

```

/* ITTextCG, Chapter 5, Exercise 4, Sample Program: 3 Cubes */
#include <GL/glut.h>

void BoxA(GLfloat boxlen);

void display(void)
{
    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);

    /* 半透明の平行六面体の三重の入れ子になった形状の表示 */
    BoxA(20.0);
    BoxA(35.0);
    BoxA(50.0);

    glFlush();
}

```

```
/* 半透明の平行六面体を表示するための関数 */  
void BoxA(GLfloat boxlen)  
  
{  
    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;  
  
    x1 = -boxlen;  
    y1 = -boxlen;  
    z1 = boxlen;  
    x2 = boxlen;  
    y2 = -boxlen;  
    z2 = boxlen;  
    x3 = boxlen;  
    y3 = boxlen;  
    z3 = boxlen;  
    x4 = -boxlen;  
    y4 = boxlen;  
    z4 = boxlen;  
    x5 = -boxlen;  
    y5 = boxlen;  
    z5 = -boxlen;  
    x6 = boxlen;  
    y6 = boxlen;  
    z6 = -boxlen;  
    x7 = boxlen;  
    y7 = -boxlen;  
    z7 = -boxlen;  
    x8 = -boxlen;  
    y8 = -boxlen;  
    z8 = -boxlen;  
}
```

```
/* 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();
```

```
    return;
```

```
}
```

```
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, 0.5 };
    GLfloat mat_specular[] = { 0.3, 0.3, 0.3, 0.5 };
    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("BlendShading");

glClearColor(0.0, 0.0, 0.0, 0.0);

LightSource();

/* 色の混合処理の設定 */
glEnable(GL_BLEND);
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);

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

}
```