Experiment No 01

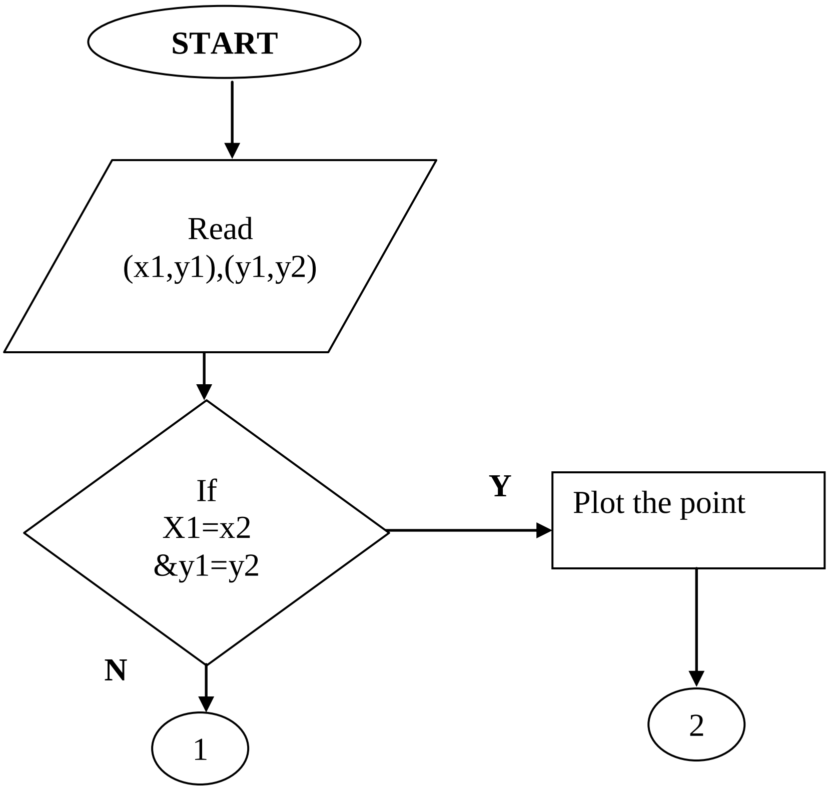
OBJECT

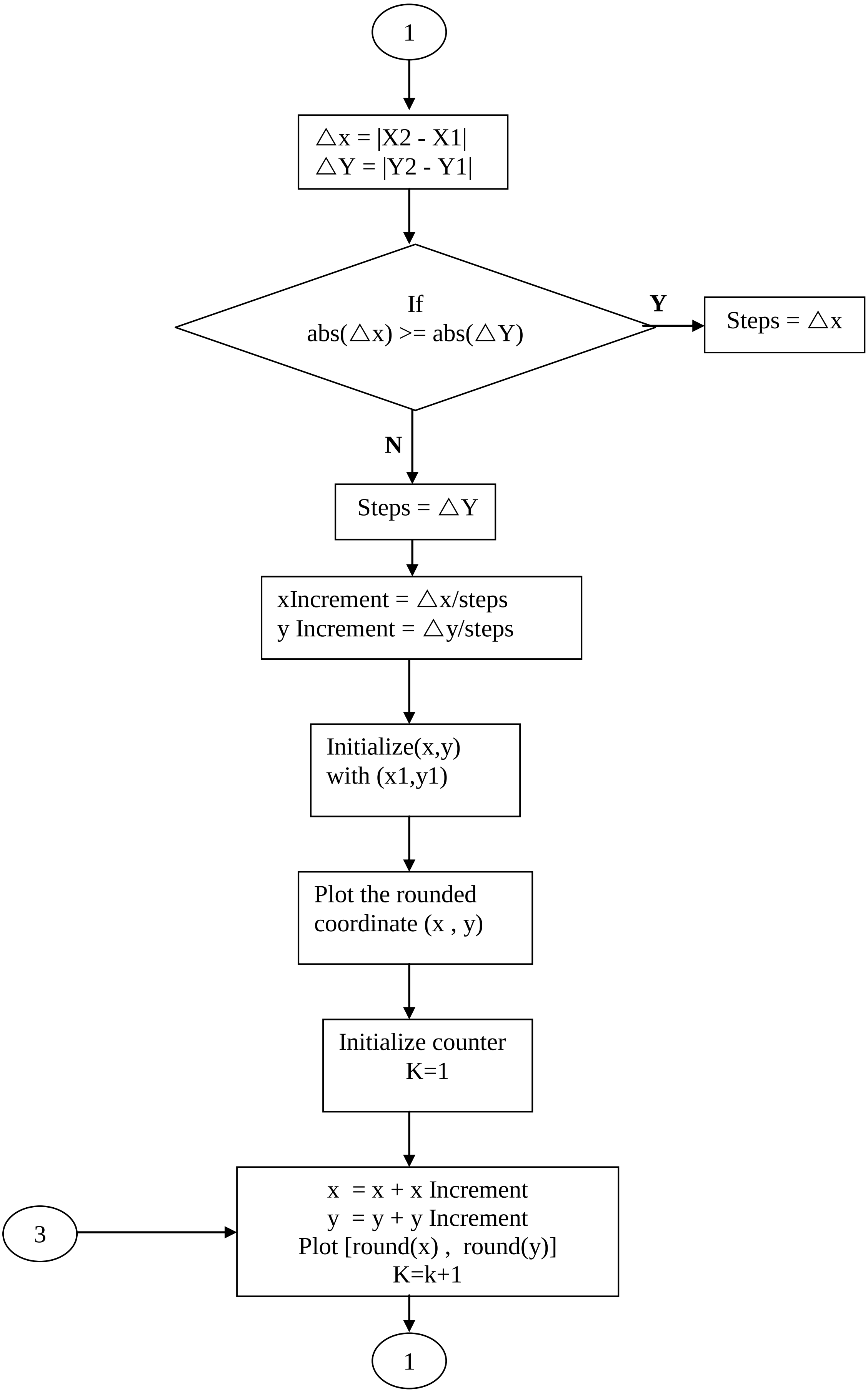
Digital Differential Analyzer Line Drawing Method

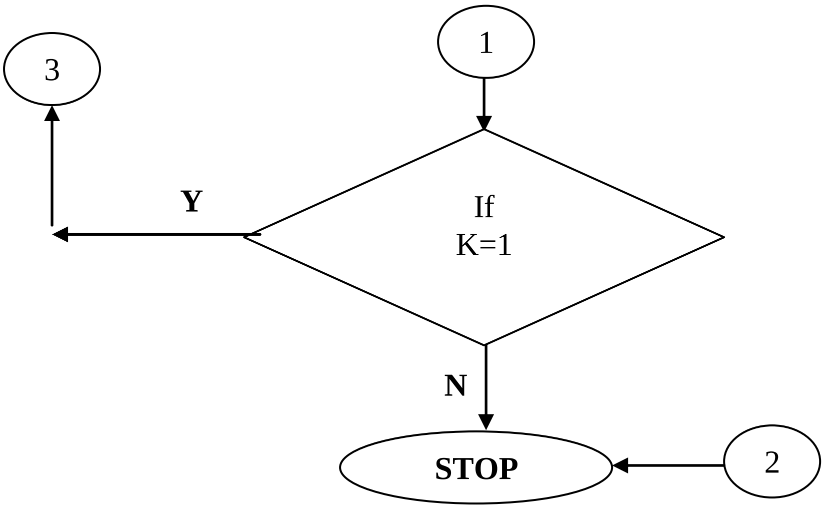
ALGORITHM

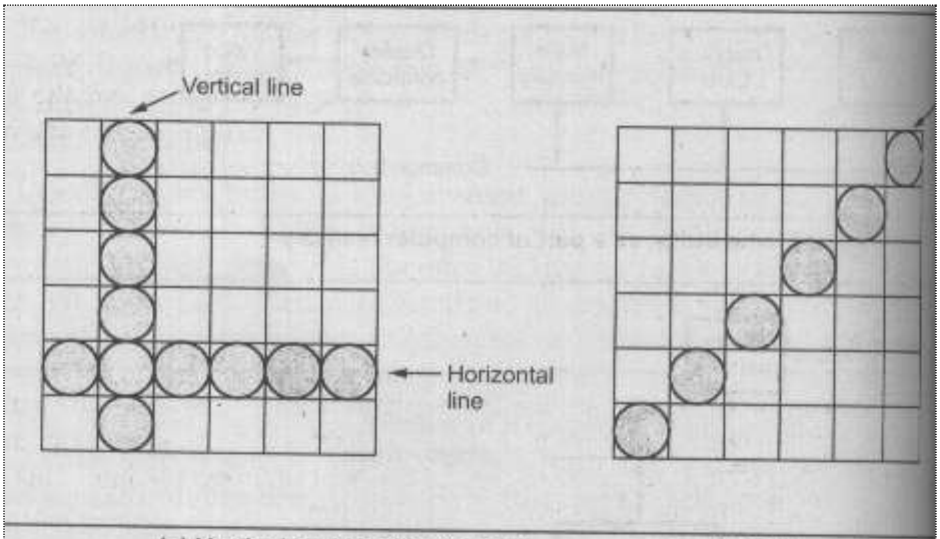
```
1.  Read the end points of a line (x1, y1) & (x2, y2).
2.  If the points are same, plot the points & exit (Enter continue with the calculation).
3.  △x = | x2-x1 | & △y = | y2-y1 |
4.  if abs(△ x) ≥ abs(△y) then
        Steps =△x
    Else
        Steps=△y
5.  x increment = △x / steps.
6.  y increment = △y /steps
7.  initialize (x, y) with (x1,y1)
        x = x1
        y = y1
8.  plot the rounded coordinate (x, y)
9.  initialize counter K=1
10. start the loop
        x = x + x increment
        y = y+ y increment
        Plot the rounded coordinate(x, y)
11. Continue the loop till the counter = steps
12. Stop.
```

FLOW CHART

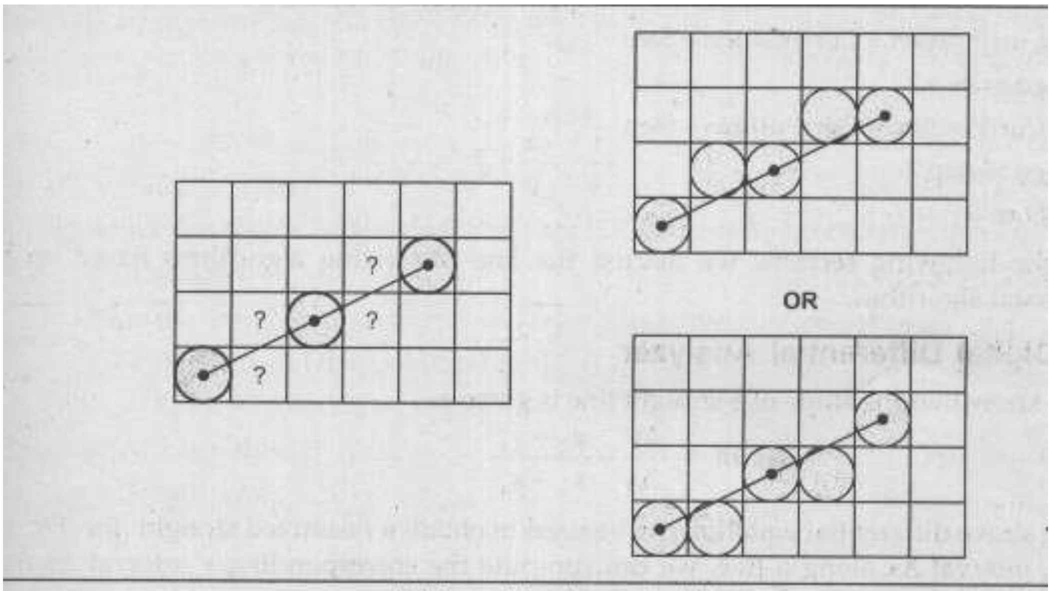



1
△x = |X2 - X1|
△Y = |Y2 - Y1|
If
abs(△x) >= abs(△Y)
Y
Steps = △x
N
Steps = △Y
xIncrement = △x/steps
y Increment = △y/steps
Initialize(x,y)
with (x1,y1)
Plot the rounded
coordinate (x , y)
Initialize counter
K=1
3
x = x + x Increment
y = y + y Increment
Plot [round(x) , round(y)]
K=k+1
1


3
1
Y
If
K=1
N
STOP
2



(a) Vertical and Horizontal lines (b) 45° line



(c) Line with other orientation

ANSWER THE FOLLOWING QUSTIONS.

