Sample Question#1. Part a

1. Prove that the following are equivalent

a) S is an infinite recursive (decidable) set.

b) S is the range of a monotonically increasing total recursive function.
Note: f is monotonically increasing means that $\forall$x f(x+1) > f(x).

a) Implies b)

Let x $\in$ S $\Leftrightarrow$ χ_S(x)

Define $f_R(0) = \mu$ x χ_S(x); $f_R(y+1) = \mu$ x [χ_S(x) && (x > f_R (y))]

Clearly, since S is non-empty, it has a least one value and so there exist a smallest value such that χ_S(x); we will enumerate this as $f_R(0) = \mu$ x χ_S(x).

Assume we have enumerated the y-th value in S as f_R (y). Since S is infinite, there will be values in S greater than f_R (y) and our search μ x [χ_S(x) && (x > f_R (y)) will find the next largest value for which χ_S(x). Thus, inductively, we will enumerate the elements of S in increasing order, as desired.

Sample Question#1 Part b

1. Prove that the following are equivalent

a) S is an infinite recursive (decidable) set.

b) S is the range of a monotonically increasing total recursive function.
Note: f is monotonically increasing means that $\forall x\ f(x+1) > f(x)$.

b) Implies a)

Let S be enumerated by the monotonically increasing algorithm f_S. Define χ_S by

$\chi_S(x) = (f_S\ ((\mu\ z\ [f_S\ (z) \geq x]) == x)$

Clearly, if x is enumerated, it must appear before any values greater than it are enumerated and consequently this is a bounded search to find the first element listed that is at least as large as x. If this element is x, then x is in S, else it is not. The fact that f_S is monotonically increasing makes S infinite. The fact that it has a characteristic function makes it decidable.

Sample Question#2

2. Let A and B be re sets. For each of the following, either prove that the set is re, or give a counterexample that results in some known non-re set.

Let A be semi decided by f_A and B by f_B

a) $A \cup B$: must be re as it is semi-decided by

$f_{A \cup B}(x) = \exists t\ [stp(f_A, x, t) \mid\mid stp(f_B, x, t)\ ]$

b) $A \cap B$: must be re as it is semi-decided by

$f_{A \cap B}(x) = \exists t\ [stp(f_A, x, t)\ \&\&\ stp(f_B, x, t)\ ]$

c) ~A: can be non-re. If ~A is always re, then all re are recursive as any set that is re and whose complement is re is decidable. However, A = K is a non-rec, re set and so ~A is not re.

Sample Question#3

3. Present a demonstration that the *even* function is primitive recursive.
   even(x) = 1 if x is even
   even(x) = 0 if x is odd
   You may assume only that the base functions are prf and that prf's are closed under a finite number of applications of composition and primitive recursion.

even(0) = 1; even(y+1) = !even(y) = 1-even(y)

Sample Question#4

4. Given that the predicate **STP** and the function **VALUE** are prf’s, show that we can semi-decide
{ f | φ_f evaluates to 0 for some input}
This can be shown re by the predicate
{f | ∃<x,t> [stp(f,x,t) && value(f,x,t) = 0] }

Sample Question#5

5. Let **S** be an re (recursively enumerable), non-recursive set, and **T** be re, non-empty, possibly recursive set. Let

E = { z | z = x + y, where x $\in$ S and y $\in$ T }.

(a) Can **E** be non re? **No as we can let S and T be semi-decided by f_S and f_T, resp., E is then semi-dec. by**
f_E (z) = $\exists$<x,y,t> [stp(f_S, x, t) && stp(f_T, y, t) &&
(z = value(f_S, x, t) + value(f_T, y, t))]

(b) Can **E** be re non-recursive? **Yes, just let T = {0}, then E = S which is known to be re, non-rec.**

(c) Can **E** be recursive? **Yes, let T = $\aleph$, then E = { x | x ≥ min (S) } which is a co-finite set and hence rec.**

