School Of Mechanical & Manufacturing Engineering, NUST

Department of Mechanical Engineering

CS-114 - Fundamental of Programing

Lab Manual # 09

**Course Instructor:** Dr Khwaja Fahad Iqbal

**Lab Instructor:** Muhammad Affan

**Student Name: ZORAIZ ABBAS**

**CMS ID: 454752**

**DATE:**

**12 DEC 2023**

Fundamental of Programing

**TASK 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 arr1[3][3]={{3,2,1},{6,5,4},{9,8,7}};

int leftsum=0,rightsum=0;

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

leftsum+=arr1[i][i];

}

rightsum=arr1[0][2]+arr1[1][1]+arr1[2][0];

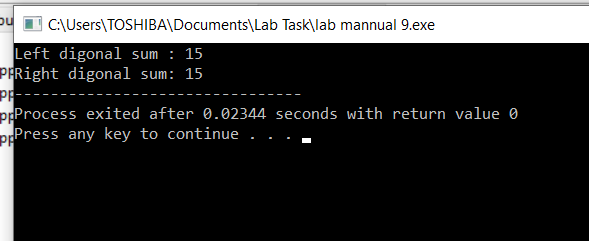
cout<<"Left digonal sum : "<<leftsum<<endl;

cout<<"Right digonal sum: "<<rightsum;

return 0;

}

**RESULT:**



**TASK 2:**

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

**CODE:**

#include<iostream>

using namespace std;

void sumofarrays(int arr1[3][3] , int arr2[3][3] ,int sum[3][3]){

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

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

sum[i][j] = arr1[i][j] + arr2[i][j];

} }

}

int main(){

int arr1[3][3]={{3,2,3},{4,5,6},{7,8,9}};

int arr2[3][3]={{7,8,8},{8,5,4},{3,5,1}};

int sum[3][3];

sumofarrays(arr1,arr2,sum);

cout<<"The sum of two arrays is: "<<endl;

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

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

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

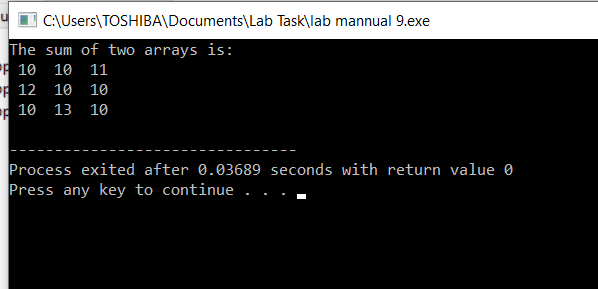
}

cout<<endl;

}

}

**RESULT:**

****

**TASK 3:**

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

**CODE:**

#include<iostream>

using namespace std;

void transpose(int arr1[3][3], int invert[3][3]){

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

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

invert[j][i]=arr1[i][j];

}

}

}

int main(){

int arr1[3][3]={{1,2,1},{4,5,8},{7,5,4}},result[3][3];

transpose(arr1,result);

cout<<"the transpose of a matrix is: "<<endl;

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

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

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

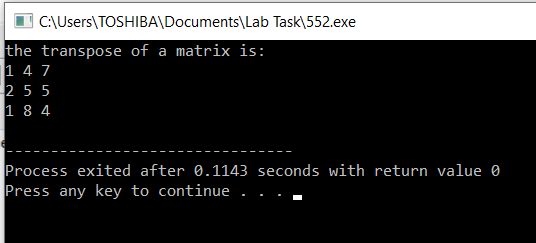
}

cout<<endl;

}

}

**RESULT:**



**TASK 4:**

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

**CODE:**

#include<iostream>

using namespace std;

void arraysmultiply(int arr1[3][3] , int arr2[3][3] , int result[3][3] ){

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

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

int res=0;

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

res+=arr1[i][k]\*arr2[k][j];

}

result[i][j]=res;

}

}

}

int main(){

int arr1[3][3],arr2[3][3],pro[3][3];

cout<<"Enter the elements of first matrix: "<<endl;

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

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

cin>>arr1[i][j];

}

}

cout<<"Enter the elements of second matrix: "<<endl;

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

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

cin>>arr2[i][j];

}

}

cout<<"Your entered matrix are: "<<endl;

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

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

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

} cout<<endl;

}

cout<<endl;

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

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

cout<<" "<<arr2[i][j]<<" ";

} cout<<endl;

}

arraysmultiply(arr1,arr2,pro);

cout<<"resultant matrix after multiplying: "<<endl;

