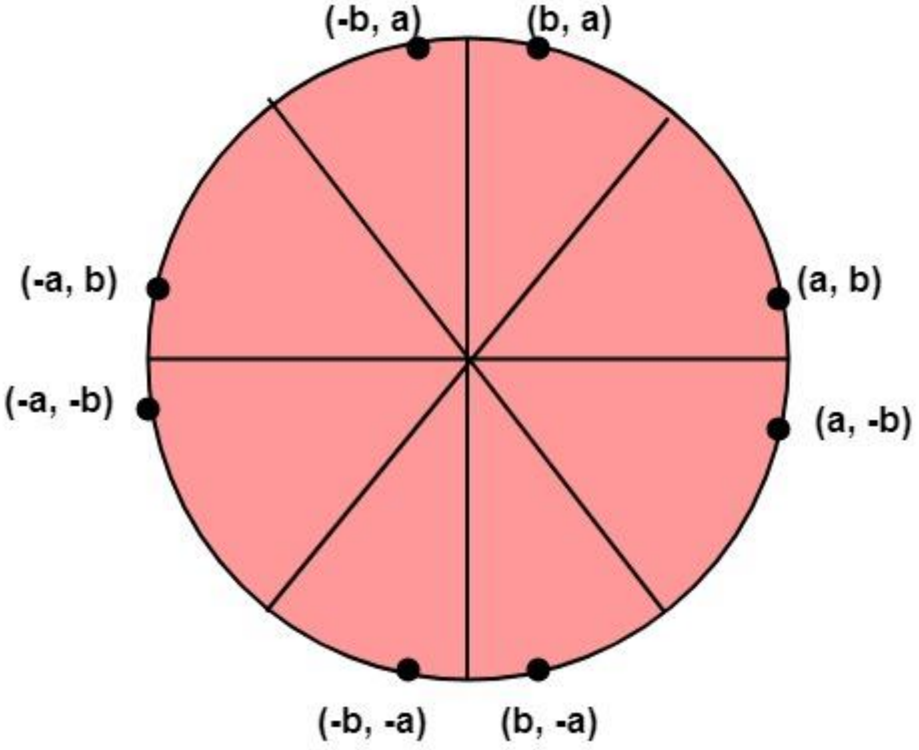
Pratctical 5 (A)

Aim:implement the midpoint circle generation algorithm to draw circle

MidPoint Circle Algorithm

It is based on the following function for testing the spatial relationship between the arbitrary point (x, y) and a circle of radius r centered at the origin:

$$f(x, y) = x^2 + y^2 - r^2 \quad \begin{bmatrix} < 0 \text{ for } (x, y) \text{inside the circle} \\ = 0 \text{ for } (x, y) \text{on the circle} \\ > 0 \text{ for } (x, y) \text{outside the circle} \end{bmatrix} \text{.....equation 1}$$