Q.N.01 WRITE “C” CODE PROGRAM FOR D.D.A. LINE DRAWING METHOD?

Q.NO.02 CONSIDER THE LINE FROM (0,0) TO (-6,-6)USE SIMPLE D.D.A. ALGORITHM TO RASTERIZE THIS LINE?

EXPERIMENT No. 02

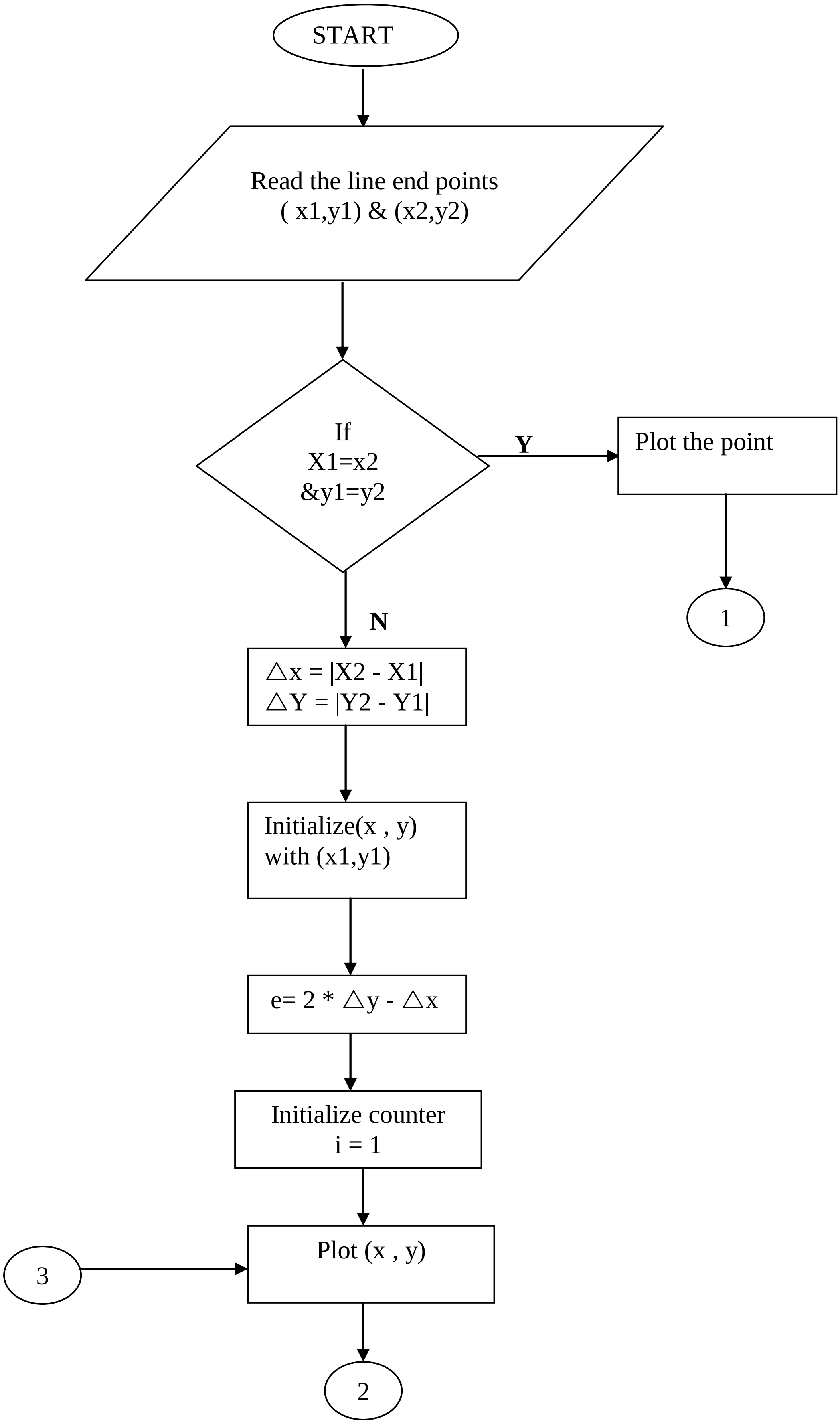
OBJECT

Bresenham's Line Drawing Method

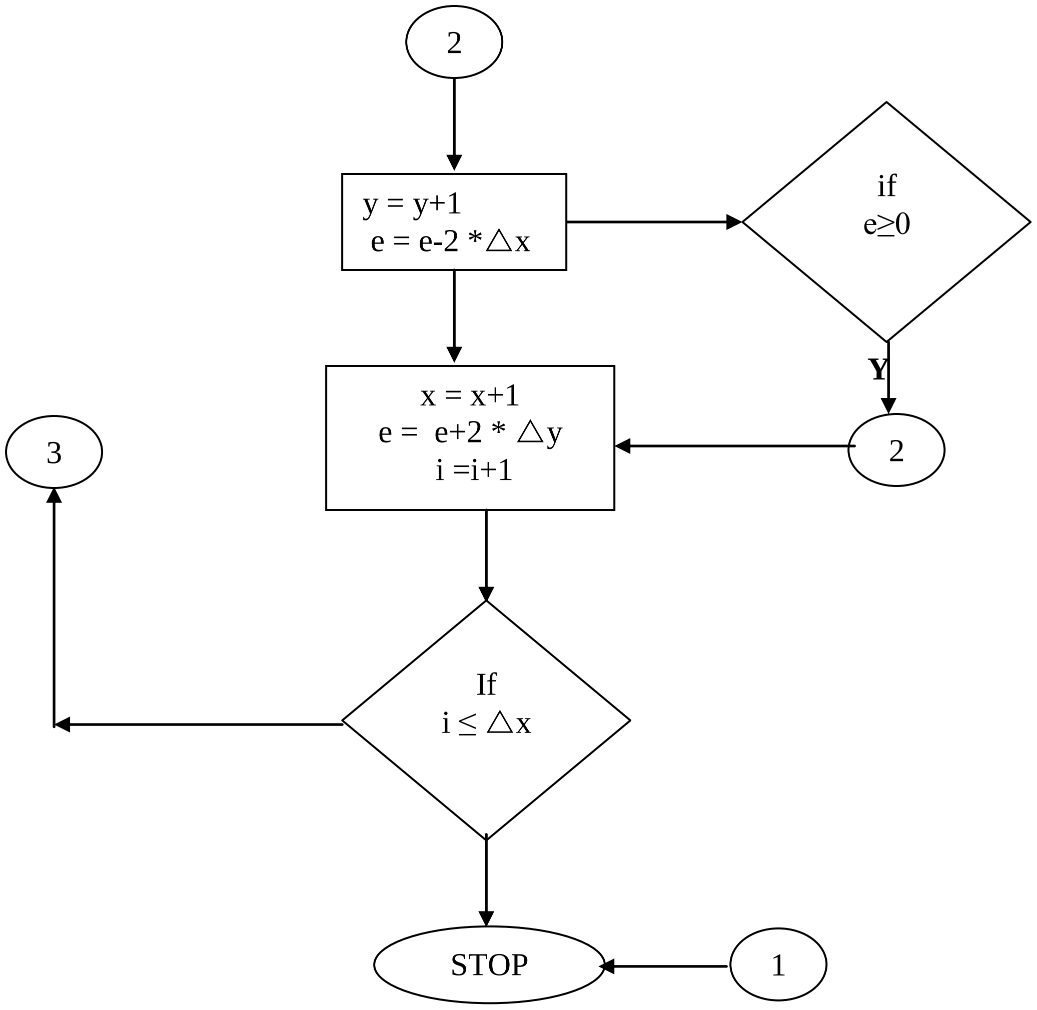
ALGORITHM

```
1.   Read the line end points (x1,y1) & (x2,y2) such that they are not equal.
2.   If the points are same, plot the points & exit (Else continue with the
     calculation)
3.   Calculate
     $\triangle x = |x_2-x_1|$
     $\triangle y = |y_2-y_1|$
4.   [initialize the starting point]
     x=x1
     y=y1
5.   e = 2 * $\triangle y$ - $\triangle x$
     [initialize value of decision variable or error to compensate for non zero
         intercepts]
6.   i = 1[initialize counter]
7.   plot(x , y)
8.   while (e $\geq$ 0)
       {
         y = y+1
         e = e – 2 * $\triangle x$
       }
         x = x+1
         e = e+2 * $\triangle y$
9.   i = i+1
10.  if ( i $\leq$ $\triangle x$ ) then go to step 6.
11.  Stop.
```

FLOW CHART



START
Read the line end points
(x1,y1) & (x2,y2)
If
X1=x2
&y1=y2
Y
Plot the point
1
N
△x = |X2 - X1|
△Y = |Y2 - Y1|
Initialize(x , y)
with (x1,y1)
e= 2 * △y - △x
Initialize counter
i = 1
3
Plot (x , y)
2


2
y = y+1
e = e-2 *△x
if
e≥0
Y
x = x+1
e = e+2 * △y
i =i+1
2
3
If
i ≤ △x
STOP
1

ANSWER THE FOLLOWING QUSTIONS.

Q.N.01- WRITE “C” CODE PROGRAM FOR BRESENHAM’S LINE DRAWING METHOD?

Q.N.02-CONCIDER THE LINE FROM (5,5) TO (13,9).USE THE BRESENHAM’S ALGORITHM ?

