

Job No: 04

Job Name: Write a program for N-Queen Problem

Theory: The N-Queens problem is a well-known puzzle in computer science. It involves placing N queens on an N×N chessboard in such a way that no two queens threaten each other. In chess, a queen can move horizontally, vertically, or diagonally, so the solution to the N-Queens problem requires that no two queens share the same row, column, or diagonal.

Code:

```
def is_safe(board, row, col, N):
    for c in range(col):
        if board[row][c] == 'Q':
            return False

    r, c = row, col
    while r >= 0 and c >= 0:
        if board[r][c] == 'Q':
            return False
        r -= 1
        c -= 1

    r, c = row, col
    while r < N and c >= 0:
        if board[r][c] == 'Q':
            return False
        r += 1
        c -= 1

    return True

def solve_n_queens_util(board, col, N, count):
    if col >= N:
        count[0] += 1
        print_solution(board, N)
        return

    for row in range(N):
        if is_safe(board, row, col, N):
            board[row][col] = 'Q'
```

```

        solve_n_queens_util(board, col + 1, N, count)

        board[row][col] = '.'

def solve_n_queens(N):
    board = [['.' for _ in range(N)] for _ in range(N)]

    count = [0]
    solve_n_queens_util(board, 0, N, count)

    print(f"Number of solutions: {count[0]}")

def print_solution(board, N):
    for row in board:
        print(' '.join(row))
    print()

N = int(input("Enter n: "))
solve_n_queens(N)

```

Input/Output:

```

Enter n: 4
. . Q .
Q . . .
. . . Q
. Q . .

. Q . .
. . . Q
Q . . .
. . Q .

Number of solutions: 2

```