

```

#include <stdio.h>
#include<math.h>

static int w1W=600;
static int w1H=500;
static float height=100.0f;
static float width=20.0f;
static float left_alpha=2.0f;
static float right_alpha=2.2f;
static float left_angle=16.0f;
static float right_angle=16.0f;
int level =0;//14
static float left_width_factor;
static float right_width_factor;
static float left_height_factor;
static float right_height_factor;
static float x,y,x1,y01;
static float turtle_x,turtle_y ,turtle_r,turtle_theta;

static void turn(float angle);
static float point (float x1,float y01,float x2,float y2);
static void step();
static void drawbranch(float width1,int x0,int y0,int x1,int y1)
{
if (width1< 1.5f)
    {
        glLineWidth(1.0);
        glBegin(GL_LINES);
            glVertex2f(x0,y0);          glVertex2f(x1,y1);
        glEnd();
    }
    glLineWidth(width1);
    glBegin(GL_LINES);
        glVertex2f(x0,y0);      glVertex2f(x1,y1);
    glEnd();
    return;
}
static void generate(float x,float y,float widtht,float height, float angle,int level)
{ float x1,y01;
    turtle_x=x;    turtle_y=y;    turtle_r=height;
    step();
    x1=turtle_x;    y01=turtle_y;    level--;
    if(level<3)
    { glColor3f(0.0,1.0,0.0);
    drawbranch(widtht,(int)x,(int)y,(int)x1,(int)y01);}
    else
    { glColor3f(0.6,0.0,0.0);
    drawbranch(widtht,(int)x,(int)y,(int)x1,(int)y01);}
}

```

```

    if(level>0)
    {
        turtle_theta=point(x,y,x1,y01);        turn (left_angle);
        generate(turtle_x,turtle_y,left_width_factor*widtht,left_height_factor*height,left_angle,level);
        turtle_theta=point(x,y,x1,y01);        turn(-right_angle);
        generate (x1,y01,left_width_factor*widtht,
                left_height_factor*height,right_angle,level);
    }
} /*end generate*/
static void turn (float angle)
{ turtle_theta +=angle;
} /*end turn*/
static float point (float x1,float y01,float x2, float y2)
{ float theta;
    if((x2-x1)==0.0)
    { if(y2>y01)theta=90.0f;
      else
          theta=270.0f;
    }
    else
    {          theta= atan((y2-y01)/(x2-x1))*57.295779f;    }
    if(x1>x2)theta+=180.0f;
    return theta;
} /*end point*/
static void step()
{
    turtle_x+=turtle_r*cos(turtle_theta*0.017453292);
    turtle_y+=turtle_r*sin(turtle_theta*0.017453292);
} /*end step*/
static void display (void)
{
    glClear(GL_COLOR_BUFFER_BIT);glLoadIdentity();glColor3f(.6,.0,.0);
    //level = leve_in;
    left_width_factor= pow(2.0,-1.0/left_alpha);
    right_width_factor= pow(2.0,-1.0/right_alpha);
    left_height_factor= pow(2.0,-2.0/(3*left_alpha));
    right_height_factor= pow(2.0,-2.0/(3*right_alpha));

    x=(float)w1W/2.0f;    y=(float)w1H/10.0f;

    x1=x;  y01= y+height;
    drawbranch(width,(int)x,(int)y,(int)x1,(int)y01);
    turtle_r=height;turtle_theta=point(x,y,x1,y01);
    turtle_x=x;    turtle_y=y;

    turn(left_angle);
    generate(x1,y01,left_width_factor*width,left_height_factor*height, right_angle,level);
}

```

```

        turtle_theta=point(x,y,x1,y01);
        turn(-right_angle);
        generate(x1,y01,left_width_factor*width,left_height_factor*height, right_angle,level);
        glFlush();
    } /* end display*/
    static void reshape (int w,int h)/*this routine handles window resizes*/
    {
        w1W=w;      w1H=h;
        glMatrixMode(GL_PROJECTION);      glLoadIdentity();
        gluOrtho2D(0.0,(float)w1W,0.0,(float)w1H);
        glMatrixMode(GL_MODELVIEW);      glViewport(0,0,w1W,w1H);
        //printf("reshape w1H=%d,w1H=%d\n",w1W,w1H);
    }
    void mouse(int button,int state,int x, int y)
    {
        switch(button)
        {
            case GLUT_LEFT_BUTTON:
                if(state==GLUT_DOWN)
                    if(level<24)
                        level++;
                        glutPostRedisplay();
                        break;
            case GLUT_RIGHT_BUTTON:
                if(state==GLUT_DOWN)
                    if(level>0)
                        level--;
                        glutPostRedisplay();
            break;
            default:
                //glutIdleFunc(NULL);
                break;
        }
    }
    static void init()
    {
        glClearColor(1.0,1.0,1.0,0.0);      glColor3f(0.0,0.0,0.0);
        glMatrixMode(GL_PROJECTION);      glLoadIdentity();
        gluOrtho2D(0.0,(float)w1W,0.0,(float)w1H);
        glMatrixMode(GL_MODELVIEW);
        glViewport(0,0,w1W,w1H);
    }
    void main(int argc,char* argv[])
    {
        glutInit(&argc,argv);
        glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
        glutInitWindowSize(w1W,w1H);      glutInitWindowPosition(100,100);
        glutCreateWindow(argv[0]);      glutDisplayFunc(display);
    }

```

```
        glutMouseFunc(mouse);          glutReshapeFunc(reshape);
    init();          glutMainLoop();
}
```