

Example Math 1

- Find out the pixels required to draw a circle with radius of 15 and centered at origin

radius = 15

x	y	d	E/SE	d update

```
func MidpointCircle(int radius, int value){
    int x, y, d;
    d = 1 - radius;
    x = 0;
    y = radius;
    Circlepoints(x, y, value);
    while (x < y) {
        if (d < 0) {
            //choose E
            d = d + 2*x + 3;
            x = x + 1;
        }
        else {
            //choose SE
            d = d + 2*x - 2*y + 5;
            x = x + 1;
            y = y - 1;
        }
        Circlepoints(x,y, value)
    }
}
```

radius = 15

x	y	d	E/SE	d update
0	15	-14		

```
func MidpointCircle(int radius, int value){
    int x, y, d;
    d = 1 - radius;
    x = 0;
    y = radius;
    Circlepoints(x, y, value);
    while (x < y) {
        if (d < 0) {
            //choose E
            d = d + 2*x + 3;
            x = x + 1;
        }
        else {
            //choose SE
            d = d + 2*x - 2*y + 5;
            x = x + 1;
            y = y - 1;
        }
        Circlepoints(x,y, value)
    }
}
```

radius = 15

x	y	d	E/SE	d update
0	15	-14	E	

```
func MidpointCircle(int radius, int value){
    int x, y, d;
    d = 1 - radius;
    x = 0;
    y = radius;
    Circlepoints(x, y, value);
    while (x < y) {
        if (d < 0) {
            //choose E
            d = d + 2*x + 3;
            x = x + 1;
        }
        else {
            //choose SE
            d = d + 2*x - 2*y + 5;
            x = x + 1;
            y = y - 1;
        }
        Circlepoints(x,y, value)
    }
}
```

radius = 15

x	y	d	E/SE	d update
0	15	-14	E	$= -14 + 2 \times 0 + 3 = -11$

```
func MidpointCircle(int radius, int value){
    int x, y, d;
    d = 1 - radius;
    x = 0;
    y = radius;
    Circlepoints(x, y, value);
    while (x < y) {
        if (d < 0) {
            //choose E
            d = d + 2*x + 3;
            x = x + 1;
        }
        else {
            //choose SE
            d = d + 2*x - 2*y + 5;
            x = x + 1;
            y = y - 1;
        }
        Circlepoints(x,y, value)
    }
}
```

radius = 15

x	y	d	E/SE	d update
0	15	-14	E	$= -14 + 2 \times 0 + 3 = -11$
1	15			

```
func MidpointCircle(int radius, int value){
    int x, y, d;
    d = 1 - radius;
    x = 0;
    y = radius;
    Circlepoints(x, y, value);
    while (x < y) {
        if (d < 0) {
            //choose E
            d = d + 2*x + 3;
            x = x + 1;
        }
        else {
            //choose SE
            d = d + 2*x - 2*y + 5;
            x = x + 1;
            y = y - 1;
        }
        Circlepoints(x,y, value)
    }
}
```

radius = 15

x	y	d	E/SE	d update
0	15	-14	E	$= -14 + 2 \times 0 + 3 = -11$
1	15	-11	E	

```
func MidpointCircle(int radius, int value){
    int x, y, d;
    d = 1 - radius;
    x = 0;
    y = radius;
    Circlepoints(x, y, value);
    while (x < y) {
        if (d < 0) {
            //choose E
            d = d + 2*x + 3;
            x = x + 1;
        }
        else {
            //choose SE
            d = d + 2*x - 2*y + 5;
            x = x + 1;
            y = y - 1;
        }
        Circlepoints(x,y, value)
    }
}
```

radius = 15

x	y	d	E/SE	d update
0	15	-14	E	$= -14 + 2 \times 0 + 3 = -11$
1	15	-11	E	$= -11 + 2 \times 1 + 3 = -6$

```
func MidpointCircle(int radius, int value){
    int x, y, d;
    d = 1 - radius;
    x = 0;
    y = radius;
    Circlepoints(x, y, value);
    while (x < y) {
        if (d < 0) {
            //choose E
            d = d + 2*x + 3;
            x = x + 1;
        }
        else {
            //choose SE
            d = d + 2*x - 2*y + 5;
            x = x + 1;
            y = y - 1;
        }
        Circlepoints(x,y, value)
    }
}
```


