National University of Sciences and Technology (NUST)

Department of Mechanical Engineering (SMME)

Fundamentals of Programming (FOP)

Home Tasks

Lab Manual 9

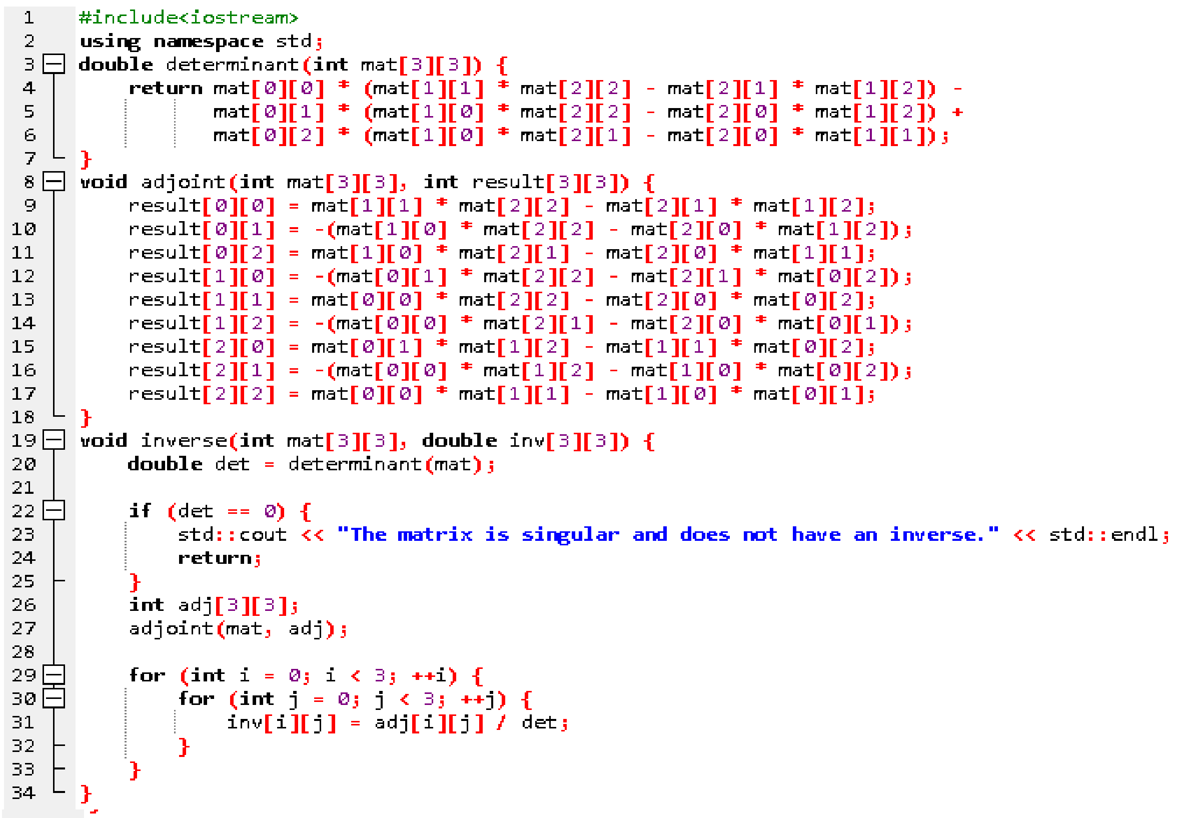
By

Muhammad Owais

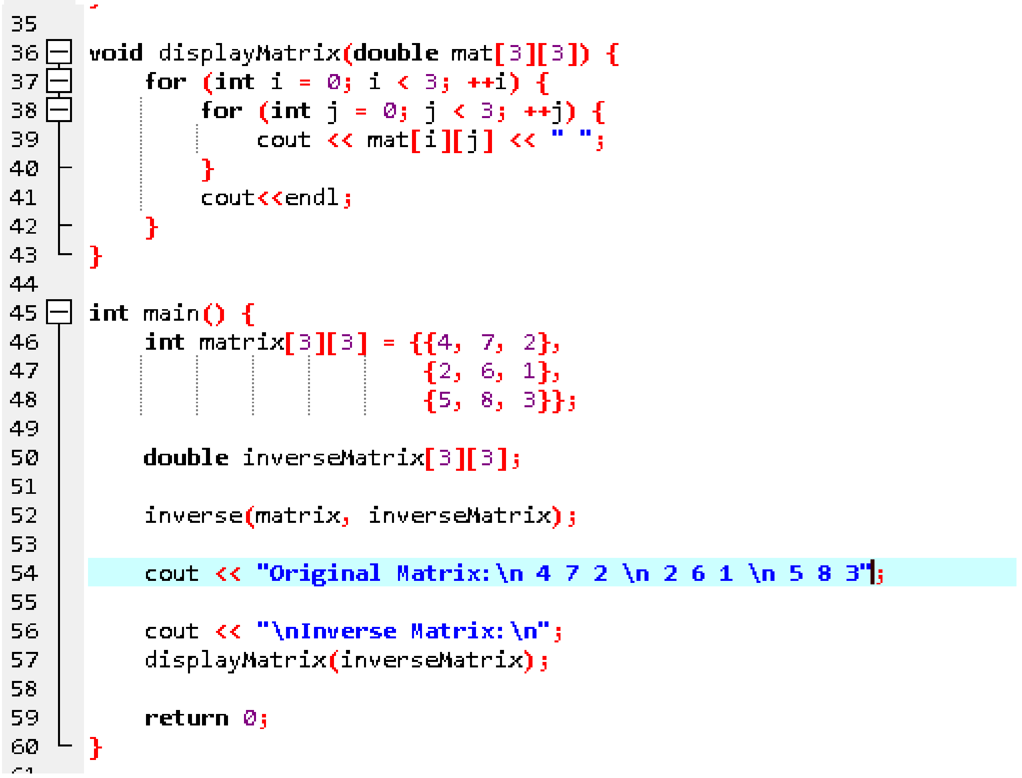
461359

Teacher: Sir Muhammad Affan

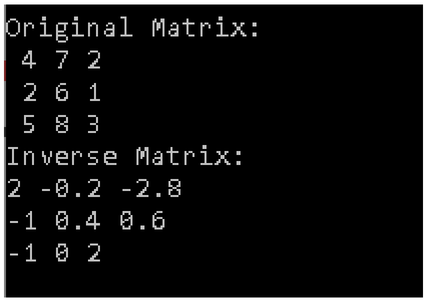
Home Task:



```cpp
#include<iostream>
using namespace std;
double determinant(int mat[3][3]) {
    return mat[0][0] * (mat[1][1] * mat[2][2] - mat[2][1] * mat[1][2]) -
           mat[0][1] * (mat[1][0] * mat[2][2] - mat[2][0] * mat[1][2]) +
           mat[0][2] * (mat[1][0] * mat[2][1] - mat[2][0] * mat[1][1]);
}
void adjoint(int mat[3][3], int result[3][3]) {
    result[0][0] = mat[1][1] * mat[2][2] - mat[2][1] * mat[1][2];
    result[0][1] = -(mat[1][0] * mat[2][2] - mat[2][0] * mat[1][2]);
    result[0][2] = mat[1][0] * mat[2][1] - mat[2][0] * mat[1][1];
    result[1][0] = -(mat[0][1] * mat[2][2] - mat[2][1] * mat[0][2]);
    result[1][1] = mat[0][0] * mat[2][2] - mat[2][0] * mat[0][2];
    result[1][2] = -(mat[0][0] * mat[2][1] - mat[2][0] * mat[0][1]);
    result[2][0] = mat[0][1] * mat[1][2] - mat[1][1] * mat[0][2];
