Ans 3:- To Implement Bresenham Circle drawing algorithm :-

- If $d \leq 0$, then $Nx+1, yx+1, y$ is to be chosen as next pixel
- if $d > 0$, then $Sx+1, y-1x+1, y-1$ is to be chosen as the next pixel

Algorithm

Step 1 - Get the coordinates of the center of the circle and radius, and store them in x, y and R respectively. Set P=0 and Q=R.

Step 2 - Set decision parameter $D = 3 - 2R$

Step 3 - Repeat through step-8 while $P \leq Q$.

Step 4 - Call Draw Circle x, y, P, Q, x, y, P, Q.

Step 5 - Increment the value of P.

Step 6 - If $D < 0$ then $D = D + 4P + 6$

Step 7 - Else set $R = R-1$, $D = D + 4P - 4Q - 4 + 10$

Step 8 - Call Draw Circle x, y, P, Q, x, y, P, Q

Program -

```
#include <stdio.h>
#include <graphics.h>
void main()
{
int gd= DETECT, gm;
int r, x, y, p, xc=320, yc=240;
printf("Enter the radius");
scanf("%d", &r);
```

```
initgraph(&gd, &gm, "");
x=0;
y=r;
Putpixel(xc+x, yc-y, 1);
Put = 3-(2*r);
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));
  }
  Put pixel(xc+x, yc-y, 1);
  Put pixel(xc-x, 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();
}
```