# Pattern1.cpp

```
#include
#include
#include
using namespace std;

class dda
{
private:
float x1,x2,y1,y2,dx,dy,xi,yi,xn,yn,length;
public:
void initialValues(float a,float b,float c,float d)
{
x1=a,y1=b,x2=c,y2=d;
dx=x2-x1;
dy=y2-y1;
if(abs(dx)>=abs(dy))
length=abs(dx);
else
length=abs(dy);
xi=dx/length;
yi=dy/length;
xn=x1;
yn=y1;
}
void drawLine()
{
for(int i=1; i<=length; i++)
{
putpixel(floor(xn),floor(yn),CYAN);
xn = xn+xi;
yn = yn+yi;
delay(100);
}
}
};

class bressenham
{
private:
int xc,yc,r,xi,yi,pi;
public:
void initialValues(int a, int b, int c)
{
xc=a,yc=b,r=c;
xi=0;
yi=r;
pi=3-(2*r);
}
```

```
void drawCircle()
{
while(xi<=yi)
{
putpixel(xc+xi,yc+yi, WHITE);
putpixel(xc-xi,yc+yi, CYAN);
putpixel(xc+xi,yc-yi, WHITE);
putpixel(xc-xi,yc-yi, CYAN);
putpixel(xc+yi,yc+xi, WHITE);
putpixel(xc-yi,yc+xi, CYAN);
putpixel(xc+yi,yc-xi, WHITE);
putpixel(xc-yi,yc-xi, CYAN);
xi++;
if(pi<0)
pi=pi+(4*xi)+6;
else
{
yi--;
pi=pi+(4*(xi-yi))+10;
}
delay(100);
}
}
};

int main()
{
int gd=DETECT,gm;
int xc,yc,r;
cout<<"Enter center the x coordinate of circle's centre : ";
cin>>xc;
cout<<"Enter center the y coordinate of circle's centre : ";
cin>>yc;
cout<<"Enter the radius of the circle : ";
cin>>r;
bressenham b1,b2;
dda line1,line2,line3;
initgraph(&gd,&gm,NULL);
b1.initialValues(xc,yc,r);
b1.drawCircle();
b2.initialValues(xc,yc,r/2);
b2.drawCircle();
line1.initialValues(xc,yc-r,xc+(r/1.154),yc+(r/2));
line2.initialValues(xc,yc-r,xc-(r/1.154),yc+(r/2));
line3.initialValues(xc+(r/1.154),yc+(r/2),xc-(r/1.154),yc+(r/2));
line1.drawLine();
line2.drawLine();
line3.drawLine();
getch();
closegraph();
```

```
return 0;
}
```