1. Recall that $L_u = \{\langle M\rangle\#w \mid M(w) \text{ accepts}\}$ is not decidable. In class we showed reductions from L_u to various languages L to show that L was undecidable. One of the languages shown undecidable was the "halting language" $L_{halt} = \{\langle M\rangle \mid M \text{ halts on blank input}\}$.

   In order to show a language L is undecidable, it is often just as easy, or even easier, to show a reduction from L_{halt} to L.

   Example: We show that $L_{1^*} = \{\langle M\rangle \mid L(M) = 1^*\}$ is not decidable by showing $L_{halt} \leq L_{1^*}$

   Reduction: We show how a decider for L_{1^*} could be used to decide L_{halt}. The reduction takes an instance of L_{halt} (i.e., a TM M that we'd like to know if it halts on blank input) and outputs an instance of M' for L_{1^*}.

   We need the following to be true: M halts on blank input if and only if $L(M') = 1^*$.

   Here is the (partial) code for M'. Fill in the blank, and then answer parts (a), (b), and (c).

```
M'(x:  string)
    run M until, if ever, it halts
    if M halted, then accept x iff x__________
```

   (a) If M doesn't halt when run on blank input, what is $L(M')$? ______________

   (b) If M halts when run on blank input, what is $L(M')$? ______________

   (c) Briefly argue that no decider for L_{1^*} can exist.

2. Let $L_{even} = \{\langle M\rangle \mid L(M) = \{w : |w| \text{ is even}\}\ \}$.
   Prove that L_{even} is not decidable by showing that $L_{halt} \leq L_{even}$.

3. Let $L_h = \{\langle M\rangle\#w \mid M(w) \text{ halts}\}$. Show how to use a decider for L_h to build a decider for L_u.

Prove that the following languages are undecidable *using Rice's Theorem*:

Rice's Theorem. The language $\{\langle M\rangle \mid L(M) \text{ satisfies property } P\}$ is undecidable for any property P that is satisfied by at least one, and not all, recursively enumerable languages.

1. AcceptRegular $:= \left\{\langle M\rangle \;\middle|\; L(M) \text{ is regular}\right\}$

   Example solution: We need to show that the property of being regular is satisfied by at least one, but not all, r.e. languages, and then Rice's theorem will be applicable. Clearly for any regular language R there is a TM M such that $L(M) = R$, so the property of being regular holds for at least one r.e. language (namely, $L(M)$). Just as clearly, there is a TM M' that accepts $\{0^n1^n \mid n \geq 0\}$, a nonregular language, so $L(M')$ is an r.e. language that is not regular. Since at least one, but not all, r.e. languages satisfy the property of being regular, we can apply Rice's theorem and conclude that AcceptRegular is not decidable.

2. AcceptIllini $:= \left\{\langle M\rangle \;\middle|\; \text{ILLINI} \in \text{L(M)}\right\}$

3. AcceptPalindrome $:= \left\{\langle M\rangle \;\middle|\; M \text{ accepts at least one palindrome}\right\}$

4. AcceptThree $:= \left\{\langle M\rangle \;\middle|\; M \text{ accepts exactly three strings}\right\}$

5. AcceptUndecidable $:= \left\{\langle M\rangle \;\middle|\; L(M) \text{ is undecidable }\right\}$