

SCHOOL OF MECHANICAL & MANUFACTURING ENGINERRING

NUST

Department of Mechanical Engineering

CS-114 – Fundamentals of Programming

LAB MANUAL #09

**Course Instructor**: Dr Khawaja Fahd Iqbal

**Lab Instructor**: Muhammad Affan

**Student Name**: Muhammad Usman Abdullah

**Section**: B

**CMS ID**: 461513

**DATE**: 12-December-2023

LAB #09 (LAB TASK)

TASK 1:

#include<iostream>

using namespace std;

int main() {

int array[3][3];

for(int i = 0; i < 3; i++) {

for(int j = 0; j < 3; j++) {

cout << "Enter element for 3D Array: ";

cin >> array[i][j];

}

}

int leftDiagonalSum = 0;

int rightDiagonalSum = 0;

for(int i = 0; i < 3; i++) {

leftDiagonalSum += array[i][i];

rightDiagonalSum += array[i][2 - i];

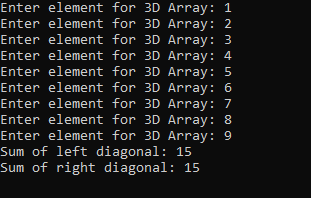
}

cout << "Sum of left diagonal: " << leftDiagonalSum << endl;

cout << "Sum of right diagonal: " << rightDiagonalSum << endl;

return 0;

}



TASK 2:

#include<iostream>

#include<iomanip>

using namespace std;

void addArrays(int array1[3][3], int array2[3][3], int result[3][3]) {

for(int i = 0; i < 3; i++) {

for(int j = 0; j < 3; j++) {

result[i][j] = array1[i][j] + array2[i][j];

}

}

}

int main() {

int array1[3][3]={{1,2,3},{4,5,6},{7,8,9}};

int array2[3][3]={{1,1,1},{3,3,3},{5,5,5}};

int result[3][3];

addArrays(array1, array2, result);

cout<<"Result array after addition: "<< endl;

for(int i = 0; i < 3; i++) {

for(int j = 0; j < 3; j++) {

cout<<setw(3)<<result[i][j];

}

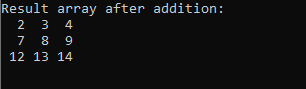
cout<<endl;

}

return 0;

}

OUTPUT:



TASK 3:

#include<iostream>

#include<iomanip>

using namespace std;

void addArrays(int array1[3][3], int array2[3][3], int result[3][3]) {

for(int i = 0; i < 3; i++) {

for(int j = 0; j < 3; j++) {

result[i][j] = array1[i][j] + array2[i][j];

}

}

}

void transposeMatrix(int matrix[3][3], int transpose[3][3]) {

for(int i = 0; i < 3; i++) {

for(int j = 0; j < 3; j++) {

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

}

}

}

int main() {

int array1[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};

int array2[3][3] = {{2, 4, 6}, {8, 10, 12}, {14, 16, 18}};

int result[3][3], transpose[3][3];

addArrays(array1, array2, result);

// Print the result array in matrix form

cout << "Result array after addition: " << endl;

for(int i = 0; i < 3; i++) {

for(int j = 0; j < 3; j++) {

cout << setw(3) << result[i][j];

}

cout << endl;

}

transposeMatrix(result, transpose);

cout << "Transpose of the result array: " << endl;

for(int i = 0; i < 3; i++) {

for(int j = 0; j < 3; j++) {

cout << setw(3) << transpose[i][j];

}

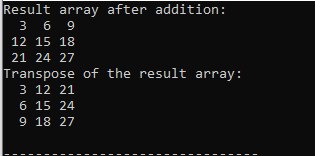
cout << endl;

}

return 0;

}

OUTPUT:



TASK 4:

#include<iostream>

#include<iomanip>

using namespace std;

void multiplyMatrices(int matrix1[3][3], int matrix2[3][3], int product[3][3]) {

for(int i = 0; i < 3; i++) {

for(int j = 0; j < 3; j++) {

product[i][j] = 0;

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

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

}

}

}

}

int main() {

int matrix1[3][3] = {{2,3,4}, {3,5,6}, {4,5,3}};

int matrix2[3][3] = {{1,2,1}, {-1,2,1}, {3,2,1}};

int product[3][3];

multiplyMatrices(matrix1, matrix2, product);

cout << "Product of the two matrices: " << endl;

for(int i = 0; i < 3; i++) {

for(int j = 0; j < 3; j++) {

cout<<setw(3)<< product[i][j];

}

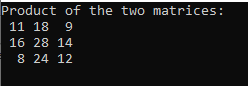
cout << endl;

}

return 0;

}

OUTPUT:



TASK 5:

#include<iostream>

#include<iomanip>

using namespace std;

void printMultiplicationTable(int n, int multiplier = 1) {

if(multiplier > n) {

return;

}

cout<<setw(3)<< n << " x " <<setw(2)<< multiplier << " = " << n \* multiplier << endl;

printMultiplicationTable(n, multiplier + 1);

}

int main() {

int number = 15;

printMultiplicationTable(number);

return 0;

}

OUTPUT:

