1. **[10 points] Sipser book, Exercise 1.4f:**

   Each of the following languages is the intersection of two simpler languages. In each part, construct DFAs for the simpler languages, then combine them using the construction discussed in footnote 3 (page 46) to give the state diagram of a DFA for the language given. In all parts $\Sigma = \{a, b\}$.

   f. $\{w \mid w \text{ has an odd number of } a \text{'s and ends with } b\}$

2. **[20 points] Sipser book, Problem 1.43b.**

   Let $A$ be any language. Define $DROP-OUT(A)$ to be the language containing all strings that can be obtained by removing one symbol from a string in $A$. Thus, $DROP-OUT(A) = \{xz \mid xyz \in A \text{ where } x, z \in \Sigma^*, y \in \Sigma\}$. Show that the class of regular languages is closed under the DROP-OUT operation. Give both a proof by picture and a more formal proof by construction as in Theorem 1.47.

   An informal proof will suffice. However, also illustrate using the DFA in 1.4f.

3. **[10 points] Sipser book, Exercise 1.5f:**

   Each of the following languages is the complement of a simpler language. In each part, construct a DFA for the simpler language, then use it to give the state diagram of a DFA for the language given. In all parts $\Sigma = \{a, b\}$.

   f. $\{w \mid w \text{ is any string not in } a^* \cup b^*\}$

4. **[10 points]** Give a regular expression for the complement of the language in problem 1 (1.4.f).

5. **[10 points] Sipser book, Exercise 1.19b:**

   Use the procedure described in Lemma 1.55 to convert the following regular expressions to nondeterministic finite automata.

   b. $((00)^* (11) \cup 01)^*$

6. **[15 points]** Convert the NFA in the previous problem to a DFA and minimize it.
7. [10 points] Sipser book, Problem 1.53:

1.53 Let $\Sigma = \{0, 1, +, =\}$ and

$ADD = \{x=y+z \mid x, y, z \text{ are binary integers, and } x \text{ is the sum of } y \text{ and } z\}.$

Show that $ADD$ is not regular.

8. [15 points] Sipser book, Problem 1.22:

1.22 In certain programming languages, comments appear between delimiters such as /# and #/. Let $C$ be the language of all valid delimited comment strings. A member of $C$ must begin with /# and end with #/ but have no intervening #/. For simplicity, we'll say that the comments themselves are written with only the symbols a and b; hence the alphabet of $C$ is $\Sigma = \{a, b, /, #\}$.

a. Give a DFA that recognizes $C$.

b. Give a regular expression that generates $C$. 