LAB MANUAL 9 – Lab tasks and hometasks

NAME:Rameen Fatima

CMS:453962

CLASS:ME-15C

**LAB TASK 1:**

**CODE:**

#include<iostream>

using namespace std;

int main(){

int matrix[3][3] = {{1,2,3},{4,5,6},{7,8,9}},suml=0,sumr=0;

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

for (int i=0;i<=2;i++){

suml = suml + matrix[i][i];

if(i==2){

cout << matrix[i][i];

}

else cout << matrix[i][i]<<" + ";

}

cout << " = "<<suml<<endl<<"Sum of right diagonal:"<<endl;

for (int i = 0;i<=2;i++){

sumr = sumr + matrix[i][2-i];

if(i==2){

cout << matrix[i][2-i];

}

else cout << matrix[i][2-i]<<" + ";

}

cout << " = "<<sumr;

}

**OUTPUT:**

**A screenshot of a computer

Description automatically generated**

**LAB TASK 2:**

**CODE:**

#include<iostream>

using namespace std;

int matrixsum(int x,int y,int b[3][3],int a[3][3]){

return a[x][y]+b[x][y];

}

int main(){

int m1[3][3]={{1,2,3},{4,5,6},{7,8,9}},m2[3][3]={{10,11,12},{13,14,15},{16,17,18}};

cout << "The sum of the two matrices is:"<<endl;

for (int i=0;i<=2;i++){

for(int j=0;j<=2;j++){

cout <<" "<< matrixsum(i,j,m1,m2)<<" ";

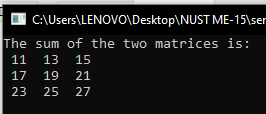
}

cout << endl;

}

}

**OUTPUT:**

****

**LAB TASK 3:**

**CODE:**

#include<iostream>

using namespace std;

int transpose(int a[3][3],int b[3][3],int x,int y){

b[x][y]=a[y][x];

return b[x][y];

}

int main(){

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

cout << "The matrix is:"<<endl;

for(int i=0;i<=2;i++){

for(int j=0;j<=2;j++){

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

}

cout << endl;

}

cout<< "The transpose is:"<<endl;

for(int i=0;i<=2;i++){

for(int j=0;j<=2;j++){

cout <<" "<< transpose(m,t,i,j)<<" ";

}

cout << endl;

}

}

**A screenshot of a computer

Description automatically generatedOUTPUT:**

**LAB TASK 4:**

**CODE:**

#include<iostream>

using namespace std;

int matrixproduct(int a[3][3],int b[3][3],int x,int y,int z){

int product = a[x][y]\*b[y][z];

return product;

}

int main(){

int m1[3][3]={{1,2,3},{4,5,6},{7,8,9}},m2[3][3]={{1,2,3},{4,5,6},{7,8,9}},p[3][3];

cout << "The first matrix is:"<<endl;

for (int i=0;i<=2;i++){

for (int j=0;j<=2;j++){

cout << m1[j][i]<< " ";

}

cout << endl;

}

cout << "The second matrix is:"<< endl;

for (int i=0;i<=2;i++){

for (int j=0;j<=2;j++){

cout << m2[j][i]<< " ";

}

cout << endl;

}

cout<< "The product of the 2 matrices is:"<<endl;

for(int k=0;k<=2;k++){

for(int i=0;i<=2;i++){

int sum = 0;

for (int j=0;j<=2;j++){

sum = sum + matrixproduct(m1,m2,i,j,k);

}

p[k][i] = sum;

cout << " "<<p[k][i]<<" ";

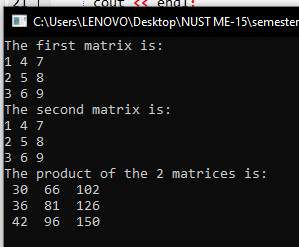
}

cout<< endl ;

}

}

**OUTPUT:**

****

**LAB TASK 5:**

**CODE:**

#include<iostream>

using namespace std;

int table(int y){

if(y<=0){

return 1;

}

else return 15\*y;

}

int main(){

int a=15,b=1;

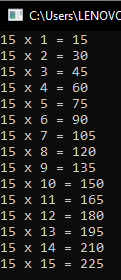
for(int i=1;i<=a;i++){

cout<<"15 x "<<i<<" = "<<table(i)<<endl;

}

}

**OUTPUT:**

****

**HOME TASK 1:**

**CODE:**

#include<iostream>

using namespace std;

int minor(int x[3][3],int row,int column){

int temp[4],a=0,m;

for(int i =0;i<=2;i++){

for(int j=0;j<=2;j++){

if (i==row || j==column)

continue;

else{

temp[a]=x[i][j];

a++;

}

}

}

m = (temp[0]\*temp[3])-(temp[1]\*temp[2]);

return m;

}

int transpose(float a[3][3],float b[3][3],int x,int y){

b[x][y]=a[y][x];

return b[x][y];

}

int power(int n,int p){

if(p>0){

return n\*power(n,p-1);

}

else

return 1;

}

int main(){

int a[3][3] = {{7,6,2},{8,9,4},{3,1,0}};

float inv[3][3],adj[3][3],cf,det;

det = a[0][0]\*((a[1][1]\*a[2][2]) - (a[1][2]\*a[2][1]))-a[0][1]\*((a[1][0]\*a[2][2]) - (a[1][2]\*a[2][0]))+a[0][2]\*((a[1][0]\*a[2][1]) - (a[1][1]\*a[2][0]));

cout<<"The matrix is:"<<endl;

for(int i=0;i<=2;i++){

for(int j=0;j<=2;j++){

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

}

cout << endl;

}

cout <<endl<< "The determinant is:"<<det<<endl<<endl;

for(int i=0;i<=2;i++){

for(int j=0;j<=2;j++){

cf = power(-1,i+j)\*minor(a,i,j);

adj[i][j]=cf;

}

}

cout<<"The inverse is:"<<endl;

for(int i=0;i<=2;i++){

for (int j=0;j<=2;j++){

inv[i][j] = transpose(adj,inv,i,j)/det;

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

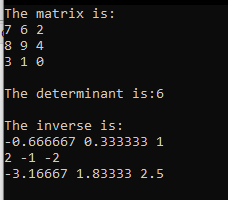
}

cout << endl;

}

}

**OUTPUT:**

****