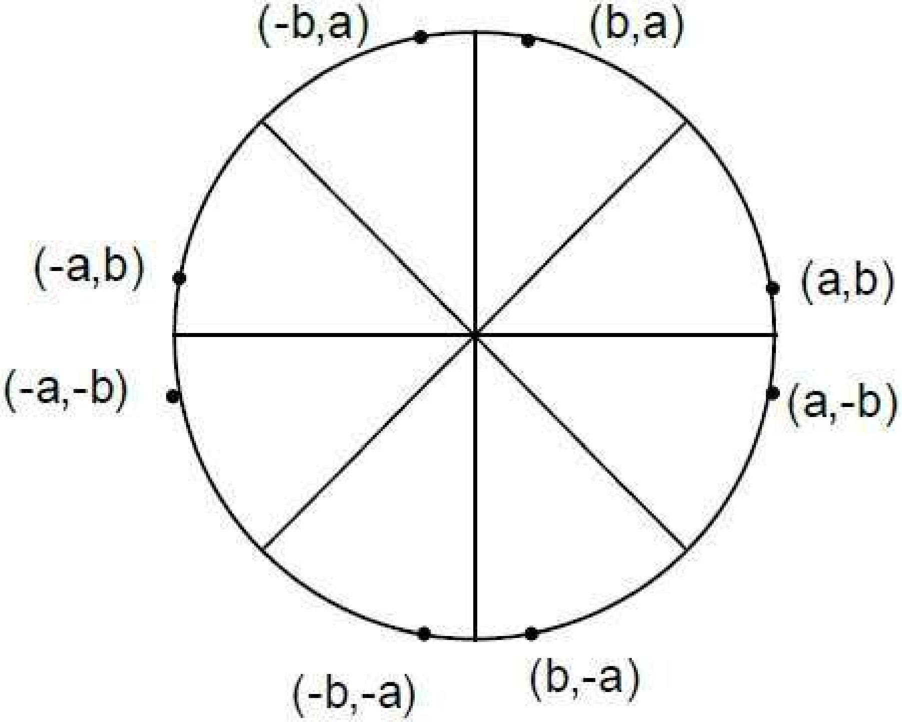
Circle Generation Algorithm

Drawing a circle on the screen is a little complex than drawing a line. There are two popular algorithms for generating a circle – **Bresenham's Algorithm** and **Midpoint Circle Algorithm**. These algorithms are based on the idea of determining the subsequent points required to draw the circle. Let us discuss the algorithms in detail –

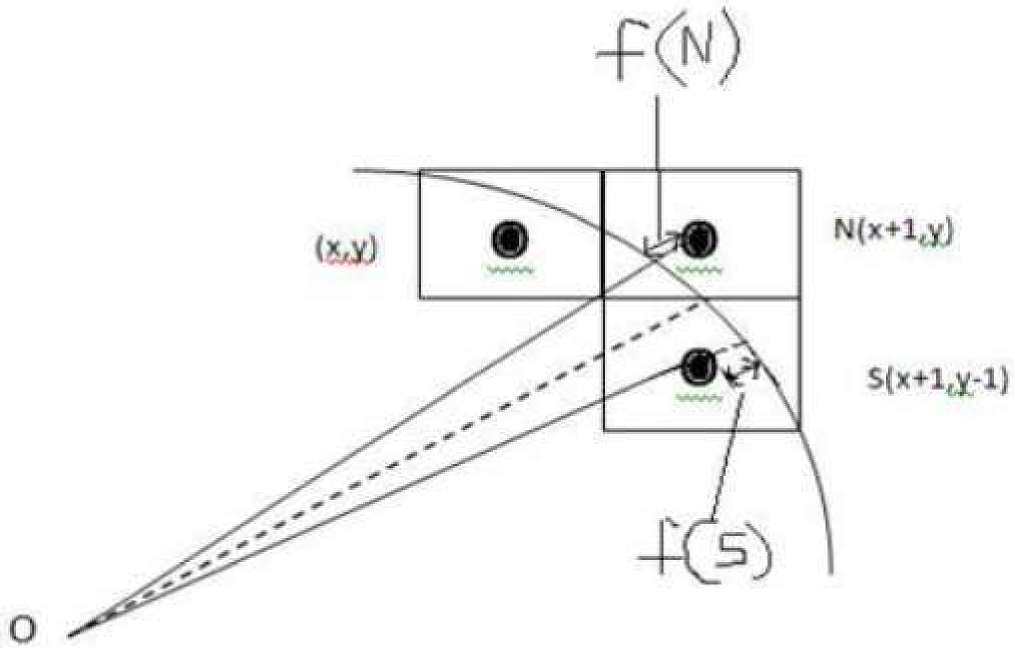
The equation of circle is $X^2 + Y^2 = r^2$, where r is radius.