radius = 15

x	y	d	E/SE	d update
0	15	-14	E	$= -14 + 2 \times 0 + 3 = -11$
1	15	-11	E	$= -11 + 2 \times 1 + 3 = -6$
2	15			

```
func MidpointCircle(int radius, int value){
    int x, y, d;
    d = 1 - radius;
    x = 0;
    y = radius;
    Circlepoints(x, y, value);
    while (x < y) {
        if (d < 0) {
            //choose E
            d = d + 2*x + 3;
            x = x + 1;
        }
        else {
            //choose SE
            d = d + 2*x - 2*y + 5;
            x = x + 1;
            y = y - 1;
        }
        Circlepoints(x,y, value)
    }
}
```

radius = 15

x	y	d	E/SE	d update
0	15	-14	E	$= -14 + 2 \times 0 + 3 = -11$
1	15	-11	E	$= -11 + 2 \times 1 + 3 = -6$
2	15	-6	E	

```
func MidpointCircle(int radius, int value){
    int x, y, d;
    d = 1 - radius;
    x = 0;
    y = radius;
    Circlepoints(x, y, value);
    while (x < y) {
        if (d < 0) {
            //choose E
            d = d + 2*x + 3;
            x = x + 1;
        }
        else {
            //choose SE
            d = d + 2*x - 2*y + 5;
            x = x + 1;
            y = y - 1;
        }
        Circlepoints(x,y, value)
    }
}
```

radius = 15

x	y	d	E/SE	d update
0	15	-14	E	$= -14 + 2 \times 0 + 3 = -11$
1	15	-11	E	$= -11 + 2 \times 1 + 3 = -6$
2	15	-6	E	$= -6 + 2 \times 2 + 3 = 1$

```
func MidpointCircle(int radius, int value){
    int x, y, d;
    d = 1 - radius;
    x = 0;
    y = radius;
    Circlepoints(x, y, value);
    while (x < y) {
        if (d < 0) {
            //choose E
            d = d + 2*x + 3;
            x = x + 1;
        }
        else {
            //choose SE
            d = d + 2*x - 2*y + 5;
            x = x + 1;
            y = y - 1;
        }
        Circlepoints(x,y, value)
    }
}
```

radius = 15

x	y	d	E/SE	d update
0	15	-14	E	$= -14 + 2 \times 0 + 3 = -11$
1	15	-11	E	$= -11 + 2 \times 1 + 3 = -6$
2	15	-6	E	$= -6 + 2 \times 2 + 3 = 1$
3	15			

```
func MidpointCircle(int radius, int value){
    int x, y, d;
    d = 1 - radius;
    x = 0;
    y = radius;
    Circlepoints(x, y, value);
    while (x < y) {
        if (d < 0) {
            //choose E
            d = d + 2*x + 3;
            x = x + 1;
        }
        else {
            //choose SE
            d = d + 2*x - 2*y + 5;
            x = x + 1;
            y = y - 1;
        }
        Circlepoints(x,y, value)
    }
}
```

radius = 15

x	y	d	E/SE	d update
0	15	-14	E	$= -14 + 2 \times 0 + 3 = -11$
1	15	-11	E	$= -11 + 2 \times 1 + 3 = -6$
2	15	-6	E	$= -6 + 2 \times 2 + 3 = 1$
3	15	1	SE	

```
func MidpointCircle(int radius, int value){
    int x, y, d;
    d = 1 - radius;
    x = 0;
    y = radius;
    Circlepoints(x, y, value);
    while (x < y) {
        if (d < 0) {
            //choose E
            d = d + 2*x + 3;
            x = x + 1;
        }
        else {
            //choose SE
            d = d + 2*x - 2*y + 5;
            x = x + 1;
            y = y - 1;
        }
        Circlepoints(x,y, value)
    }
}
```

