1) Give the sequence of configurations that the Turing machine \( M_2 \) in chapter 3 of the text (and shown in class) goes through while reading in the three following strings:

   a) 000 
   b) 0000 
   c) 000000

2) What is the flaw in the following proof to show that if a language \( L \) is Turing recognizable, then we can create an enumerator to enumerate it? Remember that the sequence \( s_1, s_2, s_3, \ldots \), is an enumerated list of all strings in \( \Sigma^* \), from shortest to longest, in lexicographical order for all strings of the same length.

Let \( M \) be a Turing machine that recognizes \( L \).

We can create an enumerator \( E \) for \( L \) as follows:

1. Repeat the following for \( i = 1, 2, 3, \ldots \)
2. Run \( M \) on \( s_i \).
3. If it accepts, print out \( s_i \).

3) A Turing machines with a stay option is similar to an ordinary Turning machine except that the transition function has the form:

\[ \delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, S\} \]

If \( \delta(q, a) = (r, b, S) \), when the machine is in state \( q \) reading an \( a \), the machine’s head stays exactly where it is. Show that Turing machines with a stay option recognize the class of Turing-recognizable languages.

4) Show that Turing-decidable languages are closed under the following operations:

   a) union 
   b) intersection 
   c) complementation 
   d) concatenation

5) Show that Turing-recognizable languages are closed under union and intersection. Why is it necessary to be more clever with these two proofs than those in question number 3?