1. // Boundary Fill algorithm implemented in C.
2. #include <stdio.h>
3. #include <conio.h>
4. #include <graphics.h>
5. // Function to implement Boundary Fill Algorithm
6. void boundary\_fill(int x, int y, int fill\_color, int boundary\_color) {
7. if (getpixel(x, y) != boundary\_color && getpixel(x, y) != fill\_color) {
   1. putpixel(x, y, fill\_color); // Fill the pixel
   2. boundary\_fill(x + 1, y, fill\_color, boundary\_color); // East
   3. boundary\_fill(x - 1, y, fill\_color, boundary\_color); // West
   4. boundary\_fill(x, y + 1, fill\_color, boundary\_color); // South
   5. boundary\_fill(x, y - 1, fill\_color, boundary\_color); // North
   6. delay(500);

8. }

9. }

1. int main() {
2. int gd = DETECT, gm;
3. //int x, y;

13. int x = 250, y = 200, radius = 10;

1. initgraph(&gd, &gm, "C:\\Turboc3\\BGI"); // Initialize graphics mode
2. setcolor(WHITE) ;
3. circle(x, y, radius);
4. // Call boundary fill algorithm
5. boundary\_fill(x, y, YELLOW, WHITE); // Fill with YELLOW, bounded by WHITE
6. getch(); // Wait for a key press
7. closegraph(); // Close the graphics mode
8. return 0;

22. }