radius = 15

x	y	d	E/SE	d update
0	15	-14	E	$= -14 + 2 \times 0 + 3 = -11$
1	15	-11	E	$= -11 + 2 \times 1 + 3 = -6$
2	15	-6	E	$= -6 + 2 \times 2 + 3 = 1$
3	15	1	SE	$= 1 + 2 \times 3 - 2 \times 15 + 5 = -18$

```
func MidpointCircle(int radius, int value){
    int x, y, d;
    d = 1 - radius;
    x = 0;
    y = radius;
    Circlepoints(x, y, value);
    while (x < y) {
        if (d < 0) {
            //choose E
            d = d + 2*x + 3;
            x = x + 1;
        }
        else {
            //choose SE
            d = d + 2*x - 2*y + 5;
            x = x + 1;
            y = y - 1;
        }
        Circlepoints(x,y, value)
    }
}
```

radius = 15

x	y	d	E/SE	d update
0	15	-14	E	$= -14 + 2 \times 0 + 3 = -11$
1	15	-11	E	$= -11 + 2 \times 1 + 3 = -6$
2	15	-6	E	$= -6 + 2 \times 2 + 3 = 1$
3	15	1	SE	$= 1 + 2 \times 3 - 2 \times 15 + 5 = -18$
4	14			

```
func MidpointCircle(int radius, int value){
    int x, y, d;
    d = 1 - radius;
    x = 0;
    y = radius;
    Circlepoints(x, y, value);
    while (x < y) {
        if (d < 0) {
            //choose E
            d = d + 2*x + 3;
            x = x + 1;
        }
        else {
            //choose SE
            d = d + 2*x - 2*y + 5;
            x = x + 1;
            y = y - 1;
        }
        Circlepoints(x, y, value)
    }
}
```

radius = 15

x	y	d	E/SE	d update
0	15	-14	E	$= -14 + 2 \times 0 + 3 = -11$
1	15	-11	E	$= -11 + 2 \times 1 + 3 = -6$
2	15	-6	E	$= -6 + 2 \times 2 + 3 = 1$
3	15	1	SE	$= 1 + 2 \times 3 - 2 \times 15 + 5 = -18$
4	14	-18	E	

```
func MidpointCircle(int radius, int value){
    int x, y, d;
    d = 1 - radius;
    x = 0;
    y = radius;
    Circlepoints(x, y, value);
    while (x < y) {
        if (d < 0) {
            //choose E
            d = d + 2*x + 3;
            x = x + 1;
        }
        else {
            //choose SE
            d = d + 2*x - 2*y + 5;
            x = x + 1;
            y = y - 1;
        }
        Circlepoints(x,y, value)
    }
}
```


radius = 15

x	y	d	E/SE	d update
0	15	-14	E	$= -14 + 2 \times 0 + 3 = -11$
1	15	-11	E	$= -11 + 2 \times 1 + 3 = -6$
2	15	-6	E	$= -6 + 2 \times 2 + 3 = 1$
3	15	1	SE	$= 1 + 2 \times 3 - 2 \times 15 + 5 = -18$
4	14	-18	E	$= -18 + 2 \times 4 + 3 = -7$

```
func MidpointCircle(int radius, int value){
    int x, y, d;
    d = 1 - radius;
    x = 0;
    y = radius;
    Circlepoints(x, y, value);
    while (x < y) {
        if (d < 0) {
            //choose E
            d = d + 2*x + 3;
            x = x + 1;
        }
        else {
            //choose SE
            d = d + 2*x - 2*y + 5;
            x = x + 1;
            y = y - 1;
        }
        Circlepoints(x, y, value)
    }
}
```

radius = 15

x	y	d	E/SE	d update
0	15	-14	E	$= -14 + 2 \times 0 + 3 = -11$
1	15	-11	E	$= -11 + 2 \times 1 + 3 = -6$
2	15	-6	E	$= -6 + 2 \times 2 + 3 = 1$
3	15	1	SE	$= 1 + 2 \times 3 - 2 \times 15 + 5 = -18$
4	14	-18	E	$= -18 + 2 \times 4 + 3 = -7$
5	14			

```
func MidpointCircle(int radius, int value){
    int x, y, d;
    d = 1 - radius;
    x = 0;
    y = radius;
    Circlepoints(x, y, value);
    while (x < y) {
        if (d < 0) {
            //choose E
            d = d + 2*x + 3;
            x = x + 1;
        }
        else {
            //choose SE
            d = d + 2*x - 2*y + 5;
            x = x + 1;
            y = y - 1;
        }
        Circlepoints(x, y, value)
    }
}
```

