Develop a program to demonstrate basic geometric operations on the 2D object

#include <stdio.h>

#include <GL/glut.h>

typedef float point2[2];

/\* initial triangle \*/

point2 v[]={{-1.0, -0.58}, {1.0, -0.58}, {0.0, 1.15}};

int n;

/\* display one triangle \*/

void triangle( point2 a, point2 b, point2 c)

{

glBegin(GL\_TRIANGLES);

glVertex2fv(a);

glVertex2fv(b);

glVertex2fv(c);

glEnd();

}

void divide\_triangle(point2 a, point2 b, point2 c, int m)

{

/\* triangle subdivision using vertex numbers \*/

point2 v0, v1, v2;

int j;

if(m>0)

{

for(j=0; j<2; j++) v0[j]=(a[j]+b[j])/2;

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

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

divide\_triangle(a, v0, v1, m-1);

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

divide\_triangle(b, v2, v0, m-1);

}

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

}

void display(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT);

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

glFlush();

}

void myinit()

{

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(-2.0, 2.0, -2.0, 2.0);

glMatrixMode(GL\_MODELVIEW);

glClearColor (1.0, 1.0, 1.0, 1.0);

glColor3f(0.0,0.0,0.0);

}

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

{

printf(" No. of Subdivisions : ");

scanf("%d",&n);

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB );

glutInitWindowSize(500, 500);

glutCreateWindow("Sierpinski Gasket 2D triangle");

glutDisplayFunc(display);

myinit();

glutMainLoop();

}

 