

Name - Saitesh Mahan

SET - B

11

Course - BCA 6th B

Roll no - 1121116 (35)

Sub - Computer graphics and animations (PBC - 602)

Date - 16/06/21

```
2. #include <stdio.h>
#include <graphics.h>
#include <conio.h>
```

```
void main()
```

```
{
```

```
int gm, gd = DETECT, ax, x1 = 100;
```

```
int x2 = 100, x3 = 200, y1 = 100;
```

```
int y2 = 200, y3 = 100;
```

```
initgraph (&gd, &gm, "");
```

```
cleardevice();
```

```
line (getmaxx() / 2, 0, getmaxx() / 2,
      getmaxy());
```

```
line (0, getmaxy() / 2, getmaxx(), getmaxy() / 2);
```

```
printf ("Before Reflection Object " "in 2nd Quadrant");
```

```
setcolor (4);
```

```
line (x1, y1, x2, y2);
```

```
line (x2, y2, x3, y3);
```

```
line (x3, y3, x1, y1);
```

```
getch();
```

```
printf ("\nAfter Reflection");
```

```
setcolor (4);
```

```
line (getmaxx() - x1, getmaxy() - y1,
```

```
      getmaxx() - x2, getmaxy() - y2);
```

```
line (getmaxx() - x2, getmaxy() - y2,
```

```
      getmaxx() - x3, getmaxy() - y3);
```

```
line (getmaxx() - x3, getmaxy() - y3,
```

```
      getmaxx() - x1, getmaxy() - y1);
```

SetColor(3);

line (getmaxx() - x1, y1,
getmaxx() - x2, y2);

line (getmaxx() - x2, y2,
getmaxx() - x3, y3);

line (getmaxx() - x3, y3,
getmaxx() - x1, y1);

~~SetColor~~

SetColor(2);

line (x1, getmaxy() - y1, x2,
getmaxy() - y2);

line (x2, getmaxy() - y2, x3,
getmaxy() - y3);

line (x3, getmaxy() - y3, x1,
getmaxy() - y1);

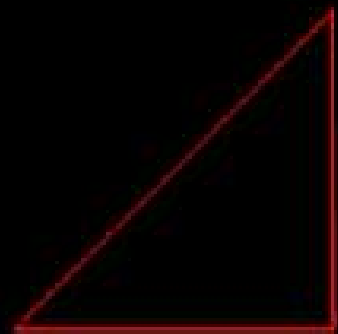
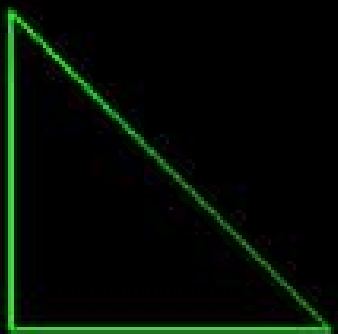
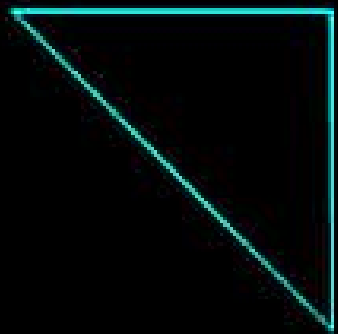
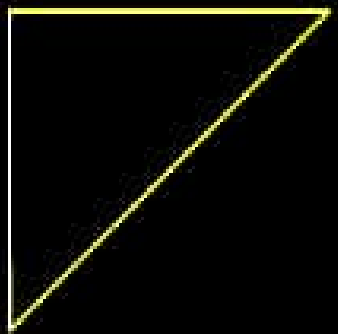
getch();

closegraph();

}



Before Reflection Object in 2nd Quadrant
After Reflection



Name - Sailesh Mahan
Course - BCA 6th 'B'

SET - B

(7)

Roll no - 1121115 (35)

Sub - Computer graphics & animations (PBC-602)

Date - 16/06/21

3.

ALGORITHM

1. Set initial values of (xc, yc) and (x, y)
2. Set decision parameter d to $d = 3 - 2 \cdot r$.
3. Call drawCircle (int xc , int yc , int x , int y) function
4. Repeat steps 5 to 8 until $x \leq y$
5. Increment value of x .
6. If $d < 0$, set $d = d + (4 \cdot x) + 6$
7. Else, set $d = d + 4 \cdot (x - y) + 10$ and decrement y by 1.
8. Call drawCircle (int xc , int yc , int x , int y) function.

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

```
#include <dos.h>
```

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

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

```
{  
    putpixel ( $xc + x$ ,  $yc + y$ , RED);  
    putpixel ( $xc - x$ ,  $yc + y$ , RED);  
    putpixel ( $xc + x$ ,  $yc - y$ , RED);  
    putpixel ( $xc - x$ ,  $yc - y$ , RED);  
    putpixel ( $xc + y$ ,  $yc + x$ , RED);  
    putpixel ( $xc - y$ ,  $yc + x$ , RED);  
    putpixel ( $xc + y$ ,  $yc - x$ , RED);  
    putpixel ( $xc - y$ ,  $yc - x$ , RED);  
}
```



②

```
void circleBres (int xc, int yc, int r)
```

```
{
```

```
    int x = 0, y = r;
```

```
    int 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 (50);
```

```
        }
```

```
    }
```

```
int main ()
```

```
{    int xc = 50, yc = 50, r = 30;
```

```
    int gd = DETECT, gm;
```

```
    initgraph (&gd, &gm, "");
```

```
    circleBres (xc, yc, r);
```

```
    return 0;
```

```
}
```