radius = 15

x	y	d	E/SE	d update
0	15	-14	E	$= -14 + 2 \times 0 + 3 = -11$
1	15	-11	E	$= -11 + 2 \times 1 + 3 = -6$
2	15	-6	E	$= -6 + 2 \times 2 + 3 = 1$
3	15	1	SE	$= 1 + 2 \times 3 - 2 \times 15 + 5 = -18$
4	14	-18	E	$= -18 + 2 \times 4 + 3 = -7$
5	14	-7	E	$= -7 + 2 \times 5 + 3 = 6$
6	14			

```
func MidpointCircle(int radius, int value){
    int x, y, d;
    d = 1 - radius;
    x = 0;
    y = radius;
    Circlepoints(x, y, value);
    while (x < y) {
        if (d < 0) {
            //choose E
            d = d + 2*x + 3;
            x = x + 1;
        }
        else {
            //choose SE
            d = d + 2*x - 2*y + 5;
            x = x + 1;
            y = y - 1;
        }
        Circlepoints(x,y, value)
    }
}
```

radius = 15

x	y	d	E/SE	d update
0	15	-14	E	$= -14 + 2 \times 0 + 3 = -11$
1	15	-11	E	$= -11 + 2 \times 1 + 3 = -6$
2	15	-6	E	$= -6 + 2 \times 2 + 3 = 1$
3	15	1	SE	$= 1 + 2 \times 3 - 2 \times 15 + 5 = -18$
4	14	-18	E	$= -18 + 2 \times 4 + 3 = -7$
5	14	-7	E	$= -7 + 2 \times 5 + 3 = 6$
6	14	6	SE	$= 6 + 2 \times 6 - 2 \times 14 + 5 = -5$
7	13			

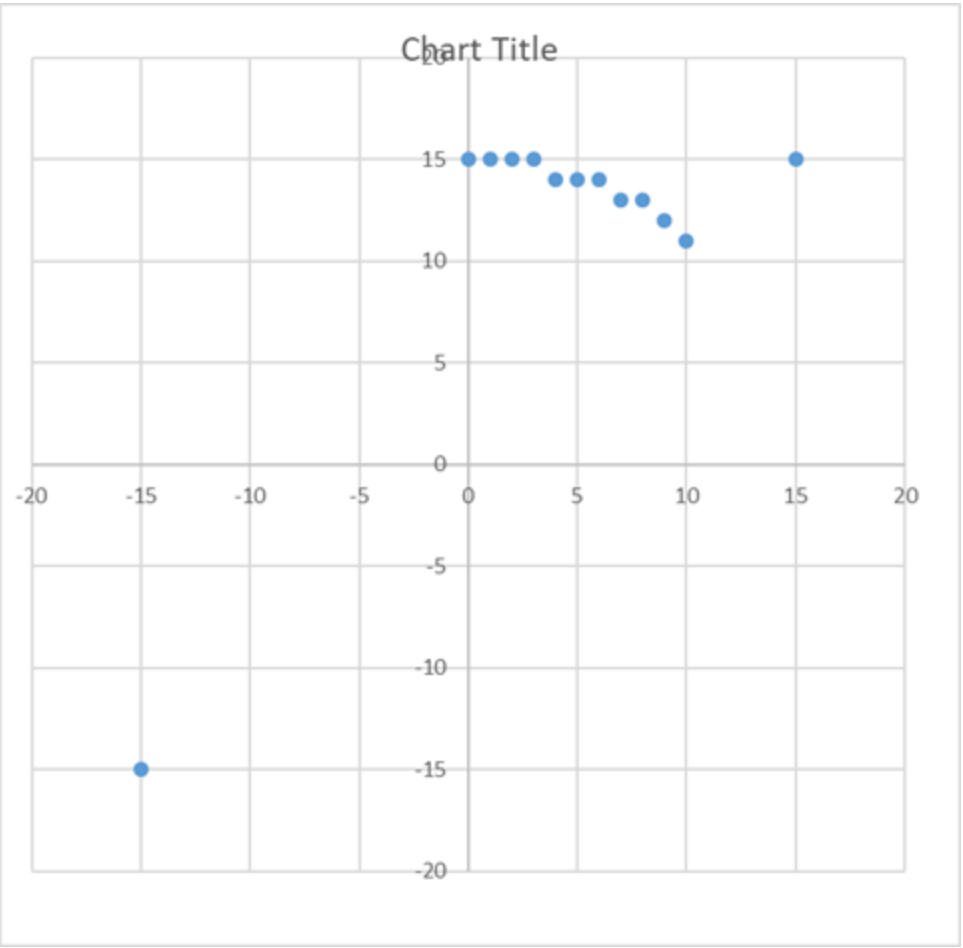
```
func MidpointCircle(int radius, int value){
    int x, y, d;
    d = 1 - radius;
    x = 0;
    y = radius;
    Circlepoints(x, y, value);
    while (x < y) {
        if (d < 0) {
            //choose E
            d = d + 2*x + 3;
            x = x + 1;
        }
        else {
            //choose SE
            d = d + 2*x - 2*y + 5;
            x = x + 1;
            y = y - 1;
        }
        Circlepoints(x,y, value)
    }
}
```