EXPERIMENT NO. 03

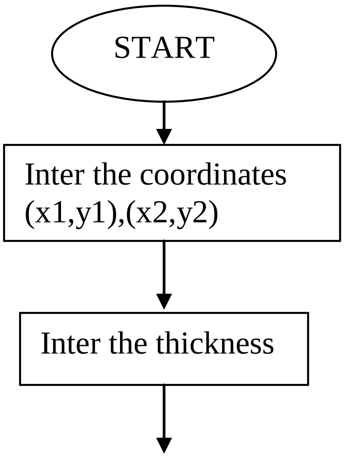
OBJECT

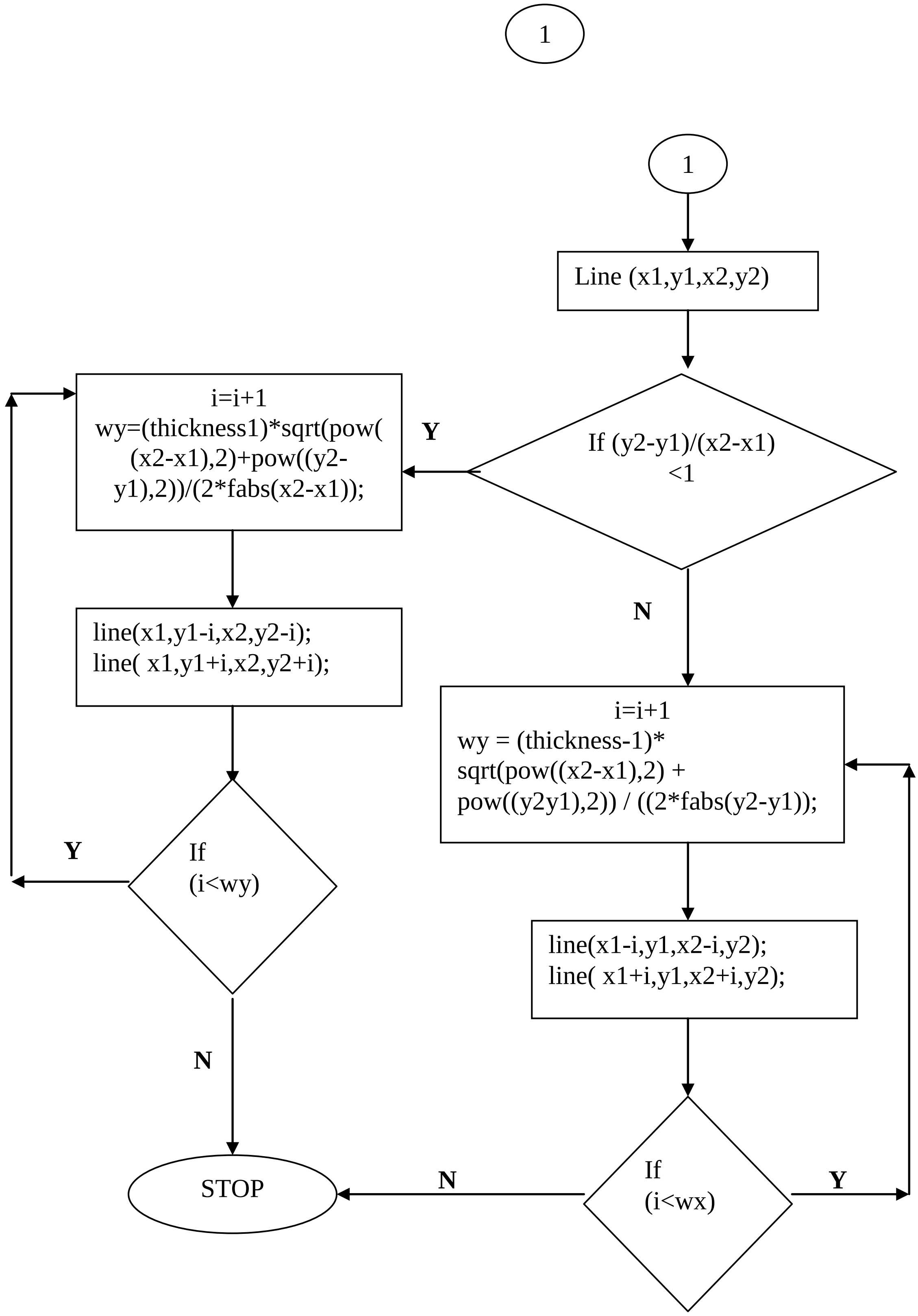
Thick Line Drawing Method

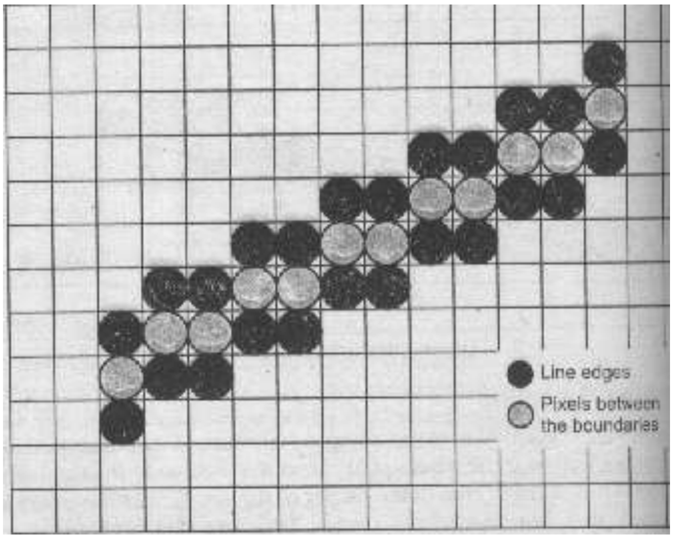
ALGORITHM

```
1.    Inter the coordinates for the line .
        p1 = (x1,y1)  ,   p2 = (x2,y2)
2.     enter the thickness.
3.     draw line (x1,y1,x2,y2).
4.     if ( (y2-y1) / (x2-x1) < 1)
       {
         wy =  (thickness-1) * sqrt (pow ( (x2-x1),2)+ pow ((y2-y1),2))/(2*fabs(x2-x1));
              while(i<wy)
         {
             i = i+1;
              line(x1,y1-i,x2,y2-i);
              line( x1,y1+i,x2,y2+i);
           }
        }
              else
          {

                  while(i<wx);
              {

                 wx=(thickness-1)*sqrt(pow((x2-x1),2)+pow((y2-y1),2))/(2*fabs(y2-y1));
                 line(x1-i,y1,x2-i,y2);
                  line( x1+i,y1,x2+i,y2);
                 i= i+1;
                 }
             }

6.     determine the thickness of line.
7.      stop.
```

FLOW CHART




1
1
Line (x1,y1,x2,y2)
If (y2-y1)/(x2-x1) <1
Y
i=i+1
wy=(thickness1)*sqrt(pow((x2-x1),2)+pow((y2-y1),2))/(2*fabs(x2-x1));
line(x1,y1-i,x2,y2-i);
line(x1,y1+i,x2,y2+i);
If (i<wy)
Y
N
STOP
N
i=i+1
wy = (thickness-1)*
sqrt(pow((x2-x1),2) +
pow((y2y1),2)) / ((2*fabs(y2-y1));
line(x1-i,y1,x2-i,y2);
line(x1+i,y1,x2+i,y2);
If (i<wx)
N
Y



THICK LINE DRAWING

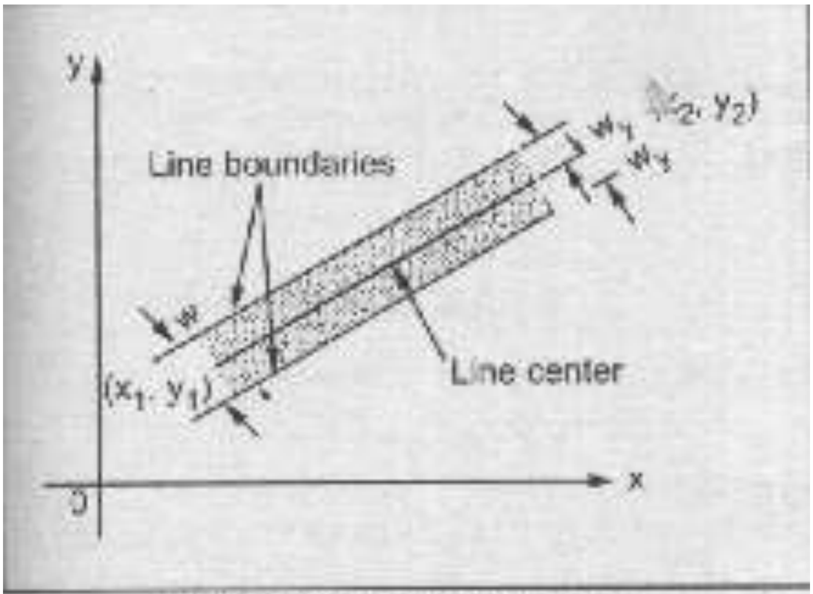


Fig. 2.18 Thick line details

ANSWER THE FOLLOWING QUSTIONS.

Q.N.01- WRITE “C” CODE PROGRAM FOR THICK LINE DRAWING METHOD?

Q.N.02- EXPLAINE THE CONCEPT OF BASIC LINE DRAWING?

Experiment no 04

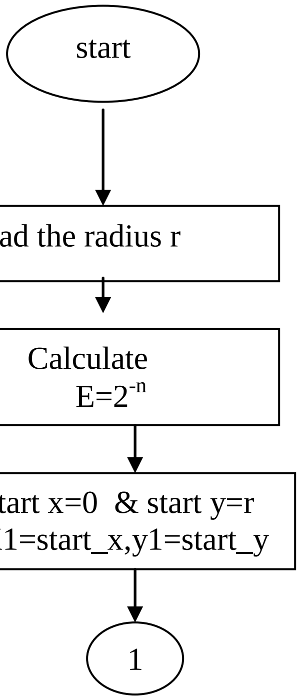
OBJECT:

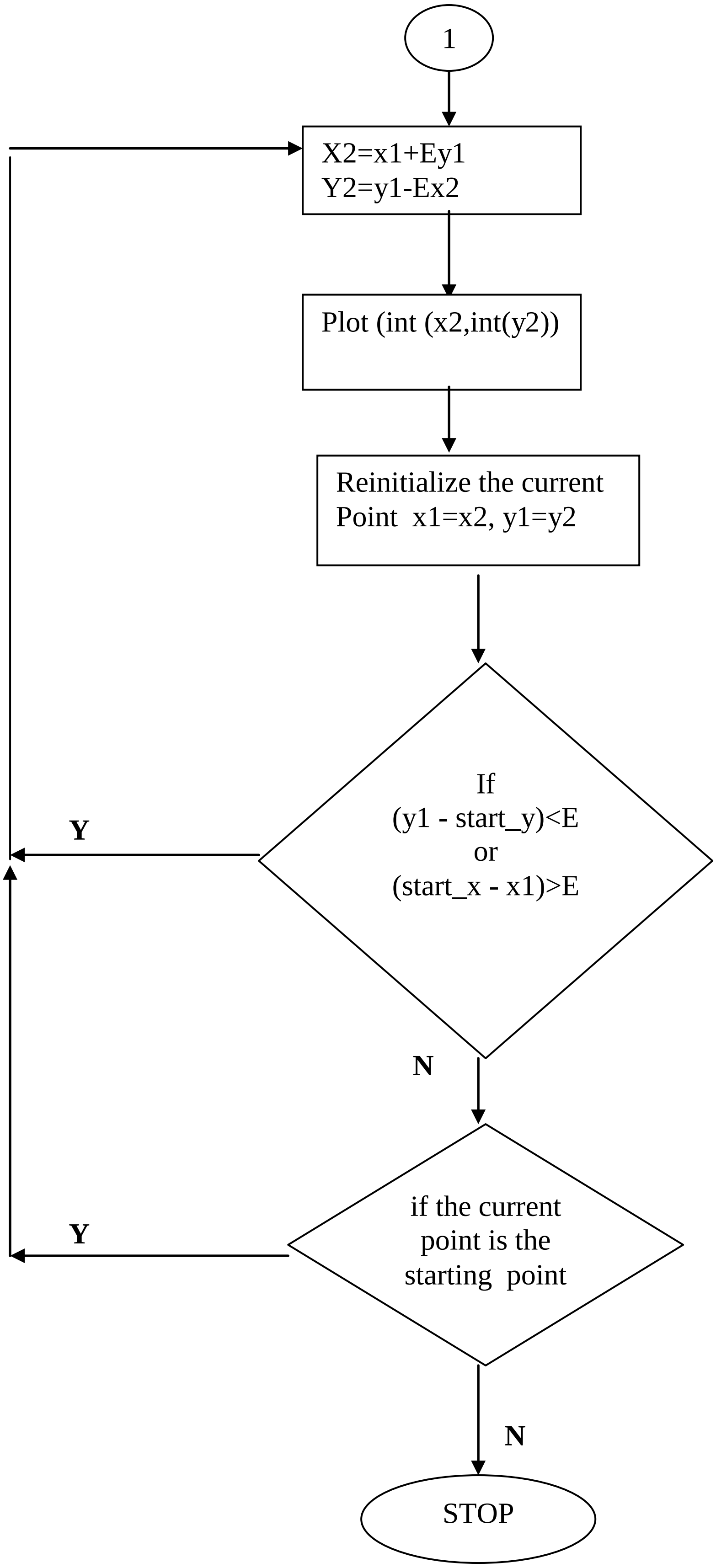
DDA CIRCLE DRAWING METHOD

ALGORITHM

```
1 .     read the radius r ,of the circle and calculate value of E
        $2^{n-1}<=r<2^n$
        $E=2^{-n}$
2  .    start_x=0
         start_y=r
3       x1=start_x
        y1=start_y
4                 do
         {
                  x2=x1+Ey1
                  y2=y1-Ex2
                  [x2 represents xn+1 and x1 presents xn]
                   plot (int(x2),int(y2))
                    x1=x2;
                   y1=y2;
                  [reinitialize the current point]
         }while (y1-start_y)<E or (start_x-x1)>E
                  [check if the current  point is the starting  point or not .if current point is
                            not starting point repeat step 4; otherwise stop]
5.                Stop.
```

FLOW CHART




1
X2=x1+Ey1
Y2=y1-Ex2
Plot (int (x2,int(y2))
Reinitialize the current
Point x1=x2, y1=y2
If
(y1 - start_y)<E
or
(start_x - x1)>E
Y
N
if the current
point is the
starting point
Y
N
STOP

ANSWER THE FOLLOWING QUSTIONS.

Q.N.01- WRITE “C” CODE PROGRAM FOR **DDA CIRCLE DRAWING METHOD** ?

Q.N.02- EXPLAINE THE BASIC CONCEPT OF CIRCLE DRAWING?

Experiment 05

OBJECT

To draw the 1/8th, of the circle of a given radius.

