

Lab 11 - Array (2 Dimension)

1. อาร์เรย์ 2 มิติ (Array 2 Dimension)

จาก Lab 10 นักเรียนได้เรียนรู้การใช้อาร์เรย์ไปแล้วซึ่งสามารถทำให้นักเรียนเก็บข้อมูลหลาย ๆ ตัวภายในตัวแปรตัวเดียวได้ เช่นต้องการเก็บคะแนนในการสอบของนักเรียน 50 คนสามารถเก็บไว้ในตัวแปรแบบอาร์เรย์ดังนี้

```
int[] scores;
score = new int[50];
```

จากการประกาศตัวแปรข้างต้นจะได้โครงสร้างดังภาพที่ 1 คราวนี้นักเรียนลองพิจารณาว่าถ้ามีการสอบทั้งหมด 3 ครั้ง เราก็ต้องประกาศตัวแปรแบบอาร์เรย์ 3 ตัวแปรเพื่อเก็บข้อมูลคะแนนสอบของนักเรียนทั้ง 3 ครั้งของนักเรียน 50 คนดังนี้

```
// คะแนนสอบครั้งที่ 1 ของนักเรียนทั้ง 50 คน
int[] scores1;
score1 = new int[50];

// คะแนนสอบครั้งที่ 2 ของนักเรียนทั้ง 50 คน
int[] scores2;
score2 = new int[50];

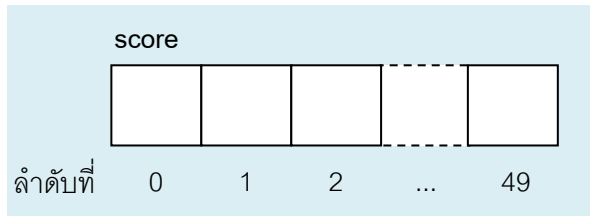
// คะแนนสอบครั้งที่ 3 ของนักเรียนทั้ง 50 คน
int[] scores3;
score3 = new int[50];
```

จะได้ตัวแปรแบบอาร์เรย์ 3 ตัวแปรขนาด 50 ช่องดังภาพที่ 2

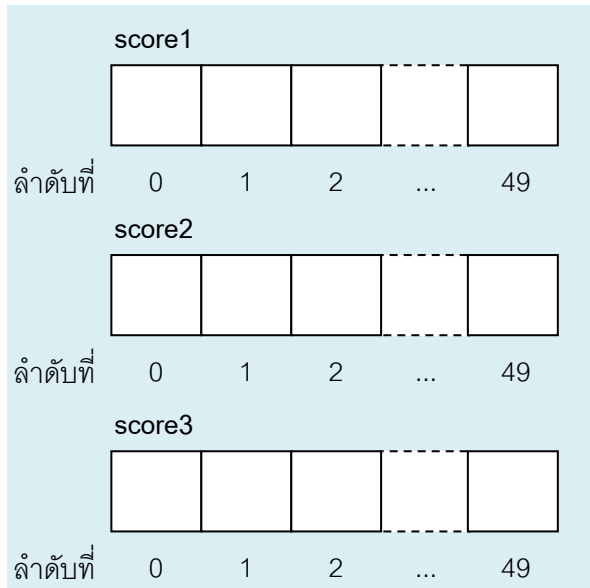
ซึ่งการประกาศตัวแปรแบบอาร์เรย์ที่มีชื่อตัวแปรคล้ายกันดังภาพที่ 2 นั้นสามารถทำให้ง่ายขึ้นได้โดยการใช้ตัวแปรแบบอาร์เรย์ 2 มิติโดยหากต้องการตัวแปรแบบอาร์เรย์ที่มีความยาวขนาด 50 ช่องจำนวน 3 ชุดสามารถใช้อาร์เรย์ 2 มิติโดยการประกาศง่าย ๆ ดังนี้

```
int[,] score;
score = new int[3,50];
```