radius = 15

x	y	d	E/SE	d update
0	15	-14	E	$= -14 + 2 \times 0 + 3 = -11$
1	15	-11	E	$= -11 + 2 \times 1 + 3 = -6$
2	15	-6	E	$= -6 + 2 \times 2 + 3 = 1$
3	15	1	SE	$= 1 + 2 \times 3 - 2 \times 15 + 5 = -18$
4	14	-18	E	$= -18 + 2 \times 4 + 3 = -7$
5	14	-7	E	$= -7 + 2 \times 5 + 3 = 6$
6	14	6	SE	$= 6 + 2 \times 6 - 2 \times 14 + 5 = -5$
7	13	-5	E	$= -5 + 2 \times 7 + 3 = 12$
8	13	12	SE	$= 12 + 2 \times 8 - 2 \times 13 + 5 = 7$
9	12	7	SE	$= 7 + 2 \times 9 - 2 \times 12 + 5 = 6$
10	11	6	SE	$= 6 + 2 \times 10 - 2 \times 11 + 5 = 8$
11	10			

```
func MidpointCircle(int radius, int value){
    int x, y, d;
    d = 1 - radius;
    x = 0;
    y = radius;
    Circlepoints(x, y, value);
    while (x < y) {
        if (d < 0) {
            //choose E
            d = d + 2*x + 3;
            x = x + 1;
        }
        else {
            //choose SE
            d = d + 2*x - 2*y + 5;
            x = x + 1;
            y = y - 1;
        }
        Circlepoints(x, y, value)
    }
}
```

x	y
0	15
1	15
2	15
3	15
4	14
5	14
6	14
7	13
8	13
9	12
10	11



Zone 1			Zone 6			Zone 5			Zone 2			Zone 0			Zone 3			Zone 4			Zone 7	
(X, Y)			(X, -Y)			(-X, -Y)			(-X, Y)			(Y, X)			(-Y, X)			(-Y, -X)			(Y, -X)	
0	15		0	-15		0	-15		0	15		15	0		-15	0		-15	0		15	0
1	15		1	-15		-1	-15		-1	15		15	1		-15	1		-15	-1		15	-1
2	15		2	-15		-2	-15		-2	15		15	2		-15	2		-15	-2		15	-2
3	15		3	-15		-3	-15		-3	15		15	3		-15	3		-15	-3		15	-3
4	14		4	-14		-4	-14		-4	14		14	4		-14	4		-14	-4		14	-4
5	14		5	-14		-5	-14		-5	14		14	5		-14	5		-14	-5		14	-5
6	14		6	-14		-6	-14		-6	14		14	6		-14	6		-14	-6		14	-6
7	13		7	-13		-7	-13		-7	13		13	7		-13	7		-13	-7		13	-7
8	13		8	-13		-8	-13		-8	13		13	8		-13	8		-13	-8		13	-8
9	12		9	-12		-9	-12		-9	12		12	9		-12	9		-12	-9		12	-9
10	11		10	-11		-10	-11		-10	11		11	10		-11	10		-11	-10		11	-10

