**LAB SESSION 5**

**Question: Find the sum of all elements in a 2D array.**

**Code:**

#include <iostream>

using namespace std;

int startlab5()

{

    cout << "Name: Saad Ali Khan(SE-23083)" << endl;

    cout << "Start of Lab 05" << endl;

    return 0;

}

int l5q1()

{

    int matrix[3][3] = {

        {1, 2, 3},

        {4, 5, 6},

        {7, 8, 9}};

    int sum = 0;

    for (int i = 0; i < 3; ++i)

    {

        for (int j = 0; j < 3; ++j)

        {

            sum += matrix[i][j];

        }

    }

    cout << "Sum of all elements: " << sum << endl;

    return 0;

}

int main()

{

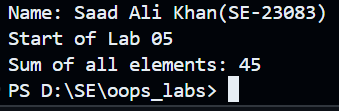
    startlab5();

    l5q1();

    return 0;

}

**OUTPUT:**

****

**Question: Calculate the transpose of a given matrix.**

**Code:**

#include <iostream>

using namespace std;

int startlab5()

{

    cout << "Name: Saad Ali Khan(SE-23083)" << endl;

    cout << "Lab 05" << endl;

    return 0;

}

int l5q2()

{

    int matrix[3][3] = {

        {1, 2, 3},

        {4, 5, 6},

        {7, 8, 9}};

    int transpose[3][3];

    for (int i = 0; i < 3; ++i)

    {

        for (int j = 0; j < 3; ++j)

        {

            transpose[j][i] = matrix[i][j];

        }

    }

    cout << "Transpose of the matrix:" << endl;

    for (int i = 0; i < 3; ++i)

    {

        for (int j = 0; j < 3; ++j)

        {

            cout << transpose[i][j] << " ";

        }

        cout << endl;

    }

    return 0;

}

int main()

{

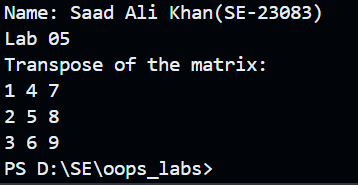
    startlab5();

    l5q2();

    return 0;

}

**Output:**

****

**Question: Check if a matrix is symmetric or not.**

**Code:**

#include <iostream>

using namespace std;

int startlab5()

{

    cout << "Name: Saad Ali Khan(SE-23083)" << endl;

    cout << "Lab 05" << endl;

    return 0;

}

bool isSymmetric(int matrix[3][3])

{

    for (int i = 0; i < 3; ++i)

    {

        for (int j = 0; j < 3; ++j)

        {

            if (matrix[i][j] != matrix[j][i])

            {

                return false;

            }

        }

    }

    return true;

}

int l5q3()

{

    int matrix[3][3] = {

        {1, 2, 3},

        {2, 5, 6},

        {3, 6, 9}};

    if (isSymmetric(matrix))

    {

        cout << "The matrix is symmetric." << endl;

    }

    else

    {

        cout << "The matrix is not symmetric." << endl;

    }

    return 0;

}

int main()

{

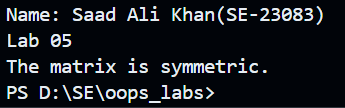
    startlab5();

    l5q3();

    return 0;

}

**Output:**

****

**Question: Multiply two matrices and print the result.**

**Code:**

#include <iostream>

using namespace std;

int startlab5()

{

    cout << "Name: Saad Ali Khan(SE-23083)" << endl;

    cout << "Lab 05" << endl;

    return 0;

}

int l5q4()

{

    int matrix1[2][3] = {

        {1, 2, 3},

        {4, 5, 6}};

    int matrix2[3][2] = {

        {7, 8},

        {9, 10},

        {11, 12}};

    int result[2][2] = {0};

    for (int i = 0; i < 2; ++i)

    {

        for (int j = 0; j < 2; ++j)

        {

            for (int k = 0; k < 3; ++k)

            {

                result[i][j] += matrix1[i][k] \* matrix2[k][j];

            }

        }

    }

    cout << "Resultant matrix after multiplication:" << endl;

    for (int i = 0; i < 2; ++i)

    {

        for (int j = 0; j < 2; ++j)

        {

            cout << result[i][j] << " ";

        }

        cout << endl;

    }

    return 0;

}

int main()

{

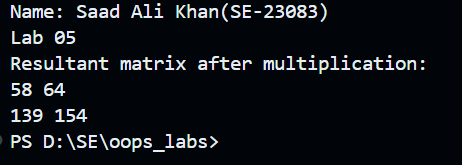
    startlab5();

    l5q4();

    return 0;

}

**Output:**

****

**Question: Find the largest element in each row of a matrix.**

**Code:**

#include <iostream>

#include <algorithm>

using namespace std;

int startlab5()

{

    cout << "Name: Saad Ali Khan(SE-23083)" << endl;

    cout << "Lab 05" << endl;

    return 0;

}

int l5q5()

{

    int matrix[3][3] = {

        {1, 2, 3},

        {4, 5, 6},

        {7, 8, 9}};

    for (int i = 0; i < 3; ++i)

    {

        int maxElement = \*max\_element(matrix[i], matrix[i] + 3);

        cout << "Largest element in row " << i + 1 << ": " << maxElement << endl;

    }

    return 0;

}

int main()

{

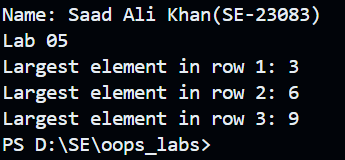
    startlab5();

    l5q5();

    return 0;

}

**Output:**

****