ALGORITHM

```
1.  Read the radius of the circle.
2.  Calculate initial decision variable.
    d= 3 – 2r
3.  Initialize start p
    x = 0, y = r
4.  do
    plot (x,y)
    {
        if d < 0 then
            d =  d + 4x + 6
        else
        {
            d = d + 4 (x-y) + 10
            y = y – 1
        }
            x = x +1
    }  while(x<y)

5.  Plot (x, y)
6.  End
```

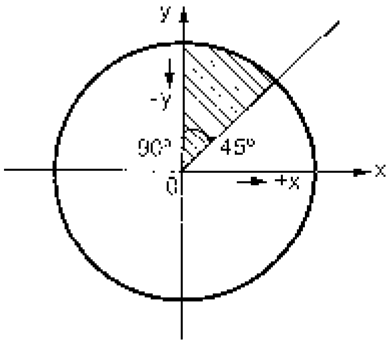
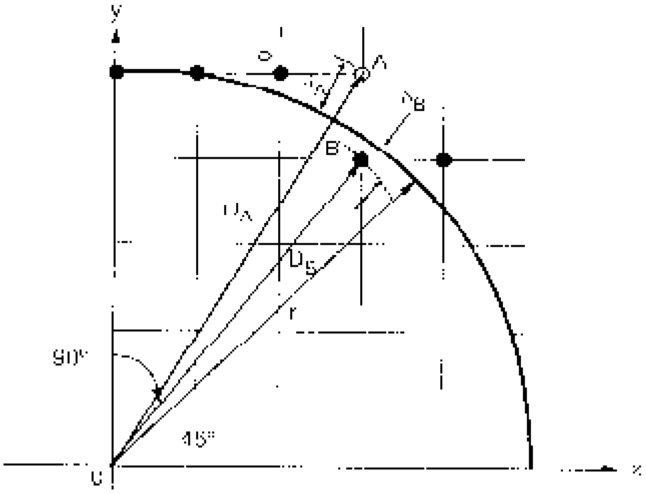
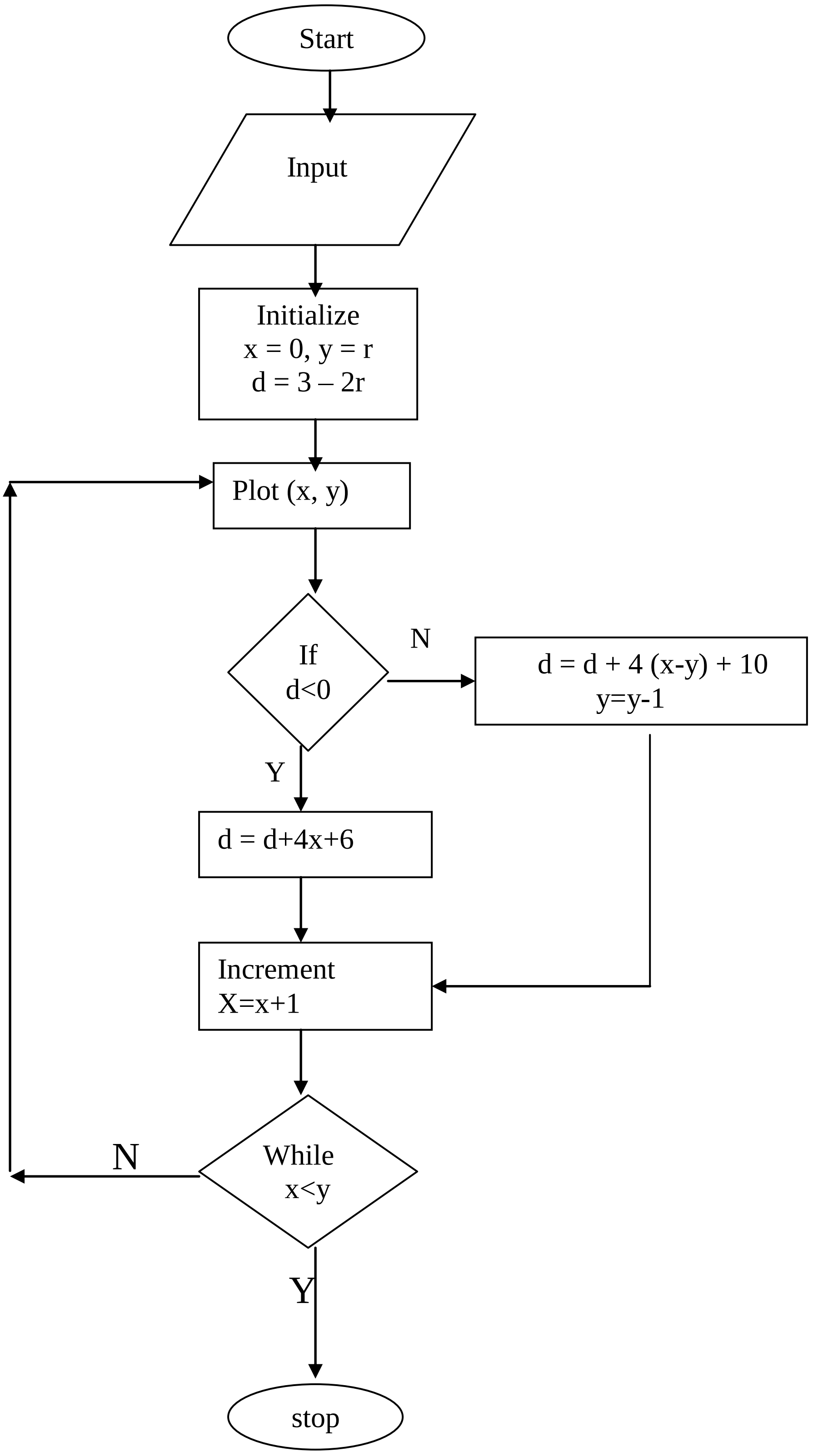


Fig. 2.22 1/8 part of circle



Scan conversion with Bresenham's algorithm

FLOW CHART


Start
Input
Initialize
x = 0, y = r
d = 3 – 2r
Plot (x, y)
If
d<0
N
d = d + 4 (x-y) + 10
y=y-1
Y
d = d+4x+6
Increment
X=x+1
N
While
x<y
Y
stop

ANSWER THE FOLLOWING QUSTIONS.

Q.N.01- WRITE “C” CODE PROGRAM FOR **BRESENHAM’S CIRCLE DRAWING METHOD** ?

Q.N.02-GIVE DIFFERENT METHODS OF REPRESENRING A CIRCLE?

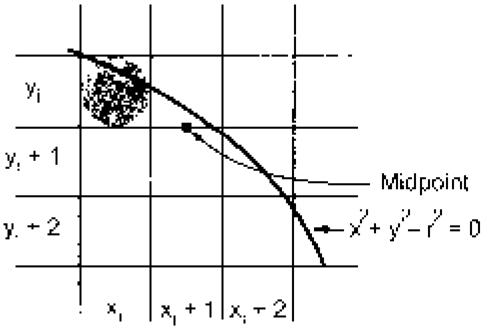
Experiment No: 06

OBJECT

Midpoint Circle Drawing Algorithm

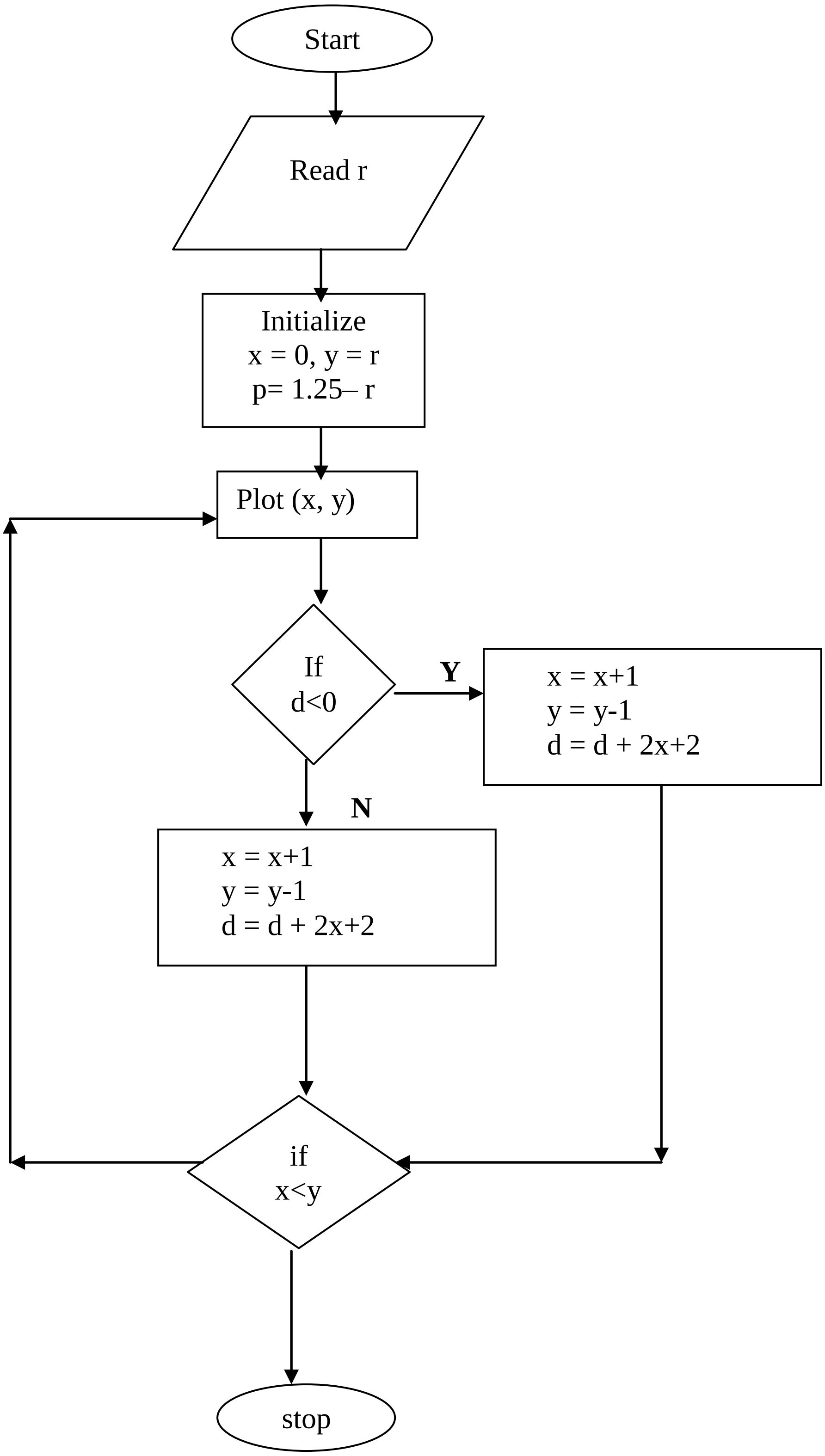
ALGORITHM

```
1.  Read the radius r of the circle.
2.  Initialize starting position as
    x=0
    y=r
3.  Calculate initial value of decision parameter as
    P=1.25-r
4.  $D_0$
    {
        plot(x , y)
        if (d<0)
        {
            x=x+1
            y=y
            =d+2x+2
        }
        else
        {
            x=x+1
            y=y-1
            d=d+2x+2y+1
        }
    }While (x<y)
5.  Determine symmetry point
6.  Stop
```



Decision parameter to select correct pixel in circle generation algorithm

FLOW CHART


Start
Read r
Initialize
x = 0, y = r
p= 1.25– r
Plot (x, y)
If
d<0
Y
x = x+1
y = y-1
d = d + 2x+2
N
x = x+1
y = y-1
d = d + 2x+2
if
x<y
stop

ANSWER THE FOLLOWING QUSTION.

