Proving that a language L is undecidable by reduction requires several steps.

1. Choose a language L' that you already know is undecidable (because we told you so in class). The simplest choice is usually the standard halting language

$$\text{HALT} := \{\langle M, w\rangle \mid M \text{ halts on } w\}$$

2. Describe an algorithm that decides L', using an algorithm that decides L as a black box. Typically your reduction will have the following form:

   Given an arbitrary string x, construct a special string y,
   such that $y \in L$ if and only if $x \in L'$.

   In particular, if $L = \text{HALT}$, your reduction will have the following form:

   Given the encoding $\langle M, w\rangle$ of a Turing machine M and a string w,
   construct a special string y, such that
   $y \in L$ if and only if M halts on input w.

3. Prove that your algorithm is correct. This proof almost always requires two separate steps:
   (a) Prove that if $x \in L'$ then $y \in L$.
   (b) Prove that if $x \notin L'$ then $y \notin L$.

Very important: Name every object in your proof, and *always* refer to objects by their names. Never refer to "the Turing machine" or "the algorithm" or "the input string" or (gods forbid) "it" or "this", even in casual conversation, even if you're "just" explaining your intuition, even when you're just *thinking* about the reduction.

Prove that the following languages are undecidable.

1 $\text{ACCEPTILLINI} := \{\langle M\rangle \mid M \text{ accepts the string } \mathit{ILLINI}\}$

2 $\text{ACCEPTTHREE} := \{\langle M\rangle \mid M \text{ accepts exactly three strings}\}$

3 $\text{ACCEPTPALINDROME} := \{\langle M\rangle \mid M \text{ accepts at least one palindrome}\}$