1. For each of the following regular expressions, describe or draw two finite-state machines:
   - An NFA that accepts the same language, using Thompson’s algorithm (described in class and in the notes)
   - An equivalent DFA, using the incremental subset construction described in class. For each state in your DFA, identify the corresponding subset of states in your NFA. Your DFA should have no unreachable states.
   
   (a) \((01 + 10)^* (0 + 1 + \epsilon)\)
   (b) \(1^* + (10)^* + (100)^*\)

2. Prove that for any regular language \(L\), the following languages are also regular:
   
   (a) \(\text{SUBSTRINGS}(L) := \{x \mid wx\ y \in L \text{ for some } w, y \in \Sigma^*\}\)
   (b) \(\text{HALF}(L) := \{w \mid ww \in L\}\)

   \[\text{Hint: Describe how to transform a DFA for } L \text{ into NFAs for } \text{SUBSTRINGS}(L) \text{ and } \text{HALF}(L). \text{ What do your NFAs have to guess? Don’t forget to explain in English how your NFAs work.}\]

3. Which of the following languages over the alphabet \(\Sigma = \{0, 1\}\) are regular and which are not? Prove your answers are correct. Recall that \(\Sigma^+\) denotes the set of all nonempty strings over \(\Sigma\).
   
   (a) \(\{wxw \mid w, x \in \Sigma^+\}\)
   (b) \(\{wxw \mid w, x \in \Sigma^+\}\)
   (c) \(\{wxwy \mid w, x, y \in \Sigma^+\}\)
   (d) \(\{wxxy \mid w, x, y \in \Sigma^+\}\)
For each of the following regular expressions, describe or draw the NFA obtained from Thompson's algorithm, and the equivalent DFA obtained from the incremental subset construction.

(a) \((0 + 1)^* (0 + 1 + \varepsilon)\)

(b) \(1^* + (10)^* + (100)^*\)
Construct a DFA for the following language over alphabet \( \{0, 1\} \):

\[
L = \left\{ w \in \{0, 1\}^* \mid \text{the number represented by binary string } w \text{ is divisible by } 19, \text{ but the length of } w \text{ is not a multiple of } 23 \right\}.
\]
Prove that each of the following languages is not regular.

1. \( \{ w \in \{0\}^* \mid \text{length of } w \text{ is a perfect square; that is, } |w| = k^2 \text{ for some integer } k \} \).

2. \( \{ w \in \{0, 1\}^* \mid \text{the number represented by } w \text{ as a binary string is a perfect square} \} \).
Suppose $L$ is a regular language which guarantees to contain at least one palindrome. Prove that if an $n$-state DFA $M$ accepts $L$, then $L$ contains a palindrome of length polynomial in $n$. What is the polynomial bound you get?