Q.N.01- WRITE “C” CODE PROGRAM FOR **MIDPOINT CIRCLE DRAWINGMETHOD?**

Experiment no :10

OBJECT

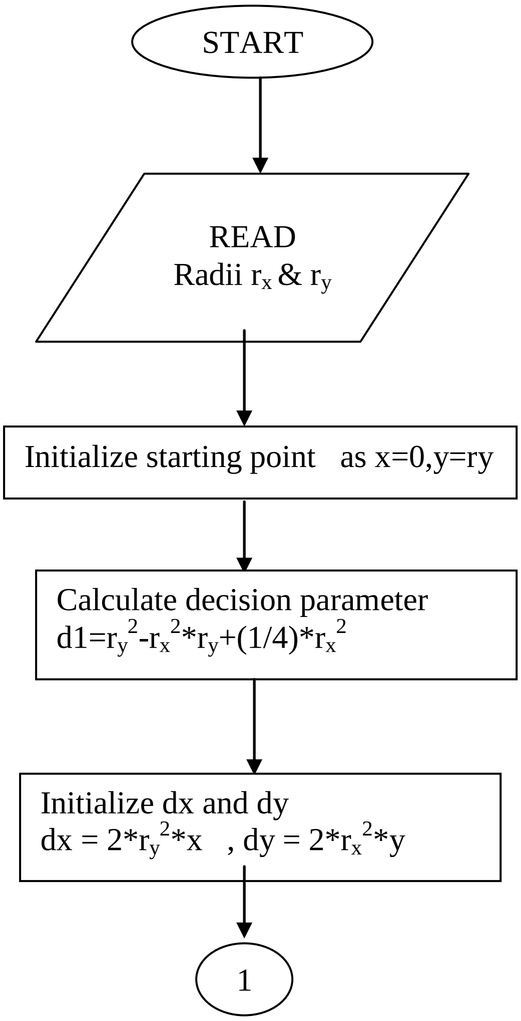
Mid point ellipse drawing algorithm

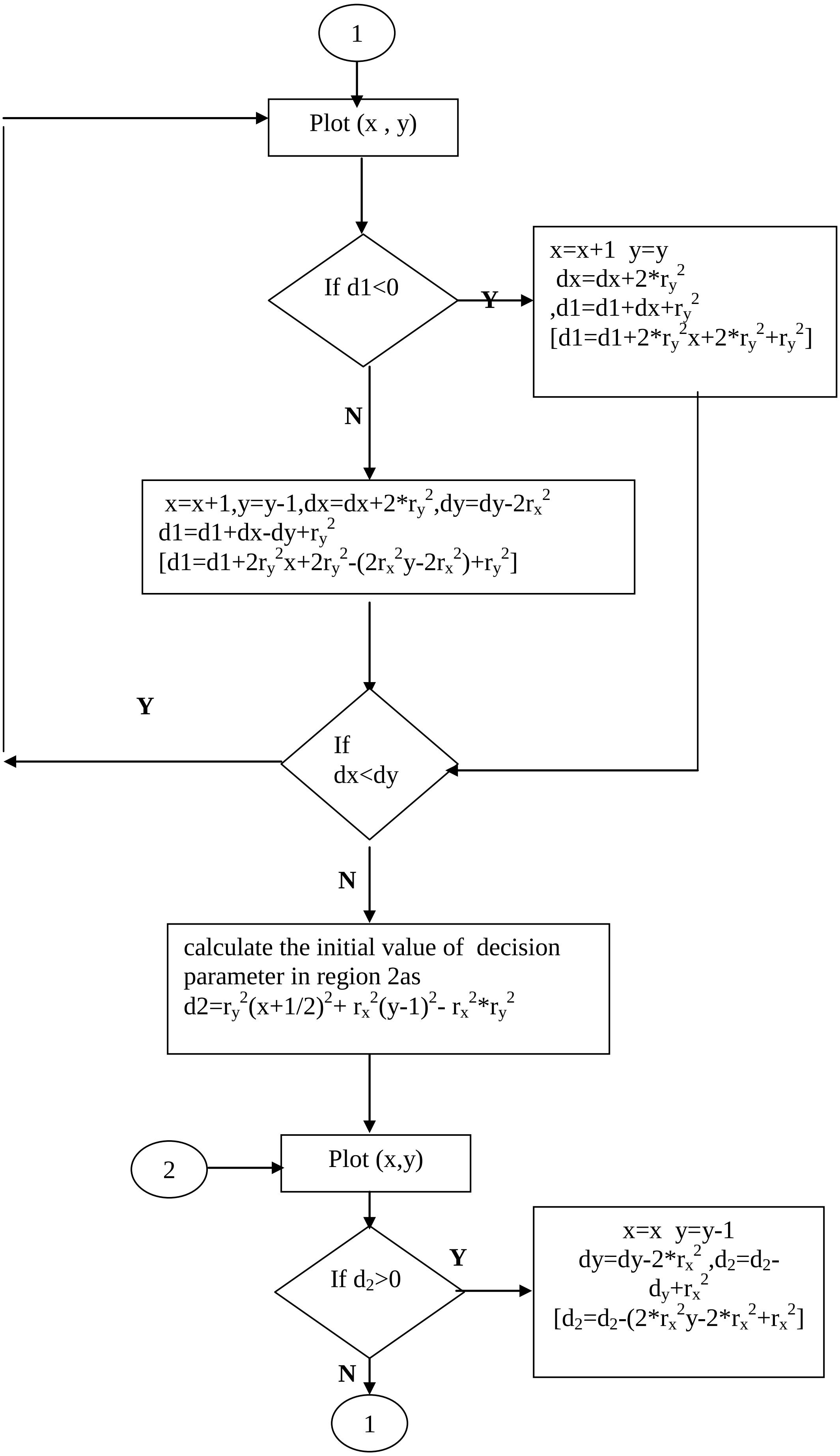
ALGORITHM

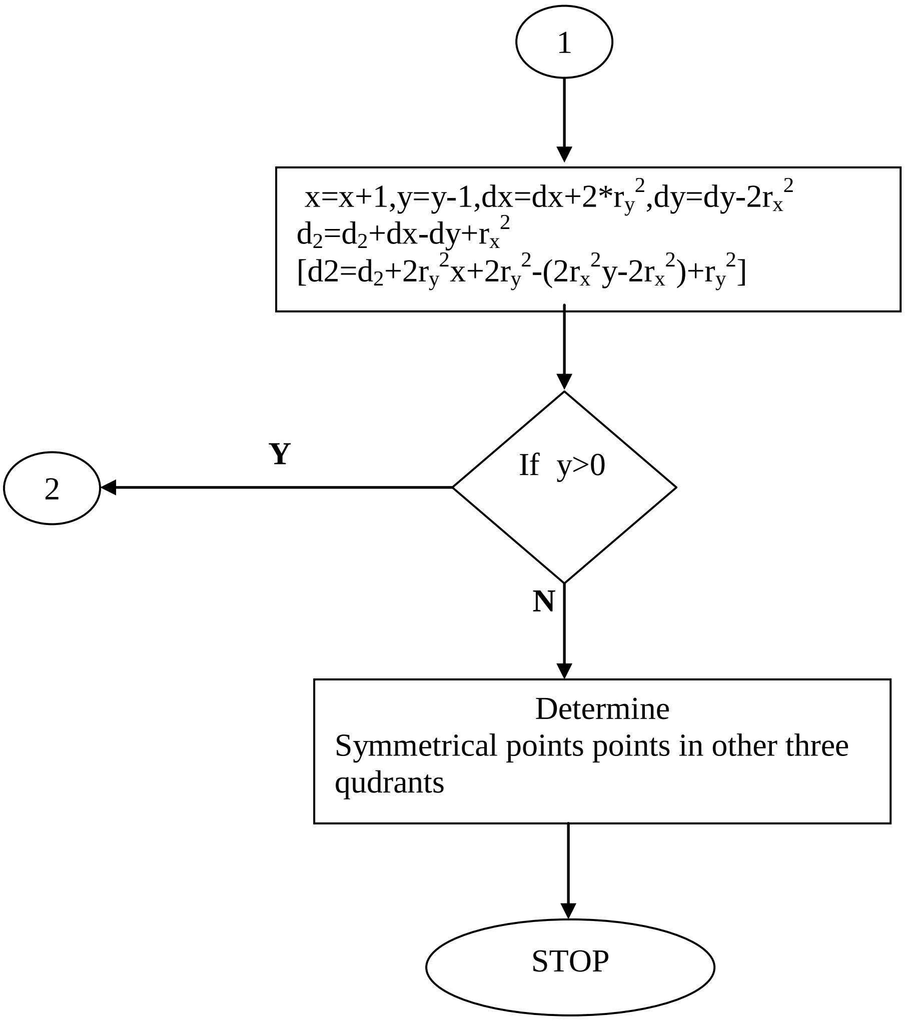
```
1.  read radii rx and ry.
2.  Initialize starting point as
    x=0
    y=ry
3.  calculate the initial value of decision parameter in region
    1 as  $d1=r_y^2-r_x^2*r_y+(1/4)*r_x^2$
4.  initialize dx and dy as
    $dx=2*r_y^2*x$
    $dy=2*r_x^2*y$
5.  do
    {

            plot (x , y)
        if (d1<0)
        {
            x=x+1
            y=y
            $dx=dx+2*r_y^2$
            $d1=d1+dx+ r_y^2$
            $[d1=d1+2*r_y^2*x+2*r_y^2]$
        }
        else
        {
             x=x+1
             y=y-1
            $dx = dx+2*r_y^2$
            $dy = dy-2r_x^2$
            $d1=d1+dx-dy+r_y^2$
            $[d1=d1+2*r_y^2*x+2r_y^2-(2r_x^2*y-2*r_x^2 )+r_y^2]$
    }  while(dx<dy)
6.  calculate the initial value of decision parameter in region 2as
    $d2=r_y^2(x+1/2)^2 + r_x^2(y-1)^2-r_x^2*r_y^2$
7.  do
    {
        plot (x,y)
        if(d2>0)
        {
            x=x
            y=y-1
```

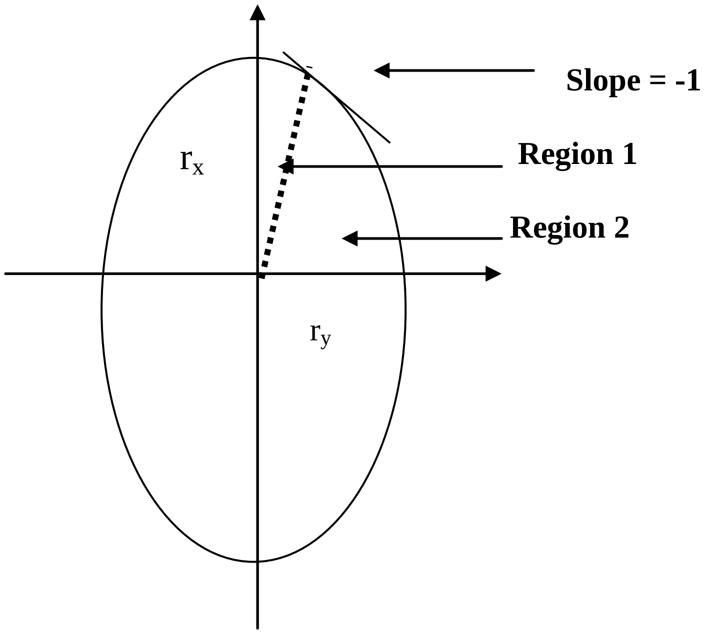
```
                $dy=dy-2*r_x^2$
                $d2=d2-dy+r_x^2$
                $[d2=d2-(2*r_x^2y-2r_x^2)+r_x^2]$
            }
            else
            {
                $x=x+1$
                $y=y-1$
                $dy=dy-2*r_x^2$
                $dx=dx+2*r_y^2$
                $d2=d2+dx-dy+r_x^2$
                $d2=[d2+2*r_y^2*x+2*r_y^2-(2*r_x^2*y-2*r_x^2)+r_y^2]$
            }
        }while(y>0)
8.      determine symmetrical points in other three quadrants
9.      stop.
```

FLOW CHART

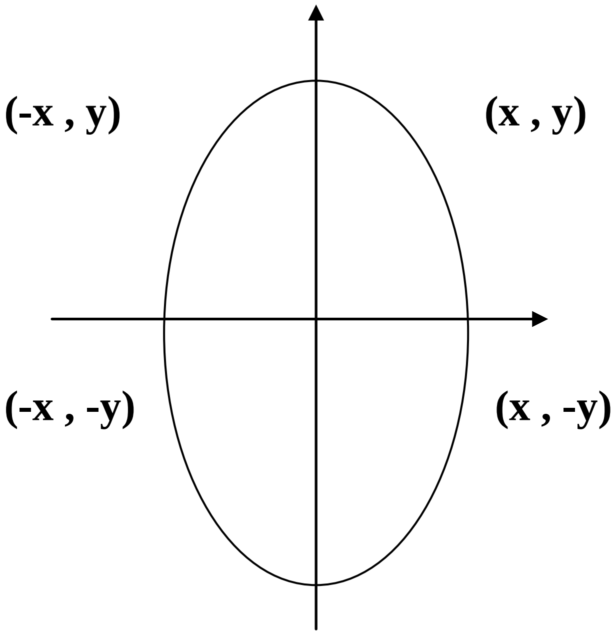

START
READ
Radii r_x & r_y
Initialize starting point as x=0,y=ry
Calculate decision parameter
$d1=r_y^2-r_x^2*r_y+(1/4)*r_x^2$
Initialize dx and dy
$dx = 2*r_y^2*x$, $dy = 2*r_x^2*y$
1


1
Plot (x , y)
If d1<0
Y
x=x+1 y=y
dx=dx+2*r_y^2
,d1=d1+dx+r_y^2
[d1=d1+2*r_y^2x+2*r_y^2+r_y^2]
N
x=x+1,y=y-1,dx=dx+2*r_y^2,dy=dy-2r_x^2
d1=d1+dx-dy+r_y^2
[d1=d1+2r_y^2x+2r_y^2-(2r_x^2y-2r_x^2)+r_y^2]
Y
If dx<dy
N
calculate the initial value of decision parameter in region 2as
d2=r_y^2(x+1/2)^2+ r_x^2(y-1)^2- r_x^2*r_y^2
2
Plot (x,y)
If d_2>0
Y
x=x y=y-1
dy=dy-2*r_x^2 ,d_2=d_2-d_y+r_x^2
[d_2=d_2-(2*r_x^2y-2*r_x^2+r_x^2]
N
1


1
$x=x+1, y=y-1, dx=dx+2*r_y^2, dy=dy-2r_x^2$
$d_2=d_2+dx-dy+r_x^2$
$[d2=d_2+2r_y^2x+2r_y^2-(2r_x^2y-2r_x^2)+r_y^2]$
If y>0
Y
2
N
Determine
Symmetrical points points in other three qudrants
STOP



Ellipse



Four Way Symmetry of ellipse

ANSWER THE FOLLOWING QUSTION.

Q.N.01- WRITE “C” CODE PROGRAM FOR Mid point Ellipse **DRAWING METHOD?**

Experiment No. 07

OBJECT

Cohen and Sutherland Subdivision Line

ALGORITHM

1. Read two end points of the line say P_1 (x_2-y_2).

2. Read two corners (left-top and right-bottom) of the window, say
(Wx_1, Wy_1 and Wx_2, Wy_2).

3. Assign the region codes for two endpoints P1 and P2 using following steps :
Initialize code with bits 0000

Set Bit 1	- if ($x<Wx_1$)
Set Bit 2	- if ($x>Wx_2$)
Set Bit 3	- if ($y<Wy_2$)
Set Bit 4	- if ($y>Wy_1$)

4. Check for visibility of line P_1 P_2

a. If region codes for both endpoints P_1 and P_2 are zero then the line is completely visible. Hence draw the line and go to step 9.

b. If region codes for endpoints are not zero and the logical ending of them is also nonzero then the line is completely invisible, so reject the line and go to step 9

c. If region codes for two end point do not satisfy the conditions in 4a) and 4b) the line is partially visible.

5. Determine the intersecting edge of the clipping window by inspecting the region codes of two end points.

a. If region codes for both end points are non-zero, find intersection points P_1 and P_2 with boundary edges of clipping window with respect to it.

b. If region code for any one end point is non zero then find intersection points P_1 or P_2 with the boundary edge of the clipping window with respect to it.

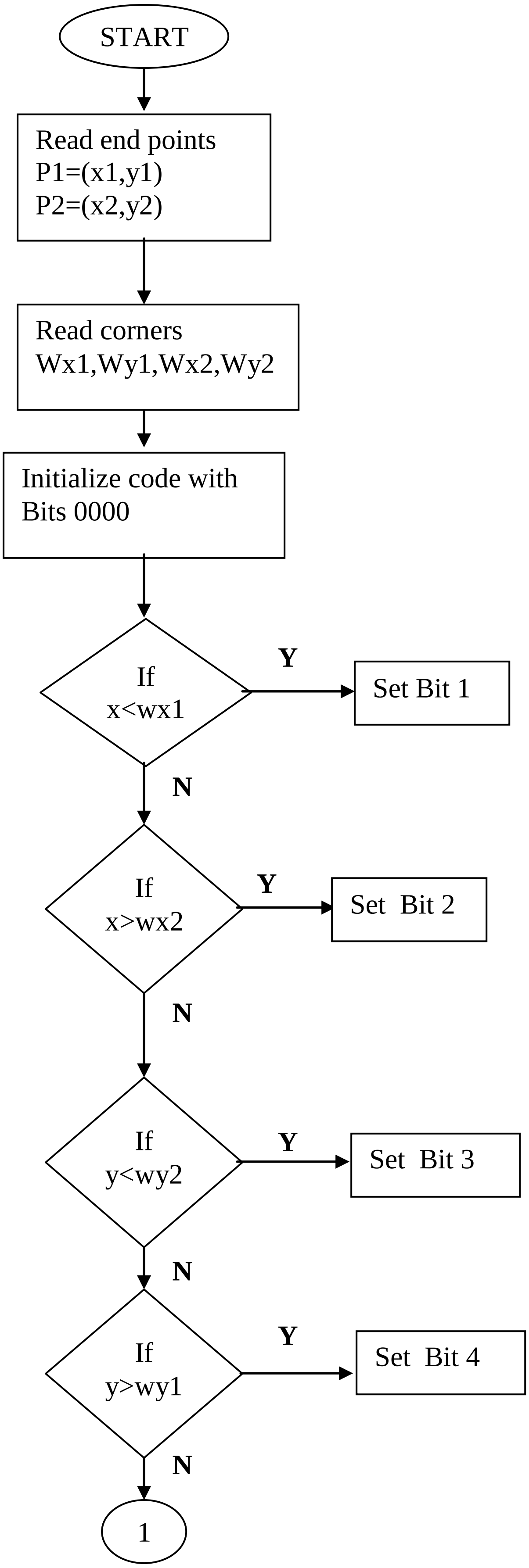
6. Divide the line segment considering intersection points.

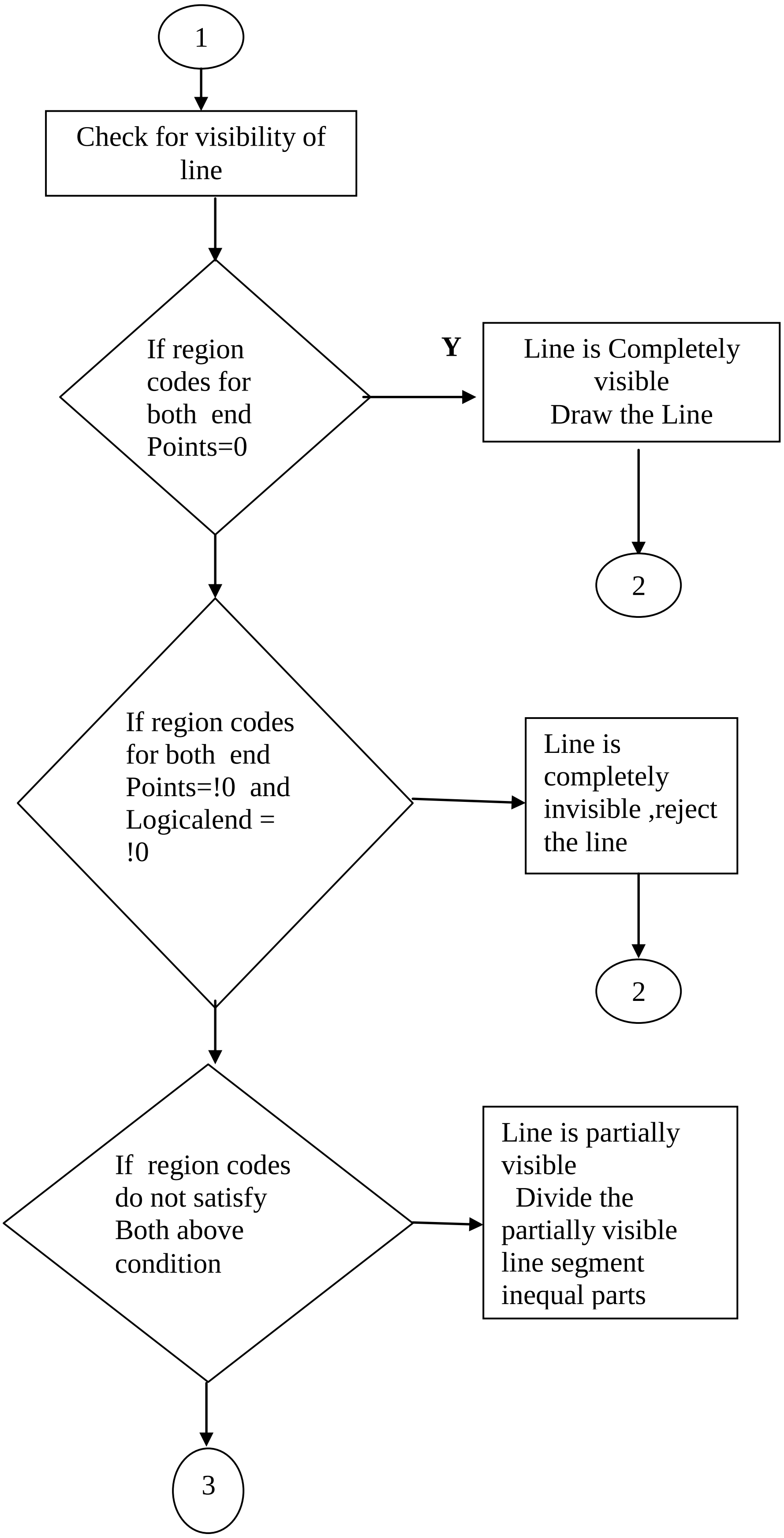
7. Reject the line segment if any one end point of it appears outsides the clipping window.

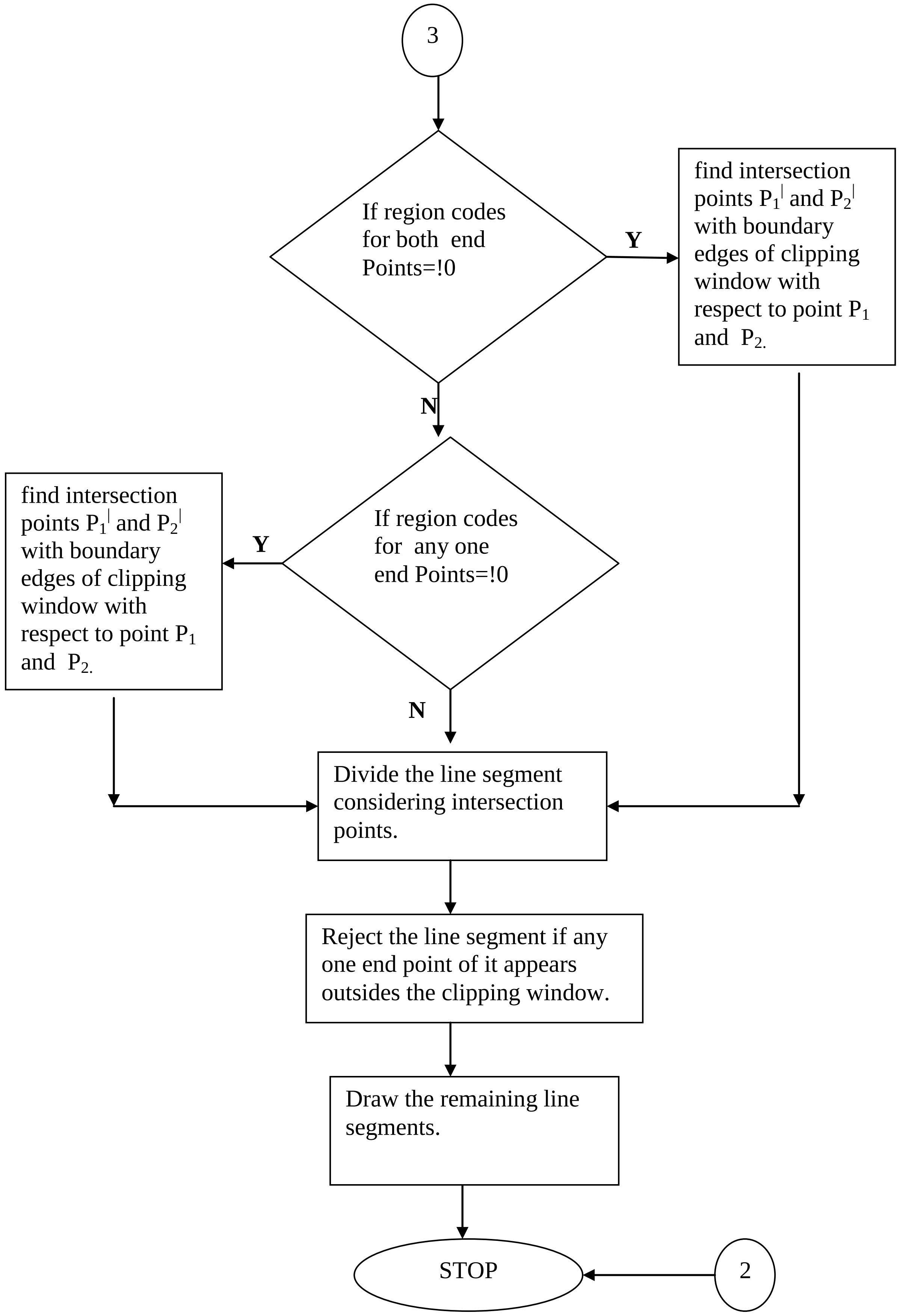
8. Draw the remaining line segments.

