def is\_safe(board, row, col):

    for i in range(row):

        if board[i][col] == 'Q':

            return False

        if col - (row - i) >= 0 and board[i][col - (row - i)] == 'Q':

            return False

        if col + (row - i) < len(board) and board[i][col + (row - i)] == 'Q':

            return False

    return True

def print\_board(board):

    for row in board:

        print(' '.join(row))

    print()

def solve\_n\_queens(board, row):

    if row == len(board):

        print\_board(board)

        return

    for col in range(len(board)):

        if is\_safe(board, row, col):

            board[row][col] = 'Q'

            solve\_n\_queens(board, row + 1)

            board[row][col] = '-'  # backtrack

# --- Main Program ---

N = 4

board = [['-' for \_ in range(N)] for \_ in range(N)]

solve\_n\_queens(board, 0)

- Q - -

- - - Q

Q - - -

- - Q -

- - Q -

Q - - -

- - - Q

- Q - -