

N = 9

```
def printing(arr):
    for i in range(N):
        for j in range(N):
            print(arr[i][j], end = " ")
        print()

def isSafe(grid, row, col, num):
    for x in range(9):
        if grid[row][x] == num:
            return False
    for x in range(9):
        if grid[x][col] == num:
            return False
    startRow = row - row % 3
    startCol = col - col % 3
    for i in range(3):
        for j in range(3):
            if grid[i + startRow][j + startCol] == num:
                return False
    return True

def solveSudoku(grid, row, col):
    if (row == N - 1 and col == N):
        return True
    if col == N:
        row += 1
        col = 0
    if grid[row][col] > 0:
        return solveSudoku(grid, row, col + 1)
    for num in range(1, N + 1, 1):
        if isSafe(grid, row, col, num):
            grid[row][col] = num
            if solveSudoku(grid, row, col + 1):
                return True
            grid[row][col] = 0
    return False

grid = [[3, 0, 6, 5, 0, 8, 4, 0, 0],
        [5, 2, 0, 0, 0, 0, 0, 0, 0],
        [0, 8, 7, 0, 0, 0, 0, 3, 1],
        [0, 0, 3, 0, 1, 0, 0, 8, 0],
        [9, 0, 0, 8, 6, 3, 0, 0, 5],
        [0, 5, 0, 0, 9, 0, 6, 0, 0],
        [1, 3, 0, 0, 0, 0, 2, 5, 0],
        [0, 0, 0, 0, 0, 0, 0, 7, 4],
        [0, 0, 5, 2, 0, 6, 3, 0, 0]]
```



```
if (solveSudoku(grid, 0, 0)):  
    printing(grid)  
else:  
    print("no solution exists ")
```

Python 3.11.4 (tags/v3.11.4:d2340ef, Ju
Type "help", "copyright", "credits" or
>>>

```
=====
```

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

>>>