Computer Graphics Lab Experiment No: - 10
Aim: Implement Sutherland Hodgeman polygon clipping method in C

Theory:

A convex polygon and a convex clipping area are given. The task is to clip polygon edges using the Sutherland–Hodgman Algorithm. Input is in the form of vertices of the polygon in clockwise order. Examples:

Input : Polygon : (100,150), (200,250), (300,200)

Clipping Area : (150,150), (150,200), (200,200),

(200,150) i.e. a Square

Output : (150, 162) (150, 200) (200, 200) (200, 174)

Example 2

Input : Polygon : (100,150), (200,250), (300,200)

Clipping Area : (100,300), (300,300), (200,100)

Output : (242, 185) (166, 166) (150, 200) (200, 250) (260, 220)

Overview of the algorithm:

Consider each edge e of clipping Area and do following:

a) Clip given polygon against e.

How to clip against an edge of clipping area? The edge (of clipping area) is extended infinitely to create a boundary and all the vertices are clipped using this boundary. The new list of vertices generated is passed to the next edge of the clip polygon in clockwise fashion until all the edges have been used.

There are four possible cases for any given edge of given polygon against current clipping edge e.

Both vertices are inside : Only the second vertex is added to the output list

First vertex is outside while second one is inside : Both the point of intersection of the edge with the clip boundary and the second vertex are added to the output list

First vertex is inside while second one is outside : Only the point of intersection of the edge with the clip boundary is added to the output list

Both vertices are outside : No vertices are added to the output list

There are two sub-problems that need to be discussed before implementing the algorithm:- To decide if a point is inside or outside the clipper polygon If the vertices of the clipper polygon are given in clockwise order then all the points lying on the right side of the clipper edges are inside that polygon. This can be calculated using :

Given that the line starts from (x_1, y_1) and ends at (x_2, y_2)

$$P = (x_2 - x_1)(y - y_1) - (y_2 - y_1)(x - x_1)$$

if $P<0$, the point is on the right side of the line

$P=0$, the point is on the line

$P>0$, the point is on the left side of the line

To find the point of intersection of an edge with the clip boundary If two points of each line(1,2 & 3,4) are known, then their point of intersection can be calculated using the formula :-

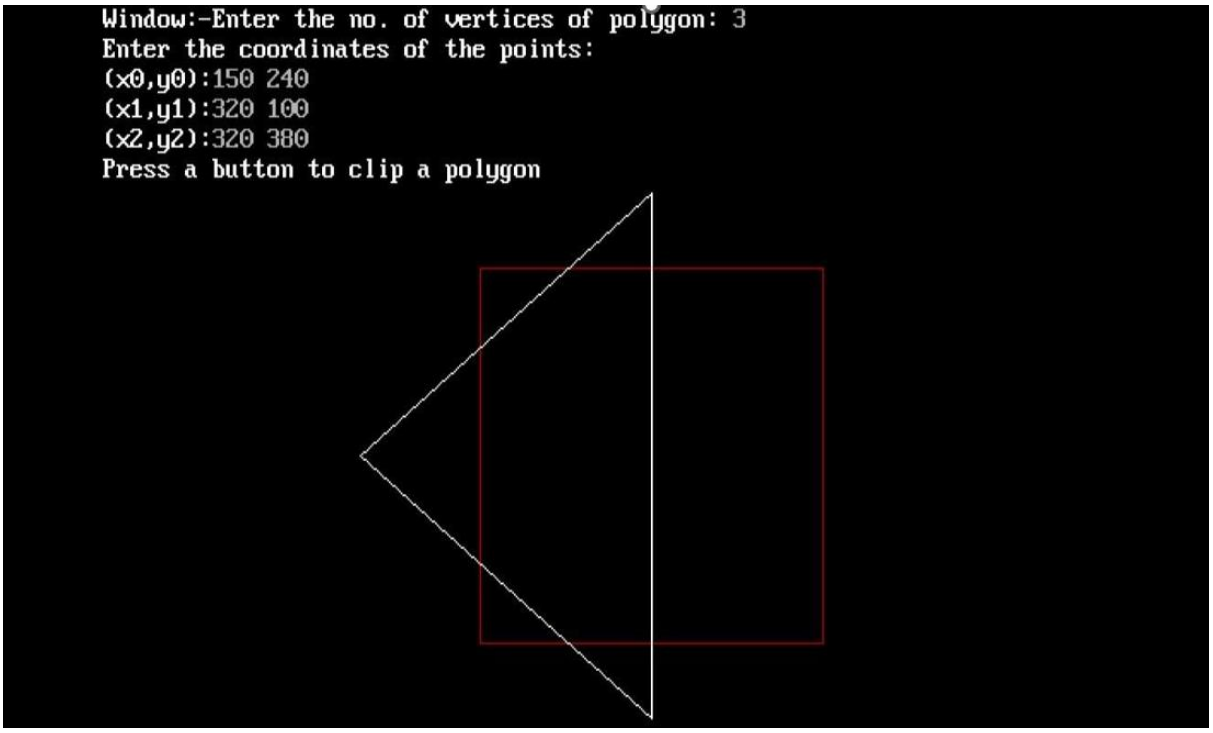
CODE:

