```
def is_safe(board, row, col, n):
    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_queens(board, row, n, solutions):
    if row == n:
        solutions.append(board[:])
        return


    for col in range(n):
        if is_safe(board, row, col, n):
            board[row] = col
            solve_queens(board, row + 1, n, solutions)
            board[row] = -1
def print_solution(board, n):
    for row in board:
        line = ['Q' if i == row else '.' for i in range(n)]
        print(" ".join(line))
    print("\n")


def eight_queens(n=8):
    solutions = []
    board = [-1] * n
    solve_queens(board, 0, n, solutions)


    print(f"Total solutions: {len(solutions)}\n")
    for idx, solution in enumerate(solutions, 1):
        print(f"Solution {idx}:")
        print_solution(solution, n)


eight_queens()
```