1. Choosing from among (REC) recursive, (RE) re non-recursive, (coRE) co-re non-recursive, (NR) non-re/non-co-re, categorize each of the sets in a) through d). Justify your answer by showing some minimal quantification of some known recursive predicate.

   a.) \{ f | \text{domain}(f) \text{ is finite} \}  \quad \text{NR}

   Justification: \exists x \forall y \forall t \lnot \text{STP}(y, f, t)

   b.) \{ f | \text{domain}(f) \text{ is empty} \}  \quad \text{CO}

   Justification: \forall x \forall t \lnot \text{STP}(x, f, t)

   c.) \{ \langle f, x \rangle | f(x) \text{ converges in at most 20 steps} \}  \quad \text{REC}

   Justification: \text{STP}(x, f, 20)

   d.) \{ f | \text{domain}(f) \text{ converges in at most 20 steps for some input } x \}  \quad \text{RE}

   Justification: \exists x \exists t \text{STP}(x, f, t)

2. Let set A be recursive, B be re non-recursive and C be non-re. Choosing from among (REC) recursive, (RE) re non-recursive, (NR) non-re, categorize the set D in each of a) through d) by listing all possible categories. No justification is required.

   a.) D = \lnot C \quad \text{RE, NR}

   b.) D \subseteq A \cup C \quad \text{REC, RE, NR}

   c.) D = \lnot B \quad \text{NR}

   d.) D = B - A \quad \text{REC, RE}

3. Prove that the Halting Problem (the set \text{HALT} = K_0 = L_\emptyset) is not recursive (decidable) within any formal model of computation. (Hint: A diagonalization proof is required here.)

   \textbf{Look at notes.}

4. Using reduction from the known undecidable HasZero, \text{HZ} = \{ f | \exists x f(x) = 0 \}, show the non-recursiveness (undecidability) of the problem to decide if an arbitrary primitive recursive function g has the property IsZero, \text{Z} = \{ f | \forall x f(x) = 0 \}. Hint: there is a very simple construction that uses \text{STP} to do this. Just giving that construction is not sufficient; you must also explain why it satisfies the desired properties of the reduction.

   \text{HZ} = \{ f | \exists x \exists [ \text{STP}(x, f, t) \& \text{VALUE}(x, f, t) = 0] \}

   Let \( f \) be the index of an arbitrary effective procedure.

   Define \( g_f(y) = 1 - \exists x \exists [ \text{STP}(x, f, t) \& \text{VALUE}(x, f, t) = 0] \)

   If \( \exists x f(x) = 0 \), we will find the x and the run-time t, and so we will return 0 \((1 - 1)\)

   If \( \forall x f(x) \neq 0 \), then we will diverge in the search process and never return a value.

   Thus, \( f \in \text{HZ} \iff g_f \in \text{Z} \).
5. Define \textit{RANGE\_ALL} = \{ f \mid \text{range}(f) = \emptyset \}.

a.) Show some minimal quantification of some known recursive predicate that provides an upper bound for the complexity of this set. (Hint: Look at c.) and d.) to get a clue as to what this must be.)
\[ \forall x \exists <y,t>[\text{STP}(y,f,t) \& \text{Value}(y,f,t)=x] \]

b.) Use Rice’s Theorem to prove that \textit{RANGE\_ALL} is undecidable.
This is non-trivial as I(x) = x \in \textit{RANGE\_ALL} and C_0(x) = 0 \notin \textit{RANGE\_ALL}
Let f,g be such that \( \forall x \varphi_t(x) = \varphi_g(x) \).
\( f \in \textit{RANGE\_ALL} \iff \text{range}(f) = \emptyset \)
\( \iff \text{range}(g) = \emptyset \quad \text{since g outputs the same value as f for any input} \)
\( \iff g \in \textit{RANGE\_ALL} \)
Since the property is non-trivial and is an I/O property, Rice’s Theorem says it is undecidable.

