Exercise: (Section 3.3.2 from textbook) 
Consider the following languages over $\Sigma = \{0, 1\}$

\[ L_e = \{ \mathcal{E}(M) \mid \mathcal{L}(M) = \emptyset \} \]
\[ L_{\neg e} = \{ \mathcal{E}(M) \mid \mathcal{L}(M) \neq \emptyset \} \]

Hence: $L_e$ is the set of all strings that encode TMs $M$ that accept the empty language.
$L_{\neg e}$ is the complement of $L_e$.

Claim 4: $L_{\neg e}$ is R.E.

Proof: construct NTM $N$ for $L_{\neg e}$.

(And then convert $N$ to an ordinary TM.)

$N$ works as follows: on input $\mathcal{E}(M)$

1) guess a string $w \in \Sigma^*$
2) simulate $M$ on $w$ (like a UTM)
3) accept $\mathcal{E}(M)$ if $M$ accepts $w$.

\[ \mathcal{E}(M) \xrightarrow{\text{guessed } w} U \xrightarrow{\text{yes}} \text{yes} \]

We have $\mathcal{E}(M) \in L_{\neg e} \iff \exists w \text{ s.t. } \langle M, w \rangle \in \mathcal{L}(U)$
\[ \iff \exists w \text{ s.t. } w \in \mathcal{L}(M) \]
\[ \iff \mathcal{E}(M) \in L_{\neg e} \]
Claim 2: \( L_{ne} \) is non-recursively

Proof: by reduction from \( L_{M_0} \) to \( L_{ne} \)

Reduction \( R \) is a function computable by a halting T.M. 

Input: instance \( \langle M, w \rangle \) of \( L_{M_0} \)

Output: instance \( \varepsilon(M') \) of \( L_{ne} \)

End set: \( \langle M, w \rangle \in L_{M_0} \iff \varepsilon(M') \in L_{ne} \)

Description of \( M' \):

- \( M' \) ignores completely its own input string \( X \)
- instead, it replaces its input by the string \( \langle M, w \rangle \) and runs \( M \) on \( w \) (see (*) below)
- if \( M \) accepts \( w \), then \( M' \) accepts \( X \)
- if \( M \) never halts on \( w \) or rejects \( w \), then \( M' \) also never halts on \( w \) or rejects \( X \)

Note:
- if \( w \in L(M) \Rightarrow L(M') = \Sigma^* \)
- if \( w \notin L(M) \Rightarrow L(M') = \emptyset \)

hence \( \langle M, w \rangle \in L_{M_0} \iff \varepsilon(M') \in L_{ne} \)

We can construct a halting T.M. \( M_R \) that, given \( \langle M, w \rangle \) as input, constructs \( \varepsilon(M') \) for an \( M' \) that behaves as above. q.e.d.

(*) \( M' \) has the following form:

\[
\begin{array}{c}
\text{Input } X \\
\text{writes } w \text{ on the tape} \\
\text{go to the beginning of } w \\
\text{runs } M \text{ on } w
\end{array}
\]

To sum up, we have that \( L_{ne} \) is RE but non-recursively.
Hence \( L_{ne} \) must be non-RE.
Exercise 3.2.1

The halting problem, \( \mathcal{H} \), is the set \( \langle M, w \rangle \) s.t.

\( M \) halts on \( w \) (with or without accepting) is \( \text{R.E.} \)

but not \( \text{recursive} \).

To show \( \text{R.E.} \), we construct a T.M. \( H \) s.t.

\( L(H) := L_H = \{ \langle M, w \rangle \mid M \text{ halts on } w \} \)

\[
\begin{array}{ccc}
\langle M, w \rangle & \rightarrow & H \\
& \uparrow & \uparrow \\
& \text{halts and says } & \text{yes} \\
& \text{no} & \\
\end{array}
\]

To show that \( L_H \) is not \( \text{recursive} \), we assume by contradiction it is \( \text{R.E.} \), and derive that \( L_H \) is \( \text{recursive} \).

By contradiction, let \( H \) be an algorithm for \( L_H \) and \( U \) a procedure for \( L_H \)

\[
\begin{array}{ccc}
\langle M, w \rangle & \rightarrow & H \\
& \uparrow & \uparrow \\
& \text{yes} & \text{triggers} \\
& \text{no} \\
\end{array}
\]

\( A_u \)

\( A_u \) would be an algorithm for \( L_H \).

\( \text{Contradiction} \)
Let $L$ be R.E. and $\overline{L}$ be non-R.E.

Consider $L' = \{0w \mid w \in L\} \cup \{1w \mid w \not\in L\}$.

What do we know about $L'$ and $\overline{L'}$?

We show that $L'$ is non-R.E.

Suppose by contradiction that we have a procedure $M_L$ for $L'$.

Then we can construct a procedure $M_{\overline{L}}$ for $\overline{L}$ as follows:

- on input $w$, $M_{\overline{L}}$ changes the input to $1w$ and simulates $M_L$.

  - if $M_L$ accepts $1w$, then $w \in \overline{L}$, and $M_{\overline{L}}$ accepts
  - if $M_L$ does not terminate or terminates and answers no, then $w \not\in \overline{L}$, and $M_{\overline{L}}$ does not terminate or terminates and answers no.

$\Rightarrow M_{\overline{L}}$ would accept exactly $\overline{L}$. Contradiction.

$\overline{L}' = \{0w \mid w \in L\} \cup \{1w \mid w \not\in L\} \cup \{\epsilon\}$.

Reasoning as for $L'$, we get that $\overline{L}'$ is non-R.E.
$\text{Fl}$, the complement of the halting problem, i.e.,
the set of pairs $\langle M, w \rangle$ such that $M$ on input $w$
does not halt, is non-$R.E.$.

**Proof:** By reduction from $\text{Fin}$, which is non-$R.E.$.

Idea: we show how to convert any TM $M$ into another
TM $M_\perp$ s.t. $M_\perp$ halts on $w$ iff $M$ accepts $w$.

**Construction:**

1) Ensure that $M_\perp$ does not halt unless $M$ accepts.
   - add to the states of $M$ a new loop state $q_0$, with
     $\delta(q_0, x) = (q_0, x, r)$ for all $x \in \Gamma$
   - for each $\delta(q, y) \text{ that is undefined and } q \in F$,
     add $\delta(q, y) = (q_0, y, r)$

2) Ensure that, if $M$ accepts, then $M_\perp$ halts
   - make $\delta(q, x)$ undefined for all $q \in F$ and $x \in \Gamma$

3) The other moves of $M_\perp$ are as those of $M$.

$q.e.d.$