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

for i in range(col):

if board[row][i] == 1:

return False

for i, j in zip(range(row, -1, -1), range(col, -1, -1)):

if board[i][j] == 1:

return False

for i, j in zip(range(row, len(board), 1), range(col, -1, -1)):

if board[i][j] == 1:

return False

return True

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

if col >= len(board):

return True

for i in range(len(board)):

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

board[i][col] = 1

if solve\_n\_queens(board, col + 1):

return True

board[i][col] = 0

return False

def print\_board(board):

for row in board:

print(" ".join(str(x) for x in row))

print()

def solve\_8\_queens():

N = 8

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

if solve\_n\_queens(board, 0):

print("Solution:")

print\_board(board)

else:

print("No solution exists")

solve\_8\_queens()

Output:

