



---

# Midterm Report

---

for Python programming



班級：電機二乙

學號：B1141065

姓名：陳昱安

## Problem 1 :

請輸入  $A(x_1, y_1)$  ,  $B(x_2, y_2)$  ,  $C(x_3, y_3)$  ,  $D(x_4, y_4)$  四個直角座標點，並判斷  $\overline{AB}$  與  $\overline{CD}$  是否有交點，若有，則輸出交點。

【解】：

### <法一>：二元一次方程式法

```
def find_intersection_point(A, B, C, D):  
    x1, y1 = A  
    x2, y2 = B  
    x3, y3 = C  
    x4, y4 = D  
  
    # 求解直線方程式  $y = mx + b$  的  $m$  和  $b$   
    if x1 == x2: # 如果 AB 線段為垂直線  
        m1 = None  
        b1 = x1  
    else:  
        m1 = (y2 - y1) / (x2 - x1)  
        b1 = y1 - m1 * x1  
        print("AB的直線方程式為:  $y =$ ", m1, " $* x +$ ", b1)  
  
    if x3 == x4: # 如果 CD 線段為垂直線  
        m2 = None  
        b2 = x3  
    else:  
        m2 = (y4 - y3) / (x4 - x3)  
        b2 = y3 - m2 * x3  
        print("CD的直線方程式為:  $y =$ ", m2, " $* x +$ ", b2)  
  
    # 判斷特殊情況  
    if m1 is None and m2 is None: # 兩條線段都是垂直線  
        if x1 != x3: # 如果兩條垂直線的  $x$  值不相等，則平行且沒有交點  
            return None  
        else: # 如果兩條垂直線的  $x$  值相等，則有無限個交點  
            return "兩線段有無限個交點"  
    elif m1 is None: # AB線段是垂直線  
        x = b1  
        y = m2 * x + b2
```

(圖一)模擬直線交點測定系統程式示意圖 (1) (from colab)

```

elif m2 is None: # CD線段是垂直線
    x = b2
    y = m1 * x + b1
elif m1 == m2: # 兩條線段平行
    return None
else:
    x = (b2 - b1) / (m1 - m2)
    y = m1 * x + b1

return x, y

# 讀取四個點的座標
x1, y1 = map(int, input("請輸入 A 的座標 (x1, y1): ").split(','))
x2, y2 = map(int, input("請輸入 B 的座標 (x2, y2): ").split(','))
x3, y3 = map(int, input("請輸入 C 的座標 (x3, y3): ").split(','))
x4, y4 = map(int, input("請輸入 D 的座標 (x4, y4): ").split(','))

A = (x1, y1)
B = (x2, y2)
C = (x3, y3)
D = (x4, y4)

# 判斷是否有交點，並輸出交點
intersection = find_intersection_point(A, B, C, D)
if intersection:
    print("AB與CD有交點，交點座標為:", intersection)
else:
    print("AB與CD無交點")

```

(圖二) 模擬直線交點測定系統程式示意圖 (2) (from colab)

```

➡ 請輸入 A 的座標 (x1, y1): 1,2
   請輸入 B 的座標 (x2, y2): 2,8
   請輸入 C 的座標 (x3, y3): 3,6
   請輸入 D 的座標 (x4, y4): 4,5
   AB的直線方程式為: y = 6.0 * x + -4.0
   CD的直線方程式為: y = -1.0 * x + 9.0
   AB與CD有交點，交點座標為: (1.8571428571428572, 7.142857142857142)

```

(圖三) 模擬直線交點測定系統程式示意圖 (3) (from colab)

## <程式解釋>

這段程式碼是用來找出兩條線段的交點的一個系統。以下是程式碼的運作理念和相關原理：

(1) `def find_intersection_point(A, B, C, D):` :

定義了一個名為 `find_intersection_point` 的函式，這個函式接收四個參數 `A`、`B`、`C`、`D`，分別代表兩條線段的端點。

(2) 分別從 `A`、`B`、`C`、`D` 四個端點取出座標值。

(3) 求解兩條線段的方程式：

1. 如果某條線段為垂直線，則直接計算出 `x` 的值。

2. 如果不是垂直線，則計算出該線段的斜率和截距，並印出直線方程式。

(4) 判斷特殊情況：

1. 如果兩條線段都是垂直線且 `x` 值不相等，則兩線平行且無交點。

2. 如果兩條線段都是垂直線且 `x` 值相等，則有無限個交點。

3. 如果其中一條線段是垂直線，則根據另一條線段的斜率和截距計算出交點座標。

4. 如果兩條線段平行，則無交點。

5. 否則計算出兩條線段的交點座標。

(5) 返回計算出的交點座標。

(6) 讀取四個點的座標。

(7) 呼叫 `find_intersection_point` 函式，並根據返回值判斷是否有交點，並印出相應的訊息和交點座標。



## <法二>：向量叉積法

```
def get_intersection(A, B, C, D):  
    def ccw(A, B, C):  
        return (C[1] - A[1]) * (B[0] - A[0]) > (B[1] - A[1]) * (C[0] - A[0])  
  
    return ccw(A, C, D) != ccw(B, C, D) and ccw(A, B, C) != ccw(A, B, D)  
  
# 輸入四個點的坐標  
A = tuple(map(int, input("請輸入點A的坐標 (x1,y1) (請用逗號分隔): ").split(',')))  
B = tuple(map(int, input("請輸入點B的坐標 (x2,y2) (請用逗號分隔): ").split(',')))  
C = tuple(map(int, input("請輸入點C的坐標 (x3,y3) (請用逗號分隔): ").split(',')))  
D = tuple(map(int, input("請輸入點D的坐標 (x4,y4) (請用逗號分隔): ").split(',')))  
  
# 檢查兩線段是否有交點  
if get_intersection(A, B, C, D):  
    print("線段AB與線段CD有交點。")  
else:  
    print("線段AB與線段CD沒有交點。")
```

請輸入點A的坐標 (x1,y1) (請用逗號分隔): 3,7  
請輸入點B的坐標 (x2,y2) (請用逗號分隔): 2,5  
請輸入點C的坐標 (x3,y3) (請用逗號分隔): 2,8  
請輸入點D的坐標 (x4,y4) (請用逗號分隔): 4,5  
線段AB與線段CD有交點。

(圖四) 模擬直線交點測定系統程式示意圖 (4) (from colab)

### <程式解釋>

這段程式碼也是用來檢查兩條線段是否有交點的函式。以下呈現的是其程式的運作方式及原理：

(1) `def get_intersection(A, B, C, D):` :

這是一個函式定義，它接收四個參數 A、B、C、D，這四個參數分別代表兩條線段的端點。

(2) `def ccw(A, B, C):` :

在 `get_intersection` 函式內部定義了另一個名為 `ccw` 的函式，用

來判斷三個點 A、B、C 是否逆時針方向排列。

(3) return ccw(A, C, D) != ccw(B, C, D) and ccw(A, B, C) != ccw(A, B, D) :

get\_intersection 函式返回兩條線段是否有交點的布林值，利用了 ccw 函式來判斷線段的排列情況。

(4) 輸入四個點的坐標：使用者被要求輸入四個點 A、B、C、D 的坐標。

(5) 呼叫 get\_intersection 函式：將使用者輸入的座標作為參數傳遞給函式，並根據返回值判斷兩條線段是否有交點。

(6) 判斷是否有交點：

1. 如果有交點，則印出相應的訊息。
2. 如果沒有交點，則印出相應的訊息。