c.) Show that \textit{TOTAL} \leq_m \textit{RANGE\_ALL}, where \textit{TOTAL} = \{ f \mid \forall y \varphi_f(y) \downarrow \}.
Let f be the index of an arbitrary effective procedure \( \varphi_f \). Define g such that g(f), denoted g_f, is the index of the function \( \varphi_g \) defined by \( \forall x \varphi_g(x) = \varphi_f(x) - \varphi_f(x) + x \).
\( f \in \textit{TOTAL} \iff \forall x \varphi_f(x) \downarrow \iff \forall x \varphi_g(x) = x \iff \forall x \delta(x) \in \text{range}(g_f) \Rightarrow g_f \in \textit{RANGE\_ALL} \)
\( f \notin \textit{TOTAL} \iff \exists x \varphi_f(x) \uparrow \iff \exists x \varphi_g(x) \uparrow \Rightarrow \exists x \in \text{range}(g_f) \Rightarrow g_f \notin \textit{RANGE\_ALL} \)
This shows that \textit{TOTAL} \leq_m \textit{RANGE\_ALL}, as was desired.

d.) Show that \textit{RANGE\_ALL} \leq_m \textit{TOTAL}.
Let f be the index of an arbitrary effective procedure \( \varphi_f \). Define g such that g(f), denoted g_f, is the index of the function \( \varphi_g \) defined by \( \forall x \varphi_g(x) = \exists y \varphi_f(y) = x \).
\( f \in \textit{RANGE\_ALL} \iff \forall x \exists y \varphi_f(y) = x \iff \forall x \varphi_g(x) \downarrow \iff g_f \in \textit{TOTAL} \)
This shows that \textit{RANGE\_ALL} \leq_m \textit{TOTAL}, as was desired.

e.) From a.) through d.) what can you conclude about the complexity of \textit{RANGE\_ALL}?
\( a \) shows that \textit{RANGE\_ALL} is no more complex than others that must use the alternating qualifiers \( \forall \exists \). \( b \) shows the problem is non-recursive. \( c \) and \( d \) combine to show that the problem is in fact of equal complexity with the non-re problem \textit{TOTAL}, so the result in \( a \) was optimal.
6. This is a simple question concerning Rice’s Theorem.

a.) State the strong form of Rice’s Theorem. Be sure to cover all conditions for it to apply.

Let P be a property of indices of partial recursive function such that the set
S_P = { f | f has property P } has the following two restrictions
(1) S_P is non-trivial. This means that SP is neither empty nor is it the set of all indices.
(2) P is an I/O behavior. That is, if f and g are two partial recursive functions whose I/O
behaviors are indistinguishable, \( \forall x f(x) = g(x) \), then either both of f and g have property P or
neither has property P.

Then P is undecidable.

b.) Describe a set of partial recursive functions whose membership cannot be shown undecidable
through Rice’s Theorem. What condition is violated by your example?

There are many possibilities here. For example \{ f | \( \exists x \sim STP(x,f,x) \} \) is not an I/O property and
\{ f | \( \exists x f(x) \neq f(x) \) \} is trivial (empty).

7. Using the definition that S is recursively enumerable iff S is either empty or the range of some
algorithm \( f_S \) (total recursive function), prove that if both S and its complement \( \sim S \) are recursively
enumerable then S is decidable. To get full credit, you must show the characteristic function for S,
\( \chi_S \), in all cases. Be careful to handle the extreme cases (there are two of them). Hint: This is not an
empty suggestion.

Let S = \( \phi \) then \( \sim S = \mathbb{N} \). Both are re and \( \forall x \chi_S(x) = 0 \) is S’s characteristic function.

Let S = \( \mathbb{N} \) then \( \sim S = \phi \). Both are re and \( \forall x \chi_S(x) = 1 \) is S’s characteristic function.

Assume then that S \( \neq \phi \) and S \( \neq \mathbb{N} \) then each of S and \( \sim S \) is enumerated by some total recursive
function. Let S be enumerated by \( f_S \) and \( \sim S \) by \( f_{\sim S} \). Define

\( \chi_S(x) = f_S( \mu y [f_S(y)==x \parallel f_{\sim S}(y)==x] ) == x. \)

Note that x must be in the range of one and only one of \( f_S \) or \( f_{\sim S} \). Thus,
\( \exists y f_S(y) == x \) or \( \exists y f_{\sim S}(y) == x. \)

The \( \mu y \) operator finds the smallest such y and the predicate
\( f_S( \mu y [f_S(y)==x \parallel f_{\sim S}(y)==x] ) == x \) checks that x is in the range of \( f_S \).

If it is, then \( \chi_S(x) = 1 \) else \( \chi_S(x) = 0 \), as desired.