จะมีโครงสร้างเป็นดังภาพที่ 3

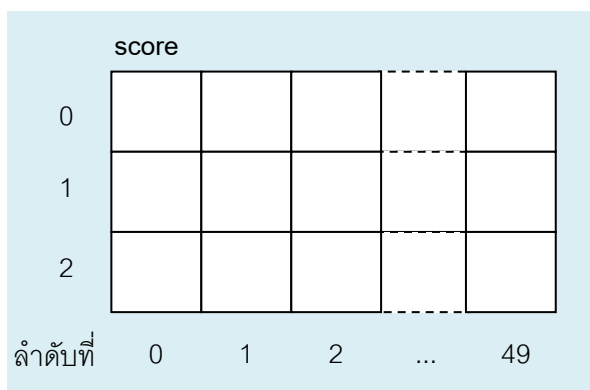


ภาพที่ 1 ตัวแปรแบบอาร์เรย์ชื่อว่า score ที่มีขนาด 50 ช่อง



ภาพที่ 2 ตัวแปรแบบอาร์เรย์ชื่อว่า score1, score2

และ score3 ที่มีขนาด 50 ช่อง



ภาพที่ 3 ตัวแปรแบบอาร์เรย์ 2 มิติ

ชื่อว่า score ที่มีขนาด 3 x 50 ช่อง

2. การประกาศตัวแปรแบบอาเรย์ 2 มิติ

ดังเช่นตัวแปรทั่ว ๆ ไปที่ก่อนจะนำตัวแปรมาใช้ต้องมีการประกาศตัวแปรเสียก่อน โดยการประกาศตัวแปรแบบอาเรย์นั้นก็มีความคล้ายกับการประกาศตัวแปรทั่ว ๆ ไปโดยมีรูปแบบดังนี้

```
DataType [ , ] ArrayName ;
```

โดย **DataType** หมายถึงชนิดของตัวแปรเช่น `int`, `double`, `char` เป็นต้น `ArrayName` หมายถึงชื่อของตัวแปรตามแต่นักเรียนจะตั้งขึ้นมา พิจารณาตัวอย่างจากแบบฝึกหัดที่ 2.1

แบบฝึกหัดที่ 2.1: การประกาศตัวแปรแบบอาเรย์

จงประกาศตัวแปรให้สอดคล้องกับเงื่อนไขที่กำหนดให้

เงื่อนไข	การประกาศ
ตัวแปรแบบอาเรย์ 1 มิติชนิด <code>int</code> ชื่อตัวแปรคือ <code>distance</code>	<code>int[] distance;</code>
ตัวแปรแบบอาเรย์ 2 มิติชนิด <code>int</code> ชื่อตัวแปรคือ <code>point</code>	<code>int[,] point;</code>
ตัวแปรแบบอาเรย์ 1 มิติชนิด <code>double</code> ชื่อตัวแปรคือ <code>height</code>	
ตัวแปรแบบอาเรย์ 2 มิติชนิด <code>double</code> ชื่อตัวแปรคือ <code>weight</code>	
ตัวแปรแบบอาเรย์ 2 มิติชนิด <code>char</code> ชื่อตัวแปรคือ <code>alpabet</code>	

3. การสร้างตัวแปรแบบอาเรย์ 2 มิติ

เช่นเดียวกันกับการประกาศตัวแปรแบบอาเรย์ 1 มิติ เมื่อนักเรียนประกาศอาเรย์ 2 มิติเสร็จแล้วจะยังไม่สามารถนำอาเรย์นั้น ๆ ไปใช้งานได้เนื่องจากการประกาศตัวแปรเปรียบเสมือนว่านักเรียนได้ทำป้ายบ้านเลขที่เตรียมมาปักลงบนที่ดิน แต่ยังไม่ได้ซื้อที่ดินพร้อมลงเสาเข็มสร้างบ้านจริง ๆ เลย ฉะนั้นนักเรียนจึงต้องทำการสร้างบ้านด้วย ซึ่งก็คือการสร้างอาเรย์นั่นเอง การสร้างอาเรย์ 2 มิติมีรูปแบบดังนี้

```
ArrayName = new DataType[num_rows, num_columns];
```

