PROGRAM -3

3. Draw a colour cube and spin it using OpenGL transformation matrices.

```
//program to rotate a cube

#include<GL/glut.h>
float v[][3]={{0,0,0},{1,0,0},{1,1,0},{0,1,0},{0,0,1},{1,0,1},{1,1,1},{0,1,1}};
float theta[]={0,0,0};
int axis=0;

void polygon(int a, int b, int c, int d)
{
glBegin(GL_POLYGON);
glColor3fv(v[a]);
glVertex3fv(v[a]);
glColor3fv(v[b]);
glVertex3fv(v[b]);
glColor3fv(v[c]);
glVertex3fv(v[c]);
glColor3fv(v[d]);
glVertex3fv(v[d]);
glEnd();
}

void colorCube()
{
polygon(0,1,2,3);
polygon(4,5,6,7);
polygon(7,6,2,3);
polygon(4,5,1,0);
polygon(4,0,3,7);
polygon(5,1,2,6);
}

void display()
{
glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
glClearColor(0,0,0,1);
glLoadIdentity();
glRotatef(theta[0],1,0,0);
```

```
glRotatef(theta[1],0,1,0);
glRotatef(theta[2],0,0,1);
colorCube();
glFlush();
glutSwapBuffers();
}

void spinCube()
{
theta[axis]+=5;
if(theta[axis]>360)
theta[axis]=0;
display();
}

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(-2,2,-2*h/w,2*h/w,-10,10);
else
glOrtho(-2*w/h,2*w/h,-2,2,-10,10);
glMatrixMode(GL_MODELVIEW);
}

int main(int argc,char **argv)
{
glutInit(&argc, argv);
```

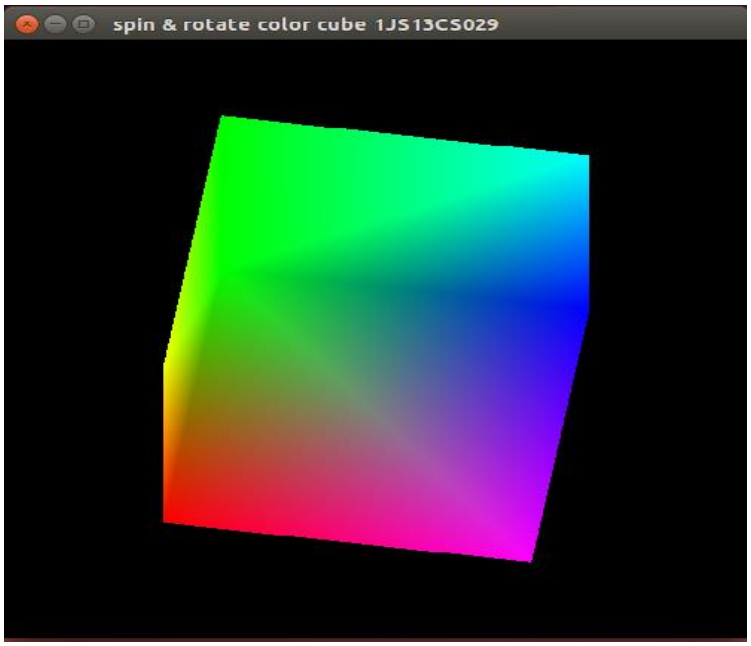
```
glutInitDisplayMode(GLUT_DOUBLE|GLUT_DEPTH);
glutInitWindowSize(500,500);
glutCreateWindow("CUBE");
glutReshapeFunc(myReshape);
glutDisplayFunc(display);
glutIdleFunc(spinCube);
glutMouseFunc(mouse);
glEnable(GL_DEPTH_TEST);
glutMainLoop();
}
```

