4 & -11 \\ 17 & 15 & 8 & -17 \end{bmatrix}$$

อาร์เรย์หลายมิติ (การบ้าน)

3. มีชั้นวางหนังสืออยู่หลังหนึ่ง มีชั้นทั้งหมด 2 ชั้น วางหนังสือได้ชั้นละ 5 เล่ม มีการเก็บข้อมูลของหนังสือราคา จำนวนหน้า และ น้ำหนัก ให้ประกาศอาร์เรย์สามมิติ ในการเก็บข้อมูล ดังนี้ `float books[2][5][3];`

- โดยดัชนีตัวแรกอ้างถึงตำแหน่งของชั้นวางหนังสือ (ชั้นวางที่ 0 หรือ 1)
- ดัชนีตัวที่สองคือตำแหน่งหนังสือ (เล่มที่ 0 ถึง 4)
- และดัชนีตัวสุดท้ายบ่งชี้ข้อมูลของหนังสือ (ราคา จำนวนหน้า และ น้ำหนัก)
- จงเขียนโปรแกรม เพื่อเก็บและแสดงผลข้อมูล ดังต่อไปนี้

ตำแหน่งชั้น (level)	ตำแหน่งหนังสือ (book position)	ราคา (price)	จำนวนหน้า (pages)	น้ำหนัก (weight)
ชั้น 0	0	280.50	350	152.5
	1	410.00	280	202.5
	2	1590.75	406	858.7
	3	296.50	310	102.5
	4	1210.50	450	952.0
ชั้น 1	0	80.50	75	45
	1	2015	500	575
	2	120	25	30.75
	3	456.50	150	200
	4	110.50	50	130

- จงเขียนโปรแกรมเพื่อค้นหา

1. ราคาสูงสุดของหนังสือทั้งหมด (maxPrice)
2. จำนวนหน้าน้อยที่สุดของหนังสือทั้งหมด (minPage) และ
3. น้ำหนักเฉลี่ย (avgWeight) ของหนังสือในแต่ละชั้น