1. ***Regular or not?*** *Prove or disprove* that each of the languages below is regular (or not). Let Σ^+ denote the set of all *nonempty* strings over alphabet Σ; in other words, $\Sigma^+ = \Sigma \cdot \Sigma^*$. Denote $n(w)$ the integer corresponding to the binary string w.

   (a) $\{ \mathtt{3}x\mathtt{=}y : x, y \in \{\mathtt{0}, \mathtt{1}\}^*, n(y) = 3n(x) \}$

   (b) $\left\{ \substack{\mathtt{3x} \\ \mathtt{=y}} : \substack{\mathtt{x} \\ \mathtt{y}} \in \left\{ \substack{\mathtt{0} \\ \mathtt{0}}, \substack{\mathtt{0} \\ \mathtt{1}}, \substack{\mathtt{1} \\ \mathtt{0}}, \substack{\mathtt{1} \\ \mathtt{1}} \right\}^*, n(y) = 3n(x) \right\}$

   (c) $\{ wxw^R : w, x \in \Sigma^+ \}$

   (d) $\{ ww^Rx : w, x \in \Sigma^+ \}$

   [Hint: To prove that a language L is regular, construct an NFA that recognizes L; to disprove that L is regular, construct a fooling set for L and argue that the construction is correct.]

2. ***Telling DFAs apart.***

   Let M_1 and M_2 be two DFAs, each with exactly n states. Assume that the languages associated with the two machines are different (that is, $L(M_1) \neq L(M_2)$), there is always some string in the symmetric difference of the two languages.

   Prove that there is a string w of length polynomial in n in the symmetric difference of $L(M_1)$ and $L(M_2)$. What is the best upper bound you can get on the length of w?

★3. ***Telling strings apart.***

   Let w_1 and w_2 be two strings over binary alphabet $\Sigma = \{\mathtt{0}, \mathtt{1}\}$, each of length exactly n. Assume that the two strings are different, there is always some DFA that accepts exactly one of the two strings.

   Prove that there is a DFA M of size $o(n)$ such that exactly one of w_1 and w_2 is in $L(M)$. What is the best upper bound you can get on the size of M?