**Assignment - 16 A Job Ready Bootcamp in C++, DSA and IOT MySirG**

**Multi-Dimensional Array in C Language**

***Abhishek Kumar***

//1. Write a program to calculate the sum of two matrices each of order 3x3

#include<stdio.h>

int main()

{

    int a[3][3]={1,2,3,4,5,6,7,8,9};

    int b[3][3]={5,6,2,9,7,3,4,2,1};

    int c[3][3];

    int i,j;

    for(i=0; i<=2; i++)

    {

        for(j=0; j<=2; j++)

        {

            c[i][j] = a[i][j]+b[i][j];

        }

    }

    printf("Sum of the two matrices:\n");

    for(i=0; i<=2; i++)

    {

        for(j=0; j<=2; j++)

        {

            printf("%d ",c[i][j]);

        }

        printf("\n");

    }

    return 0;

}

//2. Write a program to calculate the product of two matrices each of order 3x3.

#include<stdio.h>

int main()

{

    int i,j,k,sum=0;

    int a[3][3]={1,2,3,4,5,6,7,8,9};

    int b[3][3]={9,8,7,6,5,4,3,2,1};

    int c[3][3];

    for(i=0; i<=2; i++)

    {

        for(j=0; j<=2; j++)

        {

            for(k=0; k<=2; k++)

            {

                sum = sum + (a[i][k] \* b[k][j]);

                c[i][j]=sum;

                sum = 0;

            }

        }

    }

    printf("Calculate the product of two matrices:\n");

    for(i=0; i<=2; i++)

    {

        for(j=0; j<=2; j++)

        {

            printf("%d ",c[i][j]);

        }

        printf("\n");

    }

    return 0;

}

//3. Write a program in C to find the transpose of a given matrix

#include<stdio.h>

int main()

{

    int i,j;

    int a[3][3]={1,2,3,4,5,6,7,8,9};

    int b[3][3];

    for(i=0; i<=2; i++)

    {

        for(j=0; j<=2; j++)

        {

            b[i][j] = a[j][i];

        }

    }

    for(i=0; i<=2; i++)

    {

        for(j=0; j<=2; j++)

        {

            printf("%d ",b[i][j]);

        }

        printf("\n");

    }

}

//4. Write a program in C to find the sum of right diagonals of a matrix.

#include<stdio.h>

int main()

{

    int a[20][20],i,j,n,sum=0;

    printf("Enter the size of elements:");

    scanf("%d",&n);

    for(i=0; i<n; i++)

    {

        for(j=0; j<n; j++)

        {

            printf("Array Elements: [%d][%d]::",i,j);

            scanf("%d",&a[i][j]);

            if(i==j)

            sum = sum+a[i][j];

        }

    }

    printf("Matrix is:\n");

    for(i=0; i<n; i++)

    {

        for(j=0; j<n; j++)

            printf("%d ",a[i][j]);

            printf("\n");

    }

    printf("Sum of the right diagonal element is: %d\n",sum);

    return 0;

}

//5. Write a program in C to find the sum of left diagonals of a matrix.

#include<stdio.h>

int main()

{

    int a[20][20],i,j,n,sum=0;

    printf("Enter the size of elements:");

    scanf("%d",&n);

    for(i=0; i<n; i++)

    {

        for(j=0; j<n; j++)

        {

            printf("Array Elements: [%d][%d]::",i,j);

            scanf("%d",&a[i][j]);

            if(i==j)

            sum = sum+a[j][i];

        }

    }

    printf("Matrix is:\n");

    for(i=0; i<n; i++)

    {

        for(j=0; j<n; j++)

            printf("%d ",a[i][j]);

            printf("\n");

    }

    printf("Sum of the right diagonal element is: %d\n",sum);

    return 0;

}

//6. Write a program in C to find the sum of rows and columns of a Matrix

#include<stdio.h>

int main()