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

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

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

}

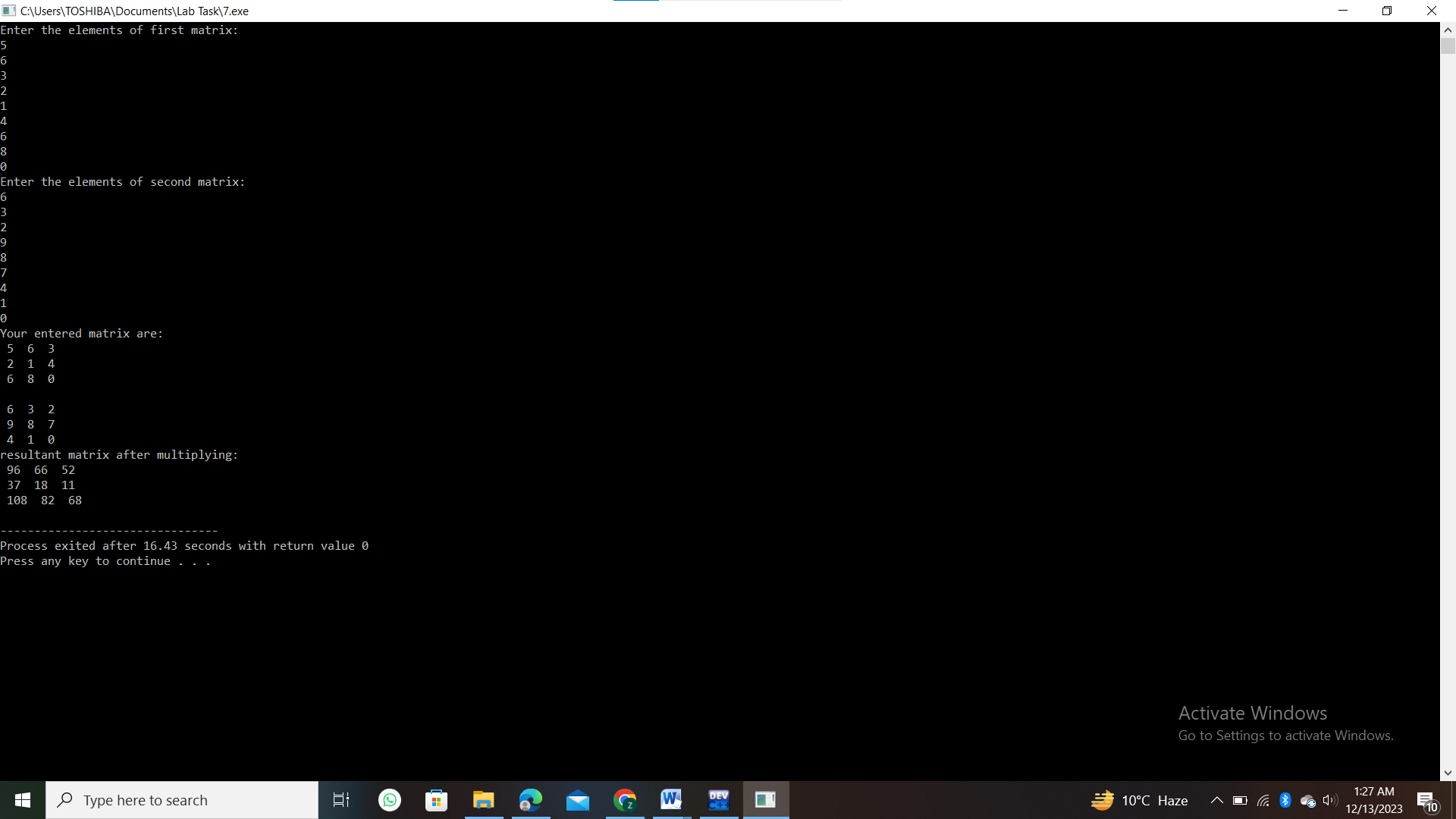
cout<<endl;

}

return 0;

}

**RESULT:**

****

**TASK 5:**

Print the multiplication table of 15 using recursion.

**CODE:**

#include<iostream>

using namespace std;

void table(int num,int res){

if(res>10){

return ;

}

cout<<" "<<num<<" \* "<<res<<" = "<<num\*res<<endl;

return table(num, res+1);

}

int main(){

int number=15,i,result;

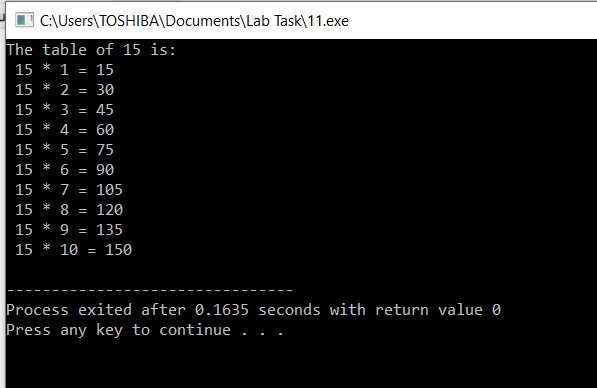
cout<<"The table of 15 is: "<<endl;

table(number,i);

return 0;

