**/\*EXPERIMENT Write C++ program to draw the following pattern. Use DDA line and Bresenham circle algorithm. Apply the concept of encapsulation. \*/**

#include <graphics.h>

#include<iostream>

#include <math.h>

using namespace std;

void circlebres(float x1,float y1,float r)

{

float x,y,p;

x=0;

y=r;

p=3-(2\*r); // initial decision parameter

while(x<=y)

{

putpixel(x1+x,y1+y,WHITE); /\* drawing pixel in each octant\*/

putpixel(x1-x,y1+y,WHITE);

putpixel(x1+x,y1-y,WHITE);

putpixel(x1-x,y1-y,WHITE);

putpixel(x1+y,y1+x,WHITE);

putpixel(x1+y,y1-x,WHITE);

putpixel(x1-y,y1+x,WHITE);

putpixel(x1-y,y1-x,WHITE);

x=x+1;

if(p<0)

{

p=p+4\*(x)+6;

}

else

{

p=p+4\*(x-y)+10;

y=y-1;

}

delay(40);

}

}

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

{

int dx,dy,m,s;

float xi,yi,x,y;

dx = x2 - x1;

dy = y2 - y1;

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

s = abs(dx);

else

s = abs(dy);

xi = dx / (float) s;

yi = dy / (float) s;

x = x1;

y = y1;

putpixel(x1, y1, WHITE);

for (m = 0; m < s; m++)

{

putpixel(x, y, WHITE);

x += xi;

y += yi;

}

delay(500);

}

int main()

{

int xc,yc,r;

cout<<" enter center coordinates : ";

cin>>xc>>yc;

cout<<"enter redius : ";

cin>>r;

int gd=DETECT,gm=DETECT,x1,y1,x2,y2;

initgraph(&gd,&gm,NULL);

circlebres(xc,yc,r); //inside circle

double height,side;

//side=r/0.577;

//height=1.73\*side;

side=1.73\*r;

height=1.73\*side;

drawline(xc-side,yc+r,xc+side,yc+r); //base line

delay(300);

drawline(xc-side,yc+r,xc,yc+r-height);// left line

drawline(xc,yc+r-height,xc+side,yc+r); // right line

circlebres(xc,yc,height-r);//outer circle

delay(3000);

closegraph();

}