**Program 16(b)**

**Write a program to implement fractal triangle design.**

#include<stdio.h>

#include<math.h>

#include<graphics.h>

#define ROUND(a) ((int)(a+0.5))

void ddaline(int x1, int y1, int x2, int y2)

{

float xsteps, ysteps, x=x1, y=y1;

int dx = x2-x1;

int dy = y2-y1;

int steps,k=1;

if(abs(dx)>=abs(dy))

steps=abs(dx);

else steps=abs(dy);

xsteps= dx/(float)steps;

ysteps= dy/(float)steps;

putpixel(ROUND(x),ROUND(y),15);

while(k<=steps)

{

x+=xsteps;

y+=ysteps;

putpixel(ROUND(x), ROUND(y),15);

k++;

}

}

void triangle(int x1, int y1, int x2, int y2)

{

float angle;

int x11,y11,x21,y21,x12,y12,x22,y22;

angle=60\*3.142/180;

x11=(x2+(2\*x1))/3;

y11=(y2+(2\*y1))/3;

x21=(x1+(2\*x2))/3;

y21=(y1+(2\*y2))/3;

x22=x11+((x21-x11)\*cos(angle)+(y21-y11)\*sin(angle));

y22=y11-((x21-x11)\*sin(angle)-(y21-y11)\*cos(angle));

if((x11-x22)\*(x11-x22)+ (y11-y22)\*(y11-y22)<100)

{

ddaline(x1, y1, x11, y11);

ddaline(x11,y11,x22,y22);

ddaline(x22,y22,x21,y21);

ddaline(x21, y21, x2, y2);

return;

}

triangle(x1, y1, x11, y11);

triangle(x11,y11,x22,y22);

triangle(x22,y22,x21,y21);

triangle(x21, y21, x2, y2);

}

int main()

{

int x1,y1,x2,y2;

int gdriver = DETECT, gmode, errorcode;

initgraph(&gdriver, &gmode, "..\\");

errorcode = graphresult();

if (errorcode != grOk)

{

printf("Graphics error: %s\n", grapherrormsg(errorcode));

printf("Press any key to halt:");

getch();

exit(1);

}

printf("Enter the Co-ordinates\n");

scanf("%d%d%d%d",&x1,&y1,&x2,&y2);

triangle(x1,y1,x2,y2);

getch();

closegraph();

return 0;

}



