Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Technology

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

**Assignment -2**

|  |  |
| --- | --- |
| **Subject** | CGVR |
| **Name** | Parth Petkar |
| **Class** | CS-C |
| **Roll No.** | 76 |

# Aim Program for drawing 3 patterns using C language

1. ***Wave Graph***

void FirstHalfWaySymmetricPlot(int xc, int yc, int x, int y)

{

putpixel(x + xc, y + yc, RED);

putpixel(-x + xc, y + yc, BLUE);

putpixel(y + xc, x + yc, RED);

putpixel(-y + xc, x + yc, BLUE);

}

void SecondHalfWaySymmetricPlot(int xc, int yc, int x, int y)

{

putpixel(x + xc, -y + yc, YELLOW);

putpixel(-x + xc, -y + yc, GREEN);

putpixel(y + xc, -x + yc, YELLOW);

putpixel(-y + xc, -x + yc, GREEN);

}

void BottemCircle(int xc, int yc, int r)

{

int x = 0, y = r, d = 3 - (2 \* r);

FirstHalfWaySymmetricPlot(xc, yc, x, y);

while (x <= y)

{

if (d <= 0)

{

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

}

else

{

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

y = y - 1;

}

x = x + 1;

FirstHalfWaySymmetricPlot(xc, yc, x, y);

}

}

void TopCircle(int xc, int yc, int r)

{

int x = 0, y = r, d = 3 - (2 \* r);

SecondHalfWaySymmetricPlot(xc, yc, x, y);

while (x <= y)

{

if (d <= 0)

{

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

}

else

{

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

y = y - 1;

}

x = x + 1;

SecondHalfWaySymmetricPlot(xc, yc, x, y);

}

}

void BresenhamCircle(int xc, int yc, int r)

{

BottemCircle(xc, yc, r);

xc += 2\*r;

TopCircle(xc, yc, r);

}

int main()

{

int gdriver = DETECT, gmode, xc, yc, r;

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

printf("Enter the values of xc and yc :");

scanf\_s("%d%d", &xc, &yc);

printf("Enter the value of radius :");

scanf\_s("%d", &r);

BresenhamCircle(xc,yc,r);

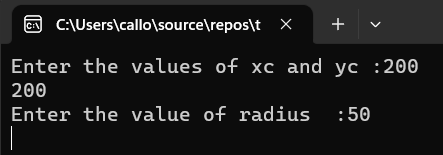
getch();

closegraph();

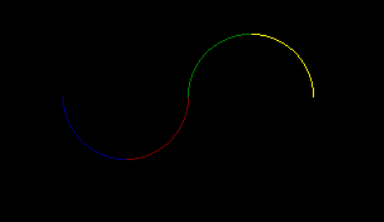
    return 0;

}

**User input:**



**Output:**



1. ***Flower Pattern***

#include <stdio.h>

#include "graphics.h"

void FirstHalfWaySymmetricPlot(int xc, int yc, int x, int y)

{

putpixel(x + xc, y + yc, RED);

putpixel(-x + xc, y + yc, BLUE);

putpixel(y + xc, x + yc, RED);

putpixel(-y + xc, x + yc, BLUE);

}

void SecondHalfWaySymmetricPlot(int xc, int yc, int x, int y)

{

putpixel(x + xc, -y + yc, YELLOW);

putpixel(-x + xc, -y + yc, GREEN);

putpixel(y + xc, -x + yc, YELLOW);

putpixel(-y + xc, -x + yc, GREEN);

}

void LeftHalfWaySymmetricPlot(int xc, int yc, int x, int y) {

putpixel(-x + xc, -y + yc, GREEN);

putpixel(-x + xc, y + yc, BLUE);

putpixel(-y + xc, -x + yc, GREEN);

putpixel(-y + xc, x + yc, BLUE);

}

void RightHalfWaySymmetricPlot(int xc, int yc, int x, int y) {

putpixel(x + xc, y + yc, RED);

putpixel(x + xc, -y + yc, YELLOW);

putpixel(y + xc, x + yc, RED);

putpixel(y + xc, -x + yc, YELLOW);

}

void TopCircle(int xc, int yc, int r)

{

int x = 0, y = r, d = 3 - (2 \* r);

yc -= r;

SecondHalfWaySymmetricPlot(xc, yc, x, y);

while (x <= y)

{

if (d <= 0)

{

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

}

else

{

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

y = y - 1;

}

x = x + 1;

SecondHalfWaySymmetricPlot(xc, yc, x, y);

}

}

void LeftCircle(int xc, int yc, int r)

{

int x = 0, y = r, d = 3 - (2 \* r);

xc -= r;

LeftHalfWaySymmetricPlot(xc, yc, x, y);

while (x <= y)

{

if (d <= 0)

{

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

}

else

{

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

y = y - 1;

}

x = x + 1;

LeftHalfWaySymmetricPlot(xc, yc, x, y);

}

}

void BottemCircle(int xc, int yc, int r)

{

int x = 0, y = r, d = 3 - (2 \* r);

yc += r;

FirstHalfWaySymmetricPlot(xc, yc, x, y);

while (x <= y)

{

if (d <= 0)

{

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

}

else

{

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

y = y - 1;

}

x = x + 1;

FirstHalfWaySymmetricPlot(xc, yc, x, y);

}

}

void RightCircle(int xc, int yc, int r)

{

int x = 0, y = r, d = 3 - (2 \* r);

xc += r;

RightHalfWaySymmetricPlot(xc, yc, x, y);

while (x <= y)

{

if (d <= 0)

{

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

}

else

{

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

y = y - 1;

}

x = x + 1;

RightHalfWaySymmetricPlot(xc, yc, x, y);

}

}

int main()

{

int gdriver = DETECT, gmode, xc, yc, r;

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

printf("Enter the values of xc and yc :");

scanf\_s("%d%d", &xc, &yc);

printf("Enter the value of radius :");

scanf\_s("%d", &r);

TopCircle(xc,yc,r);

LeftCircle(xc,yc,r);

BottemCircle(xc, yc, r);

RightCircle(xc, yc, r);

getch();

closegraph();

    return 0;

}

**User Input:**

A screenshot of a computer

Description automatically generated

**Output:**

A colorful circle with black background

Description automatically generated

1. ***Spiral***

#include "graphics.h"

#include <iostream>

#include <stdlib.h>

#include <stdio.h>

void main()

{

int p,r,x,y,r1;

int gd=DETECT,gm;

initgraph(&gd,&gm,"C:/TURBOC3/BGI");

r=10;

while(r<=100)

{

p=3-2\*r;

x=0;y=r;

r=r+20;

while(y>=x)

{

putpixel(200-x,200-y,5);

putpixel(200-y,200-x,5);

putpixel(200+y,200-x,5);

putpixel(200+x,200-y,5);

x++;

if(p>0)

{

y--;

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

}

else

{

p=p+4\*x+6;

}

}

}

r1=20;

while(r1<=100)

{

p=3-2\*r1;

x=0;y=r1;

r1=r1+20;

while(y>=x)

{

putpixel(210+x,200+y,5);

putpixel(210+y,200+x,5);

putpixel(210-y,200+x,5);

putpixel(210-x,200+y,5);

/

x++;

if(p>0)

{

y--;

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

}

else

{

p=p+4\*x+6;

}

}

}

getch();

}

**Output:**

A purple spiral on a black background

Description automatically generated