

National University of Science and technology

(NUST)

**CS-114 - Fundamental of Programming**

**Lab Manual # 09**

**Course Instructor: Dr Jawad Khan**

**Lab Instructor: Muhammad Affan**

**Student name:** Hashir Tariq

**QALAM ID:** 471240

**Date: 12/12/2023**

**Lab Tasks**

**Task No 1:**

Make 2D Array in C++ and print left diagonal and right diagonal sum of a 3x3 matrix.

**Code:**

#include <iostream>

using namespace std;

int main(){

int n,m,sum,sum1;

cout<<"Enter the Rows of the array:";

cin>>n;

cout<<"Enter the Columns of the array:";

cin>>m;

int arr[n][m];

for(int i=1;i<=n;i++){

for(int j=1;j<=m;j++){

cout<<"Enter element of "<<i<<","<<j<<":";

cin>>arr[i][j];

}

}

for(int i=1;i<=n;i++){

for(int j=1;j<=m;j++){

if (i==j){

sum=sum+arr[i][j];

}

else if(i+j==n){

sum1=sum1+arr[i][j];

}

}

}

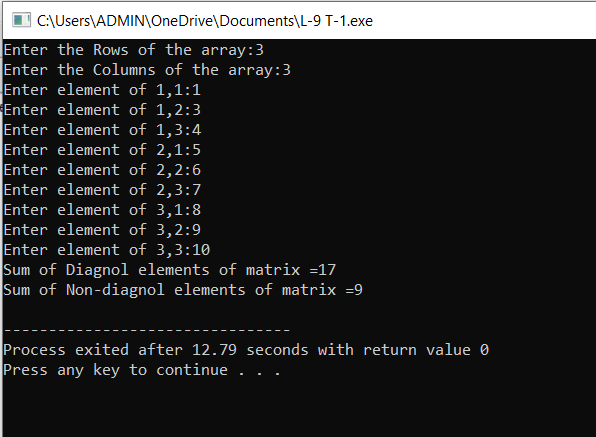
cout<<"Sum of Diagnol elements of matrix ="<<sum<<endl;

cout<<"Sum of Non-diagnol elements of matrix ="<<sum1<<endl;

return 0;

}

**Output:**

****

**Task No 2:**

Write a function to add two 2D arrays of size 3x3.

**Code:**

#include <iostream>

using namespace std;

void matrixSum() {

int n = 3, m = 3;

int a1[n][m], a2[n][m], sum[n][m];

for (int i = 1; i <= 3; i++) {

for (int j = 1; j <= 3; j++) {

cout << "Enter the Elements of First Array:" << i << "," << j << ":";

cin >> a1[i][j];

}

}

for (int i = 1; i <= 3; i++) {

for (int j = 1; j <= 3; j++) {

cout << "Enter the Elements of Second Array:" << i << "," << j << ":";

cin >> a2[i][j];

}

}

for (int i = 1; i <= 3; i++) {

for (int j = 1; j <= 3; j++) {

sum[i][j] = a1[i][j] + a2[i][j];

}

}

for (int i = 1; i <= 3; i++) {

for (int j = 1; j <= 3; j++) {

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

}

cout << endl;

}

}

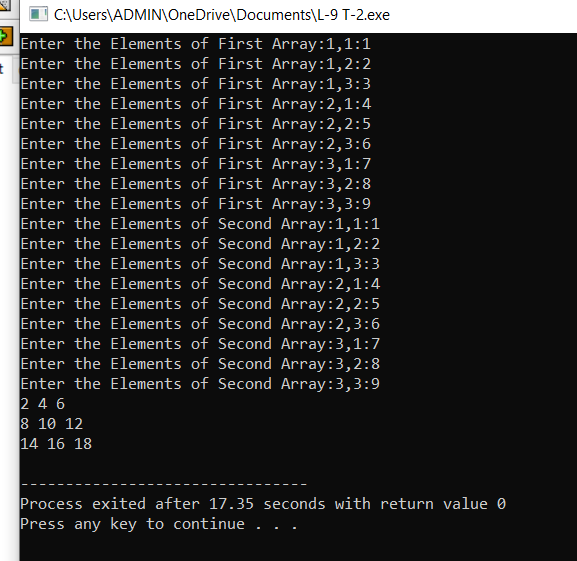
int main() {

matrixSum();

return 0;

}

**Output:**

****

**Task No 3:**

Using 2D arrays in C++, take transpose of a 3x3 matrix. Make a transpose function.

