* Assignment No :- 3 *

Aim :- write a C++ program to draw a concave polygon and fill it with desired color using scan fill algorithm.

Theory :-

Concave polygon :-

A simple polygon is considered as a concave polygon if and only if at least one angle of the polygon (interior) is a reflex angle (between 180° and 360°).

Convex polygon :-

The interior angle of a convex polygon are strictly less than 180°.

Scan line Algorithm for polygon filling :-

1) Read n, the number of vertices of polygon.

2) Read x and y co-ordinates of all vertices in array x [n] and y [n].

3) find y min and y max.

4) Store the initial x value (x1) y value y1, and y2 for two end points and x. increment Δx from scane line to scan line for each edge in the array edges.

while doing this check that $y_1 > y_2$, if not interchange y_1 and y_2 and corresponding x_1 and x_2. So that for each edge, y_1 represent it maximum y-co-ordinates and y represents its minimum y co-ordinates.

5) Sort the row of Array, edges [n] [4] in decreasing order of y, desending of y_2 and asending order of x_2.

6) set $y = y_{max}$.

7) find the active edges and update active edge list :-

  if $(y > y_2$ and $y < y_1)$
    { edge is active }
  else
    { edge is not active }.

8) Compute the x intersept for all active edge for current y value [initially] x-intercept for successive y value x_1 and x intersect can be given as

  $x_{i+1} \leftarrow x_i + \Delta x$

where $\Delta x = \dfrac{1}{m}$ and $m = \dfrac{y_2 - y_1}{x_2 - x_1}$

  i.e slope of a line segment.

9) If x intersect is vertex i.e x-intersect = x and y = y₁ then apply vertex test to check whether to consider one intersect or two intersect store all x intersects in the x-intersect [] array.

10) Sort x-intersect [] array in the ascending order.

11) Extract pair of intersects from the sorted X-intersect [] array.

12) pass pair of x value to line drawing routine to draw corresponding line segments.

13) set y = y-1

14) Repeat steps J through 13 until y > y min.

15) stop.

In step7, we have checked for y ≤ y₁ and not simply y < y₁. Hence step 9. become rebundant following program takes care of that.