**C language – Graphic design**

#include<stdio.h>

#include<dos.h>

#include<graphics.h>

#include<stdlib.h>

#include<math.h>

#include<conio.h>

#define PI 3.14159

#define NUME 30

#define DENOM 100

#define NUMBER 7

#define RAD 2.0

#define DTHETA 0.1

#define REDUX 3

#define SEGS 60

#define MIN 1

void init\_graph();

void tdisplay(int, float, int, int, int);

void cdisplay(int, int, int);

void bucket();

int gd, gm, maxcolor, errorcode;

int xo, yo, maxx, maxy, xasp, yasp;

double aspectratio;

void init\_graph()

{

detectgraph(&gd, &gm);

initgraph(&gd, &gm, "");

errorcode = graphresult();

if (errorcode != grOk) {

printf("graphics system eror : %s", grapherrormsg(errorcode));

exit(1);

}

maxcolor = getmaxcolor() + 1;

getaspectratio(&xasp, &yasp);

aspectratio = (double) (yasp) / (double) (xasp);

maxx = getmaxx();

maxy = getmaxy();

}

void cdisplay(int size, int x, int y)

{

int i, rc;

float theta;

for (i = 0; i < NUMBER; i++) {

theta = i \* 2 \* PI / NUMBER;

rc = random(16);

tdisplay(size, theta, x, y, rc);

}

}

void tdisplay(int size, float theta, int x, int y, int rcolor)

{

int i, chg, newsize;

for (i = 0; i < size; i++) {

chg = (random(DENOM) < NUME) ? -1 : 1;

theta += chg \* DTHETA;

x += RAD \* sin(theta);

y += RAD \* cos(theta);

if (size < 4)

setcolor(rcolor);

else if (size < 13)

setcolor(GREEN);

else if (size < 40)

setcolor(LIGHTGREEN);

else

setcolor(YELLOW);

lineto(x, y);

}

if (size > MIN) {

newsize = size / REDUX;

cdisplay(newsize, x, y);

}

}

void bucket()

{

setcolor(WHITE);

ellipse(xo, yo, 0, 360, 30, 9);

setfillstyle(SOLID\_FILL, RED);

fillellipse(xo, yo, 30, 9);

setfillstyle(LTBKSLASH\_FILL, BROWN);

ellipse(xo + 30, yo + 28, 90, 252, 12, 25);

delay(1150);

ellipse(xo - 30, yo + 28, -70, 94, 12, 25);

delay(1150);

ellipse(xo + 25, yo + 83, -69, 90, 12, 30);

delay(1150);

ellipse(xo - 25, yo + 83, 88, 252, 12, 30);

delay(1150);

ellipse(xo, yo + 116, 151, 388, 32, 9);

delay(1150);

floodfill(xo, yo + 50, WHITE);

}

void main()

{

int size;

init\_graph();

xo = maxx >> 1;

yo = (maxy >> 1) - 80;

xo++;

yo++;

bucket();

randomize();

size = SEGS;

delay(1150);

cdisplay(size, xo, yo);

getch();

closegraph();

}

**OUTPUT**

