# N-Queens Problem (Mini-project 2)

You are given with the two files. (Board.swift, QueensSolver.swift)
Your goal is to write a function *solve8Queens(board: inout Board, col: Int)* that accepts a Board as a parameter and tries to place 8 queens on it safely. (Feel free to modify the function prototype. You are allowed to create a function with different set of parameters)

Here are **two** tasks you need to accomplish from this mini-project.

1. Print all possible ways to place 8 queens on 8x8 chessboard.

**It should look like this...**

```
[Q - - - - - - -]
[- - - - - - Q -]
[- - - - Q - - -]
[- - - - - - - Q]
[- Q - - - - - -]
[- - - Q - - - -]
[- - - - - Q - -]
[- - Q - - - - -]

[Q - - - - - - -]
[- - - - - - Q -]
[- - - Q - - - -]
[- - - - - Q - -]
[- - - - - - - Q]
[- Q - - - - - -]
[- - - - Q - - -]
[- - Q - - - - -]

[Q - - - - - - -]
[- - - - - Q - -]
[- - - - - - - Q]
[- - Q - - - - -]
[- - - - - - Q -]
[- - - Q - - - -]
[- Q - - - - - -]
[- - - - Q - - -]

(... 92 possible ways)
```

2. Print the first possible way to place 8 queens on the 8x8 board with **less than 115 recursive calls**. Stop immediately as soon as you find the first solution.

**It should look like this...**

```
[Q - - - - - - -]
[- - - - - - Q -]
[- - - - Q - - -]
[- - - - - - - Q]
[- Q - - - - - -]
[- - - Q - - - -]
[- - - - - Q - -]
[- - Q - - - - -]
```

**Number of recursive calls: 114**

NOTE:
(You can create a global variable to count the number of recursive calls being made.)

```
var count = 0
func solve8Queens(board: inout Board, col: Int) -> Bool {
      count += 1
      ...
}

solve8Queens(...)
print(count)
```

*Reference Problem: https://leetcode.com/problems/n-queens/*