โดย `ArrayName` คือชื่อของตัวแปรแบบอาเรย์ที่เราได้ประกาศไว้ก่อนหน้านี้ `Datatype` คือชนิดของตัวแปรของตัวแปร `ArrayName` โดย `num_rows` คือจำนวนแถวของอาเรย์ 2 มิติและ `num_columns` คือจำนวนหลักของอาเรย์ 2 มิติ

สรุปก็คือการจะใช้ตัวแปรรูปแบบอาเรย์ 2 มิติจะต้องประกอบไปด้วยการประกาศและการสร้างตัวแปรแบบอาเรย์ ดังนี้

```
DataType [ , ] ArrayName ;  
ArrayName = new DataType[num_rows, num_columns];
```

ตัวอย่างเช่น ต้องการสร้างตัวแปรชื่อ Matrix ขนาด 4 แถวแต่ละแถวมี 5 หลัก สำหรับเก็บตัวเลขจำนวนเต็ม จะต้องใช้คำสั่งดังต่อไปนี้

```
int[,] Matrix;
Matrix = new int[4,5];
```

ซึ่งการประกาศและสร้างตัวแปรแบบอาเรย์ในข้างต้นสามารถเขียนย่อได้ดังนี้

```
DataType[,] ArrayName = new DataType[num_rows, num_columns];
```

ตัวอย่างเช่น

```
int[,] Matrix = new int[4,5];
```

แบบฝึกหัดที่ 3.1: การประกาศและสร้างตัวแปรแบบอาเรย์ 2 มิติ

จงประกาศและสร้างตัวแปรแบบอาเรย์ 2 มิติให้สอดคล้องกับเงื่อนไขที่กำหนด

เงื่อนไข	การประกาศ
ตัวแปรแบบอาเรย์ 2 มิติชนิด double ชื่อตัวแปรคือ lenght มีขนาด 3 แถว 10 หลัก	<code>double[,] lenght = new double[3,10];</code>
ตัวแปรแบบอาเรย์ 2 มิติชนิด char ชื่อตัวแปรคือ name มีขนาด 120 แถว 90 หลัก	
ตัวแปรแบบอาเรย์ 2 มิติชนิด int ชื่อตัวแปรคือ std_ID มีขนาด 1,235 แถว 8 หลัก	

3. โครงสร้างของตัวแปรแบบอาเรย์ 2 มิติ

ตัวอย่างที่ 3.1 : การประกาศและสร้างตัวแปรแบบอาเรย์ 2 มิติชนิด int ชื่อตัวแปรว่า score ขนาด 3 แถว 5 หลักสามารถทำได้โดยใช้คำสั่ง

```
int[,] score;
score = new int[3,5];
```

หลังจากการประกาศตัวแปรแบบอาเรย์ที่ชื่อว่า score ตามตัวอย่างที่ 3.1 เสิร์จนักเรียนจะได้ตัวแปรที่มีโครงสร้างเป็นดังภาพที่ 4

score	
0	
1	
2	
ลำดับที่	0 1 2 3 4

ภาพที่ 4 ตัวแปรแบบอาเรย์ชื่อว่า score ที่มีขนาด 3 แถว 5 หลัก

4. การกำหนดค่าเริ่มต้นให้กับตัวแปรแบบอาเรย์ 2 มิติ

จากหัวข้อที่ 1 – 3 นักเรียนสามารถประกาศและสร้างตัวแปรแบบอาเรย์ 2 มิติได้แล้ว แต่ถ้าต้องการกำหนดค่าเริ่มต้นให้กับตัวแปรดังกล่าวสามารถทำได้หลายรูปแบบดังต่อไปนี้

รูปแบบที่ 1

```
DataType[,] ArrayName = new DataType[nrows,ncols] {
    { value(0,0), value(0,1), ..., value(0,ncols-1) },
    { value(1,0), value(1,1), ..., value(1,ncols-1) },
    :
    :
    :
    { value(nrows-1,0), value(nrows-1,1), ..., value(nrows-1,ncols-1) }
};
```

