

```
/* ITTextCG, Chapter 5, Exercise 2, Polygon Color Blending */
#include <GL/glut.h>
#include <stdlib.h>
```

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

    }
}
```

```
void display(void)
{

    glClear(GL_COLOR_BUFFER_BIT);
    glLoadIdentity();
    glOrtho(-300.0, 300.0, -300.0, 300.0, -300.0, 300.0);

    /* 多角形の表示 */
    glColor4f(0.0, 1.0, 1.0, 0.5);
    glBegin(GL_POLYGON);
        glVertex2f(-60.0, -60.0);
        glVertex2f(60.0, -60.0);
        glVertex2f(60.0, 60.0);
        glVertex2f(-60.0, 60.0);
    glEnd();
}
```

```

/* 多角形の表示 */
glColor4f(1.0,1.0, 0.0, 0.5);
glBegin(GL_POLYGON);
    glVertex2f(-100.0, -100.0);
    glVertex2f(30.0, -100.0);
    glVertex2f(30.0, 30.0);
    glVertex2f(-100.0, 30.0);
glEnd();

/* 多角形の表示 */
glColor4f(1.0, 0.0, 1.0, 0.5);
glBegin(GL_POLYGON);
    glVertex2f(-30.0, -30.0);
    glVertex2f(100.0, -30.0);
    glVertex2f(100.0, 100.0);
    glVertex2f(-30.0, 100.0);
glEnd();

glFlush();
}

int main(int argc, char** argv)
{
    glutInitWindowSize(1000,1000);
    glutInitWindowPosition(0, 0);
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGBA);
    glutCreateWindow(argv[0]);
    glClearColor(0.0, 0.0, 0.0, 0.0);

```

```
/* 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();  
  
return 0;  
}
```