**Code:**

#include<iostream>

using namespace std;

int main(){

int arr1[3][3], transpose[3][3];

int i,j;

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

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

cout<<"Enter Value of Element "<<j<<","<<i<<":";

cin>>arr1[i][j];

}

}

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

cout<<endl;

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

cout<<arr1[i][j]<<" ";

}

}

cout<<endl;

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

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

transpose[j][i]=arr1[i][j];

}

}

cout<<endl<<"Transpose of Given Matrix is: "<<endl;

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

cout<<endl;

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

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

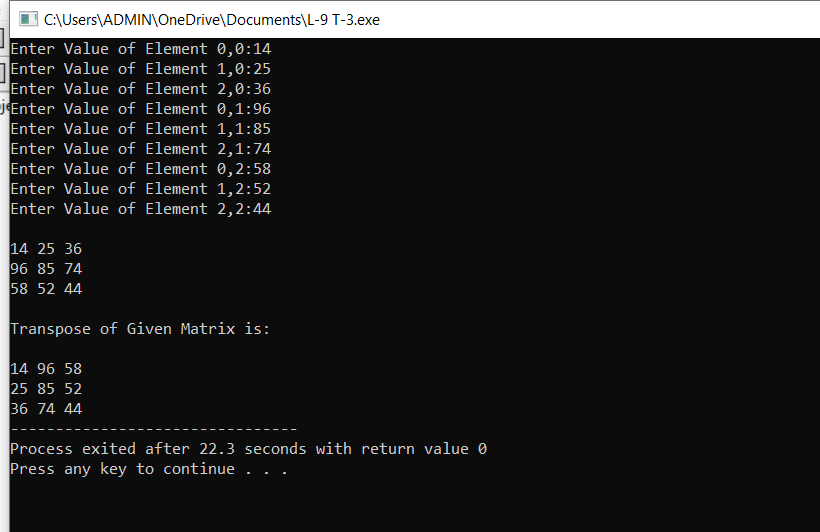
}

}

return 0;

}

**Output:**

****

**Task No 4:**

Using 2D arrays in C++, implement 3x3 matrix multiplication. Make a function.

**Code:**

#include <iostream>

using namespace std;

int main(){

int x1[3][3], x2[3][3], multiple[3][3];

int i, j;

cout<<"Enter the Values for Array 1: "<<endl;

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

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

cout<<"Enter Value of Element("<<j<<","<<i<<"):";

cin>>x1[i][j];

}

}

cout<<endl<<"Array 1 Filled! Now Input Array 2: "<<endl;

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

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

cout<<"Enter Value of Element ("<<j<<","<<i<<"): ";

cin>>x2[i][j];

}

}

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

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

multiple[i][j] = 0;

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

multiple[i][j] += x1[i][k] \* x2[k][j];

}

}

}

cout<<endl<<"Multiple of Given Array is: "<<endl;

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

cout<<endl;

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

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

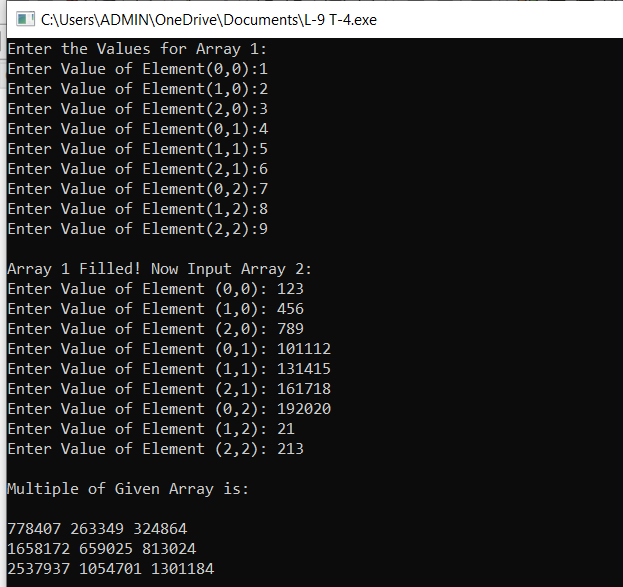
}

}

return 0;

