NAME - SUBHAM
ROLL NO - 1121147
SUBJECT - COMPUTER GRAPHICS
SUBJECT CODE - PBC-602

Ans 2

Algorithm:

step 1. ~~Put x = 0, y = r~~ ; Assign the starting point coordinates (x_0, y_0) as

- $x_0 = R$
- $y_0 = 0$

step 2. Calculate the value of initial decision parameter P_0 as -

$$P_0 = 1 - R$$

step 3.

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

case-1
$P_k < 0$

$x_{k+1} = x_k + 1$
$y_{k+1} = y_k$
$P_{k+1} = P_k + 2 \times x_{k+1} + 1$

case-2
$P_k >= 0$

$x_{k+1} = x_k + 1$
$y_{k+1} = y_k - 1$
$P_{k+1} = P_k - 2 \times y_{k+1} + 2 \times x_{k+1} + 1$

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

step 5. keep repeating step 3 and step 4 until $x_{plot} >= y_{plot}$

step 6. step 5 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 program.

Prog:

```
#include <iostream.h>
#include <graphics.h>
void drawcircle (int x0, int y0, int radius)
{
    int x = radius;
    int y = 0;
    int err = 0;
    while (x >= y)
putpixel (x0 + x, y0 + y, 7);
putpixel (x0 + y, y0 + x, 7);
putpixel (x0 - y, y0 + x, 7);
putpixel (x0 - x, y0 + y, 7);
putpixel (x0 - x, y0 - y, 7);
```

```
putpixel (x0-y, y0-x, 7);
putpixel (x0+y, y0-x, 7);
putpixel (x0+x, y0-y, 7);

if (esum<=0)
  {
    y+=1;
    esum+=2*y+1;
  }
  if (esum>0)
  {
    x-=1;
  esum=2*x+1;
}
  }
}
int main()
{
int gdriver=DETECT, gmode, esum, x, y, r;
initgraph(&gdriver, &gmode, "c:\\turboc3\\bgi");
```

```
cout << "Enter radius of circle : ";
cin >> r ;
cout << "Enter co-ordinates of centre (x and y) : ";
cin >> x >> y ;
drawcircle (x, y, r);
return 0;
}
```

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