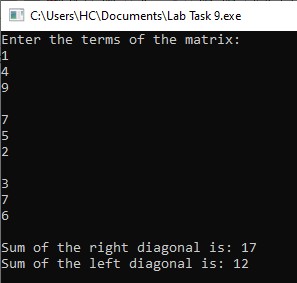
# TASK 1

int main() {

int Right=0,Left=0;

int diagonal[3][3];

cout<<"Enter the terms of the matrix:\n";

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

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

cin>>diagonal[i][j];

}

cout<<endl;

}

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

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

if (i==j) {

Left+=diagonal[i][j];

}

if (i+j==2) {

Right+=diagonal[i][j];

}

}

}

cout<<"Sum of the right diagonal is: "<<Right<<endl;

cout<<"Sum of the left diagonal is: "<<Left<<endl;

return 0;

}

# TASK 2

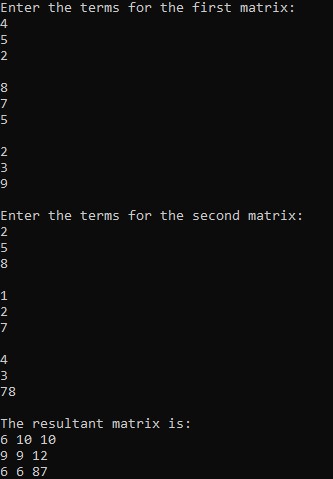
int main () {

int add[3][3];

int matrix1[3][3];

int matrix2[3][3];

cout<<"Enter the terms for the first matrix: "<<endl;

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

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

cin>>matrix1[i][j]; }

cout<<endl; }

cout<<"Enter the terms for the second matrix: "<<endl;

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

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

cin>>matrix2[i][j];

}

cout<<endl;

}

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

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

add[i][j]=matrix1[i][j]+matrix2[i][j];

}

}

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

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

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

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

}

cout<<endl;

}

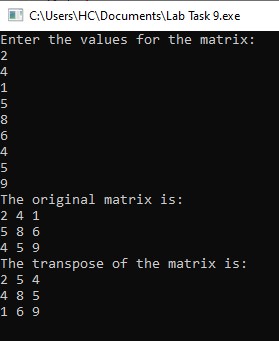
return 0;

}

# TASK 3

void transfunction(int matrix[3][3]) {

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<<"The original matrix is:\n";

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

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

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

}

cout<<endl; }

cout<<"The transpose of the matrix is:\n";

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

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

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

cout<<endl;

}}

int main () {

int matrix1[3][3];

cout<<"Enter the values for the matrix: "<<endl;

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

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

cin>>matrix1[i][j];

}}

transfunction(matrix1);

return 0;

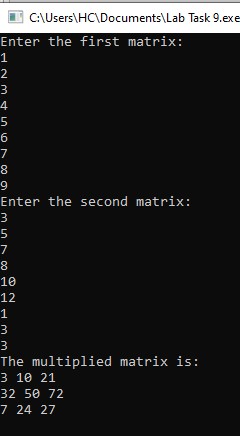
}

# TASK 4

void multi(int mat1[3][3], int mat2[3][3]) {

int mat3[3][3];

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

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

mat1[i][j]\*mat2[i][j];

mat3[i][j]=mat1[i][j]\*mat2[i][j];} }

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

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

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

cout<<mat3[i][j]<<" ";

}

cout<<endl;

}}

int main () {

int mat1[3][3],mat2[3][3];

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

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

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

cin>>mat1[i][j];

} }

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

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

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

cin>>mat2[i][j];

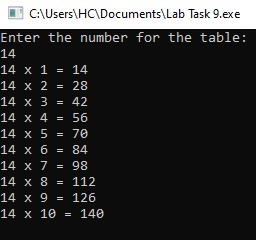
}}

multi(mat1,mat2);

return 0;

}

# TASK 5

void table(int n, int i=1) {

if (i<=10) {

cout<<n<<" x "<<i<<" = "<<n\*i<<endl;

table(n,i+1);

}

}

int main () {

int num;

cout<<"Enter the number for the table: "<<endl; cin>>num;

table(num);

return 0;

}

# HOMETASK 1

#include <iostream>

double determinant2x2(double a, double b, double c, double d) {

return a \* d - b \* c;

}

double determinant3x3(double matrix[3][3]) {

double a = matrix[0][0], b = matrix[0][1], c = matrix[0][2];

double d = matrix[1][0], e = matrix[1][1], f = matrix[1][2];

double g = matrix[2][0], h = matrix[2][1], i = matrix[2][2];

return a \* determinant2x2(e, f, h, i) - b \* determinant2x2(d, f, g, i) + c \* determinant2x2(d, e, g, h);

}

void inverse3x3(double matrix[3][3], double inverse[3][3]) {

double det = determinant3x3(matrix);

if (det == 0) {

std::cout << "Inverse does not exist as the determinant is zero.\n";

return;

}

double invDet = 1.0 / det;

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

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

int sign = ((i + j) % 2 == 0) ? 1 : -1;

inverse[j][i] = sign \* determinant2x2(matrix[(i + 1) % 3][(j + 1) % 3],

matrix[(i + 1) % 3][(j + 2) % 3],

matrix[(i + 2) % 3][(j + 1) % 3],

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

}

}

}

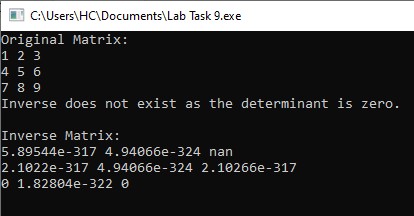
void displayMatrix(double matrix[3][3]) {

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

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

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

}

 std::cout << "\n";

}

}

int main() {

double matrix[3][3] = {{1, 2, 3},

{4, 5, 6},

{7, 8, 9}};

double inverse[3][3];

std::cout << "Original Matrix:\n";

displayMatrix(matrix);

inverse3x3(matrix, inverse);

std::cout << "\nInverse Matrix:\n";

displayMatrix(inverse);

return 0;

}