```
#include <stdlib.h> #include <GL/glut.h>
GLfloat vertices[][3] = {{-1.0,-1.0,-1.0},{1.0,-1.0,-
1.0}, {1.0,1.0,-1.0}, {-1.0,1.0,-1.0},  {-1.0,-1.0,1.0},
{1.0,-1.0,1.0}, {1.0,1.0,1.0}, {-1.0,1.0,1.0}};
GLfloat normals[][3] = {{-1.0,-1.0,-1.0},{1.0,-1.0,-
1.0}, {1.0,1.0,-1.0}, {-1.0,1.0,-1.0},   {-1.0,-1.0,1.0},
{1.0,-1.0,1.0}, {1.0,1.0,1.0}, {-1.0,1.0,1.0}};
GLfloat colors[][3] = {{0.0,0.0,0.0},{1.0,0.0,0.0},
{1.0,1.0,0.0}, {0.0,1.0,0.0},   {0.0,0.0,1.0},
{1.0,0.0,1.0}, {1.0,1.0,1.0}, {0.0,1.0,1.0}};
void displayText(float x,float y,float z,char* s){
int i; glRasterPos3f(x,y,z);
for(i=0;s[i]!='\0';i++){
glutBitmapCharacter(GLUT_BITMAP_HELVETICA
_18,s[i]); }  }
void  polygon(int a, int b, int c , int d)
{ glBegin(GL_POLYGON);glVertex3fv(vertices[a]);
glColor3fv(colors[b]); glNormal3fv(normals[b]);
glVertex3fv(vertices[b]); glColor3fv(colors[c]);
glNormal3fv(normals[c]); glVertex3fv(vertices[c]);
glColor3fv(colors[d]); glNormal3fv(normals[d]);
glVertex3fv(vertices[d]); glEnd();  }
void  colorcube(void)
{ polygon(0,3,2,1); polygon(2,3,7,6); polygon(0,4,7,3);
polygon(1,2,6,5); polygon(4,5,6,7); polygon(0,1,5,4); }
static GLfloat theta[] = {0.0,0.0,0.0};
static GLint axis = 1;
void  display(void)
{glClear(GL_COLOR_BUFFER_BIT |
GL_DEPTH_BUFFER_BIT);
glLoadIdentity(); glColor3f(1.0,1.0,1.0);
displayText(1,-1.5,0,"Cube"); glFlush();
glPushMatrix(); glRotatef(theta[0], 1.0, 0.0, 0.0);
glRotatef(theta[1], 0.0, 1.0, 0.0);
```

```
glRotatef(theta[2], 0.0, 0.0, 1.0);
colorcube(); glBegin(GL_LINES);
glVertex3f(-2.0,0.0,2.0); glVertex3f(-2.0,0.0,2.0);
glEnd(); glFlush();  glPopMatrix();  glFlush();
glutSwapBuffers();  }
void  spinCube()
{  theta[axis] += 1.5;
if( theta[axis] > 360.0 ) theta[axis] -= 360.0;
glutPostRedisplay();  }
void  mouse(int btn, int state, int x, int y)
{ if(btn==GLUT_LEFT_BUTTON && state ==
GLUT_DOWN) axis = 0;
if(btn==GLUT_MIDDLE_BUTTON && state ==
GLUT_DOWN) axis = 1;
if(btn==GLUT_RIGHT_BUTTON && state ==
GLUT_DOWN) axis = 2; }
void  myReshape(int w, int h)
{ glViewport(0, 0, w, h);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
if (w <= h)glOrtho(-2.0, 2.0, -2.0 * (GLfloat) h /
(GLfloat) w, 2.0 * (GLfloat) h / (GLfloat) w, -10.0,
10.0);
else    glOrtho(-2.0 * (GLfloat) w / (GLfloat) h, 2.0 *
(GLfloat) w / (GLfloat) h, -2.0, 2.0, -10.0, 10.0);
glMatrixMode(GL_MODELVIEW); }
void  main(int argc, char **argv)
{ glutInit(&argc, argv);
glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB
| GLUT_DEPTH);
glutInitWindowSize(500, 500);
glutCreateWindow("Rotating a Color Cube");
glutReshapeFunc(myReshape);
glutDisplayFunc(display);glutIdleFunc(spinCube);
glutMouseFunc(mouse);
glEnable(GL_DEPTH_TEST); glutMainLoop();  }
```