**Algorithm for Rotation:**

1. **Input:**
   * Take the radius r and center coordinates (xc, yc) of the circle as input.
2. **Circle Drawing (Using Midpoint Circle Algorithm):**
   * **Initialize Variables**: Start with (x = 0, y = r) and calculate the initial decision parameter p = 1 - r.
   * **Plot Circle Points**: Plot the 8 symmetric points of the circle using the current values of x and y. Add the points (x + xc, y + yc), (-x + xc, y + yc), and other symmetrically placed points in all octants around (xc, yc).
   * **Loop Until Completion**:
     + In each iteration, either increment x and update the decision parameter based on the value of p.
     + If p is negative, only update x. If p is positive or zero, update both x and y.
     + Continue this process until x becomes greater than or equal to y.
3. **Rotation Transformation:**
   * **Define Rotation Matrix**: Compute the 2D rotation matrix for rotating the circle by θ radians (here, θ = π/4).
   * **Translation Matrices**: Define two translation matrices:
     + **Translation to Origin**: To translate the circle to the origin (0,0).
     + **Translation Back**: To shift the circle back to its original center (xc, yc) after rotation.
   * **Composite Transformation**: Combine the translation to origin, rotation, and translation back into a single composite transformation matrix.
4. **Apply Rotation:**
   * Convert the circle's points into homogeneous coordinates (add a third coordinate of 1).
   * Multiply the composite transformation matrix by the points to get the rotated coordinates.
5. **Plotting:**
   * Use matplotlib to:
     + Plot the original circle's points in **blue**.
     + Plot the rotated circle's points in **red**.
     + Display the plot with titles and labels.

**Detailed Step-by-Step Algorithm:**

1. **Input:**
   * Read the radius r and center coordinates (xc, yc) of the circle.
2. **Midpoint Circle Algorithm:**
   * Initialize x = 0, y = r, and p = 1 - r.
   * Initialize empty lists xes and yes to store the circle's points.
   * For each point (x, y), plot 8 symmetric points around (xc, yc):
     + Points are (x + xc, y + yc), (-x + xc, y + yc), (x + xc, -y + yc), etc.
   * Update x and y as per the Midpoint Circle Algorithm logic:
     + If p < 0, increment x and update p.
     + Otherwise, decrement y and increment x, updating p.
3. **Apply Rotation:**
   * Define a rotation matrix to rotate by θ = π/4 radians.
   * Define two translation matrices:
     + Translate the circle to the origin.
     + Translate the circle back to the center (xc, yc).
   * Combine the translation and rotation matrices to form the composite transformation matrix.
4. **Transformation:**
   * Convert the circle points (xes, yes) into homogeneous coordinates.
   * Apply the composite transformation matrix to get the rotated points.
5. **Plotting:**
   * Create a plot with the original circle in blue and the rotated circle in red.
   * Display the plot with appropriate titles, labels, and a grid.
6. **Output:**
   * Show the plot with both the original and rotated circles.