name Pranav Narayan Pisal
roll no 63 se2

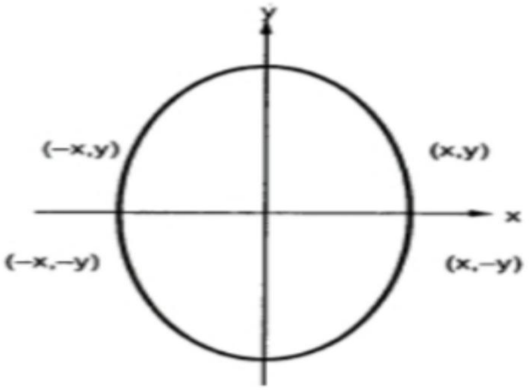
Aim-To implement midpoint Ellipse algorithm

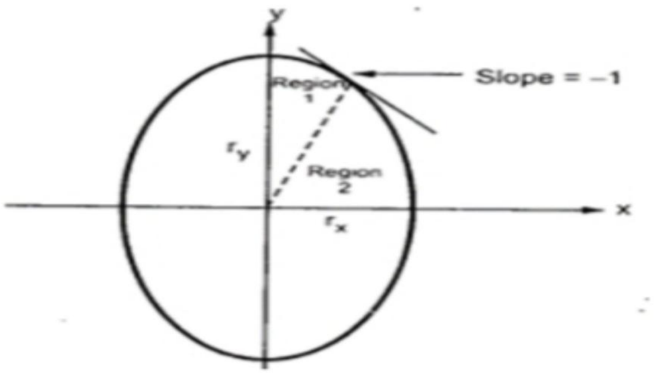
Objective:

Draw the ellipse using Mid-point Ellipse algorithm in computer graphics. Midpoint ellipse
algorithm plots (finds) points of an ellipse on the first quadrant by dividing the quadrant into
two regions.

Theory:

Midpoint ellipse algorithm uses four way symmetry of the ellipse to generate it. Figure shows
the 4-way symmetry of the ellipse.





Here the quadrant of the ellipse is divided into two regions as shown in the fig. Fig. shows the
division of first quadrant according to the slope of an ellipse with rx &lt; ry. As ellipse is

drawn
from 90 0 to 0 0 , x moves in positive direction and y moves in negative direction and ellipse
passes through two regions 1 and 2.
The equation of ellipse with center at (xc, yc) is given as -
[(x – xc) / rx] 2 + [(y – yc) / ry] 2 = 1
Therefore, the equation of ellipse with center at origin is given as -
[x / rx] 2 + [y / ry] 2 = 1
i.e. x 2 ry 2 + y 2 rx 2 = rx 2 ry 2
Let, f ellipse (x, y) = x2 ry2 + y2 rx2 - rx2 ry2
Algorithm:

1. Accept the radii of the ellipse 'r', 'ry' and center of the ellipse (XY)

2. Initialize the starting position as (x, y) = (0, ry)

3. Calculate the initial value of decision parameter for region-1, Po

as

Po = r^2-rry + 1/4 r,2

4. At each x position in region-1 starting at k = 0 perform the

following test

do

{ putpixel (x, y, COLOR)

```
If p<0
{
Xnext = Xk+1 = X + 1
Ynext = Ук+1 = Ук
Pk+1=Pk+2 r$y^2$ (X + 1) +ry2
}
else
{
Xnext = Xk+1 = X +1 Ynext=Yk+1=Yk - 1
Pk+1=Px+2$y^2$ (X + 1) + гу2-2 г$^2$ (ук-1)
} } while (2 r2x<2r$x^2$y)
```

5. Calculate the initial value of decision parameter for region-2 as

Po" = $(X + 1/2)^2$ ry^2 + $(yo-1)^{222}$. ry

6. At each x position in region-2 starting at $k = 0$ perform the

following test do { putpixel (x, y, COLOR)

If pk">0

1

Xnext=Xk+1=Xk

Ynext = Ук+1 = Ук-1 = Ук-1

Pk+1" =Pk" - 2 г Ук+3 г2

else

}

{

Xnext=Xk+1 = X +1

68-69/84

Ynext=Yk+1 = Ук-1 = Ук-1

Pk+1" =Pk" + 2ry2 [X + 1]-2 г^2 Ук+3 г2

}

} while (y > 0)

