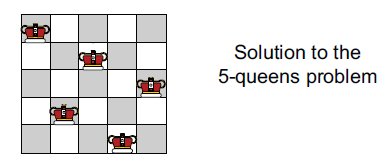
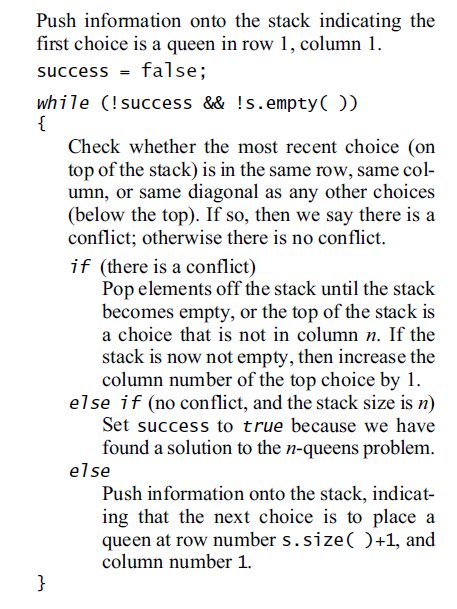
Suppose that you have n queens from a chessboard. Is it possible to place all n queens on the board so that no two queens are in the same row, no two queens are in the same column, and no two queens are on the same diagonal? For example, a solution with n = 5 is shown here:



This problem is called the n-queens problem. For this project, you are to write a function that has one integer parameter, n, and determines whether there is a solution to the n-queens problem. If a solution is found, then the procedure prints the row and column of each queen. Your program should solve the problem by making a sequence of choices, such as “Try placing the row 1 queen in column 1,” or “Try placing the row 7 queen in column 3.” Each time a choice is made, the choice is pushed onto a stack that already contains all the previously made choices.

The purpose of the stack is to make it easy to fix incorrect choices, using the following pseudocode, with a stack, s, and a boolean variable, success:



This technique is called backtracking since we keep our choices on a stack and back up to correct any mistakes that are made. Notice that when you check for a conflict, you will need access to the entire

stack (not just the top), so that you should use the seek function from Programming Project 4 or an iterator from Programming Project 3. You could also use the Standard Library stack class (which has an

iterator). I have provided a stack class with seek function too.