def solve\_8\_queens(board, row):

if row == 8:

return True # All queens are placed successfully

for col in range(8):

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

board[row] = col # Place queen in the current row and column

if solve\_8\_queens(board, row + 1):

return True # Solution found

board[row] = -1 # Backtrack: Remove queen from the current position

return False # No safe position found for the current row

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

for i in range(row):

# Check if another queen exists in the same column or diagonals

if board[i] == col or board[i] - i == col - row or board[i] + i == col + row:

return False

return True

def print\_solution(board):

for i in range(8):

for j in range(8):

if board[i] == j:

print("Q", end=" ") # Print queen symbol

else:

print(".", end=" ") # Print empty space

print()

def eight\_queens():

board = [-1, -1, -1, -1, -1, -1, -1, -1] # Initialize the chessboard

if solve\_8\_queens(board, 0):

print\_solution(board) # Print the solution

else:

print("No solution exists")

# Call the function to solve the 8 queens problem

eight\_queens()