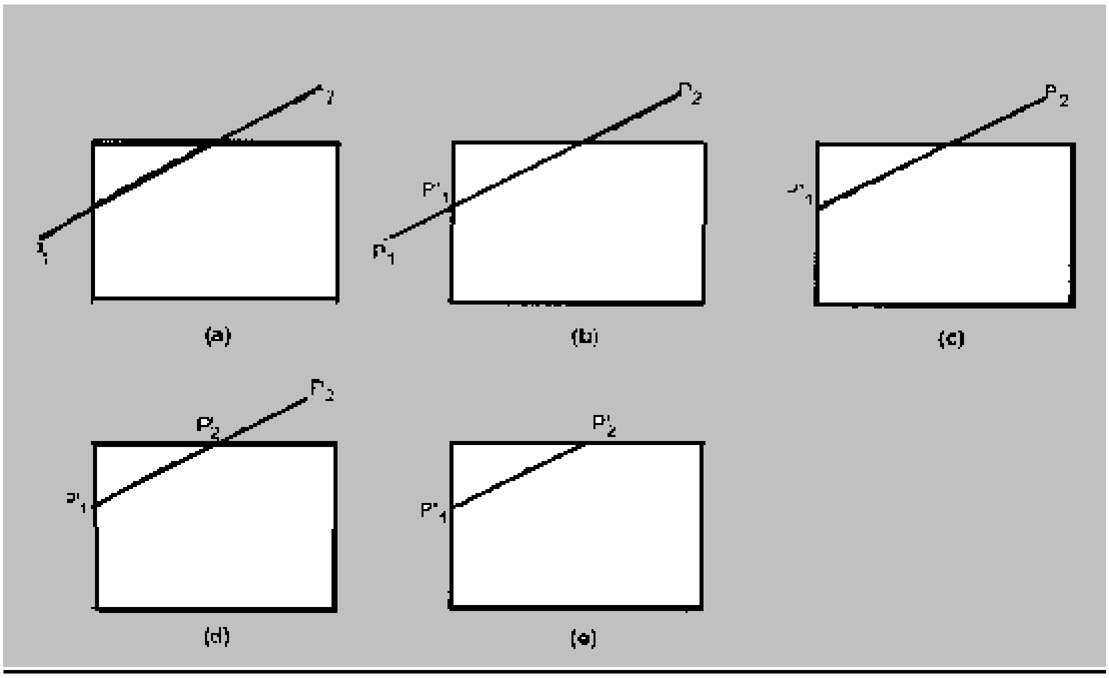
9. Stop.

FLOW CHART


START
Read end points
P1=(x1,y1)
P2=(x2,y2)
Read corners
Wx1,Wy1,Wx2,Wy2
Initialize code with
Bits 0000
If
x<wx1
Y
Set Bit 1
N
If
x>wx2
Y
Set Bit 2
N
If
y<wy2
Y
Set Bit 3
N
If
y>wy1
Y
Set Bit 4
N
1


1
Check for visibility of line
If region codes for both end Points=0
Y
Line is Completely visible
Draw the Line
2
If region codes for both end Points=!0 and Logicalend = !0
Line is completely invisible ,reject the line
2
If region codes do not satisfy Both above condition
Line is partially visible
Divide the partially visible line segment inequal parts
3


3
If region codes for both end Points=!0
Y
find intersection points P1| and P2| with boundary edges of clipping window with respect to point P1 and P2.
N
If region codes for any one end Points=!0
Y
find intersection points P1| and P2| with boundary edges of clipping window with respect to point P1 and P2.
N
Divide the line segment considering intersection points.
Reject the line segment if any one end point of it appears outsides the clipping window.
Draw the remaining line segments.
STOP
2



SUTHERLAND AND COHEN SUBDIVISION LINE CLIPPINING

ANSWER THE FOLLOWING QUSTION.

Q.N.01- WRITE “C” CODE PROGRAM FOR Cohen and Sutherland Subdivision Line Clipping Method?

Experiment No 08

OBJECT

Midpoint Subdivision of line

ALGORITHM

1. Read two endpoints of the line say $P_1(x_1,y_1)$ and $P_2(x_2,y_2)$.

2. Read two corners (left-top and right-bottom) of the window, say (Wx_1,Wy_1 and Wx_2,Wy_2).

3. Assign region codes for two end points using following steps:

4. Initialize code with bits 0000
Set Bit 1 – if ($x < Wx_1$)
Set Bit 2 – if ($x > Wx_2$)
Set Bit 3 – if ($y < Wy_1$)
Set Bit 1 – if ($y > Wy_2$)

5. Check for visibility of line

If region codes for both endpoints are zero then the line is completely visible. Hence draw the line and go to step 6.

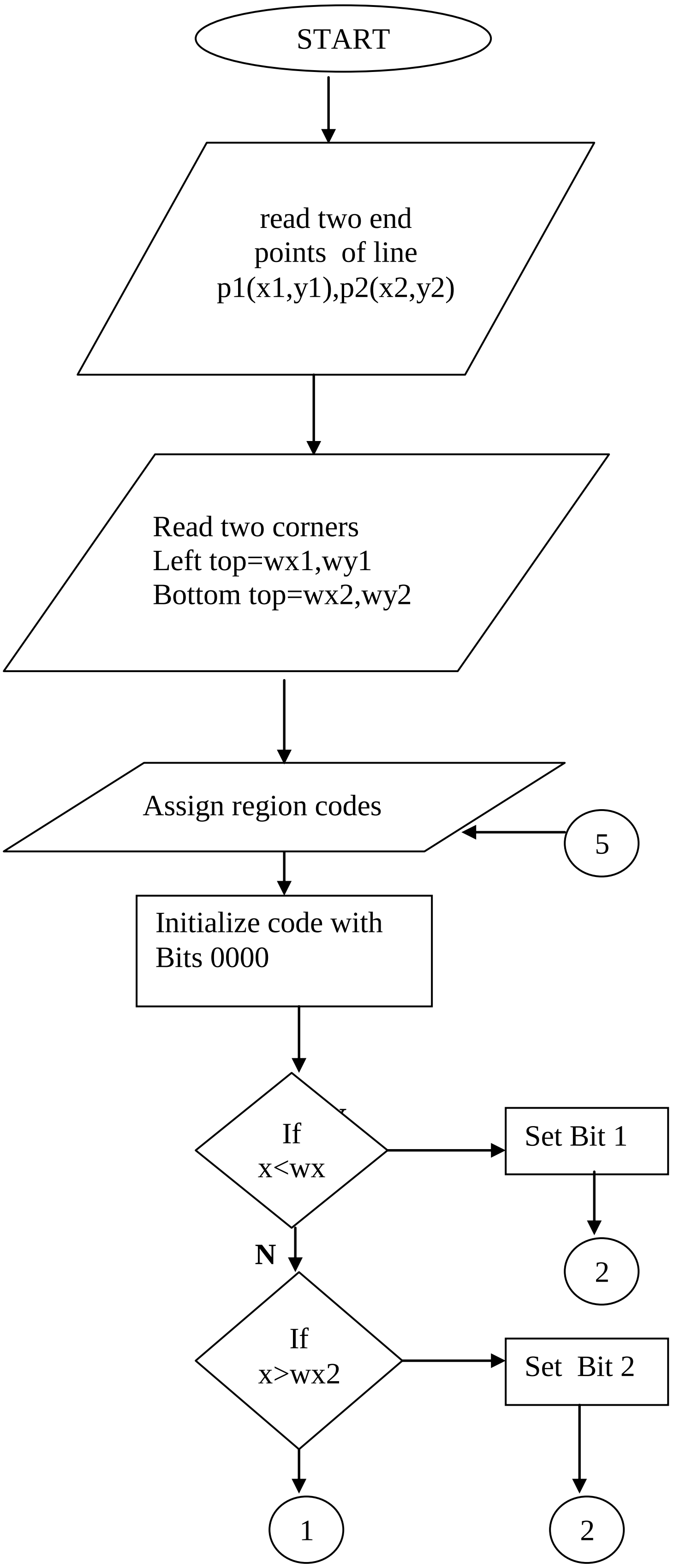
If region codes for two endpoints are not zero and the logical ending of them is also nonzero then the line is completely invisible, so reject the line and go to step 6.

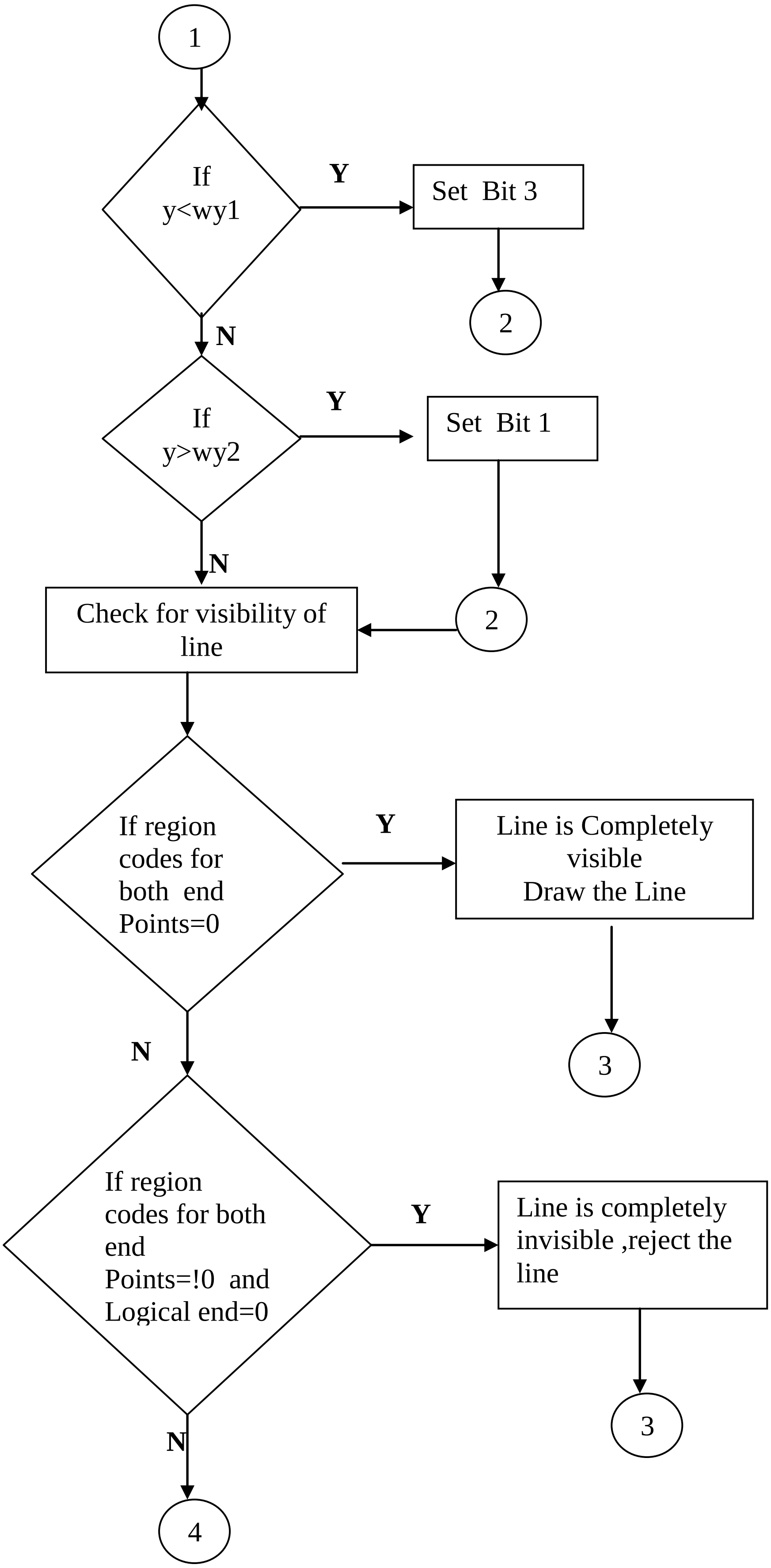
If region codes for two endpoints do not satisfy the conditions in 4a) and 4b) the line is partially visible.

