



Computer Science 2

Practical work 03: (2D array)

Exercise 01 :

Write a C program to store elements in 2D array and print it in row-major, column-major, and matrix forms.

Example :

If we have a 2D array

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

The three methods to print elements of this array it can be shown as follows

Matrix form	Row-major order	Column-major order
$X = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$	X=1 2 3 4 5 6 7 8 9	X= 1 4 7 2 5 8 9 6 3

Solution

<pre>#include <stdio.h> int main() { int a[3][3]={{1,2,3},{4,5,6},{7,8,9}}; int i,j,k,l,m,n; printf("row-major order: \n"); for(k=0;k<3;k++) { for(l=0;l<3;l++) { printf("%d ",a[k][l]); } } printf("\ncolumn-major order: \n"); for(m=0;m<3;m++) { for(n=0;n<3;n++) { printf("%d ",a[n][m]); } } }</pre>	<pre>printf("\nMatrix order: \n"); for(i=0;i<3;i++) { for(j=0;j<3;j++) { printf("%d\t",a[i][j]); } printf("\n"); } return 0; }</pre>
---	--

Exercise 02 :

Write a C program to print the element of the matrix D, where $D=2*A+B$. Where

$$A=\begin{pmatrix} 1 & 2 & 3 \\ 4 & 2 & 1 \\ 3 & 6 & 5 \end{pmatrix}, \text{ and } B=\begin{pmatrix} 2 & 4 & 3 \\ 3 & 2 & 1 \\ 7 & 10 & 6 \end{pmatrix}.$$

Solution

```
#include <stdio.h>
int main() {
    int A[3][3]={{1,2,3},{4,2,1},{3,6,5}};
    int B[3][3]={{2,4,3},{3,2,1},{7,10,6}};
    int D[3][3];
    int i,j;
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            D[i][j]=2*A[i][j]+B[i][j];
            printf("%d\t",D[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

Exercise 03 :

Write a C program for Identity matrix. The (4×4) identity matrix **I** it can be represented by

$$\mathbf{I}=\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Solution

```
#include <stdio.h>
int main() {
    int I[4][4];
    int i,j;
    for(i=0;i<4;i++)
    {
        for(j=0;j<4;j++)
        {
            if(i!=j)
            {
                I[i][j]=0;
            }
            else
            {
                I[i][j]=1;
            }
            printf("%d\t",I[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

Exercise 04 :

Write a C program for multiplying a matrix by a vector, and vector by a matrix.

Solution

Program	Output
<pre>#include <stdio.h> int main() { int M[4][3]={1,2,3,10,20,30,4,5,6,11,12,13}; int Ar[4]={1,2,3,4}; int Ar1[3]={4,5,6}; int B[3], C[4], sum1=0, sum2=0 ; int i,j,m,n; printf("multiplication of vector by matrix:\n"); for(i=0;i<3;i++) { for(j=0;j<4;j++) { sum1=sum1+Ar[j]*M[j][i]; } } }</pre>	<pre>multiplication of vector by matrix: 77 105 133 multiplication of matrix by vector: 32 320 77 182</pre>

```

    B[i]=sum1;
    printf("%d\t",B[i]);
    sum1=0;
}
printf("\nmultiplication of matrix by
vector:\n");
for(m=0;m<4;m++)
{
    for(n=0;n<3;n++)
    {
        sum2=sum2+M[m][n]*Ar1[n];

    }
    C[m]=sum2;
    printf("%d\n",C[m]);
    sum2=0;
}

return 0;
}

```

Exercise 05 :

Write a C program to multiply two matrices.

Example :

$$D=A \times B, \text{ where } A = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \end{pmatrix} \text{ and } B = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}.$$

Theoretically the elements of the matrix D are given in this way

Row-0	$D[0][0]=A[0][0]*B[0][0]+ A[0][1]*B[1][0]+ A[0][2]*B[2][0]$ $D[0][1]=A[0][0]*B[0][1]+ A[0][1]*B[1][1]+ A[0][2]*B[2][1]$ $D[0][2]=A[0][0]*B[0][2]+ A[0][1]*B[1][2]+ A[0][2]*B[2][2]$
Row-1	$D[1][0]=A[1][0]*B[0][0]+ A[1][1]*B[1][0]+ A[1][2]*B[2][0]$ $D[1][1]=A[1][0]*B[0][1]+ A[1][1]*B[1][1]+ A[1][2]*B[2][1]$ $D[1][2]=A[1][0]*B[0][2]+ A[1][1]*B[1][2]+ A[1][2]*B[2][2]$

All of this gives

$$D = \begin{pmatrix} 30 & 36 & 42 \\ 54 & 66 & 78 \end{pmatrix}$$

Solution :

Program	Output
<pre>#include <stdio.h> int main() { int A[2][3]={1,2,3,3,4,5}; int B[3][3]={1,2,3,4,5,6,7,8,9}; int D[2][3]; int i,j,k,m,n; for(m=0;m<2;m++) { for(n=0;n<3;n++) { D[m][n]=0; } } printf("\ndisplay the elements of the matrix :\n"); for(i=0;i<2;i++) { for(j=0;j<3;j++) { for(k=0;k<3;k++) { D[i][j]+=A[i][k]*B[k][j]; } } } for(m=0;m<2;m++) { for(n=0;n<3;n++) { printf("%d\t",D[m][n]); } printf("\n"); } return 0; }</pre>	<pre>display the elements of the matrix : 30 36 42 54 66 78</pre>