**PROGRAM NO:**

**AIM:**

**PROGRAM:**

#include<GL/glu.h>

#include<GL/glut.h>

GLfloat xRotated,yRotated,zRotated;

void displayOctahedron(void)

{

glMatrixMode(GL\_MODELVIEW);

glClear(GL\_COLOR\_BUFFER\_BIT);

glLoadIdentity();

glTranslatef(0.0,0.0,-4.5);

glColor3f(0.8,0.2,0.1);

glRotatef(xRotated,1.0,0.0,0.0);

glRotatef(yRotated,0.0,1.0,0.0);

glRotatef(zRotated,0.0,0.0,1.0);

glScalef(1.0,1.0,1.0);

glutSolidOctahedron();

glFlush();

}

void reshapeOctahedron(int x,int y)

{

if(y==0|x==0)

return;

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluPerspective(40.0,(GLdouble)x/(GLdouble)y,0.5,20.5);

glViewport(0,0,x,y);

}

void idleOctahedron(void)

{

yRotated+=0.03;

displayOctahedron();

}

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

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(400,350);

glutCreateWindow("Octahedron Rotating Animation");

glPolygonMode(GL\_FRONT\_AND\_BACK,GL\_LINE);

xRotated=yRotated=zRotated=30.0;

xRotated=33;

yRotated=40;

glClearColor(1.0,1.0,0.0,0.0);

glutDisplayFunc(displayOctahedron);

glutReshapeFunc(reshapeOctahedron);

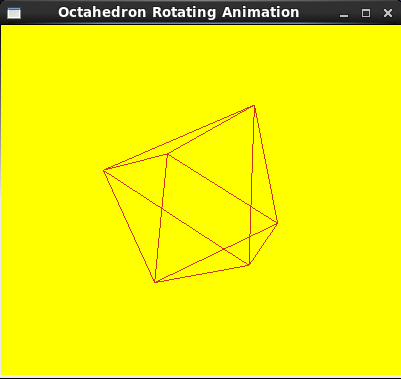
glutIdleFunc(idleOctahedron);

glutMainLoop();

return 0;

}

**OUTPUT:**



**PROGRAM NO:**

**AIM:**

**PROGRAM:**

#include<GL/glut.h>

GLint winWidth=600,winHeight=600;

GLfloat x0=100.0,y0=50.0,z0=50.0;

GLfloat xref=50.0,yref=50.0,zref=0.0;

GLfloat Vx=0.0,Vy=1.0,Vz=0.0;

GLfloat xwMin=-40,ywMin=-60,xwMax=40.0,ywMax=60.0;

GLfloat dnear=25.0,dfar=125.0;

void init(void)

{

glClearColor(0.0,0.0,0.0,0.0);

glMatrixMode(GL\_MODELVIEW);

gluLookAt(x0,y0,z0,xref,yref,zref,Vx,Vy,Vz);

glMatrixMode(GL\_PROJECTION);

glFrustum(xwMin,xwMax,ywMin,ywMax,dnear,dfar);

}

void displayFun(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(0.0,0.0,1.0);

glPolygonMode(GL\_FRONT,GL\_FILL);

glPolygonMode(GL\_BACK,GL\_LINE);

glBegin(GL\_QUADS);

glVertex3f(0.0,0.0,0.0);

glVertex3f(100.0,0.0,0.0);

glVertex3f(100.0,100.0,0.0);

glVertex3f(0.0,100.0,0.0);

glEnd();

glFlush();

}

void reshapeFun(GLint newWidth,GLint newHeight)

{

glViewport(0,0,newWidth,newHeight);

winWidth=newWidth;

winHeight=newHeight;

}

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

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowPosition(50,50);

glutInitWindowSize(winWidth,winHeight);

glutCreateWindow("Perspective view of a square");

init();

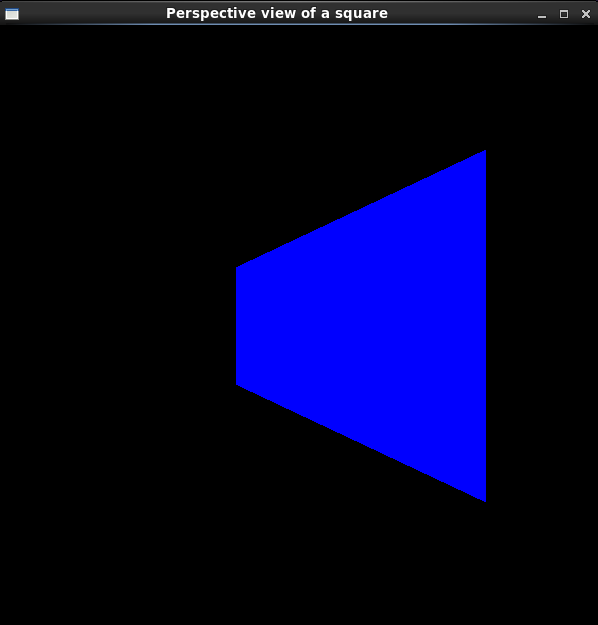
glutDisplayFunc(displayFun);