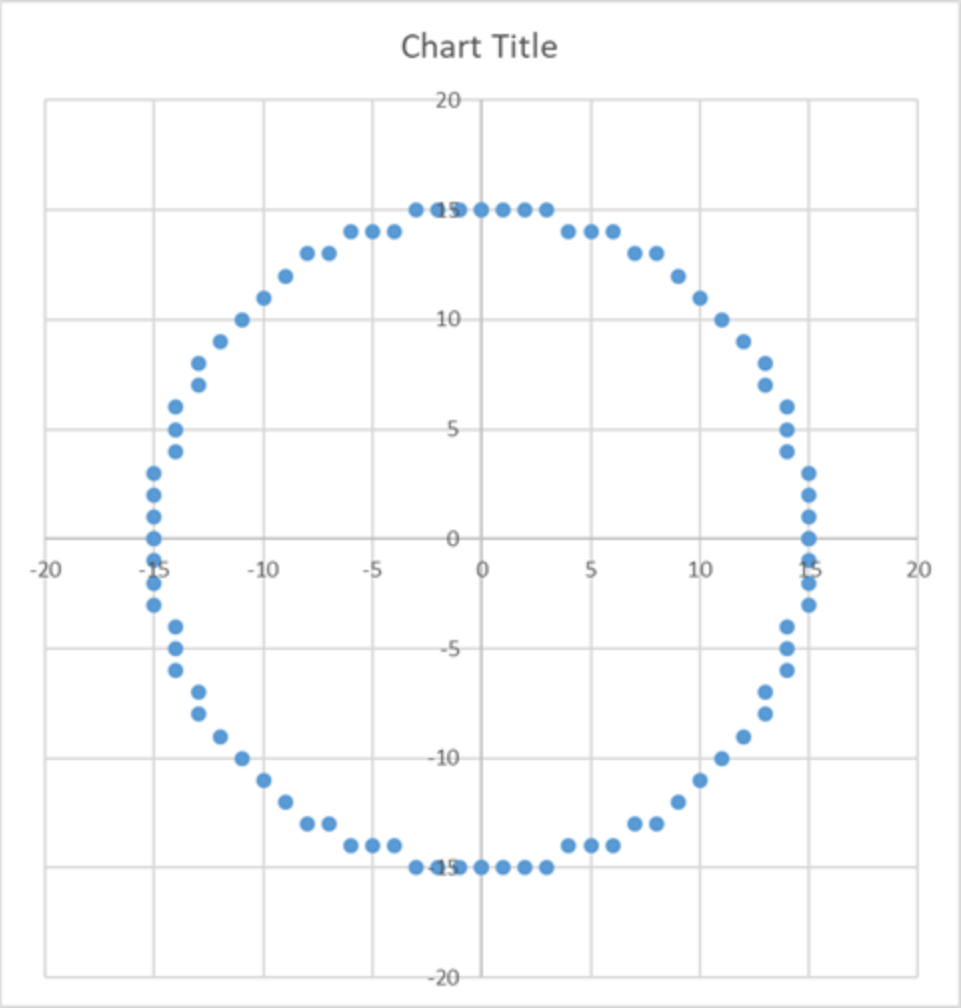
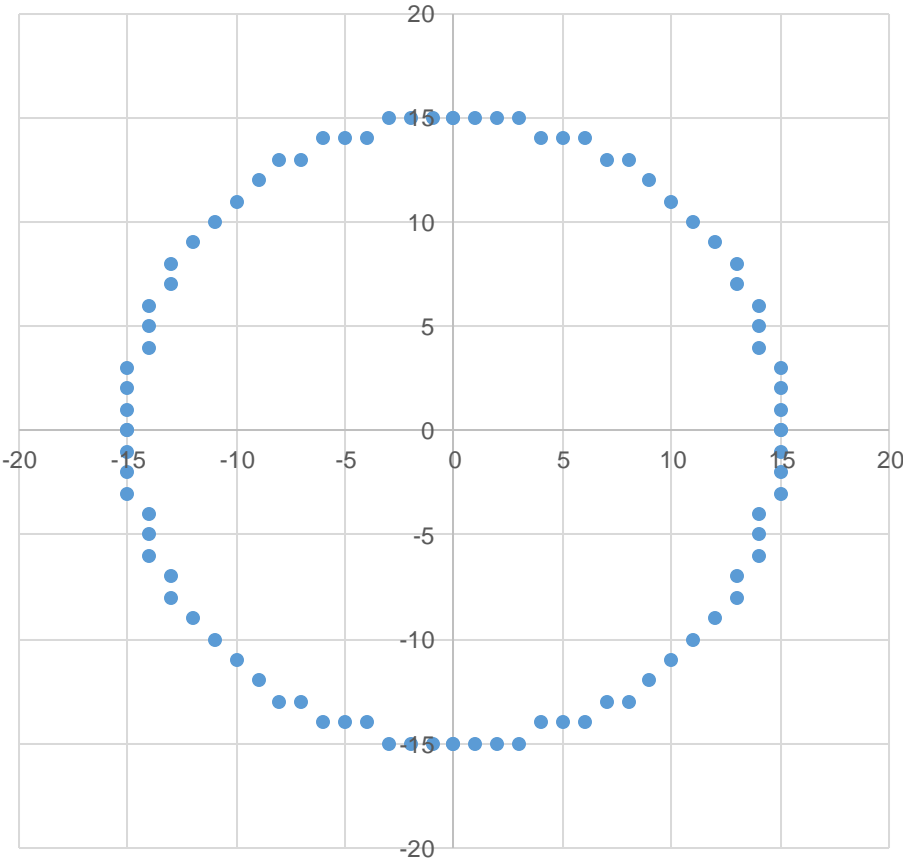
