Assignment#3 Sample Key
1. Show prfs are closed under Fibonacci induction

Fibonacci induction means that each induction step after calculating the base is computed using the previous two values. Here, 
f(0,x) = some base value;
f(1,x) is based on f(0,x) and 0 (an invented value for two steps back); and
for \( y > 1 \), \( f(y,x) \) is based on \( f(y-1,x) \) and \( f(y-2,x) \).

The formal hypothesis is:
Assume \( g \) and \( h \) are already known to be prf, then so is \( f \), where
\( f(0,x) = g(x) \);
\( f(1,x) = h(f(0,x), 0) \); and
\( f(y+2,x) = h(f(y+1,x), f(y,x)) \)

Proof is by construction
Fibonacci Recursion

Let K be the following primitive recursive function, defined by induction on the primitive recursive functions, g, h, and the pairing function.

\[
K(0, x) = B(x)
\]

\[
B(x) = \langle g(x), C_0(x) \rangle
\]

// this is just \( \langle g(x), 0 \rangle \)

\[
K(y + 1, x) = J(y, x, K(y, x))
\]

\[
J(y, x, z) = \langle h(\langle z_1, z_2 \rangle), z_1 \rangle
\]

// this is \( \langle f(y+1, x), f(y, x) \rangle \), even though f is not yet shown to be prf!!

This shows K is prf.

f is then defined from K as follows:

\[
f(y, x) = \langle K(y, x) \rangle_1
\]

// extract first value from pair encoded in K(y,x)

This shows it is also a prf, as was desired.
Fibonacci Recursion (simpler form)

Let $K$ be the following primitive recursive function, defined by induction on the primitive recursive functions, $g$, $h$, and the pairing function.

$$K(0,x) = <g(x), 0> \quad \text{// this is pair } <f(0,x), 0>$$

$$K(y+1, x) = <h(<K(y,x)>_1, <K(y,x)>_2), <K(y,x)>_1> \quad \text{// this is pair } <f(y+1,x), f(y,x)>,$$

This shows $K$ is prf.

$f$ is then defined from $K$ as follows:

$$f(y,x) = <K(y,x)>_1 \quad \text{// extract first value from pair encoded in } K(y,x)$$

This shows it is also a prf, as was desired.
2. Show $S$ inf. rec. iff $S$ is the range of a monotonically increasing function

• Let $f_S(x+1) > f_S(x)$, and $\text{Range}(f_S(x)) = S$. $S$ is decided by the characteristic function
  \[\chi_S(x) = \exists y \leq x \ [ f_S(y) == x ]\]
  The above works as $x$ must show up within the first $x+1$ numbers listed since $f_S$ is monotonically increasing.

• Let $S$ be infinite recursive. As $S$ is recursive, it has a characteristic function where $\chi_S(x)$ is true iff $x$ is in $S$.
  Define the monotonically increasing enumerating function $f_S(x)$ where
  \[f_S(0) = \mu x \ [ \chi_S(x) ]\]
  \[f_S(y+1) = \mu x > f_S(y) \ [ \chi_S(x) ]\]
  As required, this enumerates the elements of $S$ in order, low to high.
3. If $S$ is infinite re, then $S$ has an infinite recursive subset $R$

- Let $f_S$ be an algorithm where $S = \text{range}(f_S)$ is an infinite set
- Define the monotonically increasing function $f_R(x)$ by 
  
  $f_R(0) = f_S(0)$
  
  $f_R(y+1) = f_S( \mu x \ [ f_S(x) > f_R(y) ])$

  - The above is monotonically increasing because each iteration seeks a larger number and it will always succeed since $S$ is itself infinite and so has no largest value. Also, $R$ is clearly a subset of $S$ since each element is in the range of $f_S$.
- From #2, $R$ is infinite recursive as it is the range of a monotonically increasing algorithm $f_R$.
- Combining, $R$ is an infinite recursive subset of $S$, as was desired.