Lab 7 : More On Functions and Array Lab

Multiplication of two matrixes:

Rule: Multiplication of two matrixes is only possible if first matrix has size m X n and other matrix has size n x r. Where m, n and r are any positive integer.

Multiplication of two matrixes is defined as

$$[AB]_{i,j} = \sum_{s=1}^{n} A_{i,s} B_{s,j}$$

Where $1 \le i \le m$ and $1 \le j \le n$

EX: Suppose two matrixes A and B of size of 2 x 2 and 2 x 3 respectively:

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \quad B = \begin{pmatrix} 5 & 6 & 7 \\ 8 & 9 & 10 \end{pmatrix}$$

Multiplication of two matrixes:

$$A * B = \begin{pmatrix} 1*5 + 2*8 & 1*6 + 2*9 & 1*7 + 2*10 \\ 3*5 + 4*8 & 3*6 + 4*9 & 3*7 + 4*10 \end{pmatrix}$$

$$A * B = \begin{pmatrix} 21 & 24 & 27 \\ 47 & 54 & 61 \end{pmatrix}$$

การคูณเมทริกซ์ด้วยเมทริกซ์

ถ้า A และ B เป็นเมทริกซ์ 2 เมทริกซ์ใด ๆ การนำเมทริกซ์ A มาคูณกับเมทริกซ์ B จะเกิดผลขึ้นอย่างใดอย่างหนึ่งใน 2 อย่างต่อไปนี้

1. ไม่สามารถหาผลคูณได้
2. สามารถหาผลคูณได้

ปัญหาที่เราจะต้องทราบก็คือ ถ้าหาผลคูณได้ต้องมีเงื่อนไขอย่างไร และสมาชิกของเมทริกซ์ที่เป็นผลคูณจะหามาได้อย่างไร ให้ดูหลักการต่อไปนี้

1. มิติของเมทริกซ์ที่นำมาหาผลคูณ

ถ้า A เป็นเมทริกซ์ $m \times p$ B เป็นเมทริกซ์ $q \times n$

ผลคูณ AB จะเกิดขึ้นได้เมื่อ $p = p$ และ AB จะมีมิติ $m \times n$

2. ลักษณะสมาชิกของเมทริกซ์ที่เป็นผลคูณ (ถ้าหาผลคูณได้)

หลักการหาสมาชิกโดยทั่ว ๆ ไป สามารถหาได้ดังนี้

"สมาชิกของผลคูณของเมทริกซ์ในแถวที่ i หลักที่ j จะเกิดสมาชิกในแถวที่ i ของเมทริกซ์ที่อยู่หน้า คูณกับ สมาชิกในหลักที่ j ของเมทริกซ์หลักเป็นคู่ ๆ แล้วนำมาบวกกัน" ลองมาดูตัวอย่างกัน

ตัวอย่าง ให้ $A = \begin{bmatrix} 1 & 2 \\ -1 & 0 \\ 3 & 2 \end{bmatrix}, B = \begin{bmatrix} 1 & 5 & 2 \\ -2 & 0 & 1 \end{bmatrix}$ **จงหา AB และ BA**

วิธีทำ (1) $AB = \begin{bmatrix} 1 & 2 \\ -1 & 0 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} 1 & 5 & 2 \\ -2 & 0 & 1 \end{bmatrix}$

ใช้หลักการแถว คูณ หลัก

$$= \begin{bmatrix} (1)(1)+(2)(-2) & (1)(5)+(2)(0) & (1)(2)+(2)(1) \\ (-1)(1)+(0)(-2) & (-1)(5)+(0)(0) & (-1)(2)+(0)(1) \\ (3)(1)+(2)(-2) & (3)(5)+(2)(0) & (3)(2)+(2)(1) \end{bmatrix}$$

$$= \begin{bmatrix} -3 & 5 & 4 \\ -1 & -5 & -2 \\ -1 & 15 & 8 \end{bmatrix}$$

ข้อสังสังเกตในการคูณ ถ้าใช้แถวใด คูณ ผลลัพธ์ที่ได้ก็จะเป็นแถวนั้นถ้าคูณหลักใดก็ได้หลักนั้น ของเมทริกซ์ผลลัพธ์ เช่น

- แถวที่ 1 คูณหลักที่ 1 ผลลัพธ์ก็จะอยู่ตำแหน่ง แถวที่ 1 หลักที่ 1 ของเมทริกซ์ผลลัพธ์
- แถวที่ 1 คูณหลักที่ 2 ผลลัพธ์ก็จะอยู่ตำแหน่ง แถวที่ 1 หลักที่ 2 ของเมทริกซ์ผลลัพธ์
- แถวที่ 1 คูณหลักที่ 3 ผลลัพธ์ก็จะอยู่ตำแหน่ง แถวที่ 1 หลักที่ 3 ของเมทริกซ์ผลลัพธ์
- แถวที่ 2 คูณหลักที่ 1 ผลลัพธ์ก็จะอยู่ตำแหน่ง แถวที่ 2 หลักที่ 1 ของเมทริกซ์ผลลัพธ์
- แถวที่ 2 คูณหลักที่ 2 ผลลัพธ์ก็จะอยู่ตำแหน่ง แถวที่ 2 หลักที่ 2 ของเมทริกซ์ผลลัพธ์
- แถวที่ 2 คูณหลักที่ 3 ผลลัพธ์ก็จะอยู่ตำแหน่ง แถวที่ 2หลักที่ 3 ของเมทริกซ์ผลลัพธ์

<u>จงเขียนโปรแกรม การคูณของเมทริกซ์ 2 ตัว</u>

ให้เขียนเป็น ฟังก์ชั่นย่อย ดังต่อไปนี้

1. **ถามผู้ใช้ ขนาดของ เมทริกซ์ ตัวแรก**

```
void AskSizeMatrix(row,col);
```

2. **ถามผู้ใช้ ขนาดของ เมทริกซ์ ตัวที่สอง**

```
void AskSizeMatrix(row,col);
```

3. **ตรวจสอบว่า เมทริกซ์ 2 ตัว คูณกันได้หรือไม่**

```
int CheckPossible(rowA,colA,rowB,colB);
```

4. **รับการป้อนค่าของสมาชิกแต่ละตัวของ เมทริกซ์ ตัวแรก**

```
void EnterMatrix(Matrix,row,col);
```

5. **รับการป้อนค่าของสมาชิกแต่ละตัวของ เมทริกซ์ ตัวที่สอง**

```
void EnterMatrix(Matrix,row,col);
```

6. **แสดง เมทริกซ์ ตัวแรก ออกหน้าจอ**

```
void DisplayMatrix(Matrix,row,col);
```

7. **แสดง เมทริกซ์ ตัวที่สอง ออกหน้าจอ**

```
void DisplayMatrix(Matrix,row,col);
```

8. **ทำการคูณ เมทริกซ์**

```
void MultiplyMatrix(MatrixA,rowA,colA,
                    MatrixB,rowB,colB);
```

9. **แสดง เมทริกซ์ ที่เป็นผลลัพธ์การคูณ ออกหน้าจอ**

```
void DisplayMatrix(Matrix,row,col);
```

```c
#include<stdio.h>
int main(){

   // initial variables
  int a[5][5],b[5][5],c[5][5],i,j,k,sum=0,m,n,o,p;

   // Enter the dimension of two matrixs
  printf("\nEnter the row and column of first matrix");
  scanf("%d %d",&m,&n);
  printf("\nEnter the row and column of second matrix");
  scanf("%d %d",&o,&p);

  // Check if these two matrix can be multipled
  if(n!=o){
      printf("Matrix mutiplication is not possible");
      printf("\nColumn of first matrix must be same as
row of second matrix");
  }
  else{

       // Enter values to matrix#1
      printf("\nEnter the First matrix->");
      for(i=0;i<m;i++)
        for(j=0;j<n;j++)
            scanf("%d",&a[i][j]);


       // Enter values to matrix#2
      printf("\nEnter the Second matrix->");
      for(i=0;i<o;i++)
        for(j=0;j<p;j++)
            scanf("%d",&b[i][j]);


       // Show the first matrix
      printf("\nThe First matrix is\n");
      for(i=0;i<m;i++){
        printf("\n");
        for(j=0;j<n;j++){
            printf("%d\t",a[i][j]);
        }
      }
```

```
      // Show the second matrix
    printf("\nThe Second matrix is\n");
    for(i=0;i<o;i++){
     printf("\n");
     for(j=0;j<p;j++){
         printf("%d\t",b[i][j]);
     }
    }


     // Initial Matrix C to be 0
    for(i=0;i<m;i++)
     for(j=0;j<p;j++)
         c[i][j]=0;


     // Multipy 2 matrix
    for(i=0;i<m;i++){ //row of first matrix
     for(j=0;j<p;j++){  //column of second matrix
         sum=0;
         for(k=0;k<n;k++)
             sum=sum+a[i][k]*b[k][j];
         c[i][j]=sum;
     }
    }
}


   // display the result matrix C
  printf("\nThe multiplication of two matrix is\n");
  for(i=0;i<m;i++){
     printf("\n");
     for(j=0;j<p;j++){
         printf("%d\t",c[i][j]);
     }
  }


  return 0;
}
```