**CG Practical 4: Flood Fill Algorithm**

Name: Vivian Vijay Ludrick

Branch : S.E. Comps A Batch C

Roll No: **9914**

**FLOOD FILL:**

#include <stdio.h>

#include <graphics.h>

#include <conio.h>

void floodFill(int *x*, int *y*, int *fillc*, int *oldc*)

{

if (getpixel(*x*, *y*) == *oldc*)

{

putpixel(*x*, *y*, *fillc*);

floodFill(*x* + 1, *y*, *fillc*, *oldc*);

floodFill(*x* - 1, *y*, *fillc*, *oldc*);

floodFill(*x*, *y* + 1, *fillc*, *oldc*);

floodFill(*x*, *y* - 1, *fillc*, *oldc*);

}

}

int main()

{

int x, y, r;

int gd = DETECT, gm;

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

x = 200;

y = 200;

r = 25;

circle(x, y, r);

floodFill(x, y, WHITE, BLACK);

floodFill(x, y, RED, WHITE);

floodFill(x, y, BLUE, RED);

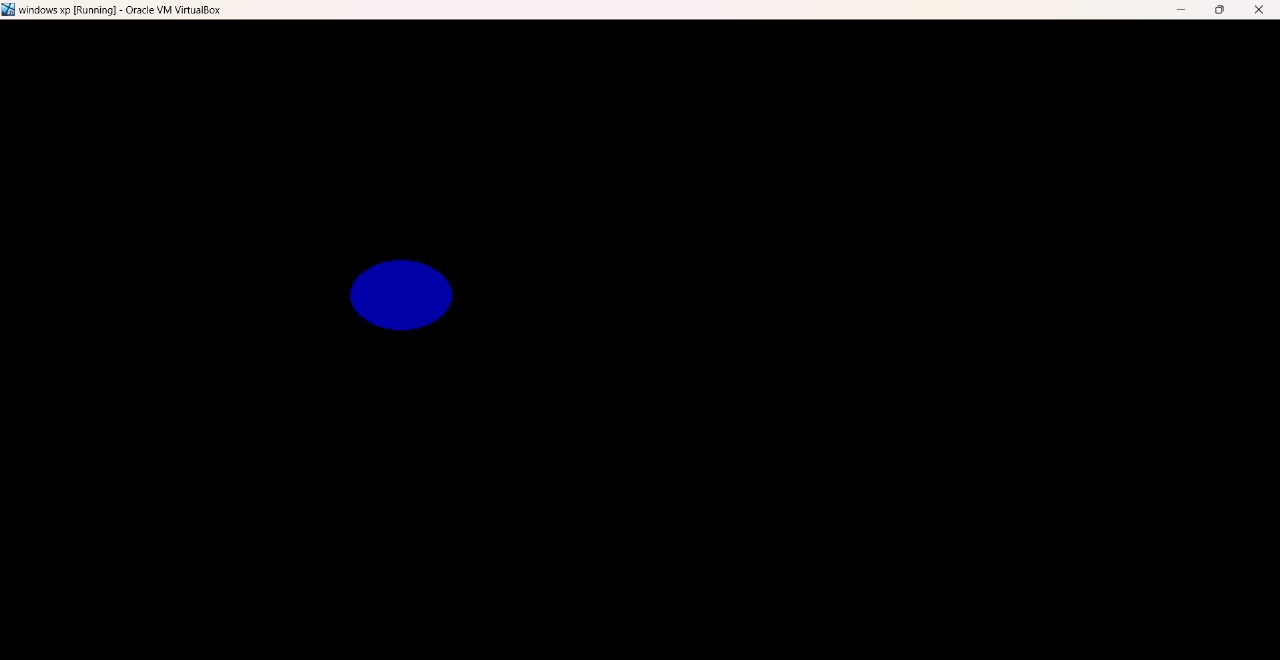
getch();

closegraph();

return 0;

