1. Exercise 9.2.4, pg. 391.
   Let \( L_1, L_2, \ldots, L_k \) be a collection of languages over alphabet \( \Sigma \) such that:
   - For all \( i \neq j \), \( L_i \cap L_j = \emptyset \); i.e., no string is in two of the languages.
   - \( L_1 \cup L_2 \cup \ldots \cup L_k = \Sigma^* \); i.e., every string is in one of the languages.
   - Each of the languages \( L_i \), for \( i = 1, 2, \ldots, k \) is recursively enumerable.
   Prove that each of the languages is therefore recursive.

2. Exercise 9.2.6(b,c,d), pg. 392.
   Determine whether the recursive and/or the recursively enumerable languages are closed under the following operations. You may give informal, but clear, constructions to show closure.
   - Intersection.
   - Concatenation.
   - Kleene closure (star).

3. The Big Computer Corp. has decided to bolster its sagging market share by manufacturing a high-tech version of the Turing machine, called BWTM, that is equipped with bells and whistles. The BWTM is basically the same as your ordinary Turing machine, except that each state of the machine is labeled either a “bell-state” or a “whistle-state”. Whenever the BWTM enters a new state, it either rings the bell or blows the whistle, depending on which state it has just entered. Prove that it is undecidable whether a given BWTM \( M \), on given input \( w \), ever blows the whistle.

4. Exercise 9.3.3, pg. 400.
   Show that the language of codes for TM’s \( M \) that, when started with a blank tape, eventually write a 1 somewhere on the tape is undecidable.

5. Exercise 9.3.6(b), pg. 400
   Let \( L \) be the set of codes for TM’s that never make a move left on any input. Show that \( L \) is decidable.