

### 3. Draw a colour cube and spin it using OpenGL transformation matrices.

```
#include<stdlib.h>
#include<GL/glut.h>

GLfloat vertices[][3] = {{-1.0,-1.0,-1.0},{1.0,-1.0,-1.0},{1.0,1.0,-1.0},{-1.0,1.0,-1.0},{-1.0,-1.0,1.0},
{1.0,-1.0,1.0},{1.0,1.0,1.0},{-1.0,1.0,1.0}};
GLfloat color[][3] = {{0.0,0.0,0.0},{1.0,0.0,0.0},{1.0,1.0,0.0},{0.0,1.0,0.0},{0.0,0.0,1.0},{1.0,0.0,1.0},
{1.0,1.0,1.0},{0.0,1.0,1.0}};

void polygon(int a, int b, int c, int d){
    glBegin(GL_POLYGON);
    glColor3fv(color[a]);
    glVertex3fv(vertices[a]);
    glColor3fv(color[b]);
    glVertex3fv(vertices[b]);
    glColor3fv(color[c]);
    glVertex3fv(vertices[c]);
    glColor3fv(color[d]);
    glVertex3fv(vertices[d]);
    glEnd();
}

void colorcube(void){
    polygon(0,3,2,1);
    polygon(2,3,7,6);
    polygon(0,4,7,3);
    polygon(1,2,6,5);
    polygon(4,5,6,7);
    polygon(0,1,5,4);
}

static GLfloat theta[] = {0.0,0.0,0.0};
static GLint axis = 2;

void display(void){
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();
    glRotatef(theta[0],1.0,0.0,0.0);
    glRotatef(theta[1],0.0,1.0,0.0);
    glRotatef(theta[2],0.0,0.0,1.0);
    colorcube();
    glutSwapBuffers();
}

void spinCube(){
    theta[axis] += 0.5;
    if(theta[axis]>36.0){
        theta[axis] -= 360.0;
    }
    glutPostRedisplay();
}

void mouse(int btn, int state, int x, int y){
    if(btn == GLUT_LEFT_BUTTON && state == GLUT_DOWN){
        axis=0;
    }
    if(btn == GLUT_MIDDLE_BUTTON && state == GLUT_DOWN){
```

```

        axis=1;
    }
    if(btn == GLUT_RIGHT_BUTTON && state == GLUT_DOWN){
        axis=2;
    }
}

void myReshape(int w, int h){
    glViewport(0,0,w,h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    if(w<=h){
        glOrtho(-2.0,2.0,-2.0*(GLfloat)h/(GLfloat)w,2.0*(GLfloat)h/(GLfloat)w,-10.0,10.0);
    }
    else{
        glOrtho(-2.0*(GLfloat)w/(GLfloat)h,2.0*(GLfloat)w/(GLfloat)h,-2.0,2.0,-10.0,10.0);
    }
    glMatrixMode(GL_MODELVIEW);
}

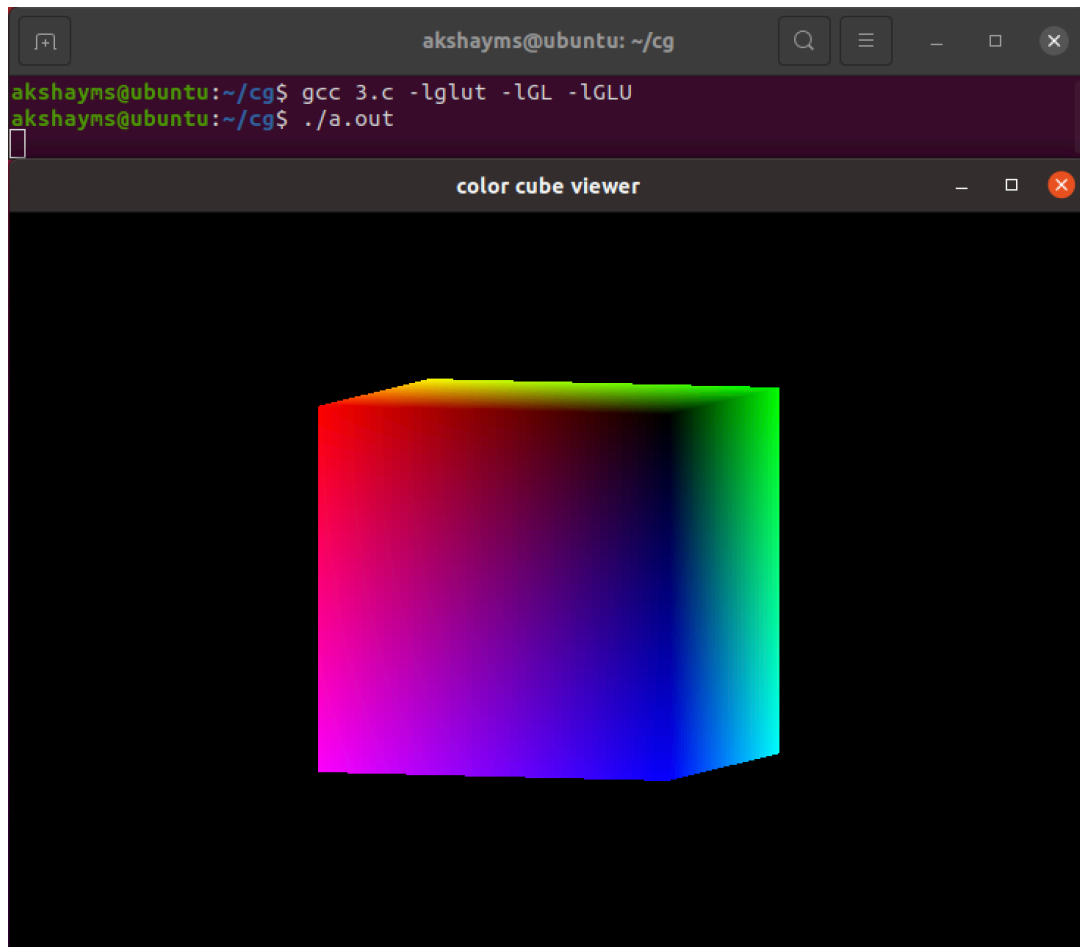
int main(int argc, char** argv){
    glutInit(&argc,argv);
    glutInitDisplayMode(GLUT_DOUBLE|GLUT_RGB|GLUT_DEPTH);
    glutInitWindowSize(500,500);
    glutCreateWindow("color cube viewer");

    glutReshapeFunc(myReshape);
    glutDisplayFunc(display);
    glutMouseFunc(mouse);
    glutIdleFunc(spinCube);

    glEnable(GL_DEPTH_TEST);
    glutMainLoop();
    return 0;
}

```

Output for spin along horizontal axis



Output for spin along vertical axis

