# ASSIGNMENT NO. 2

**NAME :-Omkar Satardekar**

**Roll No :- 123B2B327**

#include <graphics.h> #include <conio.h>

// Function to draw the house at a specified position with width and height void drawhouse(int x, int y, int w, int h) {

setcolor(BLUE); // Set the boundary color to blue

rectangle(x, y, x + w, y + h); // Base rectangle (walls) line(x, y, x + w / 2, y - h / 2); // Left side roof

line(x + w, y, x + w / 2, y - h / 2); // Right side roof

}

// Seed fill algorithm (Flood Fill)

void seedFill(int x, int y, int fillColor, int boundaryColor) { int currentColor = getpixel(x, y);

if (currentColor != boundaryColor && currentColor != fillColor) { putpixel(x, y, fillColor); // Fill the current pixel

// Recursively fill neighboring pixels

seedFill(x + 1, y, fillColor, boundaryColor); // Right seedFill(x - 1, y, fillColor, boundaryColor); // Left seedFill(x, y + 1, fillColor, boundaryColor); // Down seedFill(x, y - 1, fillColor, boundaryColor); // Up

}

}

// Boundary fill algorithm

void boundaryFill(int x, int y, int fillColor, int boundaryColor) {

int currentColor = getpixel(x, y);

if (currentColor != boundaryColor && currentColor != fillColor) { putpixel(x, y, fillColor); // Fill the current pixel

// Recursively fill neighboring pixels

boundaryFill(x + 1, y, fillColor, boundaryColor); // Right boundaryFill(x - 1, y, fillColor, boundaryColor); // Left boundaryFill(x, y + 1, fillColor, boundaryColor); // Down boundaryFill(x, y - 1, fillColor, boundaryColor); // Up

}

}

int main() {

int gd = DETECT, gm;

// Initialize graphics mode (Change the path if necessary) initgraph(&gd, &gm, "C:\\Turboc3\\bgi");

int x = 200, y = 200; // Starting coordinates for the house int w = 100, h = 100; // Width and height of the house

// Draw the house drawhouse(x, y, w, h);

// Set the color for the fill (desired color for filling) int floodFillColor = RED; // Color for Flood Fill

int boundaryColor = BLUE; // Boundary color is blue

// Seed point is inside the house

int seedX = x + 10; // A point inside the house (a bit inside the boundary) int seedY = y + 10;

// Apply Flood Fill algorithm to fill the house seedFill(seedX, seedY, floodFillColor, boundaryColor);

// Optionally, use Boundary Fill algorithm (inside the house) int boundaryFillColor = GREEN; // Color for Boundary Fill

boundaryFill(seedX, seedY, boundaryFillColor, boundaryColor); // Fill inside the house with boundary fill

getch(); // Wait for a key press closegraph(); // Close graphics mode return 0;

}

# OUTPUT

