

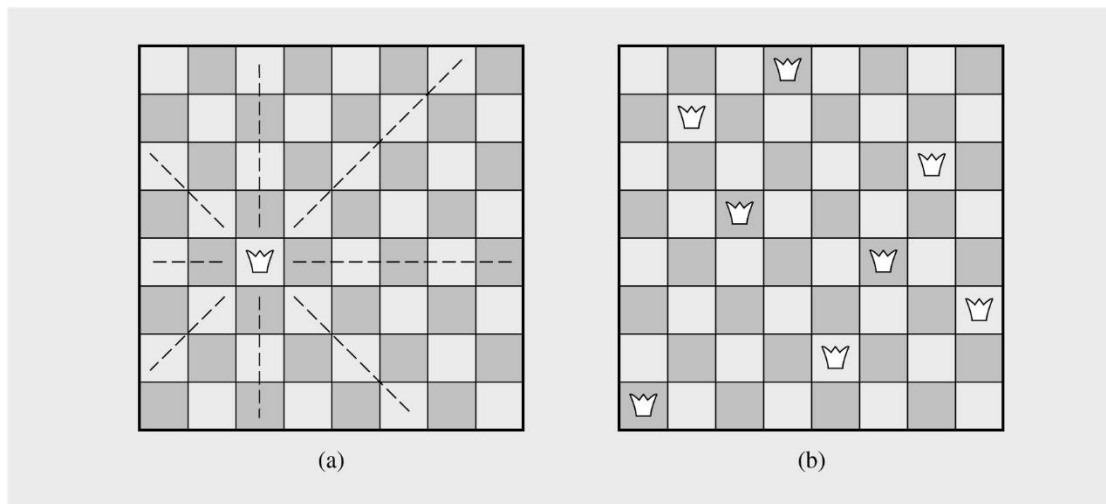
# CS 321 Programming Assignment 3

Spring 2023 – Chapter 5 Recursion

Due March 6, 2023 11:55 PM on Laulima

## Introduction

Section 5.9 in our textbook discusses backtracking – a problem solving approach that can be carried out via recursion. The problem they discuss is the “8-Queens” problem. In this problem, we attempt to place 8 queen pieces from chess on an 8x8 checkerboard in such a manner that no queen threatens another queen. Queens threaten pieces along the vertical and horizontal axes they are on, as well as on both diagonal axes, which I will refer to as the left and right diagonal axis. See the figure below for an example. On the left, we see all the positions that the queen in the figure threatens. On the right, we see one possible placement of eight queens on the board such that no queen threatens another. Your goal is to understand a recursive solution that identifies a correct solution (if it exists) to the n-queens problem for a given n, up to a given limit.



## Your Task

Figure 5.13 in your textbook contains a barebones implementation of a solution to the 8-queens problem. There is no test code given, and the code as written only says how many solutions are found – it does not show the solutions. Your task is to add test code and modify the existing code so that for a given value of n, the **first** solution found (if any solution exists) is displayed, as well

as to report the amount of work required to find a solution. Your program must adhere to the following requirements:

- N can range from 4 to 10. Ask the user what size to use.
- Display the solution output as an n x n grid. White squares can be drawn with a '.' character.  
Black squares can be drawn with an 'x'. Placed queens can be drawn with a 'Q'. Thus, a solution for the 4-queens problem might be drawn as:

```
. Q . x
x . x Q
Q x . x
x . Q .
```

- Display the number of board configurations examined before a solution was found. Any attempt to place a queen on a square, whether valid or not, and whether or not all other queens have been placed, counts as a board configuration. The first configuration consists of placing the first queen on the first square of the first column and row, for instance.
- Do not show all possible solutions. Just show the first found.
- Ask the user if they want to enter another value for n and rerun the search.
- Make sure you fully comment your code.
- All of your code for this assignment should be in a single .cpp file. Submit only that file as your solution.

For full credit, your solution will need to use recursion, and needs to use the array-based strategy employed by the book's code. That solution maintains three arrays: column, rightDiagonal, and leftDiagonal. Each array contains one entry for each column or diagonal present and indicates whether that position is currently blocked. You should read the discussion in section 5.9 clearly in order to understand how these arrays are used. This approach using 3 one dimensional arrays is much faster than an approach using a 2-dimensional array to track blocked positions would be. You need to consider how to display a solution when a solution is found, and how to break out of the recursion at that point.

## Submission

When you are finished, please submit your code on Laulima as a single .cpp file.