```
// 2D ARRAY
// Author - Harsh Dixit

#include<iostream>
using namespace std;
int main()
{
	int arr[3][3] = {{1,2,3},{4,5,6},{7,8,9}}; // Intialization
	int row = 3;
	int col = 3;

	for(int i = 0; i < row; i++)
	{
		for(int j = 0; j < col; j++)
		{
			cout << arr[i][j] << " " ;
		}
		cout << endl;
	}

	cout << "Element at row 1 and  col 2 is:" << arr[1][2];  // Access of elements in 2d array
}

// -----------------------------------------------------------------------------------------------

// Program to print 2d array (column-wise)

// This prog also work for both transpose of square and non-square matrix

#include<iostream>
using namespace std;


void columnwise(int arr[3][4] , int row ,int col)
{
	for(int i = 0; i < col; i++)
	{
		for(int j = 0; j < row; j++)
		{
			cout << arr[j][i] << "  " ;
		}
		cout << endl;
	}
}
int main()
{
	int arr[3][4] = {
	{1,2,3,4},
	{5,6,7,8},
	{9,10,11,12}
	};
	int row = 3;
	int col = 4;
```

```
        columnwise(arr,row,col);

        return 0;
}

// -------------------------------------------------------------------------------------------

// Program to take i/p elements in 2D array and printing them

#include<iostream>
using namespace std;

void inputin2Darray(int arr[3][3] , int row , int col)
{
        for(int i = 0; i < row; i++)
        {
                for(int j = 0; j < col; j++)
                {
                        cout << "Row:" << i << " " << "Col:" << j << " ";
                        cin >> arr[i][j];
                }
                cout << endl;
                }
}

int main()
{
        int arr[3][3];
        int row = 3;
        int col = 3;
        inputin2Darray(arr , row , col);

        // for printing
        for(int i = 0; i < row; i++)
        {
                for(int j = 0; j < col; j++)
                {
                        cout << arr[i][j] << " " ;
                }
                cout << endl;
        }
}

// --------------------------------------------------------------------------

// Program for linear search in 2D array

#include<iostream>
using namespace std;

int linearsearchin2DArray(int arr[3][4] , int row , int col ,int Target)
{
        int answer = 0;
        for(int i = 0; i < row; i++)
        {
                for(int j = 0; j < col; j++)
                {
                        if(arr[i][j] == Target)
                        {
                                answer++;
                        }
                }
        }
        return answer;
}

int main()
{
        int arr[3][4] = {
        {1,2,3,4},
        {5,6,7,8},
        {9,10,11,12}
        };
        int row = 3;
        int col = 4;
        int Target = 101;

        int ans = linearsearchin2DArray(arr,row,col,Target);

        if(ans == 1)
        {
                cout << "Element found.";
        }
        else
        {
                cout << "Not found..";
        }
        return 0;
}

// ----------------------------------------------------------------------

// Maximum element in 2D Array

#include<iostream>
using namespace std;

int MaximumIn2DArray(int arr[3][4] , int row , int col)
{
        int max = arr[0][0];
        for(int i = 0; i < row; i++)
        {
                for(int j = 0; j < col; j++)
                {
                        if(arr[i][j] > max)
                        {
                                max = arr[i][j];
                        }
                }
        }
        return max;
}

int main()
{
        int arr[3][4] = {
                {23,2,1,3},
                {4,6,9,89},
                {5,67,5,45}
        };
        int row = 3;
        int col = 4;

        int ans = MaximumIn2DArray(arr,row,col);
        cout <<"Maximum element in 2D Array is: " << ans;
        return 0;
}

// -----------------------------------------------------------------

// Minimum element in 2D Array

#include<iostream>
using namespace std;

int MinimumIn2DArray(int arr[3][4] , int row , int col)
{
        int min = arr[0][0];
        for(int i = 0; i < row; i++)
        {
                for(int j = 0; j < col; j++)
                {
                        if(arr[i][j] < min)
                        {
                                min = arr[i][j];
                        }
                }
        }
        return min;
}

int main()
{
        int arr[3][4] = {
                {23,2,1,3},
                {4,6,9,89},
                {5,67,5,45}
        };
        int row = 3;
        int col = 4;

        int ans = MinimumIn2DArray(arr,row,col);
        cout << "Minimum element in 2D Array is: " << ans;
        return 0;
}

// -------------------------------------------------------------

// Print column wise sum in 2D array

#include<iostream>
using namespace std;

void ColumnWiseSum(int arr[3][4] , int row , int col)
{
        for(int i = 0; i < col; i++)
        {
                int sum = 0;
                for(int j = 0; j < row; j++)
                {
                        sum = sum + arr[j][i];
                }
                cout << "Sum of column " << i << " is: " << sum << endl;;
        }
}

int main()
{
        int arr[3][4] = {
        {1,2,3,4},
        {5,6,7,8},
        {9,10,11,12}
        };
        int row = 3;
        int col = 4;

        ColumnWiseSum(arr,row,col);
        return 0;
}

// ------------------------------------------------------------------

// Row Wise Sum in 2D Array

#include<iostream>
using namespace std;

void RowWiseSum(int arr[3][4] , int row , int col)
{
        for(int i = 0; i < row; i++)
        {
                int sum = 0;
                for(int j = 0; j < col; j++)
                {
                        sum = sum + arr[i][j];
                }
                cout << "Sum of row" << i << " is: " << sum << endl;;
        }
}

int main()
{
        int arr[3][4] = {
        {1,2,3,4},
        {5,6,7,8},
        {9,10,11,12}
        };
        int row = 3;
        int col = 4;

        RowWiseSum(arr,row,col);
        return 0;
}

// -------------------------------------------------------------------

// Print diagonal sum in 2D array

#include<iostream>
using namespace std;

void DiagonalSum(int arr[3][3] , int row , int col)
{
        int sum = 0;
        for(int i = 0; i < row; i++)
        {
                        sum = sum + arr[i][i];
        }
        cout << "Sum of diagonal is: "   << sum << endl;
}

int main()
{
        int arr[3][3] = {
        {1,2,3},
        {5,6,7},
        {9,10,11}
        };
        int row = 3;
        int col = 3;

        DiagonalSum(arr,row,col);
        return 0;
}
```