

**Code:****1. Midpoint Ellipse DrawingAlgorithm:**

```
void setup()
{
  size(800,800);
  background(255);
}
Mld mld=new Mld();
void draw()
{
  mld.medg(175.0,150.0,200.0,200.0);
}
public class Mld
{
  Mld(){};
  void medg(float rx, float ry, float xc, float yc)
  {
    float dx, dy, d1, d2, x, y;
    x = 0;
    y = ry;
    d1 = (ry * ry) - (rx * rx * ry) +
          (0.25f * rx * rx);
    dx = 2 * ry * ry * x;
    dy = 2 * rx * rx * y;

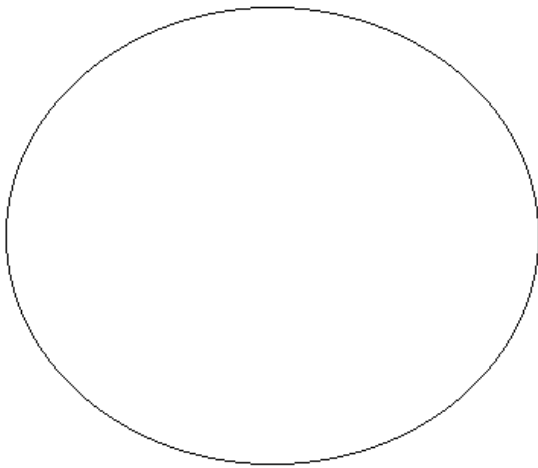
    while (dx < dy)
    {
      point((x + xc),(y + yc));
      point((-x + xc),(y + yc));
      point((x + xc),(-y + yc));
      point((-x + xc),(-y + yc));
    }
    if (d1 < 0)
    {
      x++;
      dx = dx + (2 * ry * ry);
      d1 = d1 + dx + (ry * ry);
    }
    else
    {
      x++;
      y--;
      dx = dx + (2 * ry * ry);
```

```

        dy = dy - (2 * rx * rx);
        d1 = d1 + dx - dy + (ry * ry);
    }
}
d2 = ((ry * ry) * ((x + 0.5f) * (x + 0.5f)))
      + ((rx * rx) * ((y - 1) * (y - 1)))
      - (rx * rx * ry * ry);
while (y >= 0) {
    point((x + xc),(y + yc));
    point((-x + xc),(y + yc));
    point((x + xc),(-y + yc));
    point((-x + xc),(-y + yc));
    if (d2 > 0) {
        y--;
        dy = dy - (2 * rx * rx);
        d2 = d2 + (rx * rx) - dy;
    }
    else {
        y--;
        x++;
        dx = dx + (2 * ry * ry);
        dy = dy - (2 * rx * rx);
        d2 = d2 + dx - dy + (rx * rx);
    }
}
}

```

**OUTPUT:**



## 2. Bresenham's Ellipse Drawing Algorithm:

```
void setup(){  
  size(500,500);  
}
```

```
int x1=100,y1=150,a=50,b=100,xc=250,yc=700;
```

```
void draw(){
```

```
  float x=0,y=b;  
  float p;  
  point(x1+xc/2, yc/2-y1);  
  p=2*b*b+a*a*(1-2*b);  
  while (y>0)  
  {  
    if(b*b*x<a*a*y)  
    {  
      if(p<=a*a/2)
```

```

    p+=2*b*b*(3+2*x);
    else
    { p+=2*b*b*(3+2*x)-4*a*a*(y-1);y--;}
    x++;
}
else
{
    if (p<=b*b/2)
    p+=2*a*a+4*a*a*(y-1);
    else
    { p+=2*a*a+4*a*a*(y-1)-4*b*b*(x+1);x++;}
    y--;
}
point(x+xc/2+x1,y+yc/2-y1);
point(-x+xc/2+x1 ,y+yc/2-y1);
point(x+xc/2+x1,-y+yc/2-y1);
point(-x+xc/2+x1,-y+yc/2-y1);

}

}

```

**OUTPUT:**

