

AIM: To implement midpoint ellipse algorithm in C.

ALGORITHM:

Mid-Point Ellipse Algorithm:

1. Take input radius along x axis and y axis and obtain center of ellipse.

2. Initially, we assume ellipse to be centered at origin and the first point as: $((x, y_0) =$

$((0, r_y))$.

3. Obtain the initial decision parameter for region 1 as:

$(p_{10} = r_y^2 + \frac{1}{4}r_x^2 - r_x^2 r_y)$

4. For every (x_k) position in region 1:

If $(p_{1k} < 0)$ then the next point along the is $((x_{k+1}, y_k))$ and
 $(p_{1k+1} = p_{1k} + 2r_y^2 \cdot x_{k+1} + r_y^2)$

Else, the next point is $((x_{k+1}, y_{k-1}))$

And $(p_{1k+1} = p_{1k} + 2r_y^2 \cdot x_{k+1} - 2r_x^2 \cdot y_{k+1} + r_y^2)$

5. Obtain the initial value in region 2 using the last point $((x_0, y_0))$ of region 1 as:

$[p_{20} = \{r_y^2(x_0 + \frac{1}{2})^2 + r_x^2(y_0 - 1)^2 - r_x^2 r_y^2\}]$

6. At each (y_k) in region 2 starting at $(k=0)$ perform the following task.

If $(p_{2k} > 0)$ the next point is $((x_k, y_{k-1}))$ and $(p_{2k+1} = p_{2k} - 2rx^2 y_{k+1} + rx^2)$

7. Else, the next point is $((x_{k+1}, y_{k-1}))$ and $(p_{2k+1} = p_{2k} + 2ry^2 x_{k+1} - 2ry^2 y_{k+1} + rx^2)$

8. Now obtain the symmetric points in the three quadrants and plot the coordinate

value as: $(x = x + X_C, y = y + y_C)$

9. Repeat the steps for region 1 until $(2ry^2 x \geq 2rx^2 y)$

10. Repeat steps for region 2 until $(y=0)$

CODE:

```
#include<stdio.h>

#include<graphics.h>

void main(){

    long x,y,x_center,y_center;

    long a_sqr,b_sqr, fx,fy, d,a,b,tmp1,tmp2;

    int g_driver=DETECT,g_mode;

    clrscr();


    initgraph(&g_driver,&g_mode,"C:\\TURBOC3\\BGI");
    printf("*** MID POINT ELLIPSE ALGORITHM ***");
    printf("\n\n Enter coordinate x and y = ");
    scanf("%ld%ld",&x_center,&y_center);
    printf("\n Now enter constants a and b = ");
    scanf("%ld%ld",&a,&b);
    x=0;
    y=b;
    a_sqr=a*a;
    b_sqr=b*b;
```

```
    fx=2*b_sqr*x;
    fy=2*a_sqr*y;
    d=b_sqr-(a_sqr*b)+(a_sqr*0.25);
    do
    {
        putpixel(x_center+x,y_center+y,CYAN);
        putpixel(x_center-x,y_center-y,CYAN);
        putpixel(x_center+x,y_center-y,CYAN);
        putpixel(x_center-x,y_center+y,CYAN);

        if(d<0)
        {
            d=d+fx+b_sqr;
        }
        else
        {
            y=y-1;
            d=d+fx+-fy+b_sqr;
            fy=fy-(2*a_sqr);
        }
    }
```

```
x=x+1;
fx=fx+(2*b_sqr);
delay(10);

}
while(fx<fy);
tmp1=(x+0.5)*(x+0.5);
tmp2=(y-1)*(y-1);
d=b_sqr*tmp1+a_sqr*tmp2-(a_sqr*b_sqr);
do
{
putpixel(x_center+x,y_center+y,CYAN);
putpixel(x_center-x,y_center-y,CYAN);
putpixel(x_center+x,y_center-y,CYAN);
putpixel(x_center-x,y_center+y,CYAN);

if(d>=0)
d=d-fy+a_sqr;
else
```

```
{  
x=x+1;  
d=d+fx-fy+a_sqr;  
fx=fx+(2*b_sqr);  
}  
y=y-1;  
fy=fy-(2*a_sqr);  
}  
while(y>0);  
getch();  
closegraph();  
}
```

OUTPUT:

```
*** MID POINT ELLIPSE ALGORITHM ***
```

```
Enter coordinate x and y = 100
```

```
150
```

```
Now enter constants a and b = 75
```

```
25
```

