

NAME - PRADEEP GUSAIN

ROLL NO - 22 C1121103

SECTION - B

Q=3 → P>3 → BRESENHAM CIRCLE DRAWING ALGORITHM

Code →

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

```
Void main()
```

```
{
    int gd= DETECT, gm;
    int x, x1, y, p, xc = 320, yc = 240;
    printf ("Enter 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);
        }
    }
}
```

Pradeep

```

Putpixel (xc+x, yc-y, 1);
Putpixel (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();
}

```

Bresenham Circle Drawing

Algorithm

- Step (1) → Set initial values of (xc, yc) and (x, y)
- Step (2) → Calculate decision parameter $d = 3 - (2 * r)$
- Step (3) → Call displaybresenhamcircle (int xc, int yc, int x, int y) method to display initial $(0, r)$ point.
- Step (4) → Repeat step 5 to 8 until $x \leq y$.
- Step (5) → increment value of x
- Step (6) → If $d < 0$, set $d = d + (4 * x) + 6$
- Step (7) → Else, set $d = d + 4 * (x - y) + 10$ and decrement y by 1.
- Step (8) → Call displaybresenhamcircle (int xc, int yc, int x, int y)
- Step (9) → Exit.

Pradeep

