

Programming the polygon clipping:

The declarations:

- 1- We need to define three arrays to hold the list of polygon vertices, each array is of two dimension, for the X and Y values of each vertex.

ORG (n , 2) / n is the number of vertices in the polygon

This matrix to store the coordinate of the vertices

POLYON (25,2) / the vertices input list {maximum 25 vertex }

POLYOUT (25,2) / the vertices output list{maximum 25 vertex }

- 2- We need to define a number of variables :

Vertices-no : integer ; counter of the input
vertices {same as n above}

Outlin : integer ; counter of the output vertices

$\left. \begin{array}{l} X_{min}, X_{max} \\ Y_{min}, Y_{max} \end{array} \right\}$ Integer ; the boundary of the window

V1 (1,2) : array for the first vertex

V2 (1,2) : array for the second vertex

Edge : string ; for the edge code

3- We need two functions

V3=Intersection (V1 , V2 , Edge) : used to determine the
 intersection point of the line
 between two vertices and the
 clip edge

Boolean=Inside (V , Edge) : return true if the vertex V is inside
 the edge , or return false if V is
 outside the edge

3- the algorithm

1- Store the values of the vertices coordinate in the ORG array

2- Transfer the values in ORG to POLYIN { the initial set of vertices }

For I=1 to Vertices-no

POLYIN (I,1) = ORG (I,1) { X values }

POLYIN (I,2) = ORG (I,2) { Y values }

Next

3- Edge = "L" { start from the left edge of the window }

```

For EG=1 to 4 { we have four edges in the window }
  Outlin = 0 { at first there are no output vertices }
  Take the first vertex from POLYIN and put it in V1
  For J=1 to Vertices-no
    Take the second vertex from POLYIN and put it in V2

    If Inside (V2,Edge) then { case 1 or case 4 }
      If Inside (V1,Edge) then { case 1 }
        Outlin=Outlin +1
        POLYOUT (Outlin)=V2      ] output
      Else
        V3= calculate the Intersection (V1,V2,Edge)
        Outlin=Outlin +1
        POLYOUT (Outlin)=V3      ] output

        Outlin=Outlin +1
        POLYOUT (Outlin)=V2      ] output
      End IF

    Else { case 2 or case 3 }

      If Inside (V1, Edge) Then { case 2 }
        V3=Intersection (V1,V2,Edge)

        Outlin=Outlin +1
        POLYOUT (Outlin)=V3      ] output
      End if
    End IF

    V1=V2
  Next J

  - Set Edge to the next Edge
  - Transfer the values in POLYOUT to POLYIN
  - Vertices-no=Outlin

Next EG

```

Example: Clip the polygon against the window

