1. ***Write a C++ program to take inverse of a 3x3 matrix using its determinant and adjoint.***

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

double determinant(int array[3][3], int row, int col) {
    return array[(row + 1) % 3][(col + 1) % 3] * array[(row + 2) % 3][(col + 2) % 3]
     - array[(row + 1) % 3][(col + 2) % 3] * array[(row + 2) % 3][(col + 1) % 3];
}

double calcDeterminant(int array[3][3]) {
    double det = 0.0;
    for (int i = 0; i < 3; ++i) {
        det += array[0][i] * determinant(array, 0, i);
    }
    return det;
}

void adjoint(int array[3][3], double adj[3][3]) {
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            adj[j][i] = determinant(array, i, j);
            if ((i + j) % 2 != 0)
                adj[j][i] = -adj[j][i];
        }
    }
}

bool inversemyarray(int array[3][3], double inv[3][3]) {
    double det = calcDeterminant(array);
    if (det == 0) {
        cout << "array is singular, inverse does not exist." << endl;
        return false;
    }

    double adj[3][3];
    adjoint(array, adj);

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

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

int main() {
    int myarray[3][3] = {{7, 2, 3},
                         {4, 5, 6},
                         {7, 8, 9}};

    double inverse[3][3];

    if (inversemyarray(myarray, inverse)) {
        cout << "Inverse of the array is:" << endl;
        displaymyarray(inverse);
    }

    return 0;
}
```

```
Inverse of the array is:
0.166667 0.333333 0.166667
0.333333 -2.33333 -1.66667
0.166667 -2.33333 -1.5

Process returned 0 (0x0)   execution time : 0.078 s
Press any key to continue.
```