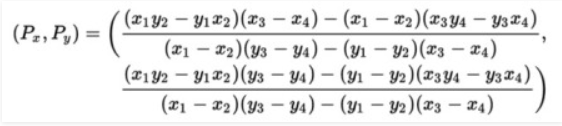
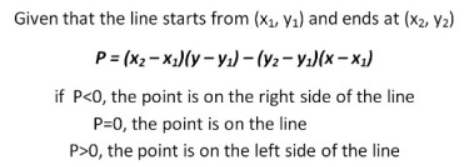
**Assignment 2 : Polygon clipping**

สืบเนื่องมาจากการเรียนบทที่แล้วเรื่อง Line clipping คือโดยการหาว่าเส้นใด อยู่ภายใน window และเส้นใดไม่อยู่ใน window หรือเส้นใดเป็น Candidate clipping โดยใช้ Cohen-Sutherland Algorithm ซึ่งในบทนี้เราศึกษาการ clipping รูปร่างต่าง ๆ ที่ประกอบไปด้วยเส้นแบบต่าง ๆ ที่ประกอบกันหลายเส้นเป็นรูปร่าง ซึ่งจุดต้นและจุดปลายจะต้องเชื่อมติดกัน หรือเราเรียกว่า Polygon ให้อยู่ภายใน Window ที่กำหนด โดยใช้ Cohen-Sutherland Algorithm

ซึ่งในครั้งนี้การคำนวณจุดตัด edge เราจะใช้สูตร



และเช็คว่าจุดอยู่ใน polygon หรือไม่ ด้วยสูตร



ซึ่งในครั้งนี้เราจะนิยามว่า

ถ้า P > 0 จะเป็นจุดที่อยู่ด้านซ้าย

ถ้า P < 0 จะเป็นจุดที่อยู่ด้านขวา

ถ้า P = 0 จะเป็นจุดที่อยู่บนเส้น ซึ่งเราจะนับว่าเป็นจุดที่อยู่ด้านซ้าย หรือเป็นจุดที่มองเห็นได้

โดยขั้นตอนการทำงาน คือ

1. ทำการวน loop เพื่อที่จะหา Edge ของ window
2. และหลังจากนั้นเก็บค่า list ไว้
3. กำหนดจุดเลือกตำแหน่งในการกรอบ
4. วน loop ตามส่วนของเส้นตรงเพื่อ get ตำแหน่งของแต่ละจุด แล้วนำมาเข้าเงื่อนไข
   1. ถ้าจุด P(x, y) > 0 แล้ว
5. P(x-1,y-1) < 0 ให้ทำการคำนวณหาจุดตัดส่วนของเส้นตรงและ edge
6. P(x-1, y-1) > 0 แสดงว่าอยู่ภายใน window จึง return current point
   1. ถ้าจุด P(x,y) < 0 แสดงว่าจุด P(x,y) อยู่ภายนอก window จึงต้องคำนวณเพื่อหาจุดตัดส่วนของเส้นตรงและ edge
   2. ถ้า P(x,y) = 0 แสดงว่าจุดอยู่บนกรอบของ window ให้ return จุดที่อยู่บนกรอบของ window
7. เมื่อทำเสร็จทำการขยับ edge ไป edge ถัดไป

\*\* Program ที่ใช้ run จะถูกแนบมาในชื่อไฟล์ Assignment2\_polygon\_7233\*\*

**Result**

*ตัวอย่างที่ 1*

Chart, line chart

Description automatically generated

Expected value :

Input : [(0,6),(0,0),(10,0),(10,6),(5,11)]

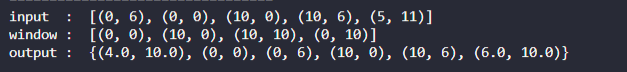
Output : [[4.0, 10.0], [-0.0, 6.0], [0.0, -0.0], [10.0, -0.0], [10.0, 6.0], [6.0, 10.0]]

Result value :

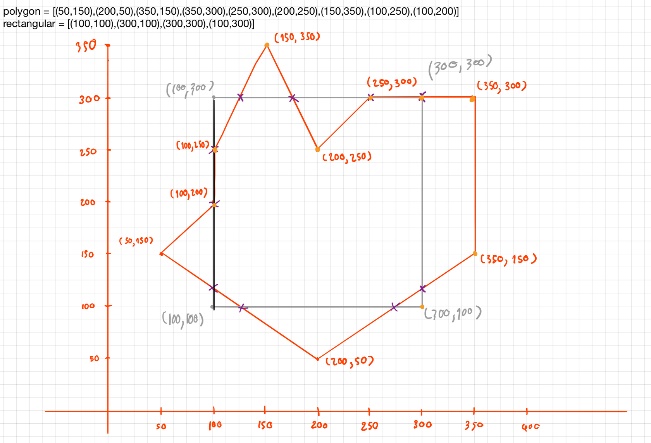
Input : [(0,6),(0,0),(10,0),(10,6),(5,11)]

Output : {(4.0, 10.0), (0, 0), (0, 6), (10, 0), (10, 6), (6.0, 10.0)}

ผลที่ได้จริง :



*ตัวอย่างที่ 2*



Expected value :

Input: [(50,150),(200,50),(350,150),(350,300),(250,300),(200,250),(150,350),(100,250),(100,200)]

Window = [(100,100),(300,100),(300,300),(100,300)]

Output : [[100.0, 116.66666666666667], [125.00000000000001, 100.0], [275.0, 100.0], [300.0, 116.66666666666667], [300.0, 299.99999999999994], [250.0, 300.0], (200, 250), [175.0, 300.0], [125.0, 300.0], [100.0, 250.0]]

Result value :

Input : [(50,150),(200,50),(350,150),(350,300),(250,300),(200,250),(150,350),(100,250),(100,200)]

Output : {(100, 200), (125.0, 300.0), (125.0, 100.0), (175.0, 300.0), (275.0, 100.0), (300.0, 116.66666666666667), (100.0, 116.66666666666667), (100, 250), (350, 300), (250, 300), (300.0, 300.0)}

ผลที่ได้จริง :

