

```
In [2]: N = int(input("Enter the number of quins :-"))

print(f"N-quin problem solution for {N} Quins :")

def printSolution(board):
    for i in range(N):
        for j in range(N):
            print(board[i][j], end = " ")
        print()

def isSafe(row, col, slashCode, backslashCode,
           rowLookup, slashCodeLookup,
           backslashCodeLookup):
    if (slashCodeLookup[slashCode[row][col]] or
        backslashCodeLookup[backslashCode[row][col]] or
        rowLookup[row]):
        return False
    return True

def solveNQueensUtil(board, col, slashCode, backslashCode,
                     rowLookup, slashCodeLookup,
                     backslashCodeLookup):

    if(col >= N):
        return True
    for i in range(N):
        if(isSafe(i, col, slashCode, backslashCode,
                 rowLookup, slashCodeLookup,
                 backslashCodeLookup)):

            board[i][col] = 1
            rowLookup[i] = True
            slashCodeLookup[slashCode[i][col]] = True
            backslashCodeLookup[backslashCode[i][col]] = True

            if(solveNQueensUtil(board, col + 1,
                               slashCode, backslashCode,
                               rowLookup, slashCodeLookup,
                               backslashCodeLookup)):

                return True
```

```
        board[i][col] = 0
        rowLookup[i] = False
        slashCodeLookup[slashCode[i][col]] = False
        backslashCodeLookup[backslashCode[i][col]] = False

    return False

def solveNQueens():
    board = [[0 for i in range(N)]
              for j in range(N)]

    slashCode = [[0 for i in range(N)]
                  for j in range(N)]
    backslashCode = [[0 for i in range(N)]
                     for j in range(N)]

    rowLookup = [False] * N

    x = 2 * N - 1
    slashCodeLookup = [False] * x
    backslashCodeLookup = [False] * x

    for rr in range(N):
        for cc in range(N):
            slashCode[rr][cc] = rr + cc
            backslashCode[rr][cc] = rr - cc + (N-1)

    if(solveNQueensUtil(board, 0, slashCode, backslashCode,
                        rowLookup, slashCodeLookup,
                        backslashCodeLookup) == False):
        print("Solution does not exist")
        return False

    printSolution(board)

    return True

solveNQueens()
```

```
Enter the number of quins :-4
N-quin problem solution for 4 Quins :
0 0 1 0
1 0 0 0
0 0 0 1
0 1 0 0
```

Out[2]: True

In []: