Sample Question#1

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 $\forall x\ f(x+1) > f(x)$.

DONE in class and notes.

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**

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

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.

DONE in class.

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 ∈ S and y ∈ 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) = ∃<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 = ℵ, 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
   $\forall x\ \varphi_f(x) - \varphi_f(x) + x = x$ iff $G_f \in$ Incr

Sample Question#7

7. Let **Incr = { f | ∀x, φ_f(x+1)>φ_f(x) }**.
   Let **TOT = { f | ∀x, φ_f(x)↓ }**.
   Prove that **Incr $\equiv_m$ TOT**. Note Q#6 starts this one.
   Let f be arb.
   Define G_f (x) = ∃t[stp(f,x,t) && stp(f,x+1,t) && (value(f,x+1,t) > value(f,x,t))]
   f ∈ Incr iff ∀x φ_f(x+1)>φ_f(x) iff
   ∀x G_f (x)↓ iff G_f ∈ 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** ⊂ **S**?

   Nothing. Just let S = ℵ, then T could be any subset of ℵ. 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 **$<x,y>$** and its two inverses **$<z>_1$** and **$<z>_2$**, where if **$z = <x,y>$**, then **$x = <z>_1$** and **$y = <z>_2$**.

Right out of Notes.

Sample Question#11

11. Assume $\mathbf{A} \leq_m \mathbf{B}$ and $\mathbf{B} \leq_m \mathbf{C}$.
    Prove $\mathbf{A} \leq_m \mathbf{C}$.

    Done in class

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.