2. Write the python program to solve 8-Queen problem and algorithm.

**AIM :** program to solve 8-Queen problem and algorithm

**ALGORITHM :**

1. Start from the leftmost column.
2. If all queens are placed, return true.
3. Try all rows in the current column. For each row, check if placing a queen in that row is safe. If safe, mark the cell and proceed recursively for the next column.
4. If placing the queen in the current row leads to a solution, return true.
5. If no row in the current column is safe, backtrack by unmarking the cell and return false.

**PROGRAM :**

N = 8

def print\_board(board):

for row in board:

print(" ".join(row))

print()

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

for i in range(row):

if board[i][col] == 'Q':

return False

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

if board[i][j] == 'Q':

return False

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

if board[i][j] == 'Q':

return False

return True

def solve\_queens(board, col):

if col >= N:

return True

for i in range(N):

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

board[i][col] = 'Q'

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

return True

board[i][col] = '.' # Backtrack if no solution is found

return False

chessboard = [['.' for \_ in range(N)] for \_ in range(N)]

if solve\_queens(chessboard, 0):

print("Solution:")

print\_board(chessboard)

else:

print("No solution found.")

**OUT PUT :**

