Fundamentals of Programming

Home Task: 9

Name: Malik Haseeb

Section: B

Roll no: 468333

Submitted to : Sir M. Affan

Task:1

Code:

```
#include <iostream>
using namespace std;
void adjoint(float array1[3][3], float adjt[3][3]) {

    adjt[0][0] = array1[1][1]*array1[2][2] - array1[2][1]*array1[1][2];
    adjt[0][1] = -(array1[1][0]*array1[2][2] - array1[2][0]*array1[1][2]);
    adjt[0][2] = array1[1][0]*array1[2][1] - array1[2][0]*array1[1][1];
    adjt[1][0] = -(array1[0][1]*array1[2][2] - array1[2][1]*array1[0][2]);
    adjt[1][1] = array1[0][0]*array1[2][2] - array1[2][0]*array1[0][2];
    adjt[1][2] = -(array1[0][0]*array1[2][1] - array1[2][0]*array1[0][1]);
    adjt[2][0] = array1[0][1]*array1[1][2] - array1[1][1]*array1[0][2];
    adjt[2][1] = -(array1[0][0]*array1[1][2] - array1[1][0]*array1[0][2]);
    adjt[2][2] = array1[0][0]*array1[1][1] - array1[1][0]*array1[0][1];
}

void showArray (float array1[3][3]) {
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            cout << array1[i][j] << "  ";
        }
        cout << endl;
    }
}
```

```
void inverse(float adjt[3][3], float det, double inv[3][3]) {

    if (det == 0) {
        cout << "Matrix can not be Singular." << endl;
        return;
    }

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



int main() {

    float array2[3][3];

    cout << "Enter the elements in array :" << endl;
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            cin >> array2 [i][j];
        }
    }

    cout << "Given array is :" << endl;
```

```
    cout << endl;
    for( int i = 0; i < 3; i++){
        for( int j = 0; j < 3; j++){
            cout<<array2[i][j]<<"  ";
        } cout << endl;
    }

    float adjArray[3][3];
    adjoint(array2, adjArray);

    cout << endl;
    cout << "The Adjoint of array is :" << endl;
    cout << endl;

    showArray(adjArray);

    float det=0;

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

            det = array2[0][0] * (array2[1][1] * array2[2][2] - array2[2][1] *
array2[1][2]) -
            array2[0][1] * (array2[1][0] * array2[2][2] - array2[2][0] * array2[1][2]) +
            array2[0][2] * (array2[1][0] * array2[2][1] - array2[2][0] * array2[1][1]);

        }
    }cout << endl;
```

```
        cout << "The determinant of array is = " << det << endl;
    cout << endl;

    double inverseArray [3][3];

    inverse(adjArray,det,inverseArray);

    cout << "The Inverse of the Array is:" <<endl;
    cout << endl;
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            cout << inverseArray [i][j] << "  ";
        }
        cout <<endl;
    }


    return 0;
}
```

Output:

```
Enter the elements in array :
1
2
3
4
5
6
7
8
9
Given array is :

1   2   3
4   5   6
7   8   9

The Adjoint of array is :

-3   6   -3
6   -12   6
-3   6   -3

The determinant of array is = 0

Matrix can not be Singular.
The Inverse of the Array is:

0   0   0
0   0   0
0   0   0
```