รูปแบบที่ 2

```
DataType[,] ArrayName = new DataType[,] {
    { value(0,0), value(0,1), ..., value(0,ncols-1) },
    { value(1,0), value(1,1), ..., value(1,ncols-1) },
    :
    :
    :
    { value(nrows-1,0), value(nrows-1,1), ..., value(nrows-1,ncols-1) }
};
```

รูปแบบที่ 3

```
DataType[,] ArrayName = {
    { value(0,0), value(0,1), ..., value(0,ncols-1) },
    { value(1,0), value(1,1), ..., value(1,ncols-1) },
    :
    :
    :
    { value(nrows-1,0), value(nrows-1,1), ..., value(nrows-1,ncols-1) }
};
```

ตัวอย่างที่ 4.1 : การประกาศ สร้าง พร้อมทั้งกำหนดค่าเริ่มต้นให้กับตัวแปรแบบอาเรย์ 2 มิติชื่อ MatrixA ขนาด 4 x 3 ที่มีสมาชิกในองค์ประกอบต่าง ๆ เป็นดังนี้

$$A = \begin{bmatrix} 4 & 3 & 2 \\ 6 & 7 & 10 \\ 11 & 2 & 5 \\ 8 & 0 & 12 \end{bmatrix}$$

รูปแบบที่ 1

```
int[,] MatrixA = new int[4,3] {
    {4, 3, 2},
    {6, 7, 10},
    {11, 2, 5},
    {8, 0, 12}
};
```

รูปแบบที่ 2

```
int[,] MatrixA = new int[,] {
    {4, 3, 2},
    {6, 7, 10},
    {11, 2, 5},
    {8, 0, 12}
};
```

รูปแบบที่ 3

```
int[,] MatrixA = {  
    {4, 3, 2},  
    {6, 7, 10},  
    {11, 2, 5},  
    {8, 0, 12}  
};
```

แบบฝึกหัดที่ 4.1: การกำหนดค่าเริ่มต้นให้กับอาเรย์ 2 มิติ

จงประกาศ สร้าง พร้อมทั้งกำหนดค่าเริ่มต้นให้กับ
ตัวแปรแบบอาเรย์ 2 มิติชนิด double ชื่อ data ซึ่งมีขนาด
2 แถว 6 หลัก และมีการกำหนดค่าเริ่มต้นเป็นดังภาพที่ 5



data						
0	3.4	7.6	2.9	8.1	1.0	9.9
1	1.6	5.0	3.5	1.3	4.2	5.0
ลำดับที่	0	1	2	3	4	5

ภาพที่ 5 โครงสร้างและข้อมูลของ
ตัวแปรแบบอาเรย์ชื่อ text

5. การอ้างอิงข้อมูลในอาเรย์ 2 มิติ

ในอาเรย์ 1 มิติเราใช้ชื่อตัวแปรและลำดับที่ใน
การอ้างอิงถึงข้อมูลช่องใด ๆ ของอาเรย์ตัวอย่างเช่น
score[4] ของตัวแปรในภาพที่ 6 จะได้ค่าเท่ากับ 1 เป็นต้น
แต่ในตัวแปรแบบอาเรย์ 2 มิติ นั้นลำดับที่มีทั้งลำดับที่ของ
แถวและลำดับที่ของหลักจะนั้นต้องใช้ลำดับที่ทั้งสองเข้ามาช่วยในการอ้างอิงด้วยดังนี้

score						
	9	-4	0	-2	1	2
ลำดับที่	0	1	2	3	4	5

ภาพที่ 6 ค่าเริ่มต้นของตัวแปรแบบอาเรย์ชื่อ score

```
ArrayName[id_row, id_column]
```

