Hints for #2 (Exercise 9.3.4)

1. To show that a problem is RE, construct a recognizer for it.

2. Suggested approach for part b): Assume that the problem is RE. Then show that the halting problem would be decidable.

The proof would go much like a proof for Rice’s theorem. Given \( M \) and \( x \), and you want to decide whether \( M \) halts on input \( x \), you would construct a second machine \( M' \) that translates the halting (or non-halting) of \( M \) into the property at hand, whether \( L(M') \) is infinite or not.

Note however that you can only choose the translation in one way: \( M \) does not halt on \( x \) \( \iff \) \( L(M') \) is infinite. The other choice, \( M \) does not halt on \( x \) \( \iff \) \( L(M') \) is finite, does not work (why?).

Hint on how to construct such an \( M' \): Instead of simulating \( M \) on \( x \) indefinitely, we would need to stop at some point and accept some strings. Where \( M' \) decides to stop the simulation of \( M \) can depend on it’s own input, say \( w \), and then \( M' \) can decide whether to accept \( w \) based on the results of the simulation. The goal is of course if \( M \) never halts no matter how long it’s simulated, then \( M' \) would accept infinitely many strings, and if \( M \) does halt if simulated long enough, then \( M' \) would accept a finite amount of strings (even if it’s possibly a very large number).