```c
#include <GL/glew.h>
#include <GL/glut.h>
#include <stdio.h>
#include <stdlib.h>

static char* vsSource = "#version 120 \n\
in vec4 aPosition; \n\
in vec4 aColor; \n\
out vec4 vColor; \n\
uniform vec4 udvec; \n\
void main(void) { \n\
  gl_Position = aPosition + udvec; \n\
  vColor = aColor; \n\
}";

static char* fsSource = "#version 120 \n\
in vec4 vColor; \n\
void main(void) { \n\
  gl_FragColor = vColor; \n\
}";

GLuint vs = 0;
GLuint fs = 0;
GLuint prog = 0;

char buf[1024];
float t = 0.0f;

GLfloat vertices[] = {
    -0.2, -0.2, -0.2, 1.0, // 0
    -0.2, -0.2, +0.2, 1.0, // 1
    -0.2, +0.2, -0.2, 1.0, // 2
    -0.2, +0.2, +0.2, 1.0, // 3
    +0.2, -0.2, -0.2, 1.0, // 4
    +0.2, -0.2, +0.2, 1.0, // 5
    +0.2, +0.2, -0.2, 1.0, // 6
    +0.2, +0.2, +0.2, 1.0, // 7
};


GLfloat colors[] = {
    1.0, 0.0, 0.0,
    0.0, 1.0, 0.0,
    0.0, 0.0, 1.0,
    1.0, 0.0, 0.0,
    0.0, 1.0, 0.0,
    0.0, 0.0, 1.0,
    1.0, 0.0, 0.0,
    0.0, 1.0, 0.0
};

GLushort indices[] = { // 36 points, 12 triangles
    0, 4, 6,
    6, 2, 0,
    4, 5, 7,
    7, 6, 4,
```

```
57      1, 3, 7,
58      7, 5, 1,
59      0, 2, 3,
60      3, 1, 0,
61      2, 6, 7,
62      7, 3, 2,
63      0, 1, 5,
64      5, 4, 0,
65  };
66
67  void myinit(void) {
68      GLuint status;
69
70      printf("***** Your student number and name *****\n");
71      vs = glCreateShader(GL_VERTEX_SHADER);
72      glShaderSource(vs, 1, &vsSource, NULL);
73      glCompileShader(vs);
74      glGetShaderiv(vs, GL_COMPILE_STATUS, &status);
75      printf("vs compile status = %s\n", (status == GL_TRUE) ? "true" :
          "false");
76      glGetShaderInfoLog(vs, sizeof(buf), NULL, buf);
77      printf("vs log = [%s]\n", buf);
78
79      fs = glCreateShader(GL_FRAGMENT_SHADER);
80      glShaderSource(fs, 1, &fsSource, NULL);
81      glCompileShader(fs);
82      glGetShaderiv(fs, GL_COMPILE_STATUS, &status);
83      printf("fs compile status = %s\n", (status == GL_TRUE) ? "true" :
          "false");
84      glGetShaderInfoLog(fs, sizeof(buf), NULL, buf);
85      printf("fs log = [%s]\n", buf);
86
87      prog = glCreateProgram();
88      glAttachShader(prog, vs);
89      glAttachShader(prog, fs);
90      glLinkProgram(prog);
91      glGetProgramiv(prog, GL_LINK_STATUS, &status);
92      printf("program link status = %s\n", (status == GL_TRUE) ? "true" :
          "false");
93      glGetProgramInfoLog(prog, sizeof(buf), NULL, buf);
94      printf("link log = [%s]\n", buf);
95      glValidateProgram(prog);
96      glGetProgramiv(prog, GL_VALIDATE_STATUS, &status);
97      printf("program validate status = %s\n", (status == GL_TRUE) ? "true" :
          "false");
98      glGetProgramInfoLog(prog, sizeof(buf), NULL, buf);
99      printf("validate log = [%s]\n", buf);
100     glUseProgram(prog);
101
102     GLuint loc;
103     GLuint vbo[1];
104     // using vertex buffer object
105     glGenBuffers(1, vbo);
106     glBindBuffer(GL_ARRAY_BUFFER, vbo[0]);
107     glBufferData(GL_ARRAY_BUFFER, 2 * 8 * 4 * sizeof(GLfloat), NULL,
          GL_STATIC_DRAW);
```

```
108     glBufferSubData(GL_ARRAY_BUFFER, 0, 8 * 4 * sizeof(GLfloat), vertices);
109     glBufferSubData(GL_ARRAY_BUFFER, 8 * 4 * sizeof(GLfloat), 8 * 4 * sizeof
          (GLfloat),
110         colors);
111
112     loc = glGetAttribLocation(prog, "aPosition");
113     glEnableVertexAttribArray(loc);
114     glVertexAttribPointer(loc, 4, GL_FLOAT, GL_FALSE, 0, (GLvoid *)0);
115
116     loc = glGetAttribLocation(prog, "aColor");
117     glEnableVertexAttribArray(loc);
118     glVertexAttribPointer(loc, 4, GL_FLOAT, GL_FALSE, 0, (GLvoid *)(3 * 4 *
          sizeof(GLfloat)));
119
120 }
121
122 void mykeyboard(unsigned char key, int x, int y) {
123     switch (key) {
124     case 27: // ESCAPE
125         exit(0);
126         break;
127     }
128 }
129
130
131
132 void myidle(void) {
133     t += 0.001f;
134     if (t > 1.5)
135         t = 0.0f;
136
137     // redisplay
138     glutPostRedisplay();
139 }
140
141
142 void mydisplay(void) {
143     GLuint loc;
144     GLfloat d[4] = { 0.2, 0.5, 0.0, 0.0 };
145     GLfloat dvec[4];
146
147     dvec[0] = t * d[0];
148     dvec[1] = t * d[1];
149     dvec[2] = t * d[2];
150     dvec[3] = t * d[3];
151
152     glClearColor(0.7f, 0.7f, 0.7f, 1.0f); // gray
153     glClear(GL_COLOR_BUFFER_BIT);
154
155     loc = glGetUniformLocation(prog, "udvec");
156     glUniform4fv(loc, 1, dvec);
157
158     glDrawElements(GL_TRIANGLES, 12 * 3, GL_UNSIGNED_SHORT, indices);
159
160     glFlush();
161     glutSwapBuffers();
```

```
162 }
163
164 int main(int argc, char* argv[]) {
165     glutInit(&argc, argv);
166     glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
167     glutInitWindowSize(500, 500);
168     glutInitWindowPosition(0, 0);
169     glutCreateWindow("*** Your Student Number and Name ***");
170     glutDisplayFunc(mydisplay);
171     glutIdleFunc(myidle);
172     glutKeyboardFunc(mykeyboard);
173     glewInit();
174     myinit();
175     glutMainLoop();
176     return 0;
177 }
178
```