NAME- Deepankar Sharma
COURSE- BCA
ROLL NO- 2092014
SUBJECT- Computer graphics lab

PRACTICLE-14

OBJECTIVE- To implement Boundary Fill Algorithm through graphics.

SYNTAX:-

```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
#include <dos.h>
void flood(int, int, int, int);

void boundary_fill(int pos_x, int pos_y, int fill_color, int
boundary_color)
{
    int current_color = getpixel(pos_x, pos_y);
// get the color of the current pixel position
    if (current_color != boundary_color && current_color !=
fill_color) // if pixel not already filled or part of the boundary
then
    {
        putpixel(pos_x, pos_y, fill_color);                        //
change the color for this pixel to the desired fill_color
        boundary_fill(pos_x + 1, pos_y, boundary_color, fill_color);
// perform same function for the east pixel
        boundary_fill(pos_x - 1, pos_y, boundary_color, fill_color);
// perform same function for the west pixel
        boundary_fill(pos_x, pos_y + 1, boundary_color, fill_color);
// perform same function for the north pixel
        boundary_fill(pos_x, pos_y - 1, boundary_color, fill_color);
// perform same function for the south pixel
    }
}


int main()
{
    int gd = DETECT, gm;

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

    setcolor(RED);

    rectangle(50, 50, 250, 250);

    // flood(55, 55, 10, 0);
```

```
    boundary_fill(105, 200, YELLOW, RED);

    getch();
}
void flood(int x, int y, int fillColor, int defaultColor)
{

    if (getpixel(x, y) == defaultColor)
    {

        // delay(1);

        putpixel(x, y, fillColor);

        flood(x + 1, y, fillColor, defaultColor);

        flood(x - 1, y, fillColor, defaultColor);

        flood(x, y + 1, fillColor, defaultColor);

        flood(x, y - 1, fillColor, defaultColor);
    }
}
```

Output: