MIDPOINT CIRCLE DRAWING ALGORITHM

1. #include <graphics.h>
2. #include <stdlib.h>
3. #include <stdio.h>

# void drawCircle(int xc, int yc, int r) {

1. int x = 0, y = r;
2. int p = 1 - r; // Initial decision parameter
3. // Plot points for each octant
4. while (x <= y) {
   1. // Drawing points in all eight octants
   2. putpixel(xc + x, yc + y, WHITE);
   3. putpixel(xc - x, yc + y, WHITE);
   4. putpixel(xc + x, yc - y, WHITE);
   5. putpixel(xc - x, yc - y, WHITE);
   6. putpixel(xc + y, yc + x, WHITE);
   7. putpixel(xc - y, yc + x, WHITE);
   8. putpixel(xc + y, yc - x, WHITE);
   9. putpixel(xc - y, yc - x, WHITE);
   10. x++; // Increment the x value
5. // Update decision parameter
   1. if (p < 0) {
   2. p = p + (2 \* x) + 1; // Inside the circle
   3. } else {
   4. y--; // Move down to the lower row
   5. p = p + (2 \* x) - (2 \* y) + 1; // Outside the circle
   6. }
6. }
7. }

# int main() {

1. int gd = DETECT, gm;
2. // Initialize graphics mode
3. initgraph(&gd, &gm, "C:\\Turboc3\\BGI");
4. int xc, yc, r;
5. printf("Enter the center coordinates (xc, yc): ");
6. scanf("%d %d", &xc, &yc);
7. printf("Enter the radius: ");
8. scanf("%d", &r);
9. drawCircle(xc, yc, r); // Invoke drawing function
10. delay(50); // Display for 5 seconds
11. getch();
12. closegraph(); // Close graphics mode
13. return 0;
14. }