โดย ArrayName คือชื่อของตัวแปรแบบอาเรย์ 2 มิติใด ๆ
id_row คือลำดับของแถวและ id_column คือลำดับของหลัก
ตัวอย่างเช่นต้องการเข้าถึงข้อมูลของตัวแปรแบบอาเรย์ชื่อ point ในแถว
ที่ 0 หลักที่ 3 ซึ่งมีโครงสร้างและข้อมูลดังภาพที่ 7 ให้ใช้ point[0, 3]
ในการอ้างอิง จะได้ข้อมูลเท่ากับ 11 เป็นต้น

point					
0	4	14	6	11	-5
1	-12	7	5	3	-9
2	10	-32	1	0	15
	0	1	2	3	4

ภาพที่ 7 ค่าเริ่มต้นของตัวแปรแบบอาเรย์
2 มิติขนาด 3 x 5 ชื่อ point

แบบฝึกหัดที่ 5.1 : การอ้างอิงข้อมูลในอาเรย์ 2 มิติ

กำหนดให้ point เป็นอาเรย์ 2 มิติขนาด 3 x 5 เก็บข้อมูลดังภาพที่ 7
จงตอบคำถามว่าการอ้างอิงต่อไปนี้ได้ค่าของข้อมูลเป็นเท่าไร

	point[0,4]	point[2,0]	point[1,4]	point[3,1]	point[0,5]
ข้อมูลที่ได้	-5				

ตัวอย่างที่ 5.1 : กำหนดให้ตัวแปรแบบอาร์เรย์ชนิด int ชื่อ MatrixA มีขนาด 3 x 4 มีโครงสร้างและค่าเริ่มต้นของข้อมูลในแต่ละช่องเป็นดังภาพที่ 8 ให้นักเรียนศึกษาการใช้คำสั่งในการทำงานต่าง ๆ ดังต่อไปนี้

- กำหนดค่า 30 ให้กับข้อมูลในแถวแรก หลักแรก ของอาร์เรย์

```
MatrixA[0,0] = 30;
```

- นำข้อมูลในแถวสุดท้าย หลักสุดท้าย ของอาร์เรย์มาแสดงผลบนหน้าจอ

```
Console.WriteLine(MatrixA[2,3]);
```

- กำหนดในทุกช่องของอาร์เรย์มีค่าเท่ากับ 0

```
for (int i = 0; i < 3; i++)
    for (int j = 0; j < 4; j++)
        MatrixA[i,j] = 0;
```

- คำนวณผลรวมของค่าทั้งหมดภายในอาร์เรย์ทุกช่อง

```
int sum = 0;
for (int i = 0; i < 3; i++)
    for (int j = 0; j < 4; j++)
        sum = sum + MatrixA[i,j];
```

MatrixA				
	0	1	2	3
0	90	5	12	-3
1	-8	15	-7	52
2	0	23	-8	1
	0	1	2	3

ภาพที่ 8 ค่าเริ่มต้นของตัวแปรแบบอาร์เรย์ 2 มิติขนาด 3 x 4 ชื่อ MatrixA

แบบฝึกหัดที่ 5.2: โปรแกรมรับและแสดงค่าของอาร์เรย์ 2 มิติ

ให้นักเรียนเติมค่าในช่องว่างให้สมบูรณ์เพื่อให้ส่วนของโปรแกรมต่อไปนี้เป็นส่วนของโปรแกรมที่กำหนดค่าของ Matrix A มีค่าดังภาพที่ 8 จากนั้นจะแสดงข้อมูลใน Matrix A ออกมาทางหน้าจอ

ตัวอย่าง

Matrix A is

90	5	12	-3
-8	15	-7	52
0	23	-8	1

```
1 static void Main()
2 {
3     int i, j;
4     _____ // ประกาศ สร้าง และกำหนดค่าเริ่มต้นให้กับ MatrixA
5     _____
6     _____
7     _____
8     _____
9     _____
10    _____;
11    for(_____;_____;_____)
12    {
13        for(_____;_____;_____)
14        {
15            Console.Write("{0,4}",MatrixA[_____]);
16        }
17        _____;
18    }
19 }
```

6. การหาจำนวนแถวและจำนวนหลักของอาเรย์ 2 มิติ

ในตัวแปรแบบอาเรย์ 1 มิติ เราใช้คำสั่ง `.Length` ในการหาจำนวนช่องของอาเรย์ แต่สำหรับอาเรย์ 2 มิติ ถ้าต้องการทราบว่าอาเรย์ 2 มิติที่มีจำนวนแถวกี่แถวหรือจำนวนหลักกี่หลัก สามารถทำได้โดยใช้คำสั่ง `.GetLength(dimension_index)` ดังนี้