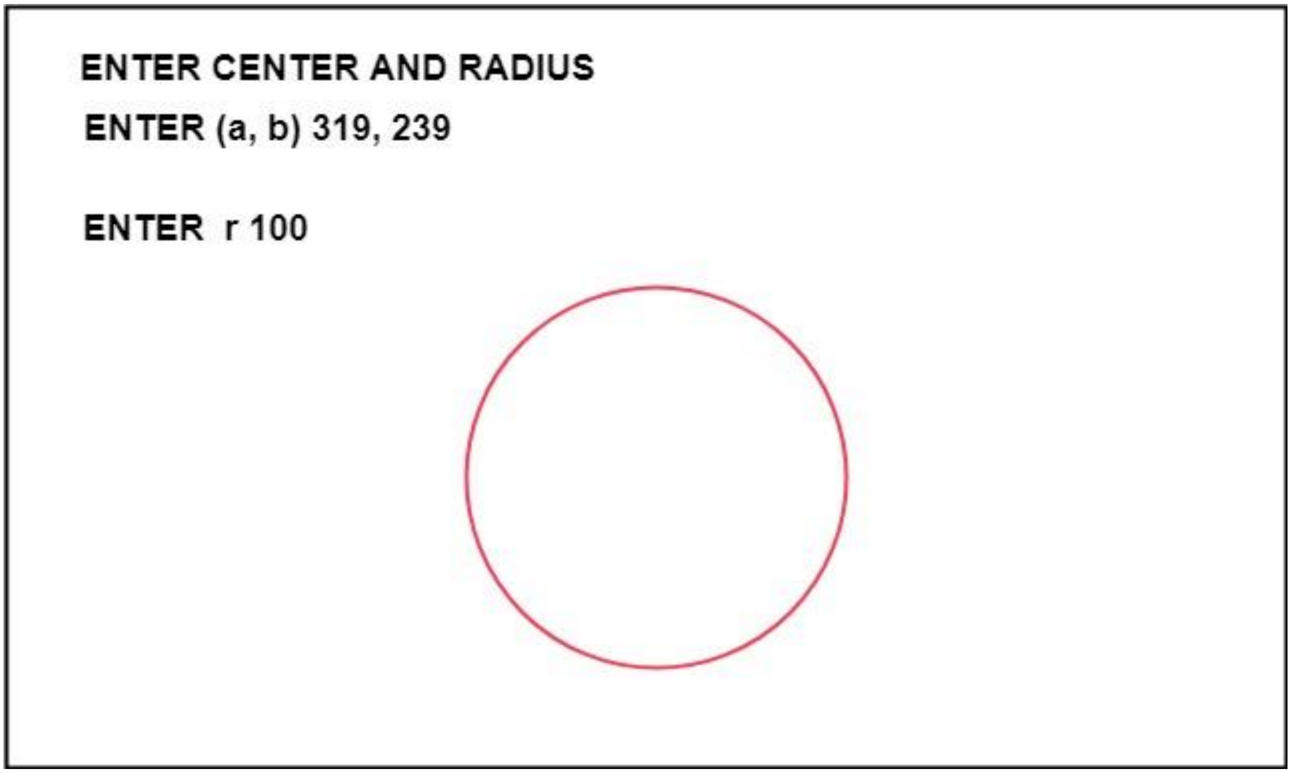


```
#include <graphics.h>
#include <stdlib.h>
#include <math.h>
#include <stdio.h>
#include <conio.h>
```

```
#include <iostream.h>

class bresen
{
        float x, y,a, b, r, p;
        public:
        void get ();
        void cal ();
    };
        void main ()
        {
        bresen b;
        b.get ();
        b.cal ();
        getch ();
       }
        Void bresen :: get ()
       {
        cout<<"ENTER CENTER AND RADIUS";
         cout<< "ENTER (a, b)";
        cin>>a>>b;
        cout<<"ENTER r";
        cin>>r;
    }
    void bresen ::cal ()
    {
        /* request auto detection */
        int gdriver = DETECT,gmode, errorcode;
        int midx, midy, i;
        /* initialize graphics and local variables */
        initgraph (&gdriver, &gmode, " ");
        /* read result of initialization */
        errorcode = graphresult ();
        if (errorcode ! = grOK)    /*an error occurred */
        {
            printf("Graphics error: %s \n", grapherrormsg ( errorcode);
            printf ("Press any key to halt:");
            getch ();
            exit (1); /* terminate with an error code */
        }
        x=0;
        y=r;
        putpixel (a, b+r, RED);
        putpixel (a, b-r, RED);
        putpixel (a-r, b, RED);
        putpixel (a+r, b, RED);
        p=5/4)-r;
        while (x<=y)
        {
            If (p<0)
            p+= (4*x)+6;
            else
            {
                p+=(2*(x-y))+5;
                y--;
            }
            x++;
            putpixel (a+x, b+y, RED);
            putpixel (a-x, b+y, RED);
            putpixel (a+x, b-y, RED);
            putpixel (a+x, b-y, RED);
            putpixel (a+x, b+y, RED);
            putpixel (a+x, b-y, RED);
            putpixel (a-x, b+y, RED);
            putpixel (a-x, b-y, RED);
        }
    }
```

Output:



```
ENTER CENTER AND RADIUS
ENTER (a, b) 319, 239

ENTER  r 100
```