

```

/* ITTextCG, Chapter 8, Exercise 1, Bezier Curve */
#include <GL/glut.h>

/* ベジエ曲線の制御点 */
GLfloat controlpoints[4][3] = { {-40.0, -40.0, 0.0},
                                {-20.0, 40.0, 0.0},
                                {20.0, 40.0, 0.0},
                                {40.0, -40.0, 0.0}};

void display(void)
{
    int i;

    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(100.0, 100.0, 100.0, 0.0, 0.0, 0.0, 0.0,
              1.0, 0.0);

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();

    /* ベジエ曲線の表示 */
    glColor3f(1.0, 0.0, 1.0);
    glBegin(GL_LINE_STRIP);
    for(i = 0; i <= 30; ++i)
    {
        glEvalCoord1f((GLfloat)(i/30.0));
    }
}

```

```

    } /* for i */
    glEnd();

    /* 制御多角形の表示 */
    glColor3f(1.0, 0.0, 0.0);
    glBegin(GL_LINE_STRIP);
    for(i = 0; i <= 3; ++i)
    {
        glVertex3f((GLfloat)controlpoints[i][0],
                    (GLfloat)controlpoints[i][1],
                    (GLfloat)controlpoints[i][2]);
    } /* for i */
    glEnd();

    glFlush();

}

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

    }
}

```

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

    glClearColor(0.0, 0.0, 0.0, 0.0);

    /* ベジエ曲線のための一次元エバリュエータの定義 */
    glMap1f(GL_MAP1_VERTEX_3, 0.0, 1.0, 3, 4,
            &controlpoints[0][0]);
    glEnable(GL_MAP1_VERTEX_3);

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