คำสั่งในการหาจำนวนแถวของอาเรย์ 2 มิติ

```
ArrayName.GetLength(0)
```

คำสั่งในการหาจำนวนหลักของอาเรย์ 2 มิติ

```
ArrayName.GetLength(1)
```

ตัวอย่างที่ 6.1 : พิจารณาตัวแปรแบบอาเรย์ 2 มิติที่ชื่อว่า `point` ในภาพที่ 7 ให้นักเรียนศึกษาการใช้คำสั่งในการทำงานต่าง ๆ ดังต่อไปนี้

- กำหนดค่าให้อาเรย์ทุกช่องในหลักแรกของอาเรย์มีค่าเท่ากับ 5

```
for (int i = 0; i < point.GetLength(0); i++)  
    point[i,0] = 5;
```

- กำหนดค่าให้อาเรย์ทุกช่องในแถวลำดับที่สองของอาเรย์มีค่าเท่ากับ 0

```
for (int j = 0; j < point.GetLength(1); j++)  
    point[2,j] = 0;
```

- แสดงค่าของอาเรย์ทุกช่องออกมาทางหน้าจอ

```
for (int i = 0; i < point.GetLength(0); i++)  
    for (int j = 0; j < point.GetLength(1); j++)  
        Console.WriteLine("point[{0},{1}]={2}", i, j, point[i, j]);
```

แบบฝึกหัดที่ 6.1: การใช้คำสั่ง `.GetLength()`

จากโปรแกรมในแบบฝึกหัดที่ 5.2 ถ้าต้องการเปลี่ยนส่วนของโปรแกรมในบรรทัดที่ 11 และ 13 โดยใช้คำสั่ง `.GetLength` ส่วนของโปรแกรมในบรรทัดดังกล่าวจะเปลี่ยนแปลงเป็นเช่นไร

บรรทัดที่ 11

บรรทัดที่ 13

แบบฝึกหัดที่ 6.2: ShowMatrix

เราจะทำการเขียนเมธอดชื่อ ShowMatrix เพื่อแสดงข้อมูลภายในเมตริกซ์ออกทางหน้าจอโดยตัวเมธอดจะรับเมตริกซ์มาในรูปของพารามิเตอร์แบบอาร์เรย์สองมิติจึงเติมส่วนที่ขาดหายไปของเมธอดเพื่อให้โปรแกรมทำงานได้อย่างสมบูรณ์ตามตัวอย่างของผลลัพธ์

ตัวอย่างผลการทำงาน

5	3	8
2	6	10
1	8	25
12	3	30

```
using System;
class Matrix {
    static _____ ShowMatrix(_____) {
        for (int i = 0; i < _____; i++) {
            for (int j = 0; j < _____; j++) {
                Console.WriteLine("{0,4}", _____);
            }
            Console.WriteLine();
        }
    }

    static void Main() {
        int[,] A = {
            { 5, 3, 8 },
            { 2, 6, 10 },
            { 1, 8, 25 },
            { 12, 3, 30 }
        };
        ShowMatrix(A);
    }
}
```

แบบฝึกหัดที่ 6.3: ReadMatrix

โปรแกรมด้านล่างจะทำการถามขนาดของเมตริกซ์จากผู้ใช้และทำการเรียกเมธอดชื่อ ReadMatrix เพื่อสร้างเมตริกซ์ตามขนาดที่กำหนดและอ่านข้อมูลของเมตริกซ์มาทีละค่าจากผู้ใช้ หลังจากนั้นโปรแกรมจึงเรียกใช้เมธอด ShowMatrix จากแบบฝึกหัดที่ 6.2 เพื่อแสดงเมตริกซ์ออกทางหน้าจอ

