**Lab Exercise Date 23-01-2019**

## Write a C program to draw ellipse with major axis as 100, minor axis as 50 and centre as (200, 200) using [Mid-point’s ellipse drawing algorithm](http://cprogramtutorials.blogspot.com/2011/09/bresenhams-ellipse-drawing-algorithm-c.html).

## Write a C program to draw a polygon with coordinates: (100, 30), (50, 70), (70, 70), (30,150), (150,150) and fill it using Scan-Line Polygon Filling Algorithm

**Instructions**

## Steps of Midpoint Ellipse Algorithm

## Midpoint ellipse algorithm plots(finds) points of an ellipse on the first quadrant by dividing the quadrant into two regions. Each point(x, y) is then projected into other three quadrants (-x, y), (x, -y), (-x, -y) i.e. it uses 4-way symmetry.

## Function of ellipse:

## *fellipse(x, y)=ry2x2+rx2y2-rx2ry2 fellipse(x, y)<0 then (x, y) is inside the ellipse. fellipse(x, y)>0 then (x, y) is outside the ellipse. fellipse(x, y)=0 then (x, y) is on the ellipse*

## Ellipse(4 way symmetry)

1. Input rx, ry and ellipse center ***(***xc, yc***)***

## Assume ellipse to be centered at origin and obtain the first point on it as: (x0, y0) = (0, ry)

1. Obtain the initial decision parameter for region 1 as: p10 = ry2 + 1/4rx2 - rx 2ry
2. For every xk position in region 1, starting at k=0, perform the following tests :

If p1k<0 then the next point along the ellipse is (xk+1, yk) and p1k+1 = p1k + 2ry2xk+1 + ry2

Else, the next point is (xk + 1, yk -1 ) and p1k+1 = p1k + 2ry2xk+1 – 2rx2yk+1 + ry2

1. Obtain the initial value in region 2 using the last point (x0, y0) of region 1 as: p20=ry2(x0+1/2)2+rx2 (y0-1)2-rx2ry2
2. At each yk in region 2 starting at k =0 perform the following task.

If p2k<0 the next point is (xk, yk+1) and p2k+1=p2k-2rx2yk+1+rx2

Else, the next point is (xk+1, yk -1) and p2k+1=p2k+2ry2xk+1 -2rx2yk+1+rx2

1. Now obtain the symmetric points in the three quadrants and plot the coordinate value as:

x=x+ xc, y=y+ yc

1. Repeat the steps for region 1 until 2ry2x >= 2rx2y

## Steps of Scan- Line Polygon Filling Algorithm:

## This algorithm works by intersecting scanline with polygon edges and fills the polygon between pairs of intersections. The following steps depict how this algorithm works.

## Draw a polygon with given coordinates

## Intersect scanline with polygon edges

## Fill between pairs of intersections

## 

## Basic algorithm:

## For y = ymin to ymax

## 1) intersect scanline y with each edge

## 2) sort intersections by increasing x [p0,p1,p2,p3]

## 3) fill pairwise (p0 −> p1, p2−> p3, ....)

## Special handling to improve the performance:

## Make sure to only fill the interior pixels

## Intersection has an integer X coordinate

## Intersection is an edge end point