

NAME - SHUBHAM THAPA

University Roll No - 1121142

Subject Code - PBC-602

Subject Name - Computer Graphics

Q2:- Write an algorithm and program to implement mid point circle drawing algorithm.

→ Algorithm :-

Step 1 :- Assign the starting point coordinates (x_0, y_0) as-

-) $x_0 = 0$
-) $y_0 = R = 10$

Step 2 :- Calculate the value of initial decision parameter P_0 as :-

$$P_0 = 1 - R$$

$$P_0 = 1 - 10$$

$$P_0 = -9$$

Step 3 :- As $P_{\text{initial}} < 0$, so case - 01 is satisfied
thus, •) $x_{k+1} = x_k + 1 = 0 + 1 = 1$

$$•) y_{k+1} = y_k = 10$$

$$•) P_{k+1} = P_k + 2 \times x_{k+1} + 1 = -9 + (2 \times 1) + 1 = -6$$

Step 4 :- This step is not applicable here as the given centre point coordinates is $(0, 0)$

W. Sign - Ans

Step 5 :- Step 3 is repeated similarly until $X_{k+1} \geq Y_{k+1}$

P_k	P_{k+1}	(X_{k+1}, Y_{k+1})
-9		(0, 10)
	-6	
-6		(1, 10)
	-1	
-1		(2, 10)
	6	
6		(3, 10)
	-3	
-3		(4, 9)
	8	
8		(5, 9)
	5	
		(6, 8)

Now Octant 2

Octant 1 points

(0, 10)
 (1, 10)
 (2, 10)
 (3, 10)
 (4, 9)
 (5, 9)
 (6, 8)

Octant 2 points

(8, 6)
 (9, 5)
 (9, 4)
 (10, 3)
 (10, 2)
 (10, 1)
 (10, 0)

Sign: 

Now the points for rest of the part are generated by following signs of other quadrants. The other points can also be generated by calculating each Octant separately.

Quadrant-1 (x, y)	Quadrant-2 ($-x, y$)	Quadrant-3 ($-x, -y$)	Quadrant-4 ($x, -y$)
(0, 10)	(0, 10)	(0, -10)	(0, -10)
(1, 10)	(-1, 10)	(-1, -10)	(1, -10)
(2, 10)	(-2, 10)	(-2, -10)	(2, -10)
(3, 10)	(-3, 10)	(-3, -10)	(3, -10)
(4, 9)	(-4, 9)	(-4, -9)	(4, -9)
(5, 9)	(-5, 9)	(-5, -9)	(5, -9)
(6, 8)	(-6, 8)	(-6, -8)	(6, -8)
(8, 6)	(-8, 6)	(-8, -6)	(8, -6)
(9, 5)	(-9, 5)	(-9, -5)	(9, -5)
(9, 4)	(-9, 4)	(-9, -4)	(9, -4)
(10, 3)	(-10, 3)	(-10, -3)	(10, -3)
(10, 2)	(-10, 2)	(-10, -2)	(10, -2)
(10, 1)	(-10, 1)	(-10, -1)	(10, -1)
(10, 0)	(-10, 0)	(-10, 0)	(10, 0)

Sign: Phy

Program :-

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

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

```
void drawcircle(int x0, int y0, int radius)
```

```
{
```

```
    int x = radius;
```

```
    int y = 0;
```

```
    int err = 0;
```

```
    while (x >= y)
```

```
    {
```

```
        putpixel(x0 + x, y0 + y, 7);
```

```
        putpixel(x0 + x, y0 + x, 7);
```

```
        putpixel(x0 - y, y0 + x, 7);
```

```
        putpixel(x0 - x, y0 + y, 7);
```

```
        putpixel(x0 - x, y0 - y, 7);
```

```
        putpixel(x0 - y, y0 - x, 7);
```

```
        putpixel(x0 + y, y0 - x, 7);
```

```
        putpixel(x0 + x, y0 - y, 7);
```

```
        if (err <= 0)
```

```
        {
```

```
            y += 1;
```

```
            err += 2 * y + 1;
```

```
        }
```

```
        if (err > 0)
```

```
        {
```

Sign :- Ph

```

{
    x = 1;
    r = 2 * x + 1;
}
}
}

```

```

int main()

```

```

{
    int gdriver = DETECT, gmode, error, x, y, r;
    printf("Enter radius : ");
    scanf("%d", &r);
    printf("Enter co-ordinates of center (x and y) : ");
    scanf("%d %d", &x, &y);
    initgraph(&gdriver, &gmode, "");
    drawcircle(x, y, r);
    delay(9999999);
    return 0;
}

```


NeuTroN DOS-C++ 0.77, Cpu speed: max 100% cycles, Frameskip 0, Program:

Enter radius of circle: 100

Enter co-ordinates of center(x and y): 150

150