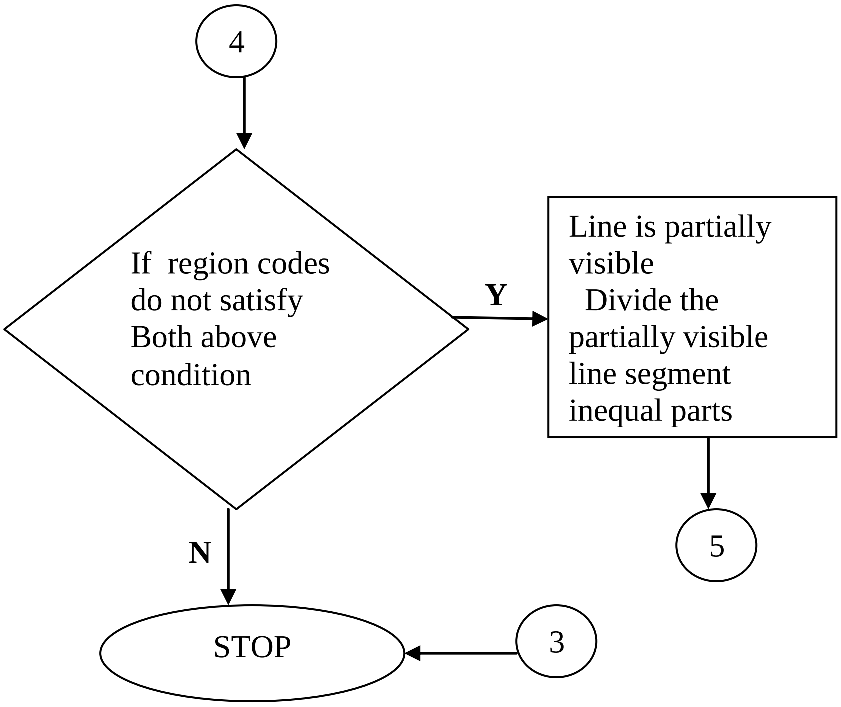
Divide the partially visible line segment in equal parts and repeat steps 3 through 5 for both subdivision line segments until you get completely visible and completely invisible line segments.

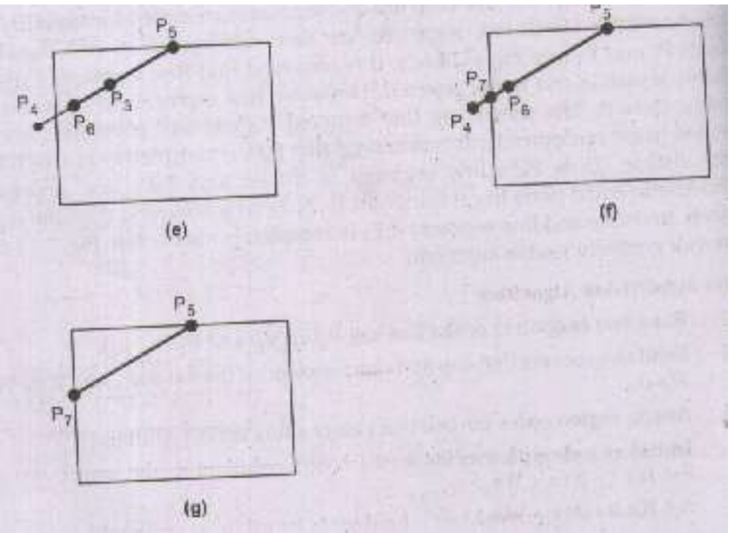
6. Stop.

FLOW CHART


START
read two end points of line p1(x1,y1),p2(x2,y2)
Read two corners
Left top=wx1,wy1
Bottom top=wx2,wy2
Assign region codes
5
Initialize code with Bits 0000
If x<wx
Set Bit 1
2
N
If x>wx2
Set Bit 2
1
2


1
If
y<wy1
Y
Set Bit 3
2
N
If
y>wy2
Y
Set Bit 1
N
Check for visibility of line
2
If region codes for both end Points=0
Y
Line is Completely visible
Draw the Line
3
N
If region codes for both end Points=!0 and Logical end=0
Y
Line is completely invisible ,reject the line
3
N
4


4
If region codes do not satisfy Both above condition
Y
Line is partially visible
Divide the partially visible line segment inequal parts
5
N
STOP
3



Midpoint Subdivision of line

ANSWER THE FOLLOWING QUSTION.

Q.N.01- WRITE "C" CODE PROGRAM FOR Midpoint Subdivision of line clipping **DRAWING METHOD?**

Experiment No:09

OBJECT

Bezier Curves

ALGORITHM

1. Get four control points say A(x_A, y_A), B(x_B, y_B), C(x_C, y_C), D(x_D, y_D).

2. Divide the curve represented by points say A, B, C and D in two sections.

$$\begin{aligned}
x_{AB} &= (x_A+x_B)/2 \\
y_{AB} &= (y_A+y_B)/2 \\
x_{BC} &= (x_B+x_C)/2 \\
y_{BC} &= (y_B+y_C)/2 \\
x_{CD} &= (x_C+x_D)2 \\
y_{CD} &= (y_C+y_D)/2 \\
x_{ABC} &= (x_{AB}+y_{BC})/2 \\
y_{ABC} &= (y_{AB}+y_{BC})/2 \\
x_{BCD} &= (x_{BC}+y_{CD})/2 \\
y_{BCD} &= (y_{BC}+y_{CD})/2 \\
x_{ABCD} &= (x_{ABC}+x_{BCD})/2 \\
y_{ABCD} &= (y_{ABC}+y_{BCD})/2
\end{aligned}$$

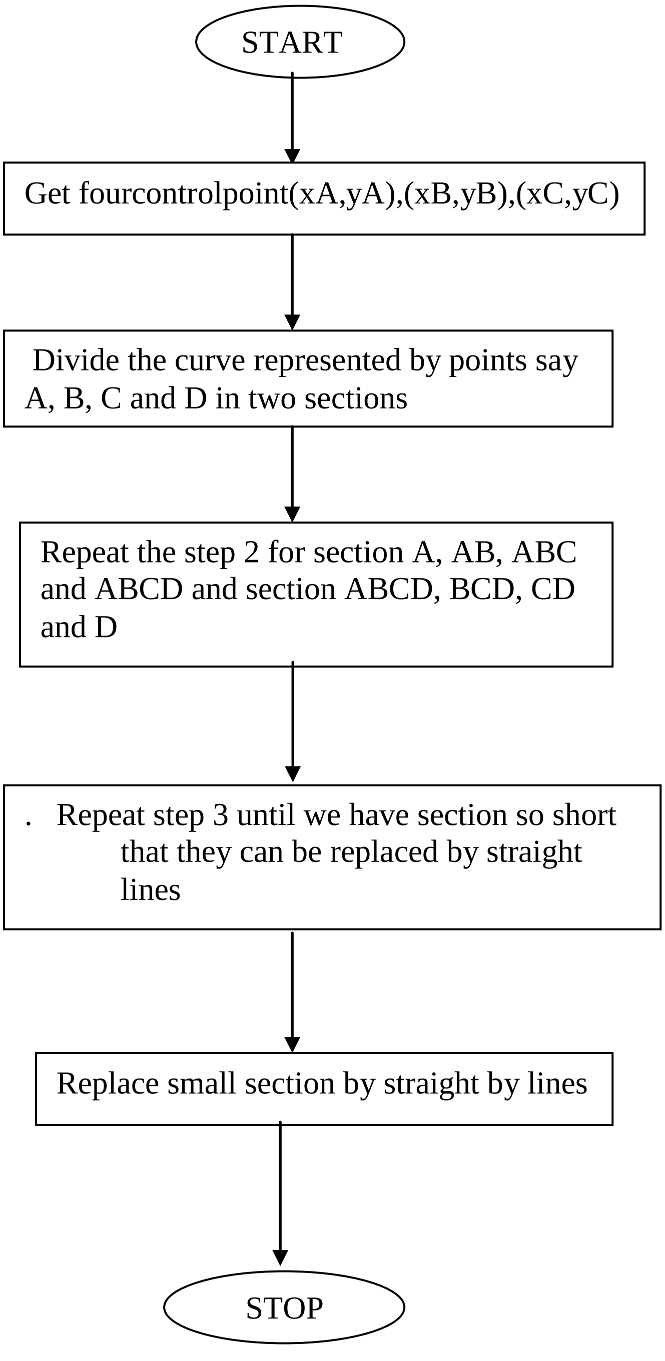
3. Repeat the step 2 for section A, AB, ABC and ABCD and section ABCD, BCD, CD and D.

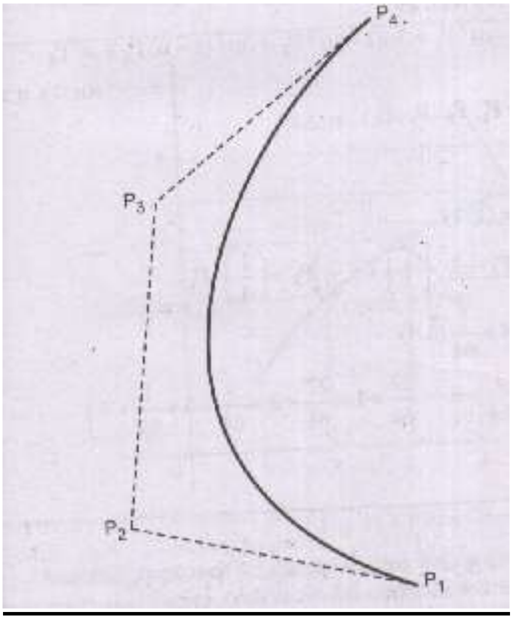
4. Repeat step 3 until we have section so short that they can be replaced by straight lines.

5. Replace small section by straight by lines.

6. Stop.

FLOW CHART


START
Get fourcontrolpoint(xA,yA),(xB,yB),(xC,yC)
Divide the curve represented by points say A, B, C and D in two sections
Repeat the step 2 for section A, AB, ABC and ABCD and section ABCD, BCD, CD and D
. Repeat step 3 until we have section so short that they can be replaced by straight lines
Replace small section by straight by lines
STOP



CUBIC BEZIER SPLINE

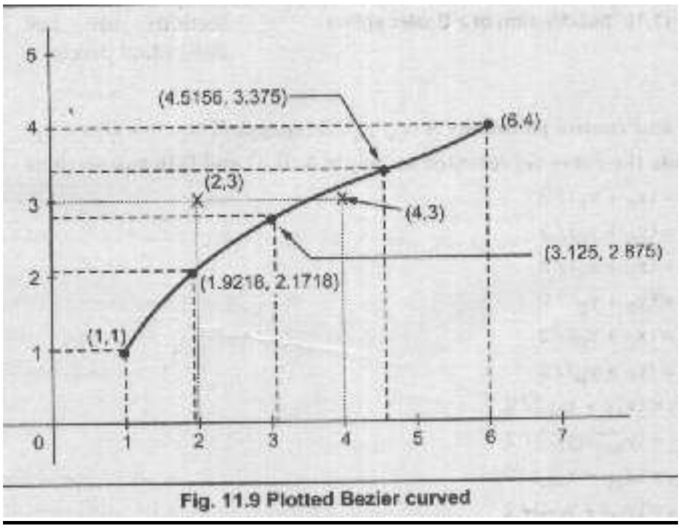


Fig. 11.9 Plotted Bezier curved

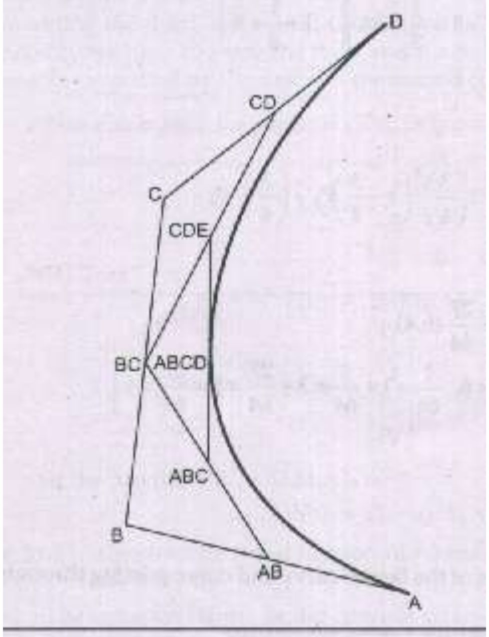


Fig. 11.10 Subdivision of a Bezier spline

ANSWER THE FOLLOWING QUSTION.

Q.N.01- WRITE “C” CODE PROGRAM FOR Bezier Curve **DRAWING METHOD?**