Ex. No. : 4 Date:

Register No.: 231701045 Name: P. Sahaana

.

Polygon Clipping using Sutherland–Hodgman Algorithm AIM:

To write a program that clips a polygon to a specified rectangular window using the Sutherland–Hodgman Polygon Clipping Algorithm and displays the clipped polygon.

Procedure:

1. Input:
   - Vertices of the polygon.
   - Clipping window boundaries (left, right, top, bottom).
2. Clip the polygon edges one by one against each window edge.
3. For each clipping edge, retain only the portion of the polygon that lies inside.
4. Display the original and the clipped polygon.

Program:

```
import matplotlib.pyplot as plt


EDGE_LEFT, EDGE_RIGHT, EDGE_BOTTOM, EDGE_TOP = 0, 1, 2, 3


def is_inside(pt, edge, win):
    x, y = pt
    xmin, xmax, ymin, ymax = win
    if edge == EDGE_LEFT:
        return x >= xmin
    elif edge == EDGE_RIGHT:
        return x <= xmax
```

```
    elif edge == EDGE_BOTTOM:
        return y >= ymin
    elif edge == EDGE_TOP:
        return y <= ymax

def intersection(a, b, edge, win):
    xmin, xmax, ymin, ymax = win
    x1, y1 = a
    x2, y2 = b
    if edge == EDGE_LEFT:
        x = xmin
        y = y1 + (y2 - y1) * (xmin - x1) / (x2 - x1)
    elif edge == EDGE_RIGHT:
        x = xmax
        y = y1 + (y2 - y1) * (xmax - x1) / (x2 - x1)
    elif edge == EDGE_BOTTOM:
        y = ymin
        x = x1 + (x2 - x1) * (ymin - y1) / (y2 - y1)
    elif edge == EDGE_TOP:
        y = ymax
        x = x1 + (x2 - x1) * (ymax - y1) / (y2 - y1)
    return (x, y)

def clip(poly, win):
    out = poly[:]
```

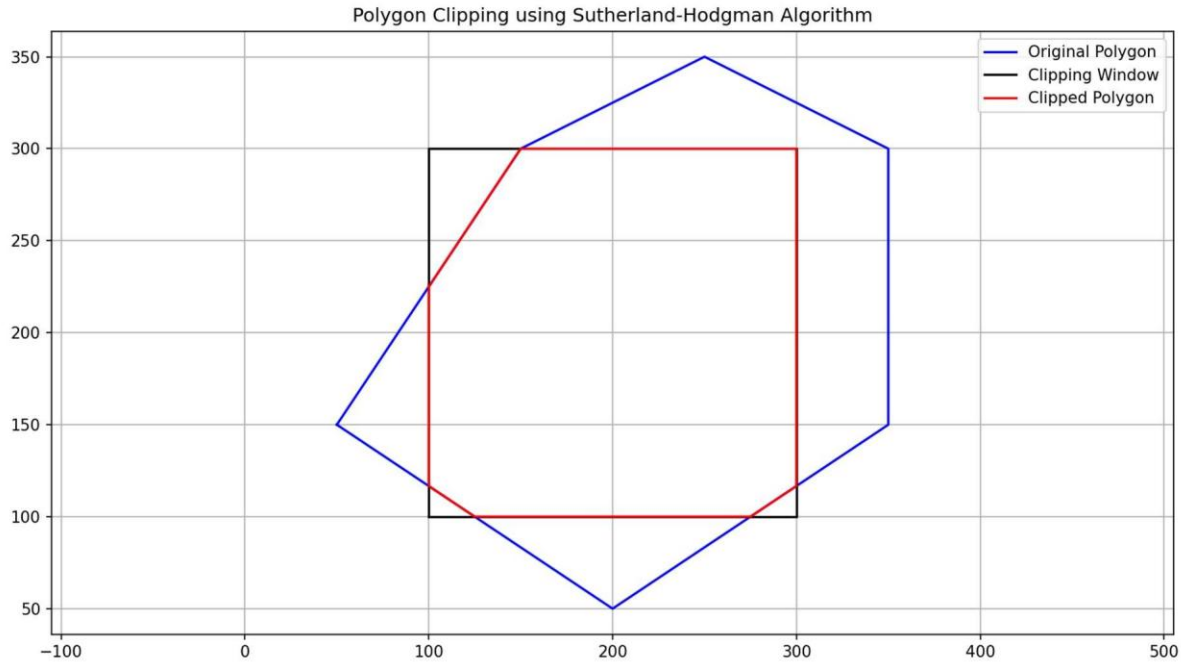
```
    for edge in [EDGE_LEFT, EDGE_RIGHT, EDGE_BOTTOM, EDGE_TOP]:
        inp = out
        out = []
        if not inp:
            break
        s = inp[-1]
        for p in inp:
            if is_inside(p, edge, win):
                if not is_inside(s, edge, win):
                    out.append(intersection(s, p, edge, win))
                out.append(p)
            elif is_inside(s, edge, win):
                out.append(intersection(s, p, edge, win))
            s = p
    return out


def draw(points, color, label):
    x, y = zip(*(points + [points[0]]))
    plt.plot(x, y, color=color, label=label)


clip_win = (100, 300, 100, 300)
poly = [(50, 150), (200, 50), (350, 150), (350, 300), (250, 350), (150, 300)]


res = clip(poly, clip_win)
```

```
plt.figure(figsize=(8, 8))
draw(poly, 'blue', "Original Polygon")
box = [
    (clip_win[0], clip_win[2]),
    (clip_win[1], clip_win[2]),
    (clip_win[1], clip_win[3]),
    (clip_win[0], clip_win[3])
]
draw(box, 'black', "Clipping Window")
draw(res, 'red', "Clipped Polygon")
plt.legend()
plt.title("Sutherland Hodgman Polygon Clipping")
plt.axis("equal")
plt.grid(True)
plt.show()
```



Result:

The polygon was successfully clipped using the Sutherland–Hodgman algorithm against a rectangular clipping window.