

Softcopy – Practical-5 (N-Queen Problem)

Code

```
def placeQueens(i, cols, leftDiagonal, rightDiagonal, cur):
    n = len(cols)

    if i == n:
        return True

    for j in range(n):
        if cols[j] or rightDiagonal[i + j] or leftDiagonal[i - j + n -
1]:
            continue

        cols[j] = 1
        rightDiagonal[i + j] = 1
        leftDiagonal[i - j + n - 1] = 1
        cur.append(j)

        if placeQueens(i + 1, cols, leftDiagonal, rightDiagonal, cur):
            return True

        cur.pop()
        cols[j] = 0
        rightDiagonal[i + j] = 0
        leftDiagonal[i - j + n - 1] = 0

    return False

def nQueen(n):
    cols = [0] * n
    leftDiagonal = [0] * (n * 2)
    rightDiagonal = [0] * (n * 2)
    cur = []
    board = [['.' for _ in range(n)] for _ in range(n)]

    if placeQueens(0, cols, leftDiagonal, rightDiagonal, cur):
        for i in range(n):
            board[i][cur[i]] = 'Q'
        return board
    else:
        return None

def printBoard(board):
    if board:
        for row in board:
            print(" ".join(row))
    else:
        print("No solution exists.")

n = int(input("Enter the number of queens:\t"))
board = nQueen(n)
printBoard(board)
```

Output

```
$ python3 Code-A5.py
Enter the number of queens: 10
Q . . . . . . . . . .
. . Q . . . . . . .
. . . . . Q . . . .
. . . . . . Q . . .
. . . . . . . Q . .
. . . . . Q . . . .
. . . . . . . . Q .
. Q . . . . . . . .
. . . Q . . . . . .
. . . . . Q . . . .
```