Logic, Computability and Incompleteness.

Exercise Set 1. Due Friday 2 March. Turn in (hardcopies) to Teaching Office, ground floor DSB.

1) Show that the set of odd positive integers is enumerable, and provide an enumerating function $f(x)$.
2) Show that the cardinal number of the set of squares of natural numbers is $\aleph_0$.
3) Show that the set of all finite strings of 0's and 1's is enumberable.
4) Show that the set of all infinite strings of 0's and 1's is not enumberable
5) Suppose that the set Γ is denumerable, that the set $\Delta \subset \Gamma$ is finite, and that Γ^0 is the result of removing all the elements of Δ from Γ. Prove that Γ^0 is still denumerable.
6) What if *denumerably* many elements are removed from Γ?
7) Using quadruple notation and the conventions introduced in the Lecture Slides (topic 2, slide 16), construct a Turing Machine that computes the 1-place function f of positive integers, such that $f(x) = x + 1$. Exhibit the sequence of configurations for computing the value $f(4)$.
8) Recall that Turing Machines are sets of instructions for syntactic manipulations, and can be *interpreted* as computing any number of different functions depending on the initial tape configuration. Using the same conventions as above, construct a single Turing Machine that computes the entire family of functions g^n of positive integers, such that for any 'arity' n and input n-tuples $x_1, ..., x_n$, $g^n(x_1, ..., x_n) = 1$. Now consider the two place function g^2 in this family. Exhibit the sequence of configurations for computing the value of $g^2(2,3)$.
9) Show that the set of all Turing Machines with a vocabulary of symbols from the finite list $S_0, S_1, S_2, \ldots, S_n$ is enumerable.
10) The *factorial* of y is the product of all the positive integers up to and including y (but is taken to be 1 when y is 0). Assuming **prod** as a primitive recursive function, provide a formal specification of the factorial function **fac**(y) in primitive recursive terms.
11) Consider the 3-place function on natural numbers $h(x_1, x_2, y) = (x_1 \cdot x_2) + y$. The function h could be expressed purely in terms of composition. Instead, formally define $h(x_1, x_2, y)$ using the primitive recursion schema [assuming **prod** and other primitive recursive functions formally defined in the lecture slides].
12) Consider the function $f(x,y)$ given in terms of the following definition by cases: $f(x,y) = \{x \dot{-} y$ if $x > y$

    $= \{x + y$ if $x < y$

    $= \{\mathbf{fac}(x)$ if $x = y$

    Provide a formal specification of f as a primitive recursive function. You can assume **dif**, **sum** and other primitive recursive functions already formally defined. Also, can use the primitive recursive format for definition by cases given in the Lecture Slides (i.e. not making all the composition operations formally explicit).