}

**Output:**

****

**Task No 5:**

Print the multiplication table of 15 using recursion

**Code:**

#include <iostream>

using namespace std;

void multiplication(int number, int multiplier = 1) {

if (multiplier <= 10) {

int result = number \* multiplier;

cout << number << " x " << multiplier << " = " << result << endl;

multiplication(number, multiplier + 1);

}

}

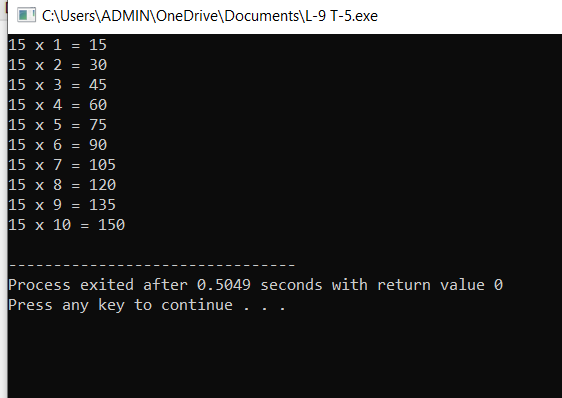
int main() {

multiplication(15);

return 0;

}

**Output:**

****

**Home Tasks**

**Home Task No 1:**

Write a C++ program to take inverse of a 3x3 matrix using its determinant and adjoint.

**Code:**

#include <iostream>

using namespace std;

int main() {

float matrix[3][3];

cout << "Enter the elements of the 3x3 matrix:" << endl;

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

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

cout<<"Enter the element of 3x3 matrix("<<i<<","<<j<<"):";

cin >> matrix[i][j];

}

}

//Taking a 3 by 3 matrix as input from user

cout << "The entered matrix is equal to :" << endl;

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

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

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

cout << endl; //Outputting the entered 3 by 3 matrix

}

float det = matrix[0][0] \* (matrix[1][1] \* matrix[2][2] - matrix[2][1] \* matrix[1][2]) -

matrix[0][1] \* (matrix[1][0] \* matrix[2][2] - matrix[2][0] \* matrix[1][2]) +

matrix[0][2] \* (matrix[1][0] \* matrix[2][1] - matrix[2][0] \* matrix[1][1]);

if (det == 0) { //Taking the dterminant of matrix and checking if det is equal to zero

cout << "The matrix is singular and cannot be inverted!" << endl;

}

else{

//If matrix is not singular than taking the adverse of the same matrix

float adj[3][3];

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

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

adj[i][j] = (matrix[(j + 1) % 3][(i + 1) % 3] \* matrix[(j + 2) % 3][(i + 2) % 3] -

matrix[(j + 1) % 3][(i + 2) % 3] \* matrix[(j + 2) % 3][(i + 1) % 3]);

float inv[3][3];

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

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

inv[i][j] = adj[i][j] / det; //applying the formula for inverse of matrix

cout << "The inverse of the matrix is:" << endl; //Outputting the inverse of matrix

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

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

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

cout << endl;

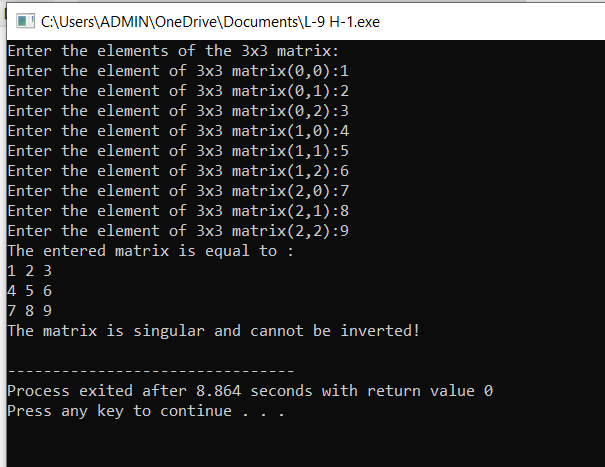
}

}

return 0;

}

**Output:**

****