Output command

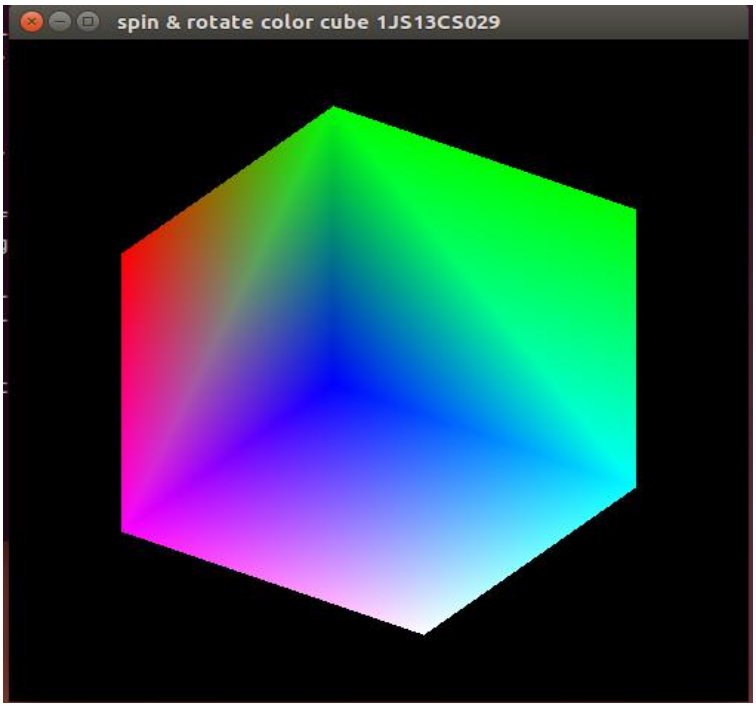
To create file - gedit filename.c

To compile file - gcc filename.c -lGL -lGLU -lglut

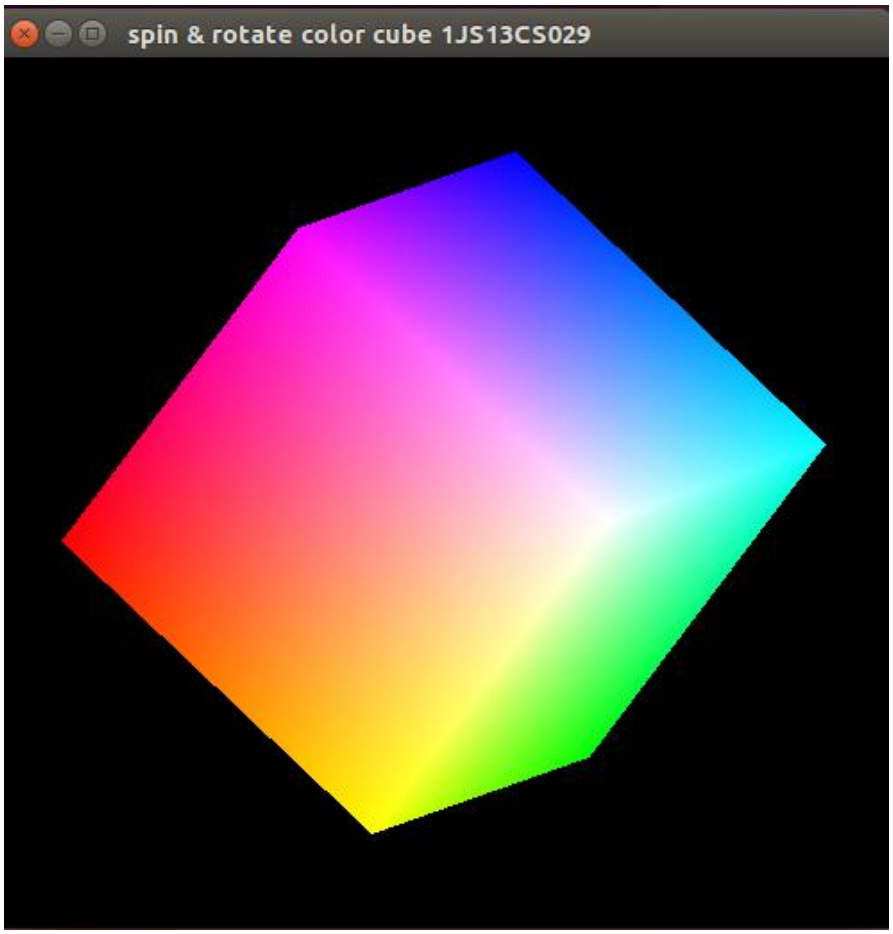
To execute - ./a.out



Rotation of cube in Z-axis (when Right button is pressed)



Rotation of cube in X-axis (when Left button is pressed)



Rotation of cube in Y-axis (when middle button is pressed)