1. Draw a finite automaton accepting each of the following languages:

   (a) \( \{ w \in \{a, b\}^* \mid w \text{ starts with } 'a' \text{ and contains 'baba' as a substring} \} \)

   (b) \( \{ w \in \{0, 1\}^* \mid w \text{ contains '111' as a substring and does not contain } 00 \text{ as a substring} \} \)

   (c) \( \{ w \in \{a, b, c\}^* \mid \text{the number of 'a's modulo 2 is equal to the number of 'b's modulo 3 in } w \} \)

2. Write a regular expression for each of the following languages \( L \):

   (a) \( L = \{ w \in \{0, 1\}^* \mid w \text{ consists of alternating '0's and '1's} \} \)

   (b) \( L = \{ w \in \{0, 1\}^* \mid w \text{ contains an even number of '0's or an even number of '1's} \} \)

   (c) \( L = \{ w \in \{a, \ldots, z\}^* \mid \text{the letters in } w \text{ appear in lexicographical order} \} \)

   (d) Given \( \Sigma = \{a, b, c, d\} \),

   \[ L = \{ w = xyzwy \mid x, w \in \Sigma^+, y \in \Sigma, z \in \Sigma \} \]

   [Each string \( xyzwy \) contains two words \( xy \) and \( wy \) built from letters in \( \Sigma \). The words end in the same letter, \( y \). They are separated by \( z \).]

   (e) \( L = \{ w \in \{+, -, \times, \div, (, ), \text{id}\}^* \mid w \text{ is an algebraic expression using addition, subtraction, multiplication, division, and parentheses over } \text{id}s \} \)

3. Using the subset construction, convert the following regular expressions to a DFA, merging equivalent states where possible:

   (a) \( (ab \mid ac)^* \)

   (b) \( (0|1)^* \ 1100 \ 1^* \)

   (c) \( (01|10|00)^* \ 11 \)

4. Show that regular languages are closed under intersection. That is, show that for any two regular languages \( L_1 \) and \( L_2 \), \( L = L_1 \cap L_2 \) is also regular.

5. The following grammar is not suitable for a top-down predictive parser. Fix the problem by rewriting the grammar. Construct the LL(1) parsing table for your new grammar.

\[
L \rightarrow Ra \quad R \rightarrow aba \quad Q \rightarrow bca \\
L \rightarrow Qba \quad R \rightarrow caba \quad Q \rightarrow bc \\
R \rightarrow Rbc
\]

6. Construct the LR(1) parse table for the following grammar:

\[
S \rightarrow Aa \\
A \rightarrow BC \\
A \rightarrow BCf \\
B \rightarrow b \\
C \rightarrow c
\]

Is this grammar LR(1)?