Implementation of Midpoint Circle Drawing Algorithm

1. Start the program.
2. Define a function plot_circle_points(xc, yc, x, y, xes, yes) to compute and store the eight symmetric points of a circle using the current values of x and y with respect to the center (xc,yc).
3. Calculate the eight symmetric points as:
   (x+xc,y+yc), (−x+xc,y+yc), (x+xc,−y+yc), (−x+xc,−y+yc),
   (y+xc,x+yc)(y+xc,x+yc), (y+xc,−x+yc), (−y+xc,−x+yc).
4. Append each computed point into the lists xes and yes.
5. Define a function midpoint_circle(r, xc, yc) to generate all the points of the circle using the midpoint circle algorithm.
6. Initialize the starting point as x=0 and y=r.
7. Initialize the decision parameter as p=1−r.
8. Create empty lists xes and yes to store the x- and y-coordinates of the circle points.
9. Plot the initial eight symmetric points using the function plot_circle_points.
10. Repeat the following steps while x<y:

    a. Increment x by 1.

    b. If the decision parameter p<0,
    update p=p+2x+1.

    c. Else, decrement y by 1 and update
    p=p+2(x−y) +1.
    d. Plot the eight symmetric points for the updated values of x and y.

11. Return the lists xes and yes containing all circle points.
12. Define a function plot_midpoint_circle(r, xc, yc) to display the circle.
13. Call the midpoint circle function to obtain the circle points.
14. Plot the points using a scatter plot, enable grid, label axes, set equal scaling, and display the circle.
15. Stop the program.