{

    int i,j,row,col;

    int a[10][10],b[10][10];

    printf("Enter the row:");

    scanf("%d",&row);

    printf("Enter the column:");

    scanf("%d",&col);

    printf("Enter the first row %d col:\n",row\*col);

    for(i=0; i<row; i++)

    {

        for(j=0; j<col; j++)

        {

            scanf("%d",&a[i][j]);

        }

    }

    printf("Enter the second row %d col:\n",row\*col);

    for(i=0; i<row; i++)

    {

        for(j=0; j<col; j++)

        {

            scanf("%d",&b[i][j]);

        }

    }

    printf("Sum of the row and columns:\n");

    for(i=0; i<row; i++)

    {

        for(j=0; j<col; j++)

        {

            printf("%d ",a[i][j]+b[i][j]);

        }

        printf("\n");

    }

    return 0;

}

//7. Write a program in C to print or display the lower triangular of a given matrix.

#include<stdio.h>

int main()

{

    int i,j,row,col;

    int a[10][10];

    printf("Enter the row:");

    scanf("%d",&row);

    printf("Enter the column:");

    scanf("%d",&col);

    printf("Enter the row %d col:\n",row\*col);

    for(i=0; i<row; i++)

    {

        for(j=0; j<col; j++)

        {

            scanf("%d",&a[i][j]);

        }

    }

    printf("Display the lower triangular matrix:\n");

    for(i=0; i<row; i++)

    {

        for(j=0; j<col; j++)

        {

            if(i>=j)

            {

                printf("%d ",a[i][j]);

            }

            else

            {

                printf("0 ");

            }

        }

        printf("\n");

    }

    return 0;

}

//8. Write a program in C to print or display an upper triangular matrix.

#include<stdio.h>

int main()

{

    int i,j,row,col;

    int a[10][10];

    printf("Enter the row:");

    scanf("%d",&row);

    printf("Enter the column:");

    scanf("%d",&col);

    printf("Enter the row %d col:\n",row\*col);

    for(i=0; i<row; i++)

    {

        for(j=0; j<col; j++)

        {

            scanf("%d",&a[i][j]);

        }

    }

    printf("Display the lower triangular matrix:\n");

    for(i=0; i<row; i++)

    {

        for(j=0; j<col; j++)

        {

            if(j>=i)

            {

                printf("%d ",a[i][j]);

            }

            else

            {

                printf("0 ");

            }

        }

        printf("\n");

    }

    return 0;

}

//9. Write a program in C to accept a matrix and determine whether it is a sparse matrix.

#include<stdio.h>

int main()

{

    int i,j,row,col;

    int a[10][10],count=0;

    printf("Enter the row:");

    scanf("%d",&row);

    printf("Enter the column:");

    scanf("%d",&col);

    printf("Enter the row %d col:\n",row\*col);

    for(i=0; i<row; i++)

    {

        for(j=0; j<col; j++)

        {

            scanf("%d",&a[i][j]);

        }

    }

    for(i=0; i<row; i++)

    {

        for(j=0; j<col; j++)

        {

            if(a[i][j]==0)

            {

                count++;

            }

        }

    }

    if(count>(i\*j)/2)

    {

        printf("Sparse Matrix:");

    }

    else

    {

        printf("Not Sparse Matrix:");

    }

    return 0;

}

//10. Write a program in C to find the row with maximum number of 1s.

#include<stdio.h>

int main()

{

    int max[4][4] = {{0,0,0,1},

                    {0,1,1,1},

                    {1,1,1,1},

                    {0,0,0,0}};

    int max\_count=0,temp=-1;

    int i,j;

    for(i=0; i<4; i++)

    {

        int count1=0;

        for(j=0; j<4; j++)

        {

            if(max[i][j]==1)

            count1++;

        }

        if(count1>max\_count)

        {

            max\_count = count1;

            temp = i;

        }

    }

    printf("Index of row with maximum 1's is %d",temp);

}