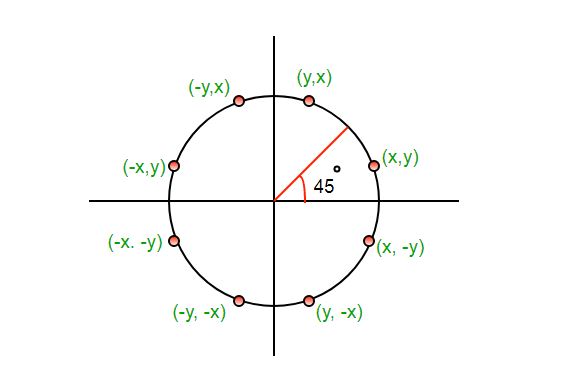
***Mid-point circle drawing algorithm***

* The **mid-point** circle drawing algorithm is an algorithm used to determine the points needed for rasterizing a circle.
* We use the **mid-point** algorithm to calculate all the perimeter points of the circle in the **first octant** and then print them along with their mirror points in the other octants. This will work because a circle is symmetric about its centre.

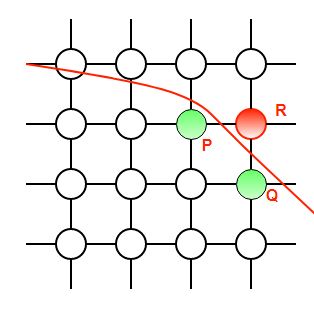


* The algorithm is very similar to the [Mid-Point Line Generation Algorithm](https://www.geeksforgeeks.org/mid-point-line-generation-algorithm/). Here, only the boundary condition is different.

* For any given pixel (x, y), the next pixel to be plotted is either **(x, y+1)** or **(x-1, y+1)**. This can be decided by following the steps below.

1. Find the mid-point **p** of the two possible pixels i.e (x-0.5, y+1)
2. If **p** lies inside or on the circle perimeter, we plot the pixel (x, y+1), otherwise if it’s outside we plot the pixel (x-1, y+1)

* **Boundary Condition :** Whether the mid-point lies inside or outside the circle can be decided by using the formula:-
* Given a circle centered at (0,0) and radius r and a point p(x,y)   
  **F(p) = x2 + y2 – r2**  
  if F(p)<0, the point is inside the circle  
  F(p)=0, the point is on the perimeter  
  F(p)>0, the point is outside the circle



* In our program, we denote F(p) with P. The value of P is calculated at the mid-point of the two contending pixels i.e. (x-0.5, y+1). Each pixel is described with a subscript k i.e. Pk.
* **Pk= x2 + y2 – r2**
* **Po=1-r**

***Algorithm:***

* 1. **Initialize the Parameters**:
* Radius r
* Center of the circle (xc,yc)
* Starting point (x0​,y0​) which is (0,r)

**2)** **Decision Parameter**:

* Initial decision parameter p0=1−r

3)**Loop k=0 to x<=y**

3.1) if pk <0 then

* next point is (xk+1,yK)
* pK+1=Pk+2xk+1+1

3.2) otherwise

* Next point (xk+1,yK-1)
* Pk+1=Pk+2xK+1-2yK+1

3.3)Determine symmetry points in the other seven octants.

3.4)move each calculated pixel position and plot

* X=x+xc
* Y=y+yc

**4)End loop**

**5)stop**