**Final Practical**

**Name :** Prasad Arun Dhobale

**Roll No :** 239

**PRN :** 202202040016

**Problem Statement :**

Develop a program for polygon filling using the boundary fill and flood fill algorithm.

**Input :**

#include<iostream>

#include <graphics.h>

using namespace std;

void boundaryFill4(int x, int y, int fill\_color,int boundary\_color)

{

if((getpixel(x, y) != boundary\_color) && (getpixel(x, y) != fill\_color))

{

putpixel(x, y, fill\_color);

boundaryFill4(x + 1, y, fill\_color, boundary\_color);

boundaryFill4(x, y + 1, fill\_color, boundary\_color);

boundaryFill4(x - 1, y, fill\_color, boundary\_color);

boundaryFill4(x, y - 1, fill\_color, boundary\_color);

}

}

FloodFill4(int x,int y,int n\_cl, int o\_cl)

{

if(getpixel(x,y)==o\_cl)

{

putpixel(x,y,o\_cl);

FloodFill4(x+1,y,n\_cl,o\_cl);

FloodFill4(x-1,y,n\_cl,o\_cl);

FloodFill4(x,y+1,n\_cl,o\_cl);

FloodFill4(x,y-1,n\_cl,o\_cl);

}

}

int main()

{

initwindow(1000, 1000);

cout<<"Boundary Fill : "<<endl;

int x = 200, y = 220, radius = 50;

circle(300, 200, 200);

circle(x, y, radius);

boundaryFill4(x, y, 6, 15);

cout<<"Flood Fill : "<<endl;

x = 400, y = 200, radius = 50;

circle(x, y, radius);

FloodFill4(x, y, 9, 0);

getch();

closegraph();

return 0;

}

**Output :**

****