1. Draw a DFA to recognize the set of strings over \{a,b\}^* that contain the same number of occurrences of the substring \textit{ab} as of the substring \textit{ba}.

2. Present the transition diagram or table for a DFA that accepts the regular set denoted by the expression \((0+1)^* (010 + 11) (0 + 1)^*\)

3. Consider the following assertion:

   Let \textbf{R} be a regular language, then any set \textbf{S}, such that \(\textbf{S} \cup \textbf{R} = \textbf{S}\), is also regular.

   State whether you believe this statement to be True or False by circling your answer.

   \[
   \begin{array}{ll}
   \text{TRUE} & \text{FALSE} \\
   \end{array}
   \]

   If you believe that this assertion is True, present a convincing argument (not formal proof) to back up your conjecture. If you believe that it is False, present a counterexample using known regular and non-regular languages, \textbf{R} and \textbf{S}, respectively.

4. Assume that \textbf{L}_1 and \textbf{L}_1 \cap \textbf{L}_2 are both regular languages. Is \textbf{L}_2 necessarily a regular language? If so, prove this, otherwise show that \textbf{L}_2 could either be regular or non-regular.
5. Let $L$ be defined as the language accepted by the finite state automaton $A$:

![Finite State Automaton Diagram]

a.) Fill in the following table, showing the $\lambda$-closures for each of $A$’s states.

<table>
<thead>
<tr>
<th>State $\lambda$-closure</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda$-closure</td>
<td>{   }</td>
<td>{   }</td>
<td>{   }</td>
<td>{   }</td>
<td>{   }</td>
</tr>
</tbody>
</table>

b.) Convert $A$ to an equivalent deterministic finite state automaton. Use states like $AC$ to denote the subset of states $\{A, C\}$. Be careful -- $\lambda$-closures are important.
6. Let \( L \) be defined as the language accepted by the finite state automaton \( A \):

Change \( A \) to a GNFA (although I don’t see the need for all those \( \phi \) transitions). Now, using the technique of replacing transition letters by regular expressions and then ripping (collapsing) states, develop the regular expression associated with \( A \) that represents the set (language) \( L \).
7. Let $L$ be defined as the language accepted by the finite state automaton $A$:

Using the technique of regular equations, develop the regular expression associated with $A$ that represents the set (language) $L$. 
8. Given a finite state automaton denoted by the transition table shown below, and assuming that 5 and 6 are final states, fill in the equivalent states matrix I have provided. Use this to create an equivalent, minimal state automaton. The start state is 1.

Don’t forget to construct and write down your new, equivalent automaton!!
9. Use the Pumping Lemma to show that the following languages are not regular:

a.) \( L = \{ a^n b^m c^t \mid n > m \text{ or } n > t, \text{ and } n, m, t \geq 0 \} \)

b.) \( L = \{ a^n b^m \mid n \leq m, \text{ and } n, m \geq 0 \} \)

c. Let \( \text{NonPrime} = \{ a^q \mid q \text{ is not a prime} \} \)
   This language is not easily shown non-regular using the Pumping Lemma. However, we already saw in class how to prove that \( \text{Prime} = \{ a^p \mid p \text{ is not a prime} \} \) is non-regular using the Pumping Lemma. Explain how you could use this latter fact to show \( \text{NonPrime} \) is non-regular.
10. DFAs accept the class of regular languages. Regular expressions denote the class of regular sets. The equivalence of these is seen by a proof that every regular set is a regular language and vice versa. The first part of this, that every regular set is a regular language, can be done by first showing that the basis regular sets \((\emptyset, \{ \lambda \}, \{ a \mid a \in \Sigma \})\) are each accepted by a DFA.

i.) Demonstrate a DFA for each of the basis regular sets.

\(\emptyset\)

\(\{ \lambda \}\)

\(\{ a \}\)

Let \(L_1\) be generated by the DFA \(A_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)\) and \(L_2\) be generated by the DFA \(A_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)\).

ii.) Present a construction from \(A_1, A_2\) that produces a DFA \(A_3\) for \(L_1 \cup L_2\).

Hint: Cross product.

iii.) What remains to be done to show that every regular set is a regular language? Don’t do the proof; just state what two steps still need to be done.