NAME: LAKSHANIKA VS

REG. NO: 241801131

EXP.NO: 1

EXP.NAME: 8 QUEEN

```python
import copy

N = 8  # Size of the chessboard (8x8)

# Function to print the solution
def printSolution(board):
    for row in board:
        for i in range(N):
            print("Q" if row[i] else ".", end=" ")
        print()
    print()

# Function to check if a queen can be placed on board[row][col]
def isSafe(board, row, col):
    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

# Modified solver to stop after first solution
def solve(board, row, solutions, found):
    if found[0]:
        return  # Stop further recursion once a solution is found
    if row == N:
        solutions.append(copy.deepcopy(board))
        printSolution(board)
        found[0] = True
        return
    for col in range(N):
        if isSafe(board, row, col):
            board[row][col] = 1
            solve(board, row + 1, solutions, found)
            board[row][col] = 0

# Main function
def eightQueens():
    board = [[0 for _ in range(N)] for _ in range(N)]
    solutions = []
    found = [False]  # Use a list to make it mutable inside recursion
    solve(board, 0, solutions, found)

# Run the program
eightQueens()
```

Output

```
Q . . . . . . .
. . . . Q . . .
. . . . . . . Q
. . . . . Q . .
. . Q . . . . .
. . . . . . Q .
. Q . . . . . .
. . . Q . . . .


=== Code Execution Successful ===
```