    result[2][1] = -(mat[0][0] * mat[1][2] - mat[1][0] * mat[0][2]);
    result[2][2] = mat[0][0] * mat[1][1] - mat[1][0] * mat[0][1];
}
void inverse(int mat[3][3], double inv[3][3]) {
    double det = determinant(mat);

    if (det == 0) {
        std::cout << "The matrix is singular and does not have an inverse." << std::endl;
        return;
    }
    int adj[3][3];
    adjoint(mat, adj);

    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            inv[i][j] = adj[i][j] / det;
        }
    }
}
```



```cpp

void displayMatrix(double mat[3][3]) {
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            cout << mat[i][j] << " ";
        }
        cout<<endl;
    }
}

int main() {
    int matrix[3][3] = {{4, 7, 2},
                         {2, 6, 1},
                         {5, 8, 3}};

    double inverseMatrix[3][3];

    inverse(matrix, inverseMatrix);

    cout << "Original Matrix:\n 4 7 2 \n 2 6 1 \n 5 8 3";

    cout << "\nInverse Matrix:\n";
    displayMatrix(inverseMatrix);

    return 0;
}
```



```
Original Matrix:
 4 7 2
 2 6 1
 5 8 3
Inverse Matrix:
2 -0.2 -2.8
-1 0.4 0.6
-1 0 2
```