

```

/* ITTextCG, Chapter 5, Exercise 3, Sample Program: Cube */
#include <GL/glut.h>

void display(void)
{
    GLfloat x1,y1,z1,x2,y2,z2,x3,y3,z3,x4,y4,z4;
    GLfloat x5,y5,z5,x6,y6,z6,x7,y7,z7,x8,y8,z8;

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glFrustum(-50.0, 50.0, -50.0, 50.0, 50.0, 1000.0);
    gluLookAt(200.0, 200.0, 200.0, 0.0, 0.0, 0.0, 0.0,
              1.0, 0.0);

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();

    /* 回転と平行移動 */
    glRotatef(30.0, 0.0, 1.0, 0.0);
    glTranslatef(50.0, 50.0, 50.0);

    x1 = -50.0;
    y1 = -50.0;
    z1 = 50.0;
    x2 = 50.0;
    y2 = -50.0;
    z2 = 50.0;
    x3 = 50.0;
    y3 = 50.0;

```

```
z3 = 50.0;
x4 = -50.0;
y4 = 50.0;
z4 = 50.0;
x5 = -50.0;
y5 = 50.0;
z5 = -50.0;
x6 = 50.0;
y6 = 50.0;
z6 = -50.0;
x7 = 50.0;
y7 = -50.0;
z7 = -50.0;
x8 = -50.0;
y8 = -50.0;
z8 = -50.0;
```

```
/* 平行六面体を構成する多角形の表示 */
```

```
glBegin(GL_POLYGON);
    glNormal3f(0.0, 0.0, 1.0);
    glVertex3f(x1, y1, z1);
    glVertex3f(x2, y2, z2);
    glVertex3f(x3, y3, z3);
    glVertex3f(x4, y4, z4);
glEnd();
```

```
/* 平行六面体を構成する多角形の表示 */
```

```
glBegin(GL_POLYGON);
    glNormal3f(1.0, 0.0, 0.0);
    glVertex3f(x3, y3, z3);
    glVertex3f(x2, y2, z2);
    glVertex3f(x7, y7, z7);
```

```

    glVertex3f(x6, y6, z6);
glEnd();

/* 平行六面体を構成する多角形の表示 */
glBegin(GL_POLYGON);
    glNormal3f(0.0, 1.0, 0.0);
    glVertex3f(x4, y4, z4);
    glVertex3f(x3, y3, z3);
    glVertex3f(x6, y6, z6);
    glVertex3f(x5, y5, z5);
glEnd();

/* 平行六面体を構成する多角形の表示 */
glBegin(GL_POLYGON);
    glNormal3f(-1.0, 0.0, 0.0);
    glVertex3f(x4, y4, z4);
    glVertex3f(x5, y5, z5);
    glVertex3f(x8, y8, z8);
    glVertex3f(x1, y1, z1);
glEnd();

/* 平行六面体を構成する多角形の表示 */
glBegin(GL_POLYGON);
    glNormal3f(0.0, -1.0, 0.0);
    glVertex3f(x7, y7, z7);
    glVertex3f(x2, y2, z2);
    glVertex3f(x1, y1, z1);
    glVertex3f(x8, y8, z8);
glEnd();

/* 平行六面体を構成する多角形の表示 */
glBegin(GL_POLYGON);

```

```

        glNormal3f(0.0, 0.0, -1.0);
        glVertex3f(x6, y6, z6);
        glVertex3f(x7, y7, z7);
        glVertex3f(x8, y8, z8);
        glVertex3f(x5, y5, z5);
    glEnd();

    glFlush();

}

void keyboard(unsigned char key, int x, int y)
{
    switch(key) {
        case 27: exit(0); break;

    }

}

/* 材質（反射率）と照明の設定を行う関数 */
void LightSource()
{
    GLfloat mat_diffuse[] = { 0.0, 0.7, 0.0, 1.0 };
    GLfloat mat_specular[] = { 0.3, 0.3, 0.3, 1.0 };
    GLfloat mat_shininess[] = { 6.0 };
    GLfloat light_diffuse[] = { 0.7, 0.7, 0.7, 1.0 };
    GLfloat light_specular[] = { 0.7, 0.7, 0.7, 1.0 };
    GLfloat light_ambient[] = { 0.3, 0.3, 0.3, 1.0 };
    GLfloat light_position[] = { -0.5, 1.0, 0.5, 0.0 };

```

```

glMaterialfv(GL_FRONT, GL_DIFFUSE, mat_diffuse);
glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
glMaterialfv(GL_FRONT, GL_SHININESS, mat_shininess);
glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse);
glLightfv(GL_LIGHT0, GL_SPECULAR, light_specular);
glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient);
glLightfv(GL_LIGHT0, GL_POSITION, light_position);

glEnable(GL_LIGHTING);
glEnable(GL_LIGHT0);
glEnable(GL_DEPTH_TEST);
}

int main(int argc, char** argv)
{
    glutInitWindowSize(1000,1000);
    glutInitWindowPosition(0,0);
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGBA | GLUT_DEPTH);
    glutCreateWindow("Shading");

    glClearColor(0.0, 0.0, 0.0, 0.0);

    LightSource();

    glutDisplayFunc(display);
    glutKeyboardFunc(keyboard);
    glutMainLoop();
}

```