

**Sehrish Mubarik**

**BSAIM-F23-001(4A)**

**Assignment no 04**

**Submitted to:**

**Sir Rasikh ali**

**Lab tasks**

**Question no 01:**

**N-queen problems**

def board\_for\_nqueen(board, n):

    for row in board:

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

    print("\n")

def check(board, row, col, n):

    for i in range(row):

        if board[i][col]:

            return False

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

        if board[i][j]:

            return False

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

        if board[i][j]:

            return False

    return True

def solve\_problem(board, row, n, solutions):

    if row == n:

        solutions.append([row[:] for row in board])

        board\_for\_nqueen(board, n)

        return

    for col in range(n):

        if check(board, row, col, n):

            board[row][col] = 1

            solve\_problem(board, row + 1, n, solutions)

            board[row][col] = 0

def n\_queens(n):

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

    y = []

    solve\_problem(board, 0, n, y)

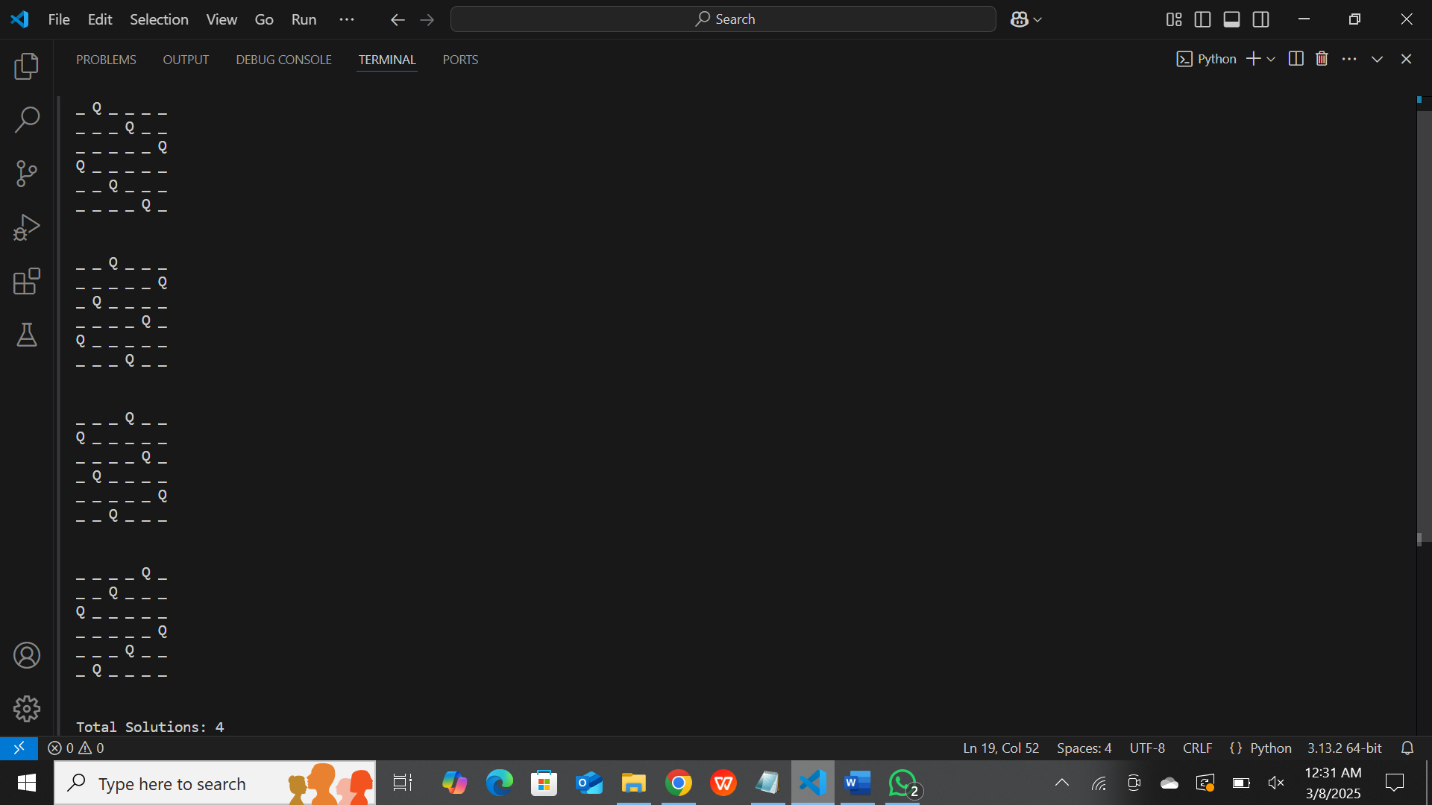
    print(f"Total Solutions: {len(y)}")

    return y

x = 6

n\_queens(x)

**output**

****