ตัวอย่างผลการทำงาน

```
How many rows? 2
How many columns? 3
Enter element[1,1]: 9
Enter element[1,2]: 8
Enter element[1,3]: 7
Enter element[2,1]: 6
Enter element[2,2]: 5
Enter element[2,3]: 4
Matrix A is
9 8 7
6 5 4
```



```

using System;
class Matrix {

    // คัดลอกเมธอด ShowMatrix จากแบบฝึกหัดที่ 6.2 มาวางในตำแหน่งนี้

    static int[,] ReadMatrix(int nrows, int ncols) {
        int[,] m = new int[_____,_____];
        for (int i = 0; i < nrows; i++) {
            for (int j = 0; j < ncols; j++) {
                Console.Write("Enter element[{0},{1}]: ", _____, _____);
                _____ = int.Parse(Console.ReadLine());
            }
        }
        return m;
    }

    static void Main() {
        int num_rows, num_cols;
        int[,] A;

        Console.Write("How many rows? ");
        num_rows = int.Parse(Console.ReadLine());
        Console.Write("How many columns? ");
        num_cols = int.Parse(Console.ReadLine());

        A = ReadMatrix(num_rows, num_cols);

        Console.WriteLine("Matrix A is");
        _____;
    }
}

```

แบบฝึกหัดที่ 6.4: TransposeMatrix

ทำการเพิ่มเมธอดชื่อ TransposeMatrix ลงในโปรแกรม จากแบบฝึกหัดที่แล้วเพื่อคำนวณทรานสโพสเมตริกซ์ (การสลับ เปลี่ยนแถวและหลัก) จากเมตริกซ์ที่รับเข้ามา โดยเมธอด TransposeMatrix จะรับเมตริกซ์ในรูปพารามิเตอร์แบบอาร์เรย์สอง มิติและส่งค่ากลับเป็นอาร์เรย์สองมิติตัวใหม่ที่เก็บค่าเมตริกซ์ที่ถูก ทรานสโพสแล้ว

ตัวอย่าง

```

How many row? 2
How many column? 3
Enter element[1,1] : 10
Enter element[1,2] : -5
Enter element[1,3] : 3
Enter element[2,1] : 9
Enter element[2,2] : 12
Enter element[2,3] : -6
Matrix A is
  10  -5  3
   9  12 -6
Transpose of Matrix A is
  10  9
 -5  12
  3  -6

```

```
using System;
```

```

class Matrix {
    // คัดลอกเมธอด ShowMatrix และ ReadMatrix
    // จากแบบฝึกหัดที่ 6.2 และ 6.3 มาวางไว้ ณ ตำแหน่งนี้

    static _____ TransposeMatrix(_____ ) {
        int[,] mt = new int[_____, _____];
        for (int i = 0; i < _____; i++)
            for (int j = 0; j < _____; j++)
                _____;
        return _____;
    }

    static void Main() {
        int num_rows, num_cols;
        int[,] A, At;

        Console.WriteLine("How many rows? ");
        num_rows = int.Parse(Console.ReadLine());
        Console.WriteLine("How many columns? ");
        num_cols = int.Parse(Console.ReadLine());

        A = _____;
        Console.WriteLine("Matrix A is");
        _____;

        Console.WriteLine("Transpose of Matrix A is");
        At = TransposeMatrix(A);
        _____;
    }
}

```

7. ฝึกหัดเขียนโปรแกรมและสนุก

โปรแกรมที่ 7.1 : ดีเทอร์มิแนนต์ของเมตริกซ์ขนาด 2 x 2

ให้นักเรียนโปรแกรมที่คำนวณค่าดีเทอร์มิแนนต์ของเมตริกซ์ขนาด 2 x 2 ซึ่งมีสูตรการคำนวณดังนี้

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

ตัวอย่าง

```

Enter matrix A
Enter element[1,1]: 1
Enter element[1,2]: 2
Enter element[2,1]: 3
Enter element[2,2]: 4
The determinant of A is -2

```

แบบฝึกหัดที่ 7.2: การคูณเมตริกซ์ขนาด 2 x 2

กำหนดเมตริก A เป็นเมตริกขนาด p x q และเมตริก B เป็นเมตริกขนาด q x r

การคูณเมตริกกับเมตริกสามารถทำได้โดยถ้า $C = AB$ สามารถหาเมตริก C ได้โดยใช้สูตรดังนี้

$$c_{ij} = \sum_{k=1}^q a_{ik} \cdot b_{kj}$$

ตัวอย่างเช่น

$$\begin{bmatrix} 2 & 0 & 1 \\ 1 & -1 & 2 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 \\ 3 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ -2 & 0 \end{bmatrix}$$

