Problem 6

There were many correct solutions submitted to this problem. Probably the
most common (and perhaps the most intuitive) was:

$$0^*(1^*(00)^*)^*0^*$$

Problem 7

This problem was tricky, and there were several correct solutions. Some
people chose to convert an NFA by one of the two methods in the book.
This approach will work but it is very error-prone and produces expressions
that are very complicated and not intuitive. It is better to construct the
regular expression directly if you can. There were two common solutions.

1.

$$(1(10)^*0)^*(\epsilon + 1(10)^*(\epsilon + 1))$$

Let $n$ be the number of 1’s minus the number of 0’s as we advance
through the string. The requirement is that $n \in \{0, 1, 2\}$ at all times.
Find the last point in the string where $n = 0$ and split the string about
that point. The first part of the expression allows $n$ to bounce between
0 and 2 arbitrarily then land on 0. It may do this zero or more times:
$(1(10)^*0)^*$. For the tail of the string, $n$ may either stay at 0 ($\epsilon$) or
increase. To increase, $n$ first goes to 1, and may return an arbitrary
number of times: $1(10)^*$. Finally, $n$ either stays at 1 or ends at two:
$(\epsilon + 1)$.

2.

$$1(10 + 01)^*(\epsilon + 0 + 1) + \epsilon$$

The logic is similar to that above, but results in a simpler expression.
In this case, we split about the last point in the string where $n$ equals
1. The string must start with 1, followed by a zero or more occurrences
of $(10 + 01)$. Finally, $n$ will either stay at 1, decrease to 0, or advance
to 2: $(\epsilon + 0 + 1)$. We must also allow $\epsilon$. 