Program to solve 8 queens problem

Aim:

To write a program to solve the 8 queens problem.

Program:

N=8  
def print\_board(board):  
    for row in board:  
        print(" ".join("Q" if x else "." for x in row))  
    print()  
  
def is\_safe(board, row, col, N):  
    for i in range(row):  
        if board[i][col]:  
            return False  
  
    for i, j in zip(range(row - 1, -1, -1), range(col - 1, -1, -1)):  
        if board[i][j]:  
            return False  
  
    for i, j in zip(range(row - 1, -1, -1), range(col + 1, N)):  
        if board[i][j]:  
            return False  
  
    return True  
  
def solve\_n\_queens(board, row, N):  
    if row == N:  
        print\_board(board)  
        return True  
     
    res = False  
    for col in range(N):  
        if is\_safe(board, row, col, N):  
            board[row][col] = True  
            res = solve\_n\_queens(board, row + 1, N) or res  
            board[row][col] = False    
     
    return res  
  
board = [[False for \_ in range(N)] for \_ in range(N)]  
if not solve\_n\_queens(board, 0, N):  
    print("Solution does not exist")  
else:  
    print("Solution found!")

Output:

Q . . . . . . .

. . . . Q . . .

. . . . . . . Q

. . . . . Q . .

. . Q . . . . .

. . . . . . Q .

. Q . . . . . .

. . . Q . . . .

Q . . . . . . .

. . . . . Q . .

. . . . . . . Q

. . Q . . . . .

. . . . . . Q .

. . . Q . . . .

. Q . . . . . .

. . . . Q . . .

Result:

Thus the code is executed successfully.