

Homework 3:Lisp Exercises 2

Now Due Tuesday September 15, 2009 in class

Be sure to follow the guidelines for Programming Assignments. Since these problems are simple, you may skip the Statement of the Problem, Approach and Algorithm - Program Flow sections and give an overall conclusion for the entire programming assignment. Make sure you give me a listing of each program and at least **three test cases for each**. You may wish to use the "dribble" option in the file menu of XLISP to capture the screen as you test your functions.

- I. Define MYLIST to what the primitive function LIST does.
- II. Define MYAPPEND to what the built-in function APPEND does.
- III. Define MYLAST to what the built-in function LAST does.
- IV. Write a function, UNNEST, which takes a single list as input and returns a simple list of all its atoms.

Ex. (unnest '(1 (2 (3)) (4 5))) ==> (1 2 3 4 5)
(unnest '((((test two)))))) ==> (test two)

- V. Write a function, LEN, which is the same as LENGTH. DO NOT use LENGTH in your function definition. Count only top level SEXEs. Make sure that NIL is counted as an atom.

Ex. (len ()) ==> 0
(len '(a b (c) ((d) ()))) ==> 5
(len '(((a ((b)))))) ==> 1

- VI. Rewrite union so that the objects remain in order. Call it UNION1.

Ex. (union1 '(a b c) '(f g c d b)) ==> (a b c f g d)
(union1 '(it is a) '(big box)) ==> (it is a big box)

- VII. Write a function, SETDIFF, which when given two sets as input, returns all elements in the first set which are not in the second set.

Ex. (setdiff '(a b c d) '(b a)) ==> (c d)
(setdiff '(a b c d) '(d e f)) ==> (a b c)
(setdiff '(a b c) '(d e f)) ==> (a b c)

- VIII. Write a predicate function, EQSET, which returns true if its two arguments are the same set.

Ex. (eqset '(this is the way) '(it is this way)) ==> nil
(eqset '(Coke is it) '(it is Coke)) ==> t

- IX. Write a function, MAKESET, which takes a simple list as input and makes a set.

Ex. (makeset '(a b a d e b f)) ==> (a d e b f)
(makeset '(1 1 1 2 2 2 3 3)) ==> (1 2 3)