1. Show the following languages are regular by creating finite automata with $L = L(M)$

   a) Strings over \{a,b\} that contain 2 consecutive a’s
   b) Strings over \{a,b\} that do not contain 2 consecutive a’s
   c) The set of strings over \{0,1\} which contain the substring 00 and the substring 11
   d) The set of strings over \{a,b\} which do not contain the substring ab.

Show your answers in both table and graph form.

   a) Strings over \{a,b\} that contain 2 consecutive a’s

   b) Strings over \{a,b\} that do not contain 2 consecutive a’s

   c) The set of strings over \{0,1\} which contain the substring 00 and the substring 11
Problem doesn’t say whether this must be a dfa and this is easier with an nfa:

\[
\begin{array}{ccc}
\lambda & 0 & 1 \\
>q_0 & q_1 \cdot q_5 & \\
q_1 & q_2 & \\
q_2 & q_3 & q_1 \\
q_3 & q_3 & q_4 \\
q_4 & q_3 & q_9 \\
q_5 & q_5 & q_6 \\
q_6 & q_5 & q_7 \\
q_7 & q_8 & q_7 \\
q_8 & q_{10} & q_7 \\
*q_9 & q_9 & q_9 \\
*q_{10} & q_{10} & q_{10}
\end{array}
\]

d) The set of strings over \{a,b\} which do not contain the substring \(ab\).

Similar to parts a and b, I will first create a fa that does accept \(a\ b\) and then I will reverse the final and the nonfinal states:
#2. Create an NFA (with $\lambda$ transitions) for all strings over \{0, 1, 2\} that are missing at least one symbol. For example, 00010, 1221, and 222 are all in L while 22102 is not in L.

a) 

\[
L(M) = \text{Alternating 0's and 1's (including none) that begin with a 0} \\
(01)^* (01 U 0)
\]

b) 

0 or more $ab$’s followed optionally by 0 or more $aab$’s 

(ab)* (aab)*

#3. a) Given an NFA with several final states, show how to convert it into one with exactly one start state and exactly one final state.

Create a new initial state and a $\lambda$-transition from it to all the original start states
Create a new final state and a $\lambda$-transition from all the original final states (which mark to no longer be final) to this new final state

b) Suppose an NFA with $k$ states accepts at least one string. Show that it accepts a string of length $k-1$ or less.

Look at a fa with 3 states:
No matter how you draw the transitions or which states are final states, to process a string of length 3 means you visited a state twice. For example:

accepts the string of length 3: aba

But just by not visiting the revisited state (q₁), this will accept aa (of length 2)

In general, if a string of length k is accepted by a fa with k states, it visits (at least) 1 state twice. By not visiting this state the 2\textsuperscript{nd} time (e.g., don’t take the loop), we can accept a string with 1 fewer symbol, i.e., of length k – 1.

#5. Let L be a regular language. Show that the language consisting of all strings not in L is also regular.

If L is regular, there is a dfa, M, such that L = L(M), that is, M accepts L. If we create a new finite automaton, M’, by reversing final and non-final states, we will accept what M didn’t and reject what M accepted; that is, C(L) = L(M’).