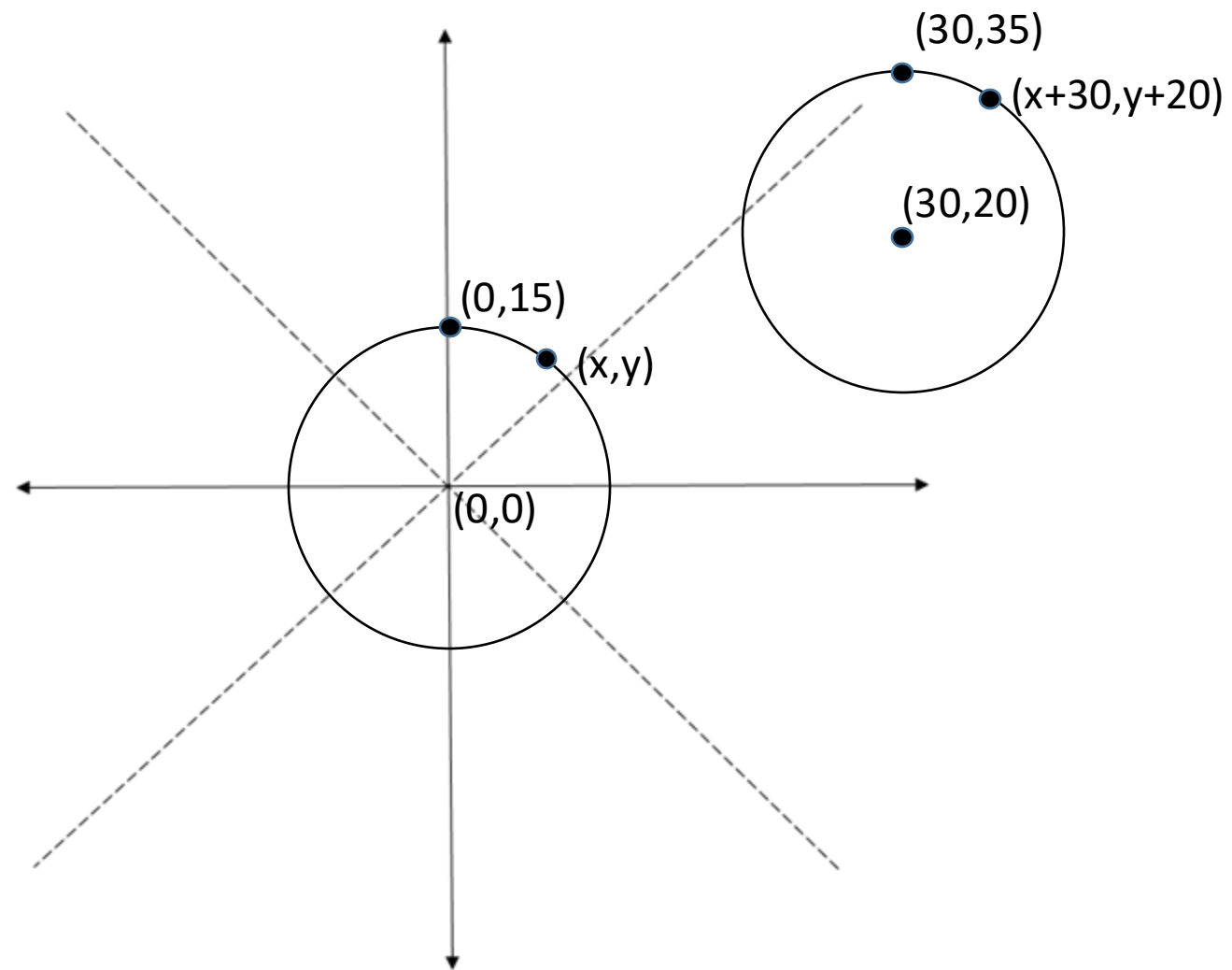


