

## Sudoku

คงเคยเห็นหรือเล่นเกม **Sudoku** กันมาบ้าง ตัวอย่างเช่น จากรูปด้านซ้าย เราต้องเติมตัวเลขให้ครบทุกช่อง โดยไม่มีแถวแนวนอนใดมีเลขซ้ำกัน ไม่มีแถวแนวตั้งใดมีเลขซ้ำกัน และไม่มีเลขซ้ำกันในกลุ่ม **3x3** เดียวกัน

5	3		7					
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

โจทย์ข้อนี้ให้เขียนโปรแกรมเพื่อหาคำตอบของกระดานเริ่มต้นที่ได้รับ (ศึกษารายละเอียดของฟังก์ชัน และจาก **comment**)

ฟังก์ชัน **solve(board)** รับตาราง **Sudoku** ที่ให้มาในรูปของสตริงของตัวเลข **81** ตัว เรียงจากซ้ายไปขวา จากบนลงล่าง ตำแหน่งที่ว่างแทนด้วยจุด เช่นตารางทางซ้ายข้างบนนี้ แทนด้วยสตริง

```
'53..7....6..195....98....6.8...6...34..8.3..17...2...6.6....28....419..5....8..79'
```

```
def same_row(i,j):          # i และ j คือ index ในสตริง 81 ตัวที่แทนตาราง 9x9
    return (i//9 == j//9)
def same_col(i,j):
    return (i-j) % 9 == 0
def same_block(i,j):
    return (i//27 == j//27 and i%9//3 == j%9//3)
def show(board):
    for i in range(3):
        print('+-+--+---+---+---+')
        for j in range(3):
            k = 9*(3*i+j)
            print('|'+board[k:k+3]+'|'+board[k+3:k+6]+'|'+board[k+6:k+9]+'|')
        print('+-+--+---+---+---+')

def solve(board):
    # หาว่า board ยังมี จุด เหลืออยู่ไหม ถ้าไม่มี ก็คืน board กลับไป

    # แต่ถ้ายังมี จุด อยู่ใน board
    # สร้างเซต S ซึ่งเก็บตัวเลขที่อยู่ในแถวแนวนอนเดียวกับจุด ตัวเลขที่อยู่ในแถวแนวตั้งเดียวกับจุด และตัวเลขที่อยู่ในกลุ่มเดียวกับจุด
    # (ทำงาน ๆ ด้วยการลู่ทุกตัวใน board แล้วใช้ฟังก์ชัน same_row, same_col, same_block ให้เป็นประโยชน์)

    # ให้ T = เซตของเลข '1' ถึง '9' ลบด้วยเซต S (T ก็คือเซตที่เก็บเลขที่อาจใช้แทนจุดได้)

    for e in T:
        newboard = board ที่แทนจุดด้วย e
        sol = solve(newboard)      # จำนวนจุดลดลงหนึ่ง ลองลู่หาคำตอบต่อ
        if sol != '' : return sol  # ถ้าลองแล้วใช้ได้ ก็คืนผล
    return '' # ถ้าลองทุกแบบ แล้วไม่สำเร็จ ก็คืน '' บอกว่าไม่พบคำตอบ

sol = solve(input().strip())
show(sol)
```

## ข้อมูลนำเข้า

สตริงของเลข 0 ถึง 9 และจุด จำนวน 81 ตัว ที่แทนตารางเริ่มต้นของเกม Sudoku โดยจุดแทนช่องที่ยังไม่เติมเลข

## ข้อมูลส่งออก

ผลลัพธ์ที่ได้จากการเติมเลข (เพื่อความง่าย input ที่ใช้ในการตรวจจะหาคำตอบได้เสมอ และมีเพียงคำตอบเดียวแน่ ๆ)

## ตัวอย่าง

input

.2.....6...3.74.8.....3..2.8..4..1.6..5.....1.78.5....9.....4.

	2							
			6					3
	7	4		8				
					3			2
	8			4			1	
6			5					
				1		7	8	
5					9			
							4	

Output

```
+---+---+---+
|126|437|958|
|895|621|473|
|374|985|126|
+---+---+---+
|457|193|862|
|983|246|517|
|612|578|394|
+---+---+---+
|269|314|785|
|548|769|231|
|731|852|649|
+---+---+---+
```

1	2	6	4	3	7	9	5	8
8	9	5	6	2	1	4	7	3
3	7	4	9	8	5	1	2	6
4	5	7	1	9	3	8	6	2
9	8	3	2	4	6	5	1	7
6	1	2	5	7	8	3	9	4
2	6	9	3	1	4	7	8	5
5	4	8	7	6	9	2	3	1
7	3	1	8	5	2	6	4	9

คำเตือน : ตัวอย่างข้างบนนี้เป็นโจทย์ Sudoku ที่ไม่ง่าย ใช้เวลาหาคำตอบพอสมควร