Develop a program to demonstrate basic geometric operations on the 3D object

#include <stdio.h>

#include <stdlib.h>

#include <GL/glut.h>

typedef float point[3];

/\* initial tetrahedron \*/

point v[]={{0.0, 0.0, 1.0}, {0.0, 0.942809, -0.33333},

{-0.816497, -0.471405, -0.333333}, {0.816497, -0.471405, -0.333333}};

static GLfloat theta[] = {0.0,0.0,0.0};

int n;

void triangle( point a, point b, point c)

/\* display one triangle using a line loop for wire frame, a single

normal for constant shading, or three normals for interpolative shading \*/

{

glBegin(GL\_POLYGON);

glNormal3fv(a);

glVertex3fv(a);

glVertex3fv(b);

glVertex3fv(c);

glEnd();

}

/\* triangle subdivision using vertex numbers

righthand rule applied to create outward pointing faces \*/

void divide\_triangle(point a, point b, point c, int m)

{

point v1, v2, v3;

int j;

if(m>0)

{

for(j=0; j<3; j++) v1[j]=(a[j]+b[j])/2;

for(j=0; j<3; j++) v2[j]=(a[j]+c[j])/2;

for(j=0; j<3; j++) v3[j]=(b[j]+c[j])/2;

divide\_triangle(a, v1, v2, m-1);

divide\_triangle(c, v2, v3, m-1);

divide\_triangle(b, v3, v1, m-1);

}

else(triangle(a,b,c)); /\* draw triangle at end of recursion \*/

}

/\* Apply triangle subdivision to faces of tetrahedron \*/

void tetrahedron( int m)

{

glColor3f(1.0,0.0,0.0);

divide\_triangle(v[0], v[1], v[2], m);

glColor3f(0.0,1.0,0.0);

divide\_triangle(v[3], v[2], v[1], m);

glColor3f(0.0,0.0,1.0);

divide\_triangle(v[0], v[3], v[1], m);

glColor3f(0.0,0.0,0.0);

divide\_triangle(v[0], v[2], v[3], m);

}

void display()

{

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

glLoadIdentity();

tetrahedron(3);

glFlush();

}

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);

glutPostRedisplay();

}

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

{

int i = 0;

printf("Enter value of N:");

scanf("%d", &i);

n = i;

glutInit(&argc, argv);

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

glutInitWindowSize(500, 500);

glutCreateWindow("3D tetrahedron Gasket");

glutReshapeFunc(myReshape);

glutDisplayFunc(display);

glEnable(GL\_DEPTH\_TEST);

glClearColor (1.0, 1.0, 1.0, 1.0);

glutMainLoop();

}

 