Comp 430 Program 1

Due Date: _____ Points: 25 Late Penalty: 10%/day

Write a CLISP function MATCH which receives a term and a pattern and returns a list of variable substitutions which, when applied to the pattern, will make the pattern identical to the term. You may assume that every term and pattern are given in prefix notation, i.e. the first member of every list is the function name, which is a constant. You may assume that the symbols, u, v, w, x, y, and z in the pattern are variables, and that all other symbols are constants or functions. You should return NIL if no match is possible, (NIL) if the pattern matches the term without substitution, and a list of pairs indicating the variable substitutions (as shown below) in all other cases.

Note that variables are not permitted in the term, only in the pattern. Either the term or pattern may be an atom or a list.

Examples:

(MATCH term pattern) -> substitution

(MATCH '(+ a b) '(+ a x)) -> ((x . b))

(MATCH '(* a b) '(* a b)) -> (nil)

(MATCH '(f a b) '(f a a)) -> nil

(MATCH '(+ a b) '(- a b)) -> nil

(MATCH '(+ (- b c) a) '(+ x y)) -> ((x - b c) (y . a))

(MATCH '(loves a b) '(loves x x)) -> nil

(MATCH '(loves joe pie) '(loves x pie)) -> ((x . joe))

(MATCH >(+ a (+ b a)) '(+ x (+ y x))) -> ((x . a) (y . b))

Turn in a well-documented copy of match.lsp, containing all needed functions via EASEL.