7. Program to recursively subdivides a tetrahedron to form 3D Sierpinski gasket. The number of recursive steps is to be specified by the user.

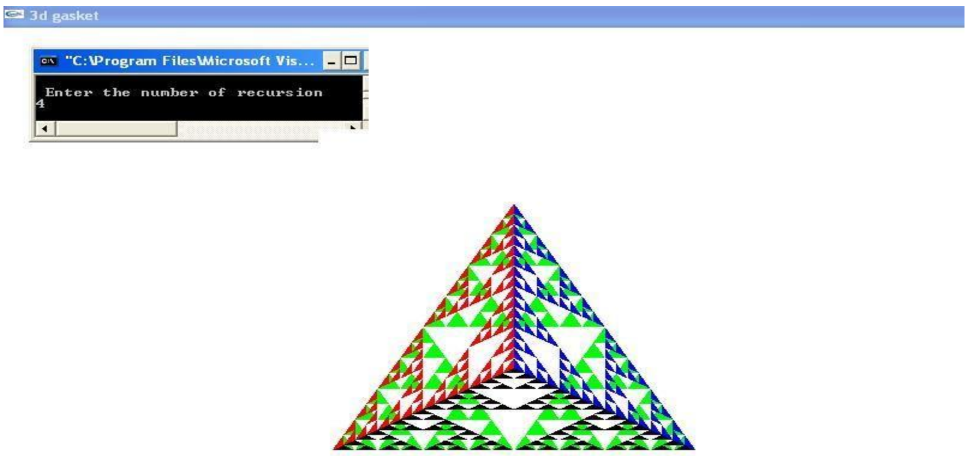
Objective:
In this program, students will learn to create window and to draw 3D Sierpinski gasket using openGL functions.

Program:

```
//Lab7: 3D sierpinski gasket
 #include<windows.h>
#include <stdio.h>
#include <GL/glut.h>

typedef float point[3];

/* initial tetrahedron */
point v[]={{0, 0, 10}, {0, 10, 0},{-10, -5, 0}, {10, -5, 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_TRIANGLES);
        glVertex3fv(a);
        glVertex3fv(b);
        glVertex3fv(c);
        glEnd();
        glFlush();
}



void divide_triangle(point a, point b, point c, int m)
{
/* triangle subdivision using vertex numbers
righthand rule applied to create outward pointing faces */
        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 */
}
```

```
 void tetrahedron( int m)
 {
 /* Apply triangle subdivision to faces of tetrahedron */
         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(void)
 {
         glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
         tetrahedron(n);
 }
 void myinit()
 {
         glOrtho(-20,20,-20,20,-20,20);
         glEnable(GL_DEPTH_TEST);
         glClearColor (0,1,1,1);
 }
 int main(int argc, char **argv)
 {
         printf("\nEnter no. of division:");
         scanf("%d",&n);
         glutInit(&argc, argv);
         glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
         glutInitWindowSize(500, 500);
         glutCreateWindow("3D Gasket");
         myinit();
         glutDisplayFunc(display);
         glutMainLoop();
         return(0);
}
```

Output:



```
Enter the number of recursion
4
```