

Name :- Streetal Vlegi

Course :- BCA

Sem :- 6

Section :- B

Rollno :- 1121131 (50)

Subject :- Computer Graphics

Bresenham's Circle Drawing Program

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

```
#include <stdlib.h>
```

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

```
#include <conio.h>
```

```
#include <math.h>
```

```
void EightWaySymmetricPlot(int xc, int yc,  
    int n, int y)
```

```
{
```

```
    putpixel(x + nc, y + yc, RED);
```

```
    putpixel(x + nc, -y + yc, YELLOW);
```

```
    putpixel(-x + nc, -y + yc, GREEN);
```

```
    putpixel(-x + nc, y + yc, YELLOW);
```

```
    putpixel(y + nc, x + yc, 12);
```

```
    putpixel(y + nc, -x + yc, 14);
```

```
    putpixel(-y + nc, -x + yc, 15);
```

```
    putpixel(-y + nc, x + yc, 6);
```

```
}
```

```
void BresenhamCircle(int xc, int yc, int r)
```

```
{
```

```
    int x = 0, y = r, d = 3 - (2 * r);
```

```
    EightWaySymmetricPlot(xc, yc, x, y);
```

```
    while (x <= y)
```

```
{
```

```
        if (d <= 0)
```

```
{
```



```
d = d + (4 * n) + 6;
```

```
{
```

```
else
```

```
{
```

```
d = d + (4 * n) - (4 * y) + 10;
```

```
y = y - 1;
```

```
{
```

```
n = n + 1;
```

```
Signature Symmetric Plot(xc, yc, n, y);
```

```
{
```

```
}
```

```
int main(void)
```

```
{
```

```
int xc, yc, r, pdriver = DETECT, gmode,  
errorcode;
```

```
initgraph(&gdriver, &gmode, "C:\\TURBOC3\\BGI");  
errorcode = graphresult();
```

```
if (errorcode != 0)
```

```
{
```

```
printf("Graphics error: %s\n",  
grapherrormsg(errorcode));
```

```
printf("Press any key to halt");  
getch();
```

```
exit(1);
```

```
}
```

```
printf("Enter value of xc and yc");
```

```
scanf("%d %d", &xc, &yc);
```

```
printf("Enter value of radius");  
scanf("%d", &r);
```

```
Bresenham Circle (x, y, r);
```

```
getch();
```

```
closegraph();
```

```
return 0;
```

```
3-
```


Bresenham's Circle Algorithm:-

- Step 1 Start algorithm
- Step 2 Declare p, q, x, y, r, d variables
 p, q are coordinates of center of circle. r is radius.
- Step 3 Enter value of r
- Step 4 Calculate $d = 3 - 2r$
- Step 5 Initialize $x = 0$ & $absy = r$
- Step 6 Check if whole circle is scan converted,
if $x \geq y$
stop
- Step 7 Plot eight points by using concepts of eight-way symmetry. The center is at (p, q) . Current active pixel is (x, y) .
- putpixel($x+p, y+q$)
 - putpixel($y+p, x+q$)
 - putpixel($-y+p, x+q$)
 - putpixel($-x+p, y+q$)
 - putpixel($-x+p, -y+q$)
 - putpixel($-y+p, -x+q$)
 - putpixel($y+p, -x+q$)
 - putpixel($x+p, -y+q$)

Step 8 Find location of next pixel to be scanned

If $d < 0$

then $d = d + 4x + 6$

increment $x = x + 1$

if $d \geq 0$

then $d = d + 4(x - y) + 10$

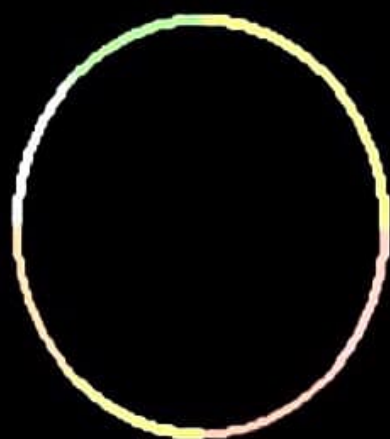
increment $x = x + 1$

decrement $y = y - 1$

Step 9 Go to step 6

Step 10 Stop algorithm

Enter the values of xc and yc :100 100
Enter the value of radius :50



2) \Rightarrow Program to implement 2-D transformation of Δ .

```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
#include <process.h>
#include <math.h>

int x1, y1, x2, y2, x3, y3, mx, my;
```

```
void draw();
```

```
void tri();
```

```
void main()
```

```
{
```

```
int gd = DETECT, gm;
```

```
int C;
```

```
initgraph(&gd, &gm, "--bgi");
```

```
printf("Enter first point");
```

```
scanf("%d %d", &x1, &y1);
```

```
printf("Enter second point");
```

```
scanf("%d %d", &x2, &y2);
```

```
printf("Enter third point");
```

```
scanf("%d %d", &x3, &y3);
```

```
cleardevice();
```

```
draw();
```

```
getch();
```

```
tri();
```

```
getch();
```

```
}
```



```
void draw()
```

```
{
```

```
line(x1, y1, x2, y2);
```

```
line(x2, y2, x3, y3);
```

```
line(x3, y3, x1, y1);
```

```
}
```

```
void tri
```

```
{
```

```
int x, y, a1, a2, a3, b1, b2, b3;
```

```
printf("Enter transaction coordinates");
```

```
scanf("%d %d", &x, &y);
```

```
cleardevice();
```

```
a1 = x1 + x;
```

```
b1 = y1 + y;
```

```
a2 = x2 + x;
```

```
b2 = y2 + y;
```

```
a3 = x3 + x;
```

```
b3 = y3 + y;
```

```
line(a1, b1, a2, b2);
```

```
line(a2, b2, a3, b3);
```

```
line(a3, b3, a1, b1);
```

```
}
```

Enter the 1st point for the triangle:100 150
Enter the 2nd point for the triangle:320 210
Enter the 3rd point for the triangle:432 320

