N = 8

def print\_solution(board):

for row in board:

print(" ".join("Q" if col else "." for col in row))

print()

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

# Check column

for i in range(row):

if board[i][col]:

return False

# Check upper left diagonal

i, j = row, col

while i >= 0 and j >= 0:

if board[i][j]:

return False

i -= 1

j -= 1

# Check upper right diagonal

i, j = row, col

while i >= 0 and j < N:

if board[i][j]:

return False

i -= 1

j += 1

return True

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

if row == N:

print\_solution(board)

return

for col in range(N):

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

board[row][col] = 1

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

board[row][col] = 0 # Backtrack

def solve():

board = [[0 for \_ in range(N)] for \_ in range(N)]

solve\_n\_queens(board, 0)

# Run the program

solve()

OUTPUT:

