NAME : ABDULLAH
CMS ID# : 461030
SECTION : ME(15-B)

HOME TASKS

TASK :

```
include <iostream>
#include <cmath>

using namespace std;

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

// Function to calculate the determinant of a 3x3 matrix
float determinant3x3(float 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(float matrix[3][3], float adj[3][3]) {
   adj[0][0] = determinant2x2(matrix[1][1], matrix[1][2], matrix[2][1], matrix[2][2]);
   adj[0][1] = -determinant2x2(matrix[1][0], matrix[1][2], matrix[2][0], matrix[2][2]);
   adj[0][2] = determinant2x2(matrix[1][0], matrix[1][1], matrix[2][0], matrix[2][1]);

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

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

```
    adj[2][1] = -determinant2x2(matrix[0][0], matrix[0][2], matrix[1][0], matrix[1][2]);
    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(float matrix[3][3], float inverse[3][3]) {
    float det = determinant3x3(matrix);

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

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

    // Calculate the inverse using the formula: inverse = adjoint / determinant
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            inverse[i][j] = adj[i][j] / det;
        }
    }
}

int main() {
    float 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];
        }
    }

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

    cout << "Inverse of the matrix:" << endl;
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            cout << inverse[i][j] << " ";
        }
        cout << endl;
    }

    return 0;
}
```