Problem 1. (Properties of Regular Languages)
(a) Show that if \( L \) is a regular language, then language \( L^n \) is regular for every \( n > 0 \).
(b) For any two languages \( L_1 \) and \( L_2 \) defined over an alphabet \( \Sigma \), we define the following language, where \( \sigma_i, \gamma_i \in \Sigma \):

\[
\text{Interleave}(L_1, L_2) = \{ w : w = \sigma_1 \gamma_1 \sigma_2 \gamma_2 \cdots \sigma_k \gamma_k, \ \sigma_1 \sigma_2 \cdots \sigma_k \in L_1, \ \gamma_1 \gamma_2 \cdots \gamma_k \in L_2, \ k \geq 0 \}.
\]

Show that if \( L_1 \) and \( L_2 \) are regular, then \( \text{Interleave}(L_1, L_2) \) is regular too.

Problem 2. (NFA)
Give non-deterministic finite automata (NFA) that accept the following:
1. Given Regular Expression: \( 0 \ 1^* \ 0^* \ 1^* \ 0 \)
    Test Strings 0110, 011010, 01000
2. To recognize floating literals in scientific notation.
    Floating point numbers
    http://linuxsoftware.co.nz/cppgrammar.html#floating-literal describes the grammar for floating-literals.
    Test strings: \(-314.151926 e - 02, 3.14, +3e + 0003.14159265358E - 000\)

For each language, draw the respective NFA in JFLAP (using the Finite Automaton option). Then run the respective test strings using the Multiple Run option of JFLAP. Your answer for each language will be the printout of your JFLAP automaton together with the result of the multiple test run.

Problem 3. (Regular Expressions)
Give regular expressions for the following languages over alphabet \( \{0,1\} \):
(a) \( L_1 = \{ \text{all strings which do not contain substring 0000} \} \).
    Test strings: 1001, 001010, 1001100, 010101101.
(b) \( L_2 = \{ \text{all strings containing both 111 and 000 as substrings} \} \).
    Test strings: 001010010, 0101001, 10010101, 01001001.
(c) Here assume the input alphabet is 0 to 9.
    \( L_3 = \{ \text{all strings when treated as a decimal number are powers of 10} \} \).
    Test strings: 1010, 01001001.

The answers are the printouts of JFLAP. Use the Regular Expression option to write your regular expression. Since JFLAP does not support testing of strings on regular expressions, you will test your strings as follows. Use the Convert option to convert your regular expression to an NFA (by selecting the Do All and then the Export buttons). Run the tests cases on the NFA using the Multiple Run option, and print the result. So, for each language you need to give the printout of the regular expression, and the multiple run on the corresponding NFA.
Problem 4. (Pumping Lemma) Prove that the following languages are not regular:

(a) $Add = \{0^x 1 0^y 1 0^{x+y}\}$.

(b) $Squares = \{\text{all strings over } a,b \text{ such that the number of } b\text{'s is the same as the twice the number of } a\text{'s}\}$

(c) $L = \{w : w \in \{1\}^*, n_1(w) \text{ is a perfect square}\}$. 