Problem 1

Give short answers to each of the following. Be sure to adequately explain your answers for full credit.

a. Give an example of a regular language $R$ and a non-regular language $N$, such that $R \cap N$ is a regular language.

$$R = ab^*$$
$$N = \{a^n b^n \mid n \geq 0\}$$
$$R \cap N = \{ab\}$$

b. True or false: If an NFA with $n$ states accepts no string with length less than $n$, then it must accept no strings at all.

True. Suppose that the NFA accepts some string. Then there is a path from the start state to a final state. This implies that there is a simple path (in which every state appears at most once) from the start state to the final state. (A simple path is obtained by removing loops.) A simple path can contain at most $n$ nodes and therefore at most $n-1$ transitions. Therefore, a simple path defines an accepted string with length at most $n-1$. If no such string exists in $L(M)$, then the NFA accepts no strings at all, since it must be possible to take at least one simple path to a final state.

c. Consider this statement: “If $L_1$ is a regular language and $L_1 \cup L_2$ is also regular, then $L_2$ must be regular.”

True or false: If the statement were true for all $L_1$ and $L_2$, then all languages would be regular.

True. Let $L_1 = \Sigma^*$, which is clearly regular because it is denoted by the regular expression $\Sigma^*$. Let $L_2$ be any language. By the definition of a language, $L_2 \subseteq L_1$. Therefore, $L_1 \cup L_2 = L_1$ and, hence, is regular. Since $L_2$ is any arbitrary language, we would have that all languages are regular.

The statement itself is clearly false though. Let $L_1 = \Sigma^*$, $L_2 = \{a^n b^n \mid n \geq 0\}$. Then $L_1$ is regular, and $L_1 \cup L_2 = L_1$ is regular, but we know $L_2$ is not regular, disproving the statement.
Problem 2

Let $M$ be the NFA-$\epsilon$ below. The input alphabet is $\{a, b, c\}$.

Using the method described in the book and in class, construct a deterministic finite automaton that is equivalent to $M$. Show your steps.

First compute the $\epsilon$-closure for each state of the NFA:

$E(\{p\}) = \{p\}$
$E(\{q\}) = \{p, q\}$
$E(\{r\}) = \{p, q, r\}$

Then compute the transition function $\delta$:

$\delta(\{p\}, a) = E(\{p\}) = \{p\}$
$\delta(\{p\}, b) = E(\{q\}) = \{p, q\}$
$\delta(\{p\}, c) = E(\{r\}) = \{p, q, r\}$
$\delta(\{p, q\}, a) = E(\{p, q\}) = E(\{p\}) \cup E(\{q\}) = \{p, q\}$
$\delta(\{p, q\}, b) = E(\{q, r\}) = E(\{q\}) \cup E(\{r\}) = \{p, q, r\}$
$\delta(\{p, q\}, c) = E(\{r\}) = \{p, q, r\}$
$\delta(\{p, q, r\}, a) = E(\{p, q, r\}) = E(\{p\}) \cup E(\{q\}) \cup E(\{r\}) = \{p, q, r\}$
$\delta(\{p, q, r\}, b) = E(\{q, r\}) = \{p, q, r\}$
$\delta(\{p, q, r\}, c) = E(\{p, r\}) = \{p, q, r\}$
This gives us the DFA:
Problem 3

Convert the following NFA-ε into an equivalent regular expression using the technique shown in class and on pages 72–76 of our book. For Step 1, add a new start and final state. For Steps 2–5, show the graph after removal of q₀, q₁, and q₂ (in that order). You must show each step of the conversion and label each arc in each transition graph for full credit.

Step 1 (add new start and final states)

Step 2 (after q₀ removed)

Step 3 (after q₁ removed)

Step 4 (after q₂ removed) = Resulting regular expression

$S \rightarrow a^* (ba^*)^* ba^*$
Problem 4

Using the construction method in the proof of Kleene’s theorem in your book, construct a nondeterministic finite automaton that accepts the language \(((a \cup b)(a \cup b)^*)^*\).

Note that this is a different construction for Kleene star than is used by Sipser. Either way is fine.
NFA for \((a \cup b)(a \cup b)^*a\)

NFA for \(((a \cup b)(a \cup b)^*)^*\)
Problem 5

For each language, prove whether it is a regular language. If you say it is regular, give a DFA, NFA, or regular expression for \( L \). If you say it is not regular, prove this using the Pumping Lemma for regular languages and/or closure properties.

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

   Non-regular. We show it does not satisfy the Pumping Lemma. Let \( p \) be the number from the Pumping Lemma for \( L \). Consider the string \( a^p b^p a^p b^p \). Because \( |xy| \leq p \), \( y \) must be all As. Therefore \( xy^2z \) will have at least one more \( a \) at the beginning of the string than there are \( b \)s at the end. The resulting string is not in the language, and therefore \( L \) cannot be regular.

b. \( L = \{ a^n \mid n \text{ is a multiple of 8} \} \)

   Regular. \((aaaaaaa)^*\) is a regular expression for \( L \).

c. \( L = \{ x = y + z \mid x, y, z \text{ are binary integers and } x \text{ is the sum of } y \text{ and } z \} \). The alphabet is \( \Sigma = \{ 0, 1, =, + \} \); an example string is \( 1011 = 101 + 110 \).

   Non-regular. We show it does not satisfy the Pumping Lemma. Let \( n \) be the number from the Pumping Lemma for \( L \). Consider the string \( 1^n = 1^n + 0^n \). Because \( |xy| \leq n \), \( y \) must be all \( 1 \)s. Choose \( i = 0 \). \( xy^i z = 1^n-|y|1^n + 0^n \notin L \) because \( |y| > 0 \), yielding more \( 1 \)s on the left side of the equal sign than the right. Contradiction, so \( L \) is not regular.
Problem 6

Let $\Sigma = \{a, b\}$. Give a regular expression for the set of all strings in $\Sigma^*$ with exactly one occurrence of the substring $aaa$.

$$(b \cup ab \cup aab)^*aaa(b \cup ba \cup baa)^*$$

The trick is to make sure that more than two $a$s do not have the opportunity to ever get next to each other, with the exception of the central $aaa$. 
