6. Draw a color cube and allow the user to move the camera suitably to experiment with perspective viewing.

```
#include<stdio.h>
#include<GL/glut.h>
float v[8][3]={{-100,-100,100},{100,-100,100},{100,100,100},{-100,100,100},{-100,-100,-100}, {100,-100,-100},{100,100,-100},{-100,100,-100}};
float viewer[3]={ 0.0,0.0,500};
void keys(unsigned char k,int x,int y)
{
if(k=='x')
viewer[0]-=10.0;
if(k=='X')
viewer[0]+=10.0;
if(k=='y')
viewer[1]+=10.0;
if(k=='Y')
viewer[1]-=10.0;
if(k=='z')
viewer[2]+=10.0;
if(k=='Z')
viewer[2]-=10.0;
glutPostRedisplay();
}
void display()
{
glClearColor(0.0,0.0,0.0,0.0);
glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
glFrustum(-200,200,-200,200,200,800);
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
gluLookAt(viewer[0],viewer[1],viewer[2],0,0,0,1,0,0);
glColor3f(1.0,0.6,0.3);
drawcube(v[0],v[1],v[2],v[3]);
glColor3f(1.0,0.7,0.3);
drawcube(v[1],v[5],v[6],v[2]);
glColor3f(1.0,0.0,0.0);
drawcube(v[3],v[2],v[6],v[7]);
glColor3f(0.0,1.0,0.0);
drawcube(v[4],v[5],v[1],v[0]);
glColor3f(0.0,0.0,1.0);
drawcube(v[7],v[6],v[5],v[4]);
glColor3f(1.0,1.0,0.3);
```

```
drawcube(v[3],v[7],v[4],v[0]);
glFlush();
}
void drawcube(GLfloat *a,GLfloat *b,GLfloat *c,GLfloat *d)
{
glBegin(GL_POLYGON);
glVertex3fv(a);
glVertex3fv(b);
glVertex3fv(c);
glVertex3fv(d);
glEnd();
}
void main(int argc, char *argv[])
{
glutInit(&argc,argv);
glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB|GLUT_DEPTH);
glutInitWindowPosition(10,10);
glutInitWindowSize(500,500);
glutCreateWindow("Perspective View");
glutDisplayFunc(display);
glEnable(GL_DEPTH_TEST);
glutKeyboardFunc(keys);
glutMainLoop();
}
```