}

**RESULT:**

****

**HOMETASK:**

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

**CODE:**

#include<iostream>

using namespace std;

int main(){

int matrix1[3][3];

cout<<"enter the elements of a 3x3 matrix"<<endl;

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

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

cin>>matrix1[i][j];

} }

cout<<" your entered matrix is: "<<endl;

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

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

cout<<matrix1[i][j]<<" ";

}

cout<<endl; }

cout<<"the determinant of the given matrix is: "<<" ";

int det=0;

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

det=det+matrix1[0][i]\*(matrix1[1][(i+1)%3]\*matrix1[2][(i+2)%3]-matrix1[1][(i+2)%3]\*matrix1[2][(i+1)%3]);

}

cout<<det<<endl;

cout<<endl;

cout<<endl;

if(det==0){

cout<<"the inverse of this matrix is not possible because its determinant is zero "<<endl;

return 0;

}

cout<<"the adjoint is the matrix is "<<endl;

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

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

cout<<((matrix1[(j+1)%3][(i+1)%3]\*matrix1[(j+2)%3][(i+2)%3])-(matrix1[(j+1)%3][(i+2)%3]\*matrix1[(j+2)%3][(i+1)%3]))<<" ";

}

cout<<endl;

}

cout<<"the inverse of the given matrix is: "<<endl;

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

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

cout<<((matrix1[(j+1)%3][(i+1)%3]\*matrix1[(j+2)%3][(i+2)%3])-(matrix1[(j+1)%3][(i+2)%3]\*matrix1[(j+2)%3][(i+1)%3])/det)<<" ";

}

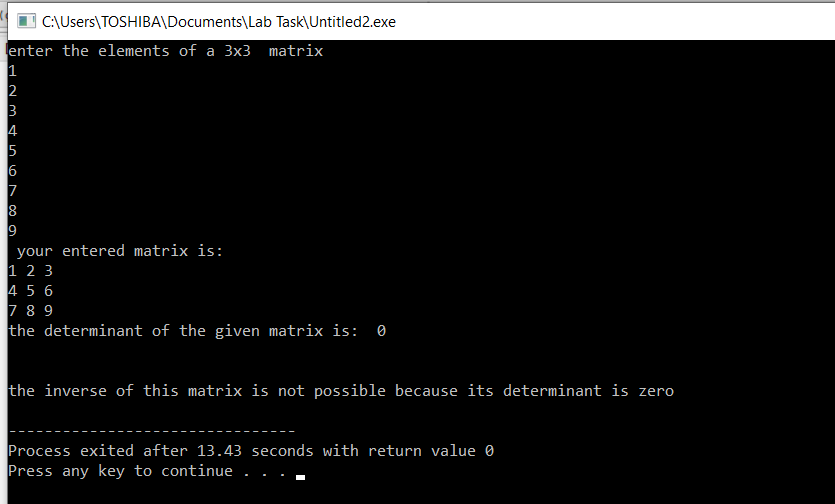
cout<<endl;

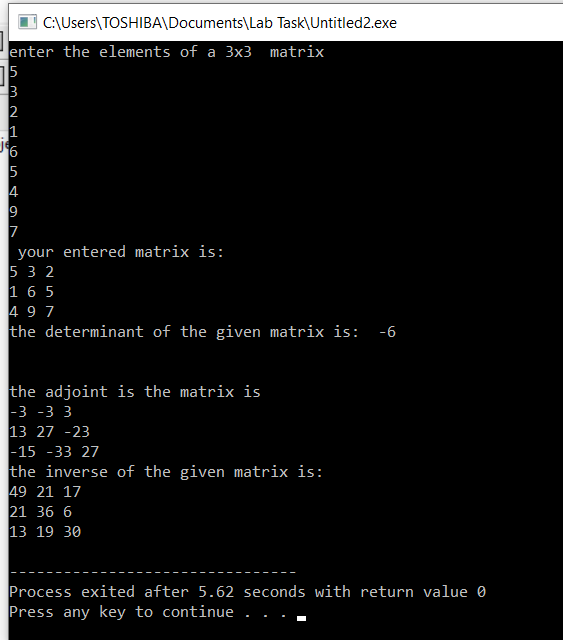
}

return 0;

}

**RESULT:**

****



**THE END**