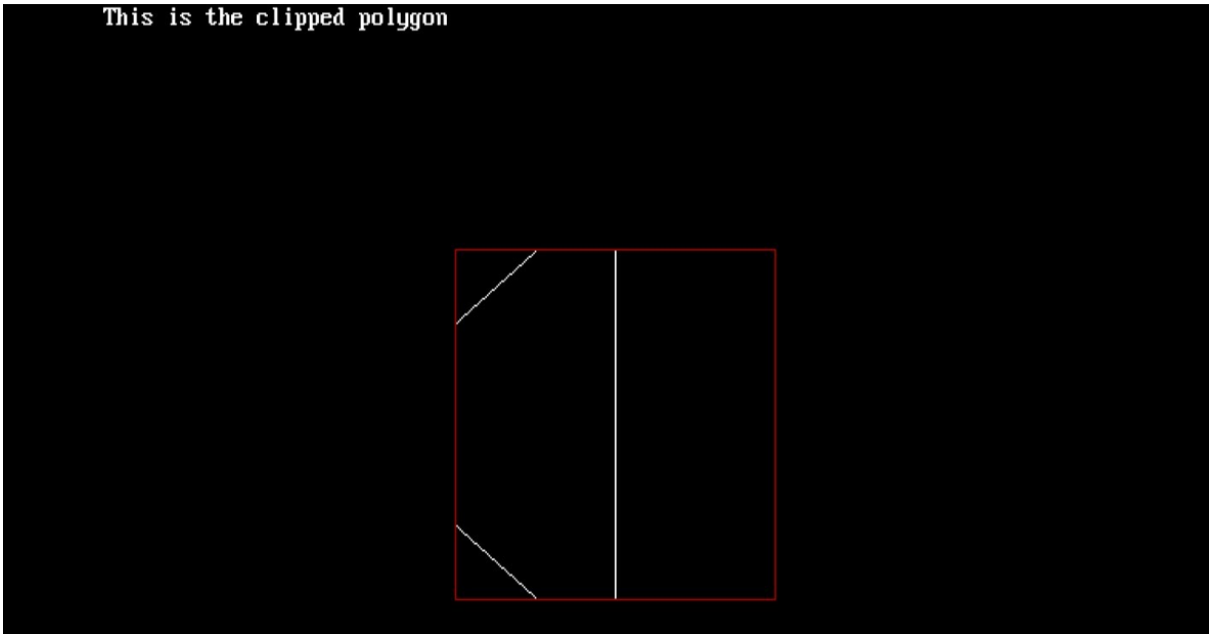
```
 #include <stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<graphies.h>

int main()
ins gd, gm, n, "x,, k = 0;
int wx1 = 220,wyl = 140, wx2 420, wy2 = 140, wx3 = 420, wx4 = 220,wy4 540;
in w() = (220, 140,420, 140,420,340,220,340, 220, 14017
detectgraph tagd, gm);
initgraph (@gd, igm, "C:\\TURBOC3\\BGT") #
printf("Window:-");
setcolor (RED);
drawpoly(3,w);
printf("Enter the no. of vertices of polygont
scanf("%d", in);
x = ma_locin 2+1) 7
printf("Enter the coordinates of the points: \n");
k = 0;
for(i=0;i<n*2;1+=2}
printf("(xid, yld):",k,k);
scanf("%d%d", &x[i], 6x[i+1])7°
*In 21
x[n*2 - 1]
setcolor (WHITE);
drawpoly(n+1,x);
printf("Prens a buston to clip a polygon");
sepcolor (RED);
drawpoly (5,w)
setfilistyle (SOLID_FILL, BLACK) #
flood!!11 (2,2, RED);
gotoxy (1,1);

printf("This is the clipped polygon");
cleardevice();
closegraph ();
return 0;
}
```

OUTPUT:





Conclusion: In conclusion, the Sutherland-Hodgman polygon clipping method is a powerful algorithm used in computer graphics to efficiently clip polygons against a window or another polygon. It helps in the removal of portions of a polygon that lie outside a specified boundary, which is a fundamental operation in rendering and image processing.