1. **To decide or not to decide**
   For each of the following languages, give a proof that it is undecidable or describe an algorithm to decide it. (You may assume that all the languages are over the alphabet \{0, 1\} and all the Turing machines have \{0, 1\} as their input alphabet.)
   
   (a) \( L_1 = \{\langle M \rangle | M \text{ is a Turing machine that rejects all inputs of even length} \} \).
   (b) \( L_2 = \{\langle M \rangle | M \text{ is a Turing machine that halts on an empty input} \} \).
   (c) \( L_3 = \{\langle M \rangle | \text{there is some input } x \in \{0, 1\}^* \text{ such that } M \text{ accepts } x \text{ in less than 100 steps} \} \).

2. **More on halting**
   \( H_{TM}^{1/2} = \{\langle (M), x, y \rangle | M \text{ halts on } x \text{ but not on } y \} \).
   
   (a) Show that \( \overline{H_{TM}} \leq_m H_{TM}^{1/2} \).
   (b) Use part (a) to show that neither \( H_{TM}^{1/2} \) nor \( \overline{H_{TM}^{1/2}} \) is recognizable.

3. **Erasers are hard to find**
   Consider the problem of testing whether a given single-tape Turing machine ever writes a blank symbol over a nonblank symbol during the course of its computation on any string input to it i.e. does it ever erase anything on the tape. Formulate this problem as a language, and show that it is undecidable.

4. **Consent in its refusal**
   Let \( L \) be a Turing-recognizable language and let \( \overline{L} \) be non-Turing recognizable. We defined the following language in the previous discussion:
   \[ L' = \{0w | w \in L\} \cup \{1w | w \notin L\} \]
   
   (a) Show that \( L' \leq_m \overline{L} \).
   (b) Show that for any undecidable language having the property that \( B \leq_m \overline{B} \), neither \( B \) nor \( \overline{B} \) is Turing-recognizable.