Sample Question#6

6. Assuming **TOTAL** is undecidable, use reduction to show the undecidability of
Incr = { f | $\forall x\ \varphi_f(x+1) > \varphi_f(x)$ }
Let f be arb.
Define $G_f(x) = \varphi_f(x) - \varphi_f(x) + x$
$f \in$ TOTAL iff $\forall x \varphi_f(x)\downarrow$ iff $\forall x\ G_f(x)\downarrow$ iff
$\forall x\ \varphi_f(x) - \varphi_f(x) + x = x$ implies $G_f \in$ Incr
$f \notin$ TOTAL iff $\exists x \varphi_f(x)\uparrow$ iff $\exists x\ G_f(x)\uparrow$ iff
$\exists x\ (\varphi_f(x) - \varphi_f(x) + x)\uparrow$ implies $G_f \notin$ Incr

Sample Question#7

7. Let **Incr = { f | $\forall$x, φ_f(x+1)>φ_f(x) }**.
Let **TOT = { f | $\forall$x, φ_f(x)$\downarrow$ }**.
Prove that **Incr $\equiv_m$ TOT**. Note Q#6 starts this one.
Let f be arb.
Define G_f (x) = $\exists$t[stp(f,x,t) && stp(f,x+1,t) && (value(f,x+1,t) > value(f,x,t))]
f $\in$ Incr iff $\forall$x φ_f(x+1)>φ_f(x) iff
$\forall$x G_f (x)$\downarrow$ iff G_f $\in$ TOT

Sample Question#8

8. Let **Incr = { f | $\forall$x φ_f(x+1)>φ_f(x) }**. Use Rice's theorem to show **Incr** is not recursive.
   Non-Trivial as
   C_0(x)=0 $\notin$ Incr; S(x)=x+1 $\in$ Incr
   Let f,g be arb. Such that $\forall$x φ_f(x)=φ_g(x)
   f $\in$ Incr iff $\forall$x φ_f(x+1)>φ_f(x) iff
   $\forall$x φ_g(x+1)>φ_g(x) iff g $\in$ Incr

Sample Question#9

9. Let **S** be a recursive (decidable set), what can we say about the complexity (recursive, re non-recursive, non-re) of **T**, where **T** $\subset$ **S**?

   Nothing. Just let S = $\aleph$, then T could be any subset of $\aleph$. There are an uncountable number of such subsets and some are clearly in each of the categories above.

Sample Question#10

10. Define the pairing function **$\langle x,y \rangle$** and its two inverses **$\langle z \rangle_1$** and **$\langle z \rangle_2$**, where if **$z = \langle x,y \rangle$**, then **$x = \langle z \rangle_1$** and **$y = \langle z \rangle_2$**.

$pair(x,y) = \langle x,y \rangle = 2^x \; (2y + 1) – 1$

with inverses

$\langle z \rangle_1 = \log_2(z+1)$

$\langle z \rangle_2 = (((z + 1) // 2^{\langle z \rangle_1}) – 1) // 2$

Sample Question#11

11. Assume **A** $\leq_m$ **B** and **B** $\leq_m$ **C**.
Prove **A** $\leq_m$ **C**. In this proof, we will assume the universe for each set **S** is U_S. In general $U_S = \aleph$

A $\leq_m$ **B** iff there exists an m-1 algorithm
f1: $U_A \rightarrow U_B$ such that **x∈A** ⟺ **f1(x)∈B**
B $\leq_m$ **C** iff there exists an m-1 algorithm
f2: $U_B \rightarrow U_C$ such that **x∈B** ⟺ **f2(x)∈C**
Define **f3(x) = f2(f1(x)), f3:** $U_A \rightarrow U_C$ is an m-1 algorithm and **x∈A** ⟺ **f3(x)∈C** implies
A $\leq_m$ **C** as was desired

Sample Question#12

12. Let **P = { f | ∃ x [STP(f, x, x)] }**. Why does Rice's theorem not tell us anything about the undecidability of **P**?

This is not an I/O property as we can have implementations of C_0 that are efficient and satisfy P and others that do not.