**Practical No. 5**

***Title:-*** Implement **Bresennham’s Algorithm**  for Circle Drawing.

***Course outcome*** :- Apply the algorithms to draw lines, circle and polygons.

***Resources Required (Hardware & Softwares):-***

1. A Desktop PC/ Laptop
2. Ansi C/ Turbo C/ (Any distribution) installed

***Theory:-***

**Brenennham’s Circle Drawing Algorithm to plot 1/8 of a Circle:-**

1. Read the radius (r) of circle
2. P=3-2r [initialize Decision variable]
3. x=0 y=r [Initialize starting point]

4. do

{

Plot(x,y)

if( p<0) then

{

p=p+4x+6

x=x+1

}

else

{

p=p+4(x-y)+10

x=x+1

y=y-1

}while(x<y)

5. Stop

The remaining part of circle can be drawn by reflecting point about y axis ,x axis can be plotted.

The remaining seven plot commands are-

Plot(y,x)

Plot(y,-x)

Plot(-y,-x)

Plot(-x,-y)

Plot(x,-y)

Plot(-y,x)

Plot(-x,y)

**Program for Bresennham’s Circle Drawing Algorithm**

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<math.h>

void main()

{

float d;

int gd=DETECT,gm,x,y,r;

clrscr();

printf("Enter the radius of the Circle\n");

scanf("%d",&r);

detectgraph(&gd,&gm);

initgraph(&gd,&gm,"c:\\turboc3\\bgi");

x=0;

y=r;

d=3-2\*r;

do

{

putpixel(250+x,250+y,15);

putpixel(250+y,250+x,15);

putpixel(250+y,250-x,15);

putpixel(250+x,250-y,15);

putpixel(250-x,250-y,15);

putpixel(250-y,250-x,15);

putpixel(250-y,250+x,15);

putpixel(250-x,250+y,15);

if(d<=0)

{

d=d+(4\*x)+6;

}

else

{

d=d+(4\*(x-y))+6;

y=y-1;

}

x=x+1;

delay(75);

}

while(x<y);

getch();

closegraph();

}

**Output:- ( Paste your own Output )**

***Conclusion:-***

Thus, we have implemented **Bresennham’s Algorithm** for Circle Drawing.