Exercise 1 (20 points). Improve your implementation of comparative Sudoku. You can use a refinement of backtracking, some partial solving using deduction followed by some other method, a constraint based solver, or some other method (or combination of methods). You are allowed to use C, C++, C#, Java, or Ruby in your implementation.

Please describe briefly the main algorithmic ideas behind your new comparative Sudoku solver.

Exercise 2 (20 points). Print out your source code. Make sure that it is well-documented.

Exercise 3 (20 points). Solve the challenge problems that will be posted. Give a table that compares the time of your solver from Problem Set 7 with the solver from this problem set. Include a printout of the solutions produced by your new comparative Sudoku solver.

Exercise 4. Let $\phi$ be a boolean formula in 3-CNF. An $\neq$-assignment to the variables of $\phi$ is one where each clause contains two literals with unequal truth values. In other words, an $\neq$-assignment satisfies $\phi$ without assigning three true literals in any clause.

(a) Show that the negation of a $\neq$-assignment to $\phi$ is also a $\neq$-assignment.

(b) Let $\neq$-SAT be the collection of boolean formulas in 3-CNF that have an $\neq$-assignment. Show that $3 - SAT \leq_p \neq SAT$, and that $\neq$-SAT is NP-complete. [Hint: Replace each clause $(y_1 \lor y_2 \lor y_3)$ by two clauses $(y_1 \lor y_2 \lor z_i) \land (\neg z_i \lor y_3 \lor b)$, where $z_i$ is a new variable for each clause and $b$ is a single additional new variable.]

Exercise 5 (20 points). Problem 34-3 d,e,f (that is, prove that 3-COLOR is NP-complete using the reduction 3-SAT$\leq_p$ 3-COLOR) on page 1103 of our textbook.