0. Notes on Conceptual Review

1. RegEx, CFGs, and DFAs
Let $\Sigma = \{0, 1, 2\}$. Consider the language "all strings with an even number of 2's."

(a) Design a regular expression for this language.

(b) Design a CFG for this language.

(c) Design a DFA for this language.
2. Constructing Regular Expressions
For each of the following, construct a regular expression for the specified language.
(a) Strings from the language $\Sigma := \{a\}^*$ with an even number of $a$'s.

(b) Strings from the language $\Sigma := \{a, b\}^*$ with an even number of $a$'s.

(c) Strings from the language $\Sigma := \{a, b\}^*$ with odd length.

(d) (Challenge) Strings from the language $\Sigma := \{a, b\}^*$ with an even number of $a$'s or an odd number of $b$'s.

3. Context Free Grammars
Consider the following CFG which generates strings from the language $V := \{0, 1, 2, 3, 4\}^*$

$$
S \rightarrow 0X4 \\
X \rightarrow 1X3 \mid 2
$$

List 5 strings generated by the CFG and 5 strings from $V$ not generated by the CFG. Then, summarize this CFG in your own words.
4. Constructing CFGs
For each of the following, construct a CFG for the specified language.

(a) Strings from the language $\Sigma := \{a\}^*$ with an even number of $a$'s.

(b) Strings from the language $\Sigma := \{a, b\}^*$ with odd length.

(c) Strings from the language $\Sigma := \{a, b\}^*$ with an even number of $a$'s or an odd number of $b$'s.

(d) Strings from the language $\Sigma := \{a, b\}^*$ with an equal number of $a$'s and $b$'s.
5. Constructing DFAs
For each of the following, construct a DFA for the specified language.
(a) Strings from the language $\Sigma := \{a\}^*$ with an even number of $a$’s.

(b) Strings from the language $\Sigma = \{a, b\}$ with an even number of $a$’s or an odd number of $b$’s.

(c) Strings from the language $\Sigma = \{a, b\}$ with odd length.
6. Challenge: All the Machines!

Using the alphabet \( \Sigma = \{0, 1, 2, 3, 4, 5\} \), define the language \( L \) as follows. If \( x \) is a string from \( \Sigma^* \) with characters \( x_0, \ldots, x_n \), then \( x \in L \) iff. for every \( i \) between 0 and \( n \), if \( x_i \) is an odd digit, then \( x_k > x_i \) for every \( k > i \). For example, if one of the digits is a 3, every digit after it must be a 4 or higher.

(a) List 3 strings in \( L \) and 3 strings from \( \Sigma^* \) not in \( L \).

(b) Construct a regular expression for the language \( L \).

(c) Construct a CFG for the language \( L \).

(d) Construct a DFA for the language \( L \).