

```
In [17]: def solve_n_queens():
n=4
def is_safe(board, row, col):
    for i in range(row):
        if board[i]== col or \
            board[i]-i==col- row or \
            board[i]+i== col+ row:
            return False
    return True
def solve(board, row):
    if row== n:
        result.append(board[:])
        return
    for col in range(n):
        if is_safe(board, row, col):
            board[row]=col
            solve(board,row + 1)
            board[row]=-1
    result=[]
    solve([-1]*n, 0)
    return result
solutions=solve_n_queens()
print(f"Number of solutions for 8-Queens:{len(solutions)}")
for solution in solutions:
    print(solution)
```

Number of solutions for 8-Queens: 2

[1, 3, 0, 2]

[2, 0, 3, 1]

In []: