1. The rules of Sudoku are as follows:
   • Each puzzle of order $N$ is presented in an $N^2 \times N^2$ grid, which is divided into $N \times N$ squares with thick border lines.
   • Some cells are pre-populated with integers between 1 and $N^2$
   • All remaining cells must be filled so that each row, column, and $N \times N$ square has exactly one of each integer from 1 through $N^2$

![Problem](image1.png) ![Solution](image2.png)

Figure 1: Simple example of a sudoku puzzle ($N = 2$), and its complete solution.

1a. Write a code to solve the blank puzzle below, analogous to the example solution above. Output the full solution as a 9 x 9 matrix.

![Blank Puzzle](image3.png)

Figure 2: Blank puzzle ($N = 3$) to solve.

1b. Is this problem $P$, $NP$, or uncomputable? How should the execution time of your program scale with the order $N$ (use big-oh notation)?
2. Consider an analog computer that obeys the laws of quantum mechanics. In particular, it consists of \( N \) spin-1/2 particles (e.g., electrons), each of which can exist in a spin-up, spin-down, or superposition state (partially spin-up and partially spin-down). These are known as quantum bits, or *qubits*. See the figure below for an illustration. Furthermore, each of these qubits can be *entangled* with one another, which means that the spins can be correlated. The full range of states allowed is known as a Hilbert space.

![Figure 3: Chain of spin-1/2 particles that are subject to the laws of quantum mechanics, known as qubits.](image)

**2a.** Assume that propositional logic can be performed on this qubit array. How long would it take to evaluate the satisfiability (i.e., truth for at least one choice of input values) of a non-trivial logical proposition, as a function of \( N \)?

**2b.** How long would it take to evaluate the validity (i.e., truth for all choices of input values) of a non-trivial logical proposition, as a function of \( N \)?

**2c.** Would you consider this example to contradict the Church-Turing thesis? Why or why not?