7. Determine the symmetry points in the other three quadrant.

8. Translate each pixel position by (x, y)

X = Xk+1 + Xe

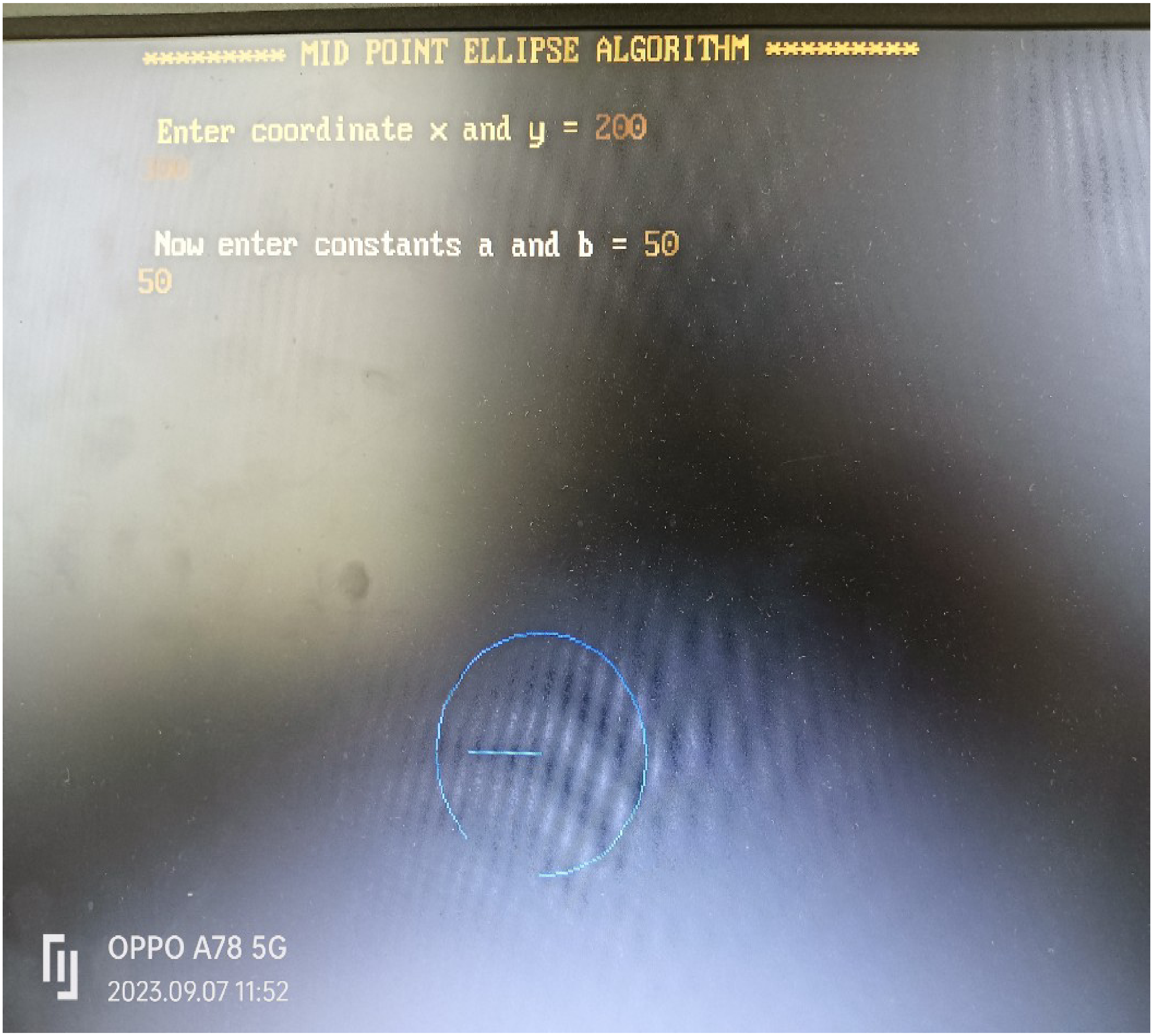
```
y = yk+1 + yc
putpixel (x, y, COLOR)
9. Stop
```

program

```
include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main(){
      long x,y,xc,yc;
      long a_sqr,b_sqr, dx,dy, d,a,b,tmp1,tmp2;
      int gd=DETECT,gm;
      clrscr();
    initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
    printf("********* MID POINT ELLIPSE ALGORITHM *********");
    printf("\n\n Enter coordinate x and y = ");
    scanf("%ld%ld",&xc,&yc);
    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;
    dx=2*b_sqr*x;
    dy=2*a_sqr*y;
  d=b_sqr-(a_sqr*b)+(a_sqr*0.25);
  do
   {
  putpixel(xc+x,yc+y,1);
  putpixel(xc-x,yc-y,1);
  putpixel(xc+x,yc-y,1);
  putpixel(xc-x,yc,1);
if(d<0)
    {
  d=d+dx+b_sqr;

    }
   else
  {
  y=y-1;
  d=d+dx+-dy+b_sqr;
  dy=dy-(2*a_sqr);
  }
  x=x+1;
  dx=dx+(2*b_sqr);


   }
   while(dx<dy);
   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(xc+x,yc+y,1);
  putpixel(xc-x,yc-y,1);
  putpixel(xc+x,yc-y,1);
  putpixel(xc-x,yc+y,1);

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

  {
  x=x+1;
  d=d+dx-dy+a_sqr;
  dx=dx+(2*b_sqr);
  }
   y=y-1;
   dy=dy-(2*a_sqr);
   }
   while(y>0);
```

```
while(y>0);
   getch();
   closegraph();
}
```

Output



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

     Enter coordinate x and y = 200
    300

     Now enter constants a and b = 50
    50
```