

Q3) Bresenham circle generation algorithm

sol)

Step 1: Start algorithm

Step 2: Declare x, x, y, P and initialize $x_c = 320$ and $y_c = 240$

Step 3: Enter the value of radius r

Step 4: Initialize x to 0 and y to r (radius)

Step 5: Calculate $P = 3 - 2r$

Step 6:

check the next pixel to be scanned

if $P < 0$

then $y = y$ and

$P = P + 4x + 6$

else

increment x &

$P = P + (4 * (x - y) + 10);$

Step 7: Go to step 6 till x become $\leq y$

Step 8: Plot eight points by using concept of eight way symmetry. The centre is at (x_c, y_c) .

Putpixel $(x_c + x, y_c - y, 1);$

Putpixel $(x_c - x, y_c - y, 2);$

Putpixel $(x_c + x, y_c + y, 3);$

Putpixel $(x_c - x, y_c + y, 4);$

Putpixel ($x_c + y, y_c - x, 5$);
Putpixel ($x_c - y, y_c - x, 6$);
Putpixel ($x_c + y, y_c + x, 7$);
Putpixel ($x_c - y, y_c + x, 8$);

step 9 : Go to step 8

step 10 : stop algorithm.

Neeraj Pant

Program :

```
#include <stdio.h>
#include <graphics.h>
void main()
{
    int gd = DETECT, gm;
    int x, xc, y, p, xc = 320, yc = 240;
    printf("Enter the radius");
    scanf("%d", &x);
    initgraph(&gd, &gm, "");
    xc = 0;
    y = x;
    putpixel(xc + x, yc - y, 1);
    p = 3 - (2 * x);
    for (x = 0; x <= y; x++)
    {
        if (p < 0)
        {
            y = y;
            p = (p + (4 * x) + 6);
        }
        else
        {
            y = y + 1;
            p = p + (4 * (x - y) + 10);
        }
    }
}
```

{

putpixel (xc+x, yc-y, 1);

putpixel (xc-c, yc-y, 2);

putpixel (xc+x, yc+y, 3);

putpixel (xc-x, yc+y, 4);

putpixel (xc+y, yc-x, 5);

putpixel (xc-y, yc-x, 6);

putpixel (xc+y, yc+x, 7);

putpixel (xc-y, yc+x, 8);

}

getch();

closegraph();

}

neeraj lant

