Bresenham's Circle Algorithm:

Step1: Start Algorithm

Step2: Declare p, q, x, y, r, d variables

```
    p, q are coordinates of the center of the circle
    r is the radius of the circle
```

Step3: Enter the value of r

Step4: Calculate d = 3 - 2r

Step5: Initialize   x=0

```
     &nbsy= r
```

Step6: Check if the whole circle is scan converted

```
      If x > = y
      Stop
```

Step7: 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)
```

Step8: Find location of next pixels to be scanned

```
      If d < 0
      then d = d + 4x + 6
      increment x = x + 1
      If d ≥ 0
      then d = d + 4 (x - y) + 10
```

increment $x = x + 1$
decrement $y = y - 1$

Step9: Go to step 6

Step10: Stop Algorithm

Example: Plot 6 points of circle using Bresenham Algorithm. When radius of circle is 10 units. The circle has centre (50, 50).

Solution: Let $r = 10$ (Given)

Step1: Take initial point (0, 10)
$d = 3 - 2r$
$d = 3 - 2 * 10 = -17$
$d < 0 \therefore d = d + 4x + 6$
$= -17 + 4 (0) + 6$
$= -11$

Step2: Plot (1, 10)
$d = d + 4x + 6$ ($\because d < 0$)
$= -11 + 4 (1) + 6$
$= -1$

Step3: Plot (2, 10)
$d = d + 4x + 6$ ($\because d < 0$)
$= -1 + 4 x 2 + 6$
$= 13$

Step4: Plot (3, 9) d is > 0 so $x = x + 1$, $y = y - 1$
$d = d + 4 (x-y) + 10$ ($\because d > 0$)
$= 13 + 4 (3-9) + 10$
$= 13 + 4 (-6) + 10$
$= 23-24=-1$

Step5: Plot (4, 9)
$d = -1 + 4x + 6$
$= -1 + 4 (4) + 6$
$= 21$

Step6: Plot (5, 8)

$$d = d + 4\,(x-y) + 10 \quad (\because d > 0)$$
$$= 21 + 4\,(5-8) + 10$$
$$= 21-12 + 10 = 19$$

So $P_1\ (0,0) \Longrightarrow (50,50)$

$P_2\ (1,10) \Longrightarrow (51,60)$

$P_3\ (2,10) \Longrightarrow (52,60)$

$P_4\ (3,9) \Longrightarrow (53,59)$

$P_5\ (4,9) \Longrightarrow (54,59)$

$P_6\ (5,8) \Longrightarrow (55,58)$

Program to draw a circle using Bresenham's circle drawing algorithm:

```
#include <iostream>

#include <dos.h>

#include <graphics.h>

// Function for print circle

using namespace std;

void drawCircle(int xc, int yc, int x, int y)

{

  putpixel(xc+x, yc+y, 15);

  putpixel(xc-x, yc+y, 15);

  putpixel(xc+x, yc-y, 15);

  putpixel(xc-x, yc-y, 15);

  putpixel(xc+y, yc+x, 15);

  putpixel(xc-y, yc+x, 15);

  putpixel(xc+y, yc-x, 15);

  putpixel(xc-y, yc-x, 15);

}

int main()
```

```
{
  int xc = 100, yc = 100, r = 70,d,x,y;
  int gd = DETECT, gm;
  initgraph(&gd, &gm, ""); // initialize graph
  cout<<"\n\nBresenham Circle Drawing Algorithm Example in C Graphics\n\n";
  x = 0, y = r;
  d = 3 - 2 * r;
  drawCircle(xc, yc, x, y);
  while (y >= x)
  {
    x++;
    if (d > 0)
    {
      y--;
      d = d + 4 * (x - y) + 10;
    }
    else
      d = d + 4 * x + 6;
  drawCircle(xc, yc, x, y);
  delay(70);
  }

  getch();
  return 0;
}
```

Program to draw a circle using midpoint circle drawing algorithm:

```
Program to draw Circle using Midpoint Circle Algorithm
        **Check the initgraph() path in your directory if this programs generates er
ror**
        Author: Darshan Gajara           Author Link: http://darshangajara.wordpress
.com/
        www.pracspedia.com
*/
#include<iostream.h>
#include<conio.h>
#include<graphics.h>
void drawCircle(int x, int y, int xc, int yc);
void main()
{
        int gd = DETECT, gm;
        int r, xc, yc, pk, x, y;
        initgraph(&gd, &gm, "C:\\TC\\BGI");
        cout<<"Enter the center co-ordinates\n";
        cin>>xc>>yc;
        cout<<"Enter the radius of circle\n";
        cin>>r;
        pk = 1 - r;
        x = 0;
        y = r;
        while(x < y)
        {
                drawCircle(x,y,xc,yc);
                ++x;
                if(pk < 0)
                {
                        pk = pk + (2*x) + 1;
                }
                else
                {
                        --y;
                        pk = pk + (2*x) + 1 - (2*y);
                }
        }
        getch();
        closegraph();
}

void drawCircle(int x, int y, int xc, int yc)
{
        putpixel(x+xc,y+yc,GREEN);
        putpixel(-x+xc,y+yc,GREEN);
        putpixel(x+xc, -y+yc,GREEN);
        putpixel(-x+xc, -y+yc, GREEN);
        putpixel(y+xc, x+yc, GREEN);
        putpixel(y+xc, -x+yc, GREEN);
        putpixel(-y+xc, x+yc, GREEN);
        putpixel(-y+xc, -x+yc, GREEN);
}
```