PRACTICAL – 5

PROGRAM -1

AIM- WAPP TO PERFORM BOUNDARY FILL ALGORITHM.

CODE

```
import matplotlib.pyplot as plt
import numpy as np
def boundary_fill(x, y, fill_color, boundary_color, screen):
    if (x < 0 or x >= len(screen[0]) or y < 0 or y >= len(screen) or
        screen[y][x] == fill_color or screen[y][x] == boundary_color):
        return
    screen[y][x] = fill_color
    boundary_fill(x + 1, y, fill_color, boundary_color, screen)
    boundary_fill(x - 1, y, fill_color, boundary_color, screen)
    boundary_fill(x, y + 1, fill_color, boundary_color, screen)
    boundary_fill(x, y - 1, fill_color, boundary_color, screen)
def create_plot(screen, colors):
    height = len(screen)
    width = len(screen[0])
    data = np.zeros((height, width, 3), dtype=np.uint8)
    for y in range(height):
        for x in range(width):
            data[y, x] = colors[screen[y][x]]
    plt.imshow(data)
    plt.axis('off')
    plt.show()
# Initialize the screen
width, height = 10, 10
screen = [['blue' for i in range(width)] for j in range(height)]
# Define colors
colors = {
    'black': [0, 0, 0],
    'blue': [0, 0, 255]
}
# Draw a rectangle
for i in range(width):
    screen[0][i] = 'black'
    screen[height-1][i] = 'black'
for i in range(height):
    screen[i][0] = 'black'
    screen[i][width-1] = 'black'
# Apply the boundary fill algorithm
boundary_fill(5, 4, 'blue', 'black', screen)
# Create and show the plot
```

create_plot(screen, colors)

OUTPUT