Experiment 1

Bresenham's Line Algorithm, Midpoint Circle Algorithm, and Midpoint Ellipse Algorithm

import matplotlib.pyplot as plt

---------- Bresenham's Line Algorithm ----------

def bresenham_line(x1, y1, x2, y2): points = [] dx = abs(x2 - x1) dy = abs(y2 - y1) sx = 1 if x1 < x2 else -1 sy = 1 if y1 < y2 else -1 err = dx - dy

```
while True:
    points.append((x1, y1))
    if x1 == x2 and y1 == y2:
        break
    e2 = 2 * err
    if e2 > -dy:
        err -= dy
        x1 += sx
    if e2 < dx:
        err += dx
        y1 += sy
return points
```

---------- Midpoint Circle Algorithm ----------

def midpoint_circle(xc, yc, r): points = [] x = 0 y = r d = 1 - r

```
while x <= y:
    # 8-way symmetry
    points.extend([
        (xc + x, yc + y), (xc - x, yc + y),
        (xc + x, yc - y), (xc - x, yc - y),
        (xc + y, yc + x), (xc - y, yc + x),
        (xc + y, yc - x), (xc - y, yc - x)
    ])
    if d < 0:
        d += 2 * x + 3
    else:
        d += 2 * (x - y) + 5
        y -= 1
    x += 1
return points
```

---------- Midpoint Ellipse Algorithm ----------

def midpoint_ellipse(xc, yc, rx, ry): points = [] x = 0 y = ry rx2 = rx * rx ry2 = ry * ry dx = 2 * ry2 * x dy = 2 * rx2 * y

```
# Region 1
d1 = ry2 - (rx2 * ry) + (0.25 * rx2)
while dx < dy:
    points.extend([
        (xc + x, yc + y), (xc - x, yc + y),
        (xc + x, yc - y), (xc - x, yc - y)
    ])
    if d1 < 0:
        x += 1
        dx += 2 * ry2
```

```
            d1 += dx + ry2
        else:
            x += 1
            y -= 1
            dx += 2 * ry2
            dy -= 2 * rx2
            d1 += dx - dy + ry2

    # Region 2
    d2 = (ry2 * ((x + 0.5) ** 2)) + (rx2 * ((y - 1) ** 2)) -
    (rx2 * ry2)
    while y >= 0:
        points.extend([
            (xc + x, yc + y), (xc - x, yc + y),
            (xc + x, yc - y), (xc - x, yc - y)
        ])
        if d2 > 0:
            y -= 1
            dy -= 2 * rx2
            d2 += rx2 - dy
        else:
            y -= 1
            x += 1
            dx += 2 * ry2
            dy -= 2 * rx2
            d2 += dx - dy + rx2
    return points
```

---------- Visualization ----------

if **name** == "**main**": # Line example line_points = bresenham_line(2, 3, 15, 10) lx, ly = zip(*line_points)

```
# Circle example
circle_points = midpoint_circle(30, 30, 10)
cx, cy = zip(*circle_points)

# Ellipse example
ellipse_points = midpoint_ellipse(60, 60, 20, 10)
ex, ey = zip(*ellipse_points)

# Plot all
plt.figure(figsize=(8, 8))
plt.scatter(lx, ly, color="red", s=10, label="Line")
plt.scatter(cx, cy, color="blue", s=10, label="Circle")
plt.scatter(ex, ey, color="green", s=10,
label="Ellipse")
plt.legend()
plt.gca().set_aspect("equal", adjustable="box")
plt.title("Bresenham Line, Midpoint Circle & Ellipse")
plt.show()
```