Chart Title

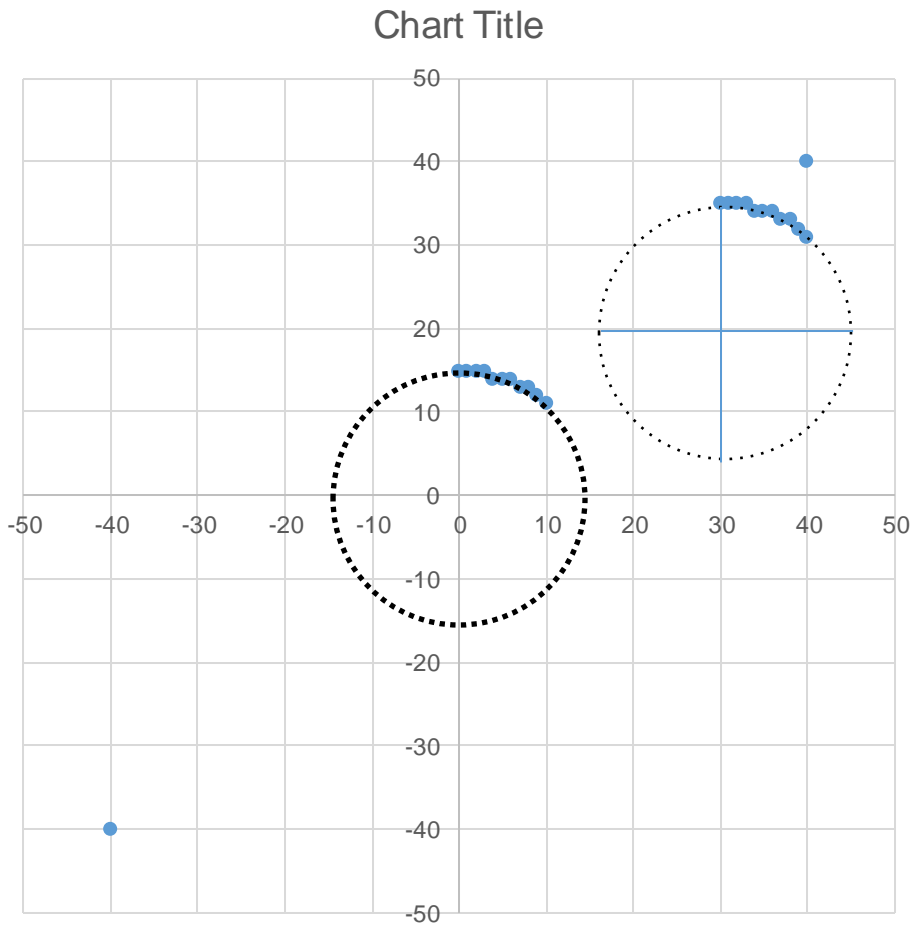


Example Math 2

Find out the pixels required to draw a circle with radius of 15 and centered at (30, 20)



Circle with center at(0, 0)		Circle with center at(30, 20)	
x	y	x+30	y+20
0	15	30	35
1	15	31	35
2	15	32	35
3	15	33	35
4	14	34	34
5	14	35	34
6	14	36	34
7	13	37	33
8	13	38	33
9	12	39	32
10	11	40	31



Steps to find out the pixels of a circle in another Zone rather than the Zone 1 where center is not on the origin:

1. Find out the pixels in Zone - 1 with center (0,0)
2. Convert the pixels of Zone - 1 into the given Zone
3. Add the center to the pixels found in Step-2