```
/* view4.c */

#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <assert.h>

#ifdef __APPLE__
#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif


/* forward declarations of functions */
void display(void);
void renderScene(void);
void keypress(unsigned char key, int x, int y);


void draw_xy_rect(GLfloat z, GLfloat r, GLfloat g, GLfloat b) {
  glBegin(GL_POLYGON);
  glColor3f(r,g,b);
  glVertex3f(-1,-1, z);
  glVertex3f(-1,1, z);
  glVertex3f(1,1, z);
  glVertex3f(1,-1, z);
  glEnd();
}


void draw_cube(GLfloat z1, GLfloat z2) {
  glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
  glColor3f(1.0, 1.0, 1.0);

  draw_xy_rect(z2,0,0,1);

  glBegin(GL_POLYGON);
  glVertex3f(-1,-1, z1);
  glVertex3f(-1,1, z1);
  glVertex3f(-1,1, z2);
  glVertex3f(-1,-1, z2);
  glEnd();

  glBegin(GL_POLYGON);
  glVertex3f(1,-1, z1);
  glVertex3f(1,1, z1);
  glVertex3f(1,1, z2);
  glVertex3f(1,-1, z2);
  glEnd();

  draw_xy_rect(z1,1,0,0);

}



void display(void) {
  draw_cube(-5, -7);
}


void renderScene(void) {
  glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

  display();
  glFlush();
}

void init_view() {

  /* camera is at (0,0,0) looking along negative y axis */
  glMatrixMode(GL_PROJECTION);
  glLoadIdentity();
  gluPerspective(60, 1 /* aspect */, 1, 10.0); /* the frustrum is from z=-1 to z=-10 *
/

  glMatrixMode(GL_MODELVIEW);
  if(0) {
    /* set up a different initial viewpoint */
    glLoadIdentity();
    gluLookAt(0, -2, 2,/* eye */  0, 0, 0,/* center */ 0, 0, 1);/* up */
  }
  if(0)glTranslatef(0,0,-5);


}



void
keypress(unsigned char key, int x, int y) {
  switch(key) {

  case 'i':/* reset to start view */
    init_view();
    glutPostRedisplay();
    break;
  case 'I':/* look at back */
    glLoadIdentity();
    glRotatef(180, 1, 0, 0);
    glutPostRedisplay();
    break;

    /* various rotations: */
  case 'x':
    glRotatef(5.0, 1, 0, 0);
    glutPostRedisplay();
    break;
  case 'y':
    glRotatef(5.0, 0, 1, 0);
    glutPostRedisplay();
    break;
  case 'z':
    glRotatef(5.0, 0, 0, 1);
    glutPostRedisplay();
    break;
  case 'X':
    glRotatef(-5.0, 1, 0, 0);
    glutPostRedisplay();
    break;
  case 'Y':
    glRotatef(-5.0, 0, 1, 0);
    glutPostRedisplay();
    break;
  case 'Z':
    glRotatef(-5.0, 0, 0, 1);
    glutPostRedisplay();
    break;

  case 'b':
    glTranslatef(0,0, -0.5);
    glutPostRedisplay();
    break;
```

```
  case 'f':
    glTranslatef(0,0, 0.5);
    glutPostRedisplay();
    break;
  case 'q':
    exit(0);
    break;
  }
}


int main(int argc, char** argv) {

    /* open a window */
  glutInit(&argc, argv);
  glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
  glutInitWindowSize(500, 500);
  glutInitWindowPosition(100,100);
  glutCreateWindow(argv[0]);

  /* OpenGL init */
  glClearColor(0, 0, 0, 0);   /* set background color black*/

  /* callback functions */
  glutDisplayFunc(renderScene);
  glutKeyboardFunc(keypress);

  /* initial view */
  init_view();

  /* event handler */
  glutMainLoop();
  return 0;
}
```