EX: 1
Date:

N-Queen Problem

Aim:

To Place N queen on a NXN chessboard such that no two queen be in same row, column or diagonal

Algorithm:

- Creak nxn with empty position
- Place queen recursively one oneach column
- Before Placing check - no same row
  - no left & right dragonal
- If safe Place & move to next column
- If no solution backtrack
- Continue upto final step

Code:

```
def is_safe(board, row, col, n):
    for i in range(col):
        if board[row][i] == 1:
            return false
    for i,j in zip(range(row,-1,-1), range(col,-1,-1)):
        if board[i][j] == 1:
            return false
    for i,j in zip(range(row,n), range(col,-1,-1)):
        if board[i][j] == 1:
            return false
    return True
```

```
def Solve_nqueen(board, col, n):
    if col >= n:
        return True
    for i in range(n):
        if is_safe(board, i, col, n):
            board[i][col] = 1
            if solve_nqueen(board, col+1, n):
                return True
            board[i][col] = 0
    return false

def Print_board(board, n):
    for i in range(n):
        for j in range(n):
            Print("Q" if board[i][j] == 1 else '.', end=" ")
        Print()

def n_queens(n):
    board = [[0] * n for _ in range(n)]
    if not solve_nqueen(board, 0, n):
        Print("solution does not exist")
        return false
    Print_board(board, n)
    return True
n = int(input("Enter no of Queen: "))
n_queen(n)
```

Output:

Enter no of queen: 8

Result:

Thus NQueen Problem is solved & executed successfully & output is verified