Q1: Given the alphabet $\Sigma = \{0, 1, 2\}$, let $L$ be the language denoted by the regular expression $(0|1)^* \ 2 \ (0|1)^*$. I attempted to work my way backwards from NFA/DFA to a regex. And, failed. A FSM for this expression would look like the following:

![ FSM Diagram ]

Q2: $L = \{ a^n b^m \mid n \geq 1, m \geq 1, n \cdot m \geq 4 \}$

The minimum choices for $n$ and $m$ are: $1 \ n \ & \ 4 \ m$, $4n \ & \ 1m$, $2n \ & \ 2m$. So $a b b b b \mid (a a a \ a) \mid (a a b)$ /* spaces for clarity */

But we’re not done yet, $n$ and $m$ can be larger than the values above so each set of chars could be followed by zero or more of the same character. So $(a a^* b b b \ b^*) \mid (a a a^* \ b \ b^*) \mid (a a^* b \ b^*)$

$L = $ all strings having at least two occurrences of substring $bb$. Note that $bb$ fits the criterion.

So this is telling us that $bb$ is part of the solution! We can have any number of $a$’s or $b$’s in the beginning and end of the string and $bb$ or two “other” occurrences of $bb$.

(a|b)* (bb | (bb XXX bb) (a|b)* and xxx is any number of $a$’s or $b$’s in between!

(a|b)* (bbb | (bb (a|b)* bb) (a|b)*

$L = \{ w \mid w=y b b a^n, \ n \geq 1, \ y \in \Sigma^* \}$

So this is telling us that $y$ is zero or more occurrences of $a$ or $b$ and $a^n$ is the same as $aa^*$ (a|b)*bbaa*
Q3: “all strings that do not have a pair of ones separated by an odd number of symbols.”

Create the reverse of this, find the FSM and then reverse the states on the FSM.

Q4: $a^m a^*$ is the form of the regex. So, you would apply the $\{a\}$ definition $m$ times and the $*$ definition. It is alright if the value $m$ is unknown, it is just a constant value.

Two or more consecutive a’s is $(aa) (aa)^*$ and you would begin and follow this regex with $(a|b)^*$. Use the $\{a\}$ definition and the $*$ definition again along with the union definition for $a|b$. 
Q5: Use the state elimination method to convert the following automaton to a regular expression. Show all your steps, and be sure to note the final regular expression that results.

1. Modify to create a unique start and end state

2. Eliminate state 1: path from s to 2 is \( b*a \); path from 2 to 2 is \( ab*a \)

3. Eliminate state 2: concatenate path from s to 2 with \( (ab*a)^* \) (loop label) and path from 2 to 3

4. Eliminate state 3: concatenate path from s to 3 with \( (a+b)^* \)