Haley Keizur

This project is a simple sudoku solving program, which uses backtracking to ensure no number is used more than once in a single square or line (diagonal, horizontal or vertical). To start, the program welcomes the user and asks whether they'd like a random puzzle output, or if the user would like to enter their own puzzle.
If the user types "random," the program chooses one of four pre-input puzzles. It calls the method fillBoard, which calls emptyCheck and doesItWork. These methods check whether there are any empty spaces in the puzzle, and if a number will work in those spots, respectively. The fillBoard uses backtracking, and attempts to fill spaces one by one. If a digit doesn;t work, it's eliminated and the next digit is tested. This is much simpler than compiling all possible digits and trying one by one.
If the fillBoard returns true, the program outputs the puzzle. If the user chooses "own," they are prompted to enter 81 digits, 9 for each row, including 0's as blank spaces. That matrix is then input and the process repeats, starting by calling the fillBoard.
If the user types incorrectly, they are told so. If their puzzle doesn't work, they get a message telling them that it is invalid and there is no solution.

Runtime for "random" option: 4.5055 seconds
Runtime for "own" option (including time to type in numbers):31.1348 seconds
Time and Space complexity: O(n^2) (for inside for loop)