Problem 1

Let $C_n = \{ x \mid x \text{ is a binary number that is a multiple of } n \}$

Show that $C_n$ is regular for any $n$.

Solution: We will assume that the input string, $x$, begins with the most significant bit.

$\Sigma = \{0, 1\}$

For a given $n$, our FSA will have $n$ states, each representing a possible remainder when $n$ is divided into $x$. The start state, $q_0$, represents a remainder of 0, and is thus also the only accept state.

The transition function is defined as $\delta(q_i, x_k) = q_j$ where $j = (2i + x_k) \mod n$. The intuition here is that doubling the current remainder, $i$, corresponds to the left shift that occurs when a new digit is found in the input string. Consider the binary representation for the number 24, 11000. When our FSA reads the first two digits of this input, 11, it looks like the number 3. When the next digit is read, 11 becomes 110, and 3 becomes 6. When we read the next 0 we double again to 12, and then once more to 24 when we read the final digit.

Examples:
$8 = 1000$, $24 = 11000$, $40 = 101000$
Problem 2

The following language is the intersection of two simpler languages. Construct FSAs for the simpler languages and then use these FSAs to construct an FSA for their intersection.

\[ L = \{ w \mid w \text{ has an odd number of 0's and at least two 1's } \} \]