**EXERCISE - 2**

**AIM :** To implement Bresenham's circle-drawing algorithm in C to draw a circle on a graphical interface.**.**

**Procedure (Using Bresenham’s Algorithm)**

1. Input the radius and center coordinates (xc,yc).
2. Initialize parameters:
   * Start from the point (0,r)on the circle.
   * Compute the initial decision parameter:  
     p=3−2r.
3. Plot initial points:
   * Using symmetry, plot points in all 8 octants of the circle based on (x,y).
4. Update decision parameter:
   * If p<0, the next point is (x+1,y). Update p as:  
     p=p+4x+6
   * Otherwise, the next point is (x+1,y−1). Update p as:  
     p=p+4(x−y)+10.
5. Repeat until x≥y:
   * Continue plotting points in all octants.

**SAMPLE CODE:**

#include <stdio.h>

#include <graphics.h>

void plotCirclePoints(int xc, int yc, int x, int y) {

putpixel(xc + x, yc + y, WHITE); // Octant 1

putpixel(xc - x, yc + y, WHITE); // Octant 2

putpixel(xc + x, yc - y, WHITE); // Octant 3

putpixel(xc - x, yc - y, WHITE); // Octant 4

putpixel(xc + y, yc + x, WHITE); // Octant 5

putpixel(xc - y, yc + x, WHITE); // Octant 6

putpixel(xc + y, yc - x, WHITE); // Octant 7

putpixel(xc - y, yc - x, WHITE); // Octant 8

}

void bresenhamCircle(int xc, int yc, int r) {

int x = 0, y = r;

int p = 3 - 2 \* r; // Initial decision parameter

plotCirclePoints(xc, yc, x, y);

while (x <= y) {

x++;

if (p < 0) {

p = p + 4 \* x + 6; // Mid-point inside or on the perimeter

} else {

y--;

p = p + 4 \* (x - y) + 10; // Mid-point outside the perimeter

}

plotCirclePoints(xc, yc, x, y);

}

}

int main() {

int gd = DETECT, gm;

int xc, yc, r;

// Initialize graphics mode

initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");

// Input center and radius

printf("Enter the center of the circle (xc, yc): ");

scanf("%d %d", &xc, &yc);

printf("Enter the radius of the circle: ");

scanf("%d", &r);

// Draw the circle

bresenhamCircle(xc, yc, r);

// Wait for user input and close the graphics window

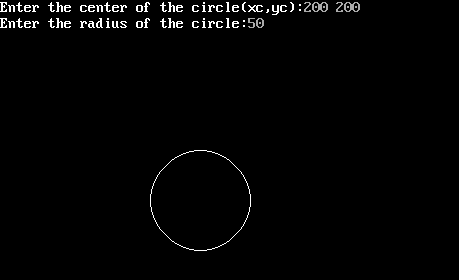
getch();

closegraph();

return 0;

}

**OUTPUT**

****