1. Provide two valid strings in the languages described by each of the following regular expressions, with alphabet $\Sigma = \{0, 1, 2\}$.

(a) $0(010)^*1$

(b) $(21 \cup 10)^*0012^*$
   Examples: 001, 001222, 21001, 10001, 210012, 2121001222, 102121001

(c) $1^*(200)^* \cup 100*01$

2. For each of the following DFAs, provide a Regular Expression to describe the language, with alphabet $\Sigma = \{a, b\}$.

(a) **RED QUESTION**

(b) **BLUE QUESTION**
3. Create a DFA to accept each of the following languages.
A={w | last number in w is even}, given alphabet $\Sigma = \{0,1,2,3\}$

B={w | at least three symbols in w}, given alphabet $\Sigma = \{a, b, c\}$

C={w | sum of digits in w equals 2}, given alphabet $\Sigma = \{0,1,2\}$

4. Convert each of the following NFAs to a DFA, with alphabet $\Sigma = \{a, b\}$. 

(a) RED QUESTION
5. Prove the following languages are not regular.
   (a) \( A=\{b^k a \mid k > 0\} \)
(b) \( B = \{ 0^k 1^{2k} 0^k \mid k > 0 \} \)

Pumping lemma!

\( w = 0^p 1^{2p} 0^p \quad x = 0^m \quad y = 0^n \quad z = 0^{p-(m+n)} 1^{2p} 0^p \quad p \geq n > 0 \)

If \( w \in B \), then must be \( xy^2z \in B \)

\( xy^2z = 0^{p+n} 1^{2k} 0^p \quad \) First number of 0’s now is not half the number of 1’s, so \( xy^2z \) is NOT in language B. This means \( w \) was not pumpable and B is not regular!

7. Provide two valid strings for each of the following CFGs.

(a) \( G_1: \)

\[
S \rightarrow A \mid B \\
A \rightarrow DC \mid C \\
B \rightarrow EF \mid F \\
C \rightarrow \text{dog} \mid \text{cat} \mid \text{mouse} \\
D \rightarrow \text{big} \mid \text{small} \mid \text{red} \mid \text{white} \\
E \rightarrow \text{quickly} \mid \text{slowly} \\
F \rightarrow \text{runs} \mid \text{swims} \mid \text{jumps} \mid \text{barks}
\]

(b) \( G_2: \)

\[
S \rightarrow BA \mid B \\
B \rightarrow xBx \mid \epsilon \\
A \rightarrow c \mid de \mid f \\
B \rightarrow \epsilon \\
B \rightarrow xBx \rightarrow xx \epsilon xx \rightarrow xxxx \\
BA \rightarrow \epsilon de \rightarrow \text{de} \\
BA \rightarrow x \epsilon xc \rightarrow xxc
\]

(c) \( G_3: \)

\[
S \rightarrow CaC \mid C \\
C \rightarrow yCy \mid y
\]
8. Convert the following CFGs to CNF (same as Q7).

(a) G1: (for G1, each word is a terminal)
    S -> A | B
    A -> DC | C
    B -> EF | F
    C -> dog | cat | mouse
    D -> big | small | red | white
    E -> quickly | slowly
    F -> runs | swims | jumps | barks

(b) G2:
    S -> BA | B
    B -> xBx | ε
    A -> c | de | f

(c) G3:
    S -> CaC | C
    C -> yBy | y

    S -> CaC | yBy | y replace C
    C -> yBy | y

    S -> CAC | YBY | y replace literals with variables
    A -> a
    Y -> y
    C -> YBY | y

    S -> CD | YE | y replace 3-variable terms with 2-var terms
    D -> AC
    E -> BY
    A -> a
    Y -> y
    C -> YE | y
9. Express each of the following languages as a CFG.
(a) \( A = \{x^ky^{2k}z\} \)
   \[ S \to Bz \]
   \[ B \to xByy \mid \varepsilon \]

(b) \( B = \{w \mid w \text{ is described by } (ab)^*ba\} \)

(c) \( C = \{010^k101^{k+2} \mid k > 0\} \)

10. Describe the PDA to accept each of the following languages (languages from Q9).
(a) \( A = \{x^ky^{2k}z\} \)

(b) \( B = \{w \mid w \text{ is described by } (ab)^*ba\} \)

(c) \( C = \{010^k101^{k+2} \mid k > 0\} \)
11. What is the response of PDA P1 to each input: i.e., does it reach an accept state?

Input 1: bbba
**Does not reach accept state!** (It starts with b, so quickly departs NFA.)

Input 2: aaa

Input 3: abb

Input 4: aaaaaabba
12. Describe the configurations resulting from each of the input tapes specified below for the following Turing Machine.

(a) aabb
q₀ aabb
q₁ babb
q₁ babb
q₀ baXb
reject baXb~

(b) abaaa

(c) aaaba

13. Express the following problems as languages.

(a) Determine if two specified CFG’s accept complementary inputs – every accepted input for the first CFG is rejected by the second CFG and vice versa.
(b) Determine if a specified DFA accepts a specified string repeated zero or more times.

(c) Determine if a specified Turing machine accepts the same language as a specified PDA.
L = {<P,T> | L(P) = L(T)}

14. Prove the follow languages are decidable.
   (a) Determine if a specified DFA accepts a specified string repeated zero or more times.

   (b) Determine if a specified CFG is in Chomsky Normal Form.

   (c) Determine if a specified CFG does not accept a specified word. Generate all words of length |w|. If one of these words is the originally specified word, reject. Otherwise accept.

15. Provide a big-O and a little-o complexity for each function.
   (a) f(n) = 20 n log n + 5n + 2
      Smallest: O(n log n)      Also: O(n^2), O(n^3), O(2^n)
      Near-smallest: o(n^2), o(n log^2 n); also: o(2^n), o(n^6)

   (b) f(n) = 30 n^3 + 6 n^5 + log n
(c) \( f(n) = 5n^2 + n^3 \log n + 4^n + 8 \)

16. Compute the complexity for each algorithm described below.

(a) Algorithm 1: (State the complexity based on \( r \) and \( c \))
   Start with a table of \( r \) rows and \( c \) columns
   1. Sum the elements in each row
      - Use a running sum with a loop across all columns
   2. Find the row with the maximum sum
      - Loop through all rows, saving biggest sum and its row in two separate variables

(b) Algorithm 2: (State the complexity based on \( n \))
   Start with a list of \( n \) elements
   1. While list is longer than 1 element long
      - Replace each pair of elements with the product of the two elements
        (elements 1 and 2 replaced by single product, elements 3 and 4 replaced by single product, elements 5 and 6 replaced by single product, etc.)
   Number of loop repeat: \( \log_2 n \); time to compute products: \( O(n/2) = O(n) \)
   In total: \( O(n \log n) \)

17. Determine if the following problems are in P and/or NP.

(a) Given a directed graph and two nodes \( a \) and \( b \), determine if there are at least two different paths to get from node \( a \) to node \( b \). Paths are “different” if they differ by at least one edge.

(b) In an undirected graph, determine if every node is attached to every other node.
This is effectively finding a clique of size $n$ where $n$ is the number of nodes. However, you only need to test ONE clique – the one containing ALL nodes. Testing one solution takes polynomial time. So this problem actually is in P (and also in NP since all P problems are also in NP).

(c) Determine if the language of a DFA is empty.