Ex. No. : 4 **Date:**

Register No.: **Name:**

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
LEFT, RIGHT, BOTTOM, TOP = 0, 1, 2, 3
def inside(p, edge, clip_win):
    x, y = p
    xmin, xmax, ymin, ymax = clip_win
    if edge == LEFT:
        return x >= xmin
    elif edge == RIGHT:
        return x <= xmax
    elif edge == BOTTOM:
        return y >= ymin
    elif edge == TOP:
```

```
        return y <= ymax

    def intersect(p1, p2, edge, clip_win):
        xmin, xmax, ymin, ymax = clip_win
        x1, y1 = p1
        x2, y2 = p2
        if edge == LEFT:
            x = xmin
            y = y1 + (y2 - y1) * (xmin - x1) / (x2 - x1)
        elif edge == RIGHT:
            x = xmax
            y = y1 + (y2 - y1) * (xmax - x1) / (x2 - x1)
        elif edge == BOTTOM:
            y = ymin
            x = x1 + (x2 - x1) * (ymin - y1) / (y2 - y1)
        elif edge == TOP:
            y = ymax
            x = x1 + (x2 - x1) * (ymax - y1) / (y2 - y1)
        return (x, y)

    def clip_polygon(polygon, clip_win):
        output = polygon
        for edge in [LEFT, RIGHT, BOTTOM, TOP]:
            input_list = output
            output = []
            if not input_list:
                break
            s = input_list[-1]
            for p in input_list:
                if inside(p, edge, clip_win):
                    if not inside(s, edge, clip_win):
```

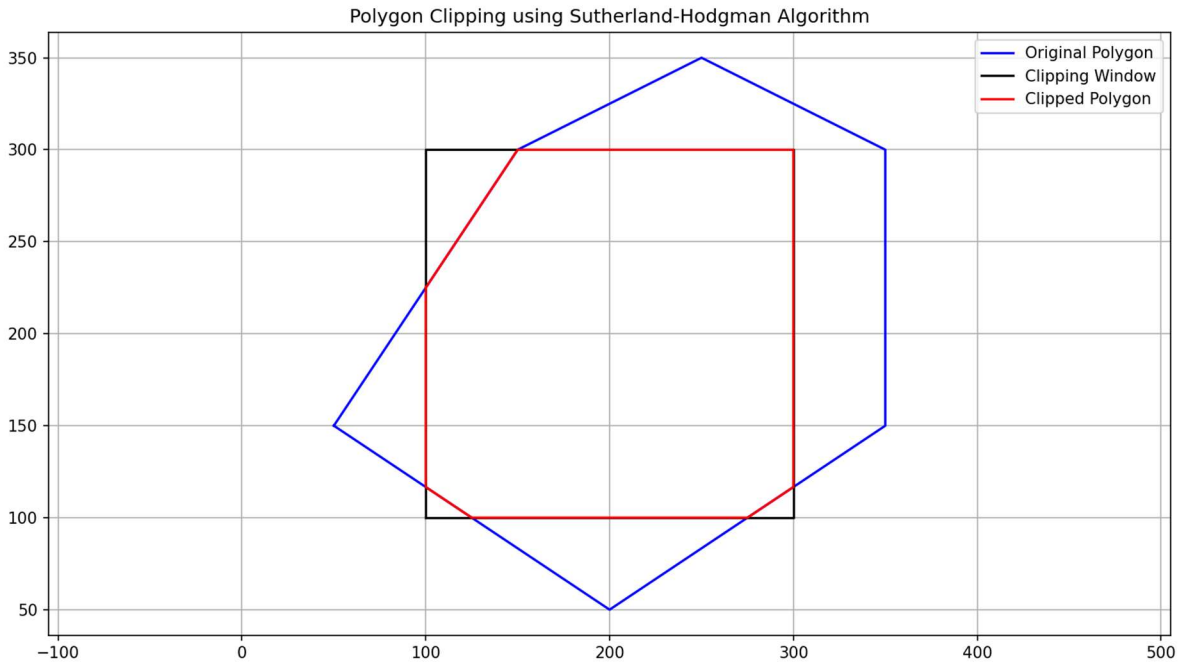
```
                output.append(intersect(s, p, edge, clip_win))
            output.append(p)
        elif inside(s, edge, clip_win):
            output.append(intersect(s, p, edge, clip_win))
        s = p
    return output

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

# Main
clip_window = (100, 300, 100, 300)  # xmin, xmax, ymin, ymax
polygon = [(50, 150), (200, 50), (350, 150), (350, 300), (250, 350), (150, 300)]

clipped_poly = clip_polygon(polygon, clip_window)

plt.figure(figsize=(8, 8))
draw_polygon(polygon, 'blue', "Original Polygon")
draw_polygon([(clip_window[0], clip_window[2]), (clip_window[1], clip_window[2]),
              (clip_window[1], clip_window[3]), (clip_window[0], clip_window[3])],
             'black', "Clipping Window")
draw_polygon(clipped_poly, 'red', "Clipped Polygon")
plt.legend()
plt.title("Polygon Clipping using Sutherland-Hodgman Algorithm")
plt.grid(True)
plt.axis("equal")
plt.show()
```



Result:

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