}

**OUTPUT :**

****

**BOUNDARY FILL:**

#include <stdio.h>

#include <graphics.h>

void boundaryfill(int *x*, int *y*, int *f\_color*, int *b\_color*)

{

if (getpixel(*x*, *y*) != *b\_color* && getpixel(*x*, *y*) != *f\_color*)

{

putpixel(*x*, *y*, *f\_color*);

boundaryfill(*x* + 1, *y*, *f\_color*, *b\_color*);

boundaryfill(*x*, *y* + 1, *f\_color*, *b\_color*);

boundaryfill(*x* - 1, *y*, *f\_color*, *b\_color*);

boundaryfill(*x*, *y* - 1, *f\_color*, *b\_color*);

}

}

*// getpixel(x,y) gives the color of specified pixel*

int main()

{

int gm, gd = DETECT, radius;

int x, y;

printf("Enter x and y positions for circle\n");

scanf("%d%d", &x, &y);

printf("Enter radius of circle\n");

scanf("%d", &radius);

initgraph(&gd, &gm, "c:\\turboc3\\bgi");

circle(x, y, radius);

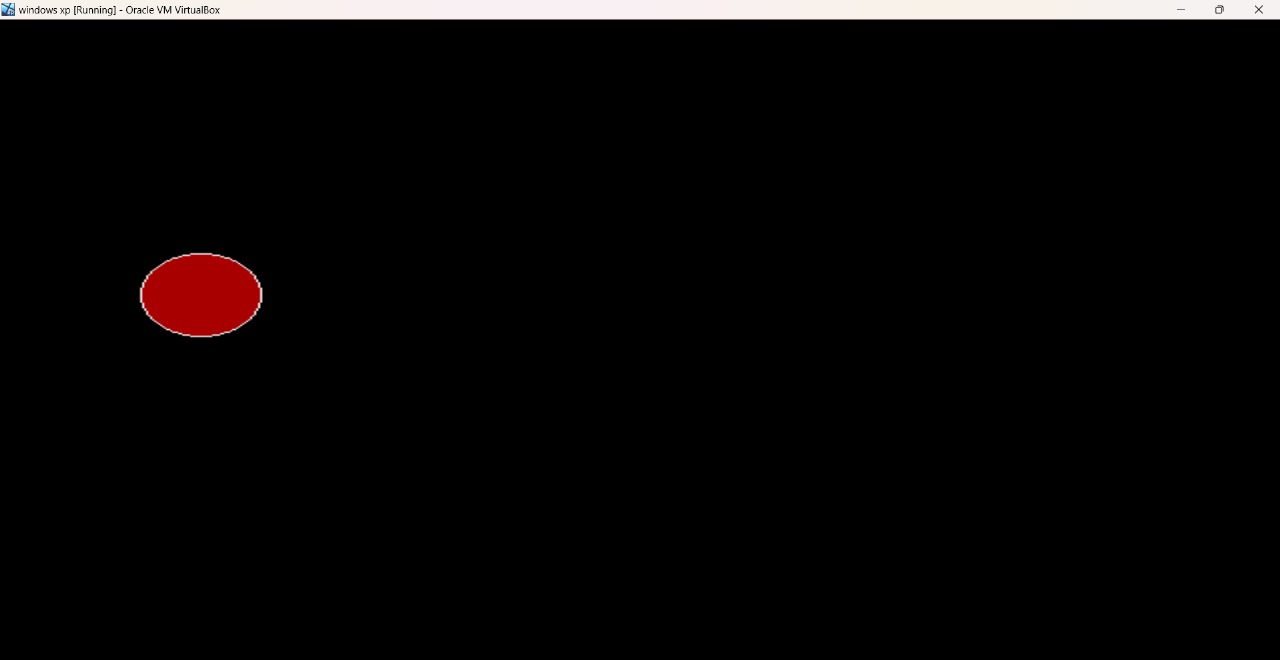
boundaryfill(x, y, 4, 15);

closegraph();

return 0;

}

**OUTPUT:**

****