```
#include <stdio.h>
#include <stdarg.h>
#include <math.h>
#include <GL/glut.h>

int W = 800, H = 800;
double Ry = 0;
double Rx = 0;
void display();
void specialKeys();

void display() {
	glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
	glLoadIdentity();
	glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
	glRotatef(Rx, 1.0, 0.0, 0.0);
	glRotatef(Ry, 0.0, 1.0, 0.0);
	//FRONT
	glBegin(GL_POLYGON);
	glColor3f(1.0, 0.5, 0.0);
	glVertex3f(-0.3, -0.3, -0.3);
	glVertex3f(-0.3, 0.3, -0.3);
	glVertex3f(0.3, 0.3, -0.3);
	glVertex3f(0.3, -0.3, -0.3);
	glEnd();
	//BACK
	glBegin(GL_POLYGON);
	glColor3f(1.0, 1.0, 0.0);
	glVertex3f(0.3, -0.3, 0.3);
	glVertex3f(0.3, 0.3, 0.3);
	glVertex3f(-0.3, 0.3, 0.3);
	glVertex3f(-0.3, -0.3, 0.3);
	glEnd();
	//RIGHT
	glBegin(GL_POLYGON);
	glColor3f(1.0, 0.0, 1.0);
	glVertex3f(0.3, -0.3, -0.3);
	glVertex3f(0.3, 0.3, -0.3);
	glVertex3f(0.3, 0.3, 0.3);
	glVertex3f(0.3, -0.3, 0.3);
	glEnd();
	//LEFT
	glBegin(GL_POLYGON);
	glColor3f(0.0, 1.0, 0.0);
	glVertex3f(-0.3, -0.3, 0.3);
	glVertex3f(-0.3, 0.3, 0.3);
	glVertex3f(-0.3, 0.3, -0.3);
	glVertex3f(-0.3, -0.3, -0.3);
	glEnd();
	//UP
	glBegin(GL_POLYGON);
	glColor3f(0.0, 0.0, 1.0);
	glVertex3f(0.3, 0.3, 0.3);
	glVertex3f(0.3, 0.3, -0.3);
	glVertex3f(-0.3, 0.3, -0.3);
	glVertex3f(-0.3, 0.3, 0.3);
	glEnd();
	//DOWN
```

```
    glBegin(GL_POLYGON);
    glColor3f(1.0, 0.0, 0.0);
    glVertex3f(0.3, -0.3, -0.3);
    glVertex3f(0.3, -0.3, 0.3);
    glVertex3f(-0.3, -0.3, 0.3);
    glVertex3f(-0.3, -0.3, -0.3);
    glEnd();

    glFlush();
    glutSwapBuffers();
}
void specialKeys(int key, int x, int y) {
    if (key == GLUT_KEY_RIGHT) {
        Ry += 5;
    }
    else if (key == GLUT_KEY_LEFT) {
        Ry-= 5;
    }
    else if (key == GLUT_KEY_UP) {
        Rx += 5;
    }
    else if (key == GLUT_KEY_DOWN) {
        Rx -= 5;
    }
    glutPostRedisplay();
}

int main(int argc, char* argv[]) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(800, 800);
    glutCreateWindow("Cubo");
    glEnable(GL_DEPTH_TEST);
    glutDisplayFunc(display);
    glutSpecialFunc(specialKeys);
    glutMainLoop();
    return 0;
}
```