HOME TASK 1

#include <iostream>

#include <cmath>

using namespace std;

float det(float a, float b, float c, float d) {

return a \* d - b \* c;

}

float cf(int a, int b, const float mat[3][3]) {

int sign = (a + b) % 2 == 0 ? 1 : -1;

float minorMat[2][2];

int minorI = 0, minorJ = 0;

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

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

if (i != a && j != b) {

minorMat[minorI][minorJ++] = mat[i][j];

if (minorJ == 2) {

minorJ = 0;

++minorI;

}

}

}

}

return sign \* det(minorMat[0][0], minorMat[0][1], minorMat[1][0], minorMat[1][1]);

}

void adjoint(const float mat[3][3], float adj[3][3]) {

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

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

adj[i][j] = cf(j, i, mat);

}

}

}

void inverse(const float mat[3][3], float inv[3][3]) {

float dt = mat[0][0] \* cf(0, 0, mat)

- mat[0][1] \* cf(0, 1, mat)

+ mat[0][2] \* cf(0, 2, mat);

if (dt == 0) {

cout << "Matrix is singular. Inverse does not exist." << endl;

return;

}

float adj[3][3];

adjoint(mat, adj);

float inverseFac = 1.0 / dt;

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

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

inv[i][j] = adj[i][j] \* inverseFac;

}

}

}

void display(const float mat[3][3], const string& label) {

cout << label << " matrix:" << endl;

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

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

cout << mat[i][j] << " ";

}

cout << endl;

}

cout << endl;

}

int main() {

float mat[3][3]={{1,2,3},{1,2,3},{0,5,7}};

float inv[3][3];

display(mat, "Original");

inverse(mat, inv);

display(inv, "Inverse");

return 0;

}

The console is:

