Assignment 2: More Programs, and Primitive Recursive Functions

This assignment is due Friday, February 5th, at the beginning of class (9:00am).

1. Let $P(x)$ be a computable predicate. If f is defined by

$$f(x_1, x_2) = \begin{cases} x_1 + x_2 & \text{if } P(x_1 + x_2); \\ \uparrow & \text{otherwise.} \end{cases}$$

   show that f is partially computable.

2. For any isomorphism $f : X \to Y$, one can define an inverse function $f^{-1} : X \to Y$, where $f^{-1}(x)$ is the unique number y such that $f(y) = x$. Suppose that $f : \mathcal{N} \to \mathcal{N}$ is an isomorphism and computable. Prove that f^{-1} is also computable.

3. The language $\mathcal{P}$ has only three instructions: increment, decrement, and loop if a variable is non-zero. But there are other reasonable choices for a simple programming language. Another choice is a language $\mathcal{P}'$ which has variables and labels just like $\mathcal{P}$, but has these three instructions:
   $V \leftarrow V'$
   $V \leftarrow V + 1$
   `IF` $V \neq V'$ `GOTO L`
   (where V, V' are variables, and L is a label). Show that $\mathcal{P}'$ is *equivalent* to $\mathcal{P}$, in the sense that a function f is partially computable in $\mathcal{P}$ if and only if it is partially computable in $\mathcal{P}'$.

4. (a) For any n, prove that the function $f(x) = x^n$ is primitive recursive.
   (b) Using part (a) and induction, prove that any polynomial function

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

   is primitive recursive (each a_i is a natural number).

5. Let $\pi(x)$ be the number of primes less than or equal to x. Show that $\pi(x)$ is computable.