glutReshapeFunc(reshapeFun);

glutMainLoop();

}

**OUTPUT:**



**PROGRAM NO:**

**AIM:**

**PROGRAM:**

#include<GL/glu.h>

#include<GL/glut.h>

void init()

{

GLfloat mat\_specular[]={1.0,1.0,1.0,1.0};

GLfloat mat\_shininess[]={50.0};

GLfloat light\_position[]={1.0,1.0,1.0,0.0};

glClearColor(0.0,0.0,1.0,0.0);

glShadeModel(GL\_SMOOTH);

glMaterialfv(GL\_FRONT,GL\_SPECULAR,mat\_specular);

glMaterialfv(GL\_FRONT,GL\_SHININESS,mat\_shininess);

glLightfv(GL\_LIGHT0,GL\_POSITION,light\_position);

glEnable(GL\_LIGHTING);

glEnable(GL\_LIGHT0);

glEnable(GL\_DEPTH\_TEST);

}

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT|GL\_DEPTH\_BUFFER\_BIT);

glutSolidSphere(1.0,20,16);

glFlush();

}

void reshape(int w,int h)

{

glViewport(0,0,(GLsizei)w,(GLsizei)h);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

if(w<=h)

glOrtho(-1.5,1.5,-1.5\*(GLfloat)h/(GLfloat)w,1.5\*(GLfloat)h/(GLfloat)w,-10.0,10.0);

else

glOrtho(-1.5\*(GLfloat)w/(GLfloat)h,1.5\*(GLfloat)w/(GLfloat)h,-1.5,1.5,-10.0,10.0);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

}

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

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB|GLUT\_DEPTH);

glutInitWindowSize(500,500);

glutInitWindowPosition(100,100);

glutCreateWindow(argv[0]);

init();

glutDisplayFunc(display);

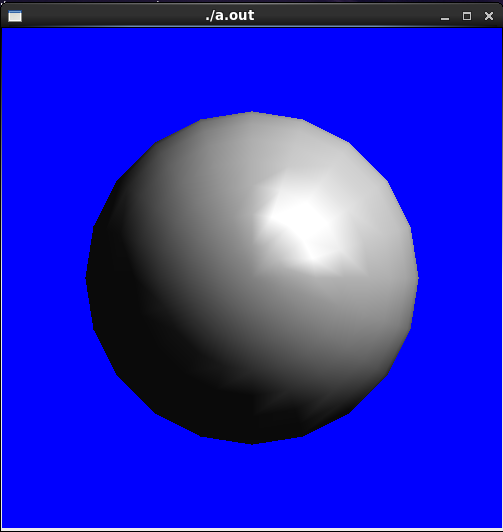
glutReshapeFunc(reshape);

glutMainLoop();

return 0;

}

**OUTPUT:**



**PROGRAM NO:**

**AIM:**

**PROGRAM:**

#include<GL/gl.h>

#include<GL/glut.h>

void mydraw()

{

glClearColor(0.0,0.0,0.0,0.0);

glClear(GL\_COLOR\_BUFFER\_BIT|GL\_DEPTH\_BUFFER\_BIT);

glColor3f(0.0,1.0,0.0);

glutSolidCube(7.0);

glTranslatef(0.0,0.0,-20.0);

glColor3f(0.0,0.0,1.0);

glutSolidTeapot(7.0);

glutSwapBuffers();

}

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

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_DOUBLE|GLUT\_RGB|GLUT\_DEPTH);

glutInitWindowSize(400,300);

glutCreateWindow("Depth Test");

glClearColor(0.0,0.0,0.0,1.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluPerspective(45.0,1.333,0.01,10000.0);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

glTranslatef(0,0,-30);

glEnable(GL\_DEPTH\_TEST);

glutDisplayFunc(mydraw);

glutMainLoop();

return 0;

}

**OUTPUT:**

