NATIONAL UNIVERSITY OF SCIENCES AND TEHNOLOGY

CS-114-FUNDAMENTAL OF PROGRAMING

LAB MANUAL # 9

- **NAME:** Muhammad Uzair

- **CLASS:** ME 15

- **SECTION:** B

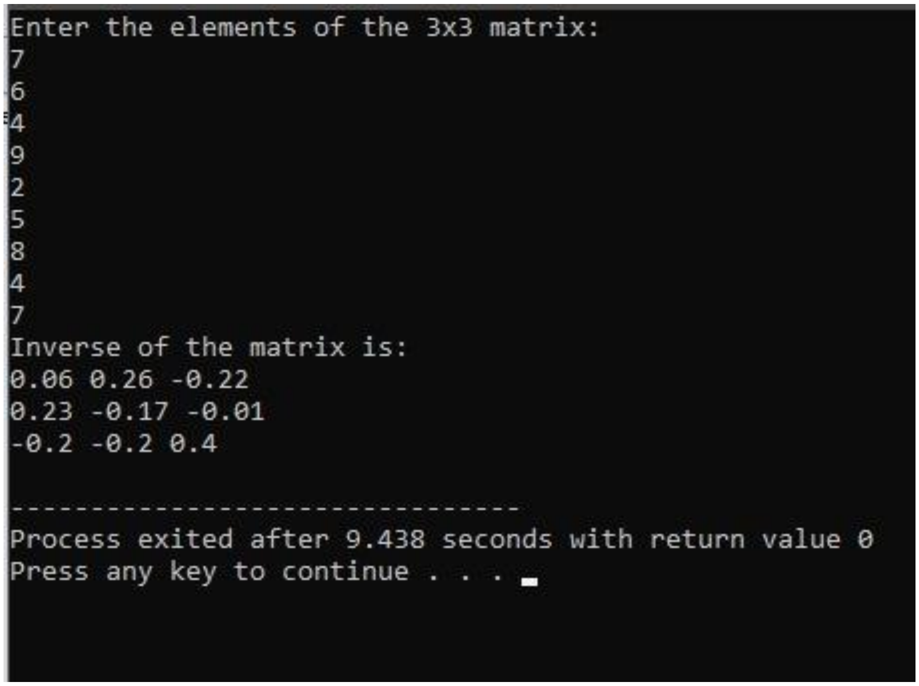
- **CMS ID:** 466092

TASK # 1

CODE:

```
#include <iostream>
using namespace std;

// Function to calculate the determinant of a 2x2 matrix
double determinant2x2(double a, double b, double c, double d) {
    return a * d - b * c;
}

// Function to calculate the determinant of a 3x3 matrix
double determinant3x3(double matrix[3][3]) {
    return matrix[0][0] * determinant2x2(matrix[1][1], matrix[1][2], matrix[2][1], matrix[2][2]) -
           matrix[0][1] * determinant2x2(matrix[1][0], matrix[1][2], matrix[2][0], matrix[2][2]) +
           matrix[0][2] * determinant2x2(matrix[1][0], matrix[1][1], matrix[2][0], matrix[2][1]);
}

// Function to calculate the adjoint of a 3x3 matrix
void adjoint3x3(double matrix[3][3], double adj[3][3]) {
    adj[0][0] = determinant2x2(matrix[1][1], matrix[1][2], matrix[2][1], matrix[2][2]);
    adj[0][1] = -determinant2x2(matrix[0][1], matrix[0][2], matrix[2][1], matrix[2][2]);
    adj[0][2] = determinant2x2(matrix[0][1], matrix[0][2], matrix[1][1], matrix[1][2]);

    adj[1][0] = -determinant2x2(matrix[1][0], matrix[1][2], matrix[2][0], matrix[2][2]);
    adj[1][1] = determinant2x2(matrix[0][0], matrix[0][2], matrix[2][0], matrix[2][2]);
    adj[1][2] = -determinant2x2(matrix[0][0], matrix[0][2], matrix[1][0], matrix[1][2]);

    adj[2][0] = determinant2x2(matrix[1][0], matrix[1][1], matrix[2][0], matrix[2][1]);
    adj[2][1] = -determinant2x2(matrix[0][0], matrix[0][1], matrix[2][0], matrix[2][1]);
    adj[2][2] = determinant2x2(matrix[0][0], matrix[0][1], matrix[1][0], matrix[1][1]);
}

// Function to calculate the inverse of a 3x3 matrix
void inverse3x3(double matrix[3][3], double inverse[3][3]) {
    double det = determinant3x3(matrix);

    if (det == 0) {
        cout << "Inverse does not exist as the determinant is zero." << endl;
        return;
    }

    double adj[3][3];
    adjoint3x3(matrix, adj);

    // Calculate the inverse by dividing each element of the adjoint by the determinant
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            inverse[i][j] = adj[i][j] / det;
        }
    }
}

// Function to display a 3x3 matrix
void displayMatrix(double matrix[3][3]) {
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            cout << matrix[i][j] << " ";
        }
        cout << endl;
    }
}

int main() {
    double 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) {
            cin >> matrix[i][j];
        }
    }

    double inverse[3][3];
    inverse3x3(matrix, inverse);

    cout << "Inverse of the matrix is:" << endl;
    displayMatrix(inverse);

    return 0;
}
```

RESULT:



```
Enter the elements of the 3x3 matrix:
7
6
4
9
2
5
8
4
7
Inverse of the matrix is:
0.06 0.26 -0.22
0.23 -0.17 -0.01
-0.2 -0.2 0.4

--------------------------------
Process exited after 9.438 seconds with return value 0
Press any key to continue . . .
```