

```

#include <stdlib.h> #include <stdio.h> #include <GL/glut.h>
typedef GLfloat point[3];
point v[]={{-1.0,-0.5,0.0},{1.0,-0.5,0.0},{0.0,1.0,0.0},
{0.0,0.0,1.0}};
GLfloat colors[4][3]={1.0,0.0,0.0},{0.0,1.0,0.0},{0.0,0.0,1.0},
{1.0,1.0,0.0}};
int n;
void triangle(point a,point b,point c
{ glBegin(GL_POLYGON);
glVertex3fv(a); glVertex3fv(b); glVertex3fv(c); glEnd();
} void tetra(point a,point b,point c,point d)
{ glColor3fv(colors[0]); triangle(a,b,c);
glColor3fv(colors[1]); triangle(a,c,d);
glColor3fv(colors[2]); triangle(a,d,b);
glColor3fv(colors[3]); triangle(b,d,c);
void divide_tetra(point a,point b,point c,point d,int m)
{ point mid[6]; int j;
if(m>0)
{ for(j=0;j<3;j++) {
mid[0][j]=(a[j]+b[j])/2.0; mid[1][j]=(a[j]+c[j])/2.0;
mid[2][j]=(a[j]+d[j])/2.0; mid[3][j]=(b[j]+c[j])/2.0;
mid[4][j]=(c[j]+d[j])/2.0; mid[5][j]=(b[j]+d[j])/2.0;
} divide_tetra(a,mid[0],mid[1],mid[2],m-1);
divide_tetra(mid[0],b,mid[3],mid[5],m-1);
divide_tetra(mid[1],mid[3],c,mid[4],m-1);
divide_tetra(mid[2],mid[5],mid[4],d,m-1); }

```

```

else
tetra(a,b,c,d); } void display()
{ glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
glClearColor(1.0,1.0,1.0,1.0);
divide_tetra(v[0],v[1],v[2],v[3],n); glFlush(); }
void myReshape(int w,int h)
{ glViewport(0,0,w,h);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
if(w<=h)
glOrtho(-1.0,1.0,-1.0*((GLfloat)h/(GLfloat)w),
1.0*((GLfloat)h/(GLfloat)w),-1.0,1.0);
else
glOrtho(1.0*((GLfloat)w/(GLfloat)h),1.0*((GLfloat)w/(GLfloat)h),-1.0,1.0,-1.0,1.0);
glMatrixMode(GL_MODELVIEW); glutPostRedisplay(); }
void main(int argc,char ** argv)
{ printf( "No of Division?: ");
scanf("%d",&n);
glutInit(&argc,argv);
glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB|GLUT_DEPTH);
glutInitWindowSize(500,500); glutCreateWindow( "3D
gasket" ); glutDisplayFunc(display);
glutReshapeFunc(myReshape); glEnable(GL_DEPTH_TEST);
glutMainLoop(); }

```