// C Implementation for Boundary Filling Algorithm

#include <graphics.h>

#include <iostream>

using namespace std;

// Function for 4 connected Pixels

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);

}

}

//driver code

int main()

{

// gm is Graphics mode which is a computer display mode that generates image using pixels.

// DETECT is a macro defined in "graphics.h" header file

int gd = DETECT, gm;

// initgraph initializes the graphics system by loading a graphics driver from disk

initgraph(&gd, &gm, NULL);

// rectangle coordinate

int top, left, bottom, right;

top = left = 50;

bottom = right = 300;

// rectangle for print rectangle

rectangle(left, top, right, bottom);

// filling start coordinate

int x = 51;

int y = 51;

// Function calling

boundaryFill4(x, y, 6, 15);

delay(100);

getch();

// closegraph function closes the graphics mode and deallocates all memory allocated by graphics system .

closegraph();

return 0;

}