จงเขียนโปรแกรมเพื่อทำการคูณเมตริกซ์ขนาด 2 x 2 และ

แสดงผลลัพธ์ออกทางหน้าจอ

ตัวอย่าง

```
Enter matrix A
Enter element[1,1]: 1
Enter element[1,2]: 2
Enter element[2,1]: 3
Enter element[2,2]: 4
```

```
Enter matrix B
Enter element[1,1]: 5
Enter element[1,2]: 6
Enter element[2,1]: 7
Enter element[2,2]: 8
```

Matrix A*B is

```
19 22
43 50
```

แบบฝึกหัดที่ 7.3: การคูณเมตริกซ์ขนาด $n \times n$

จงปรับปรุงแก้ไขโปรแกรมในโปรแกรมที่ 7.2 ให้สามารถหาผลลัพธ์ของการคูณระหว่างเมตริกซ์ A (มีขนาดเป็น $p \times q$) กับเมตริกซ์ B (มีขนาดเป็น $q \times r$) มีลักษณะการทำงานเป็นดังตัวอย่าง

ตัวอย่าง

```
Input p : 2
Input q : 3
Input r : 2
Enter matrix A
Enter element[1,1]: 2
Enter element[1,2]: 0
Enter element[1,3]: 1
Enter element[2,1]: 1
Enter element[2,2]: -1
Enter element[2,3]: 2

Enter matrix B
Enter element[1,1]: 1
Enter element[1,2]: 2
Enter element[2,1]: 3
Enter element[2,2]: 0
Enter element[3,1]: 0
Enter element[3,2]: -1

A*B is
  2  3
 -2  0
```

แบบฝึกหัดที่ 7.2: ตัดเกรดนักเรียน

ในการเรียนวิชาศิลปะแห่งมหาวิทยาลัยผู้สอนได้จัดให้มีการสอบเก็บคะแนนทั้งหมด 3 ครั้งแต่ละครั้งมีคะแนนเต็ม 100 คะแนน เมื่อเก็บคะแนนครบทั้งสามครั้ง อาจารย์จะนำคะแนนทั้งสามครั้งมาหาคะแนนเฉลี่ยเลขคณิตรวมโดยให้น้ำหนักของการสอบแต่ละครั้งตามตารางที่ 1 และนำคะแนนเฉลี่ยรวมที่หาได้มาตัดเกรดให้นักเรียนตามตารางที่ 2 ให้นักเรียนเขียนโปรแกรมที่ช่วยอาจารย์ในการตัดเกรดนักเรียนทั้งหมด n คนดังตัวอย่าง

ตารางที่ 1 ค่าน้ำหนักของการสอบ

การสอบ	น้ำหนัก
ครั้งที่ 1	30%
ครั้งที่ 2	30%
ครั้งที่ 3	40%
รวม	100%

ตารางที่ 2 เกณฑ์ในการตัดเกรด

คะแนนเฉลี่ยรวม (\bar{x})	เกรด
$80 \leq \bar{x}$	A
$70 \leq \bar{x} < 80$	B
$60 \leq \bar{x} < 70$	C
$50 \leq \bar{x} < 60$	D
$\bar{x} < 50$	F

ตัวอย่าง

Numbers of students : 4
Please input score of student1.
Test1 : 78
Test2 : 79
Test3 : 79

Please input score of student2.
Test1 : 60
Test2 : 62
Test3 : 70

Please input score of student3.
Test1 : 81
Test2 : 89
Test3 : 85

Please input score of student4.
Test1 : 45
Test2 : 53
Test3 : 41

Average score of Student1 is 78.7, Grade B.
Average score of Student2 is 64.6, Grade C.
Average score of Student3 is 85.0, Grade A.
Average score of Student4 is 45.8, Grade F.

