

N QUEENS PROBLEM

AIM:

ALGORITHM:

220071601141

OUTPUT:

```
IDLE Shell 3.12.6
File Edit Shell Debug Options Window Help
Python 3.12.6 (tags/v3.12.6:a4a2d2b, Sep  6 2024, 20:11:23) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Irfan/AppData/Local/Programs/Python/Python312/220071601141.py
Enter the number of queens (N): 4
. . Q .
Q . . .
. . . Q
. Q . .
>>> |
```

PROGRAM:

```
def print_board(board):
    for row in board:
        print(" ".join("Q" if cell else "." for cell in row))
    print()

def is_safe(board, row, col):
    for i in range(col):
        if board[row][i]:
            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, len(board)), range(col, -1, -1)):
        if board[i][j]:
            return False
    return True

def solve_n_queens(board, col):
    if col >= len(board):
        return True
    for i in range(len(board)):
        if is_safe(board, i, col):
            board[i][col] = True
            if solve_n_queens(board, col + 1):
                return True
            board[i][col] = False
    return False
```

220071601141

```
def n_queens(n):  
    board = [[False] * n for _ in range(n)]  
    if solve_n_queens(board, 0):  
        print_board(board)  
    else:  
        print("No solution exists")  
if __name__ == "__main__":  
    try:  
        n = int(input("Enter the number of queens (N): "))  
        if n <= 0:  
            print("Number of queens must be a positive integer.")  
        else:  
            n_queens(n)  
    except ValueError:  
        print("Invalid input. Please enter a positive integer.")
```

RESULT:

220071601141