NAME - YOGENDRA SINGH ASWAL

UNIVERSITY ROLL NO - 1121174

SUBJECT - COMPUTER GRAPHICS LAB

SUBJECT CODE - PBC 602

P2 Algorithm :-

Given,

Centre point of Circle = (X_0, Y_0)

Radius of circle = R

Step1: Assign the starting point coordinates (X_0, Y_0) as -

- $X_0 = 0$
- $Y_0 = R$

Step2: Calculate the value of initial decision parameter P_0 as:-

$$P_0 = 1 - R$$

Step3: Suppose the current point is (X_K, Y_K) and the next point is (X_{K+1}, Y_{K+1})

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

two cases :-

① $p_k < 0 \rightarrow$

$$x_{k+1} = x_k + 1$$

$$y_{k+1} = y_k$$

$$p_{k+1} = p_k + 2 \times x_{k+1} + 1$$

② $p_k >= 0 \longrightarrow$

$$x_{k+1} = x_k + 1$$

$$y_{k+1} = y_k - 1$$

$$p_{k+1} = p_k - 2 \times y_{k+1} + 2 \times x_{k+1} + 1$$

Step 4: If the given centre point (x_0, y_0) is not $(0,0)$, then do the following and plot the point

- $x_{plot} = x_c + x_0$
- $y_{plot} = y_c + y_0$

Here, (x_c, y_c) denotes the current value of X and Y coordinates.

Step5: Keep repeating Step-3 and step-4 until $x_{plot} >= y_{plot}$

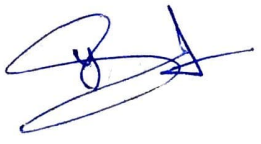
Step6: Step5 generates all the points for one octant

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

Program:

```
#include <stdio.h>
#include <graphics.h>

void drawcircle(int xo, int yo, int radius)
{
    int x=radius;
    int y=0;
    int err=0;

    while(x>=y)
    {
        putpixel(xo+x, yo+y, 7);
        putpixel(xo+y, yo+x, 7);
        putpixel(xo-y, yo+x, 7);
        putpixel(xo-x, yo+y, 7);
        putpixel(xo-x, yo-y, 7);
        putpixel(xo-y, yo-x, 7);
        putpixel(xo+y, yo-x, 7);
        putpixel(xo+x, yo-y, 7);

        if(err<=0)
        {
            y+=1;
            err+=2*y+1;
        }
        if(err>0)
```

```
    {
        x--;
        err -= 2*x+1;
    }
  }
}

int main()
{
    int gdriver = DETECT, gmode, error, x, y, r;
    printf("Enter radius:");
    scanf("%d", &r);
    printf("Enter co-ordinates of center (x and y):");
    scanf("%d %d", &x, &y);
    initgraph(&gdriver, &gmode, "");
    drawcircle(x, y, r);
    delay(9999999);
    return 0;
}
```



NeuTroN DOS-C++ 0.77, Cpu speed: max 100% cycles, Frameskip 0, Program:

```
Enter radius of circle: 100
Enter co-ordinates of center(x and y): 150
150
```