board = [  
 [7,8,0,4,0,0,1,2,0],  
 [6,0,0,0,7,5,0,0,9],  
 [0,0,0,6,0,1,0,7,8],  
 [0,0,7,0,4,0,2,6,0],  
 [0,0,1,0,5,0,9,3,0],  
 [9,0,4,0,6,0,0,0,5],  
 [0,7,0,3,0,0,0,1,2],  
 [1,2,0,0,0,7,4,0,0],  
 [0,4,9,2,0,6,0,0,7]  
]  
  
  
def solve(bo):  
 find = find\_empty(bo)  
 if not find:  
 return True  
 else:  
 row, col = find  
  
 for i in range(1,10):  
 if valid(bo, i, (row, col)):  
 bo[row][col] = i  
  
 if solve(bo):  
 return True  
  
 bo[row][col] = 0  
  
 return False  
  
  
def valid(bo, num, pos):  
 *# Check row* for i in range(len(bo[0])):  
 if bo[pos[0]][i] == num and pos[1] != i:  
 return False  
  
 *# Check column* for i in range(len(bo)):  
 if bo[i][pos[1]] == num and pos[0] != i:  
 return False  
  
 *# Check box* box\_x = pos[1] // 3  
 box\_y = pos[0] // 3  
  
 for i in range(box\_y\*3, box\_y\*3 + 3):  
 for j in range(box\_x \* 3, box\_x\*3 + 3):  
 if bo[i][j] == num and (i,j) != pos:  
 return False  
  
 return True  
  
  
def print\_board(bo):  
 for i in range(len(bo)):  
 if i % 3 == 0 and i != 0:  
 print(**"- - - - - - - - - - - - - "**)  
  
 for j in range(len(bo[0])):  
 if j % 3 == 0 and j != 0:  
 print(**" | "**, end=**""**)  
  
 if j == 8:  
 print(bo[i][j])  
 else:  
 print(str(bo[i][j]) + **" "**, end=**""**)  
  
  
def find\_empty(bo):  
 for i in range(len(bo)):  
 for j in range(len(bo[0])):  
 if bo[i][j] == 0:  
 return (i, j) *# row, col* return None  
  
print\_board(board)  
solve(board)  
print(**"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"**)  
print\_board(board)