|  |
| --- |
|  |

**Mid Point Circle Drawing Algorithm-**

The points for other octacts are generated using the eight symmetry property.

**Procedure-**

Given-

* Centre point of Circle = (X0, Y0)
* Radius of Circle = R

The points generation using Mid Point Circle Drawing Algorithm involves the following steps-

**Step-01:**

Assign the starting point coordinates (X0, Y0) as-

* X0 = 0
* Y0 = R

**Step-02:**

Calculate the value of initial decision parameter P0 as-

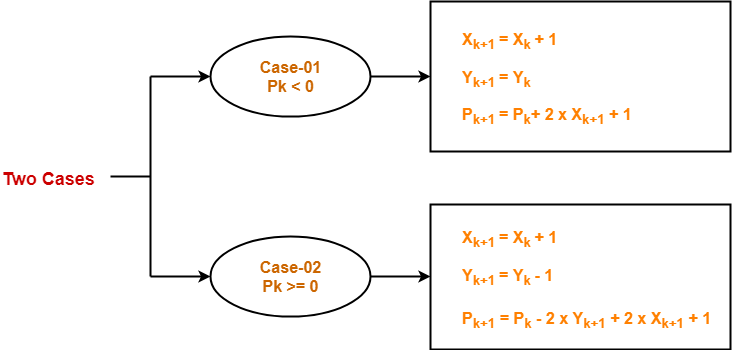
P0 = 1 – R

**Step-03:**

Suppose the current point is (Xk, Yk) and the next point is (Xk+1, Yk+1).

Find the next point of the first octant depending on the value of decision parameter Pk.

Follow the below two cases-



**Step-04:**If the given centre point (X0, Y0) is not (0, 0), then do the following and plot the point-

* Xplot = Xc + X0
* Yplot = Yc + Y0

Here, (Xc, Yc) denotes the current value of X and Y coordinates.

**Step-05:**

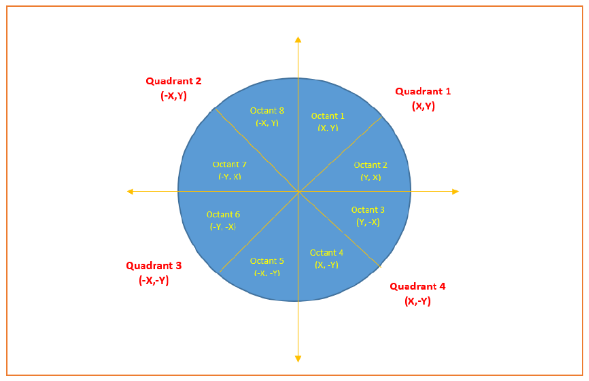
Keep repeating Step-03 and Step-04 until Xplot >= Yplot.

**Step-06:**

**Step-05** generates all the points for one octant.

To find the points for other seven octants, follow the eight symmetry property of circle.

This is depicted by the following figure-



**Source Code**

1. #include<graphics.h>
2. #include<conio.h>
3. #include<stdio.h>
4. void main()
5. {
6. int x,y,x\_mid,y\_mid,radius,dp;
7. int g\_mode,g\_driver=DETECT;
8. clrscr();
9. initgraph(&g\_driver,&g\_mode,"C:\\TURBOC3\\BGI");
10. printf("\* MID POINT Circle drawing algorithm \*\n\n");
11. printf("\nenter the coordinates= ");
12. scanf("%d %d",&x\_mid,&y\_mid);
13. printf("\n now enter the radius =");
14. scanf("%d",&radius);
15. x=0;
16. y=radius;
17. dp=1-radius;
18. do
19. {
20. putpixel(x\_mid+x,y\_mid+y,YELLOW);
21. putpixel(x\_mid+y,y\_mid+x,YELLOW);
22. putpixel(x\_mid-y,y\_mid+x,YELLOW);
23. putpixel(x\_mid-x,y\_mid+y,YELLOW);
24. putpixel(x\_mid-x,y\_mid-y,YELLOW);
25. putpixel(x\_mid-y,y\_mid-x,YELLOW);
26. putpixel(x\_mid+y,y\_mid-x,YELLOW);
27. putpixel(x\_mid+x,y\_mid-y,YELLOW);
28. if(dp<0) {
29. dp+=(2\*x)+1;
30. }
31. else{
32. y=y-1;
33. dp+=(2\*x)-(2\*y)+1;
34. }
35. x=x+1;
36. }while(y>x);
37. getch();
38. }

**Output:-**