Bresenham's Algorithm

We cannot display a continuous arc on the raster display. Instead, we have to choose the nearest pixel position to complete the arc.

From the following illustration, you can see that we have put the pixel at X, Y location and now need to decide where to put the next pixel – at N $X + 1, Y$ or at S $X + 1, Y - 1$.



This can be decided by the decision parameter **d**.

- If d <= 0, then N$X + 1, Y$ is to be chosen as next pixel.
- If d > 0, then S$X + 1, Y - 1$ is to be chosen as the next pixel.

Algorithm

Step 1 − Get the coordinates of the center of the circle and radius, and store them in x, y, and R respectively. Set P=0 and Q=R.

Step 2 − Set decision parameter D = 3 – 2R.

Step 3 − Repeat through step-8 while P ≤ Q.

Step 4 − Call Draw Circle X, Y, P, Q.

Step 5 − Increment the value of P.

Step 6 − If D < 0 then D = D + 4P + 6.

Step 7 − Else Set R = R - 1, D = D + 4$P - Q$ + 10.

Step 8 − Call Draw Circle X, Y, P, Q.

```
Draw Circle Method(X, Y, P, Q).
```

```
Call Putpixel (X + P, Y + Q).
Call Putpixel (X - P, Y + Q).
Call Putpixel (X + P, Y - Q).
Call Putpixel (X - P, Y - Q).
Call Putpixel (X + Q, Y + P).
Call Putpixel (X - Q, Y + P).
Call Putpixel (X + Q, Y - P).
Call Putpixel (X - Q, Y - P).
```

Mid Point Algorithm

Step 1 − Input radius **r** and circle center $(x_{c,} y_c)$ and obtain the first point on the circumference of the circle centered on the origin as

