

```

#include <iostream>
#include <GL/glut.h>
float ax = 10, ay = 10;
float bx = 1.1, by = 1.1;
float cx = 0.9, cy = 0.9;
float A = 0.05;
float m = 205, n = 204;
float P[66][3] = {
    {205,204},{221,188},{226,170},{221,143},{208,132},{214,118},{252,117},{264,107},{2
53,98},{216,96},{206,88},{204,53},{192,26},{180,20},{188,48},{185,75},{155,57},{121,61},{
84,81},{68,69},{53,76},{83,96},{70,110},{83,122},{57,140},{57,156},{94,135},{110,149},{14
1,165},{178,156},{200,145},{210,167},{206,203},{221,189},{212,167},{227,171},{211,166},{2
20,144},{201,145},{208,132},{177,131},{191,146},{203,133},{177,131},{206,86},{215,117},{2
44,97},{252,114},{253,98},{220,95},{206,85},{186,74},{204,53},{187,48},{193,25},{181,19},
{188,47},{184,74},{176,129},{96,81},{93,135},{183,75},{142,164},{122,61},{110,148},{181,1
04}
};
float right[3][3]{ {1,0,0},{0,1,0},{ax,0,1} };
float left[3][3]{ {1,0,0},{0,1,0},{-ax,0,1} };
float up[3][3]{ {1,0,0},{0,1,0},{0,-ay,1} };
float down[3][3]{ {1,0,0},{0,1,0},{0,ay,1} };
float zin[3][3]{ {bx,0,0},{0,by,0},{0,0,1} };
float zout[3][3]{ {cx,0,0},{0,cy,0},{0,0,1} };
float r[3][3]{ {cos(A),sin(A),0},{-sin(A),cos(A),0},{m * (1 - cos(A)) + n * sin(A),n * (1
- cos(A)) - m * sin(A),1} };
float r1[3][3]{ {cos(-A),sin(-A),0},{-sin(-A),cos(-A),0},{m * (1 - cos(-A)) + n * sin(-
A),n * (1 - cos(-A)) - m * sin(-A),1} };
float P1[66][3] = {
    {205,204},{221,188},{226,170},{221,143},{208,132},{214,118},{252,117},{264,107},{2
53,98},{216,96},{206,88},{204,53},{192,26},{180,20},{188,48},{185,75},{155,57},{121,61},{
84,81},{68,69},{53,76},{83,96},{70,110},{83,122},{57,140},{57,156},{94,135},{110,149},{14
1,165},{178,156},{200,145},{210,167},{206,203},{221,189},{212,167},{227,171},{211,166},{2
20,144},{201,145},{208,132},{177,131},{191,146},{203,133},{177,131},{206,86},{215,117},{2
44,97},{252,114},{253,98},{220,95},{206,85},{186,74},{204,53},{187,48},{193,25},{181,19},
{188,47},{184,74},{176,129},{96,81},{93,135},{183,75},{142,164},{122,61},{110,148},{181,1
04}
};
void display() {
    glClear(GL_COLOR_BUFFER_BIT);
    glFlush();
}
void start() {
    glClearColor(0, 0, 0, 0);
    gluOrtho2D(0, 800, 600, 0);
}
void Line(int x1, int y1, int x2, int y2) {
    glBegin(GL_LINES);
    glClearColor(255, 255, 255, 0);
    glVertex2f(x1, y1);
    glVertex2f(x2, y2);
    glLineWidth(2);
    glEnd();
}
void Poligono() {
    for (int i = 1; i < 66; i++) {
        Line(P1[i - 1][0], P1[i - 1][1], P1[i][0], P1[i][1]);
    }
    glFlush();
}

```

```

}
void trans(float M[3][3]) {
    glClear(GL_COLOR_BUFFER_BIT);
    for (int i = 0; i < 66; i++) {
        P1[i][0] = (P1[i][0] * M[0][0]) + (P1[i][1] * M[1][0]) + (P1[i][2] *
M[2][0]);
        P1[i][1] = (P1[i][0] * M[0][1]) + (P1[i][1] * M[1][1]) + (P1[i][2] *
M[2][1]);
        P1[i][2] = 1;
    }
    Poligono();
}
void key(unsigned char button, int x1, int y1) {
    switch (button) {
        case 'p':
            Poligono();
            break;
        case 'd':
            trans(right);
            glFlush();
            break;
        case 'a':
            trans(left);
            glFlush();
            break;
        case 'w':
            trans(up);
            glFlush();
            break;
        case 's':
            trans(down);
            glFlush();
            break;
        case '+':
            glClear(GL_COLOR_BUFFER_BIT);
            trans(zin);
            glFlush();
            break;
        case '-':
            glClear(GL_COLOR_BUFFER_BIT);
            trans(zout);
            glFlush();
            break;
        case 'q':
            glClear(GL_COLOR_BUFFER_BIT);
            trans(r);
            glFlush();
            break;
        case 'e':
            glClear(GL_COLOR_BUFFER_BIT);
            trans(r1);
            glFlush();
            break;
    }
}
int main(int argc, char* args[]) {
    glutInit(&argc, args);

```

```
glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE);
glutInitWindowPosition(650, 50);
glutInitWindowSize(800, 600);
glutCreateWindow("POLIGONO");
start();
glutDisplayFunc(display);
glutKeyboardFunc(key);
glutMainLoop();
return 0;
}
```