1. **Self-denial.** Prove that there is no Turing machine that computes the language

$$\textsc{NotAcceptItself} := \Big\{ \langle M \rangle \mid M \text{ is a Turing machine which does not accept } \langle M \rangle \Big\}.$$

2. **Self-acceptance.** Prove that there is no Turing machine that computes the language

$$\textsc{AcceptItself} := \Big\{ \langle M \rangle \mid M \text{ is a Turing machine which accepts } \langle M \rangle \Big\}.$$

   Can you design a Turing machine that *accepts* the language AcceptItself? In other words, construct a machine that accepts the encoding of all the self-accepting machines (but might loop forever on other inputs).

3. **Accepting a string?** Prove that the language

$$\textsc{Accept} := \Big\{ \langle M, w \rangle \mid M \text{ is a Turing machine and } w \in L(M) \Big\}$$

   is incomputable.

Takeaway. It is not always safe to write "if some program exhibits a specific behavior, then ..." in your pseudocode, because you might not have a Turing machine checking that behavior.

Wait. Assuming we are Turing machines, trying to write a proof to the above practice problems. By the Curry–Howard correspondence, proofs are programs. As we write the solution, we are writing a program checking the correctness of the problem statement. But didn't *we* write the sentence "if machine M rejects input $\langle M \rangle$ ..." several times?