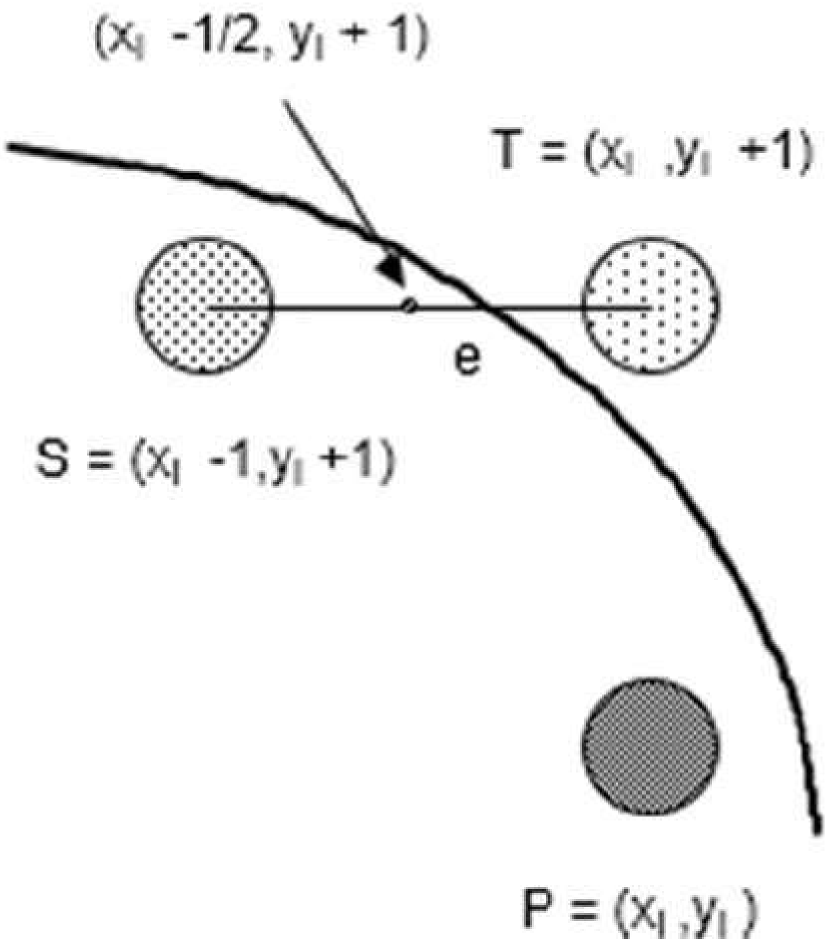
```
$(x_0, y_0) = (0, r)$
```

Step 2 − Calculate the initial value of decision parameter as

P_0 = 5/4 – r $Seethefollowingdescriptionforsimplificationofthisequation.$

```
f(x, y) = $x^2 + y^2 - r^2$ = 0

f($x_i$ - 1/2 + e, $y_i$ + 1)
        = $(x_i - 1/2 + e)^2 + (y_i + 1)^2 - r^2$
        = $(x_i- 1/2)^2 + (y_i + 1)^2 - r^2 + 2(x_i - 1/2)e + e^2$
        = f($x_i$ - 1/2, $y_i$ + 1) + 2($x_i$ - 1/2)e + $e^2$ = 0
```



```
Let $d_i = f(x_i - 1/2, y_i + 1) = -2(x_i - 1/2)e - e^2$
Thus,

If e < 0 then di > 0 so choose point $S = (x_i - 1, y_i + 1)$.
$d_{i+1}$    $= f(x_i - 1 - 1/2, y_i + 1 + 1) = ((x_i - 1/2) - 1)^2 + ((y_i + 1) + 1)^2 - r$
         $= d_i - 2(x_i - 1) + 2(y_i + 1) + 1$
         $= d_i + 2(y_{i + 1} - x_{i + 1}) + 1$

If e >= 0 then di <= 0 so choose point $T = (x_i, y_i + 1)$
   $d_{i+1} = f(x_i - 1/2, y_i + 1 + 1)$
       $= d_i + 2y_{i+1} + 1$

The initial value of di is
   $d_0 = f(r - 1/2, 0 + 1) = (r - 1/2)^2 + 1^2 - r^2$
      $= 5/4 - r$ {1-r can be used if r is an integer}

When point $S = (x_i - 1, y_i + 1)$ is chosen then
```

```
   $d_{i+1} = d_i + -2x_{i+1} + 2y_{i+1} + 1$

When point T = $(x_i, y_i + 1)$ is chosen then
   $d_{i+1} = d_i + 2y_{i+1} + 1$
```

Step 3 − At each X_K position starting at K=0, perform the following test −

```
If $P_K < 0$ then next point on circle (0,0) is $(X_{K+1},Y_K)$ and
   $P_{K+1} = P_K + 2X_{K+1} + 1$
Else
   $P_{K+1} = P_K + 2X_{K+1} + 1 - 2Y_{K+1}$

Where, $2X_{K+1} = 2X_{K+2}$ and $2Y_{K+1} = 2Y_{K-2}$.
```

Step 4 − Determine the symmetry points in other seven octants.

Step 5 − Move each calculate pixel position X, Y onto the circular path centered on (X_C, Y_C) and plot the coordinate values.

$X = X + X_C, \quad Y = Y + Y_C$

Step 6 − Repeat step-3 through 5 until X >= Y.