

UCF

School of computer Science

COT 4020: Programming Languages

Spring 2007

Homework 1 (Lisp)

Due Monday 2007-02-12 by 5:00pm

Instructions

Write the functions that are described below. Function calls are provided to show argument order, along with a description and examples where necessary. For functions 1-3, you can test your function against the analogous built-in functions in your Lisp implementation. In case of confusion I am available for clarifications. I will update the website in case of mass confusion and hysteria.

You may define extra functions to help you.

Include everything in one file named hw1-`first initial + lastname`.lisp

E.g., I would submit a file called hw1-cmillward.lisp

Submit no later than 5:00pm on 2007-02-12 via email; attach in email to cmillward@cs.ucf.edu.

Late homework WILL NOT be accepted.

Imperative programming should not be used.

Acceptable functions: cons, car, cdr, list, cond, defun, eq, eql, equal, null, atom, listp, +, -, /, *, >, <

Functions 1-4 are worth 15 pts each.

Function 5 is worth 40 pts.

Functions to implement

1. Define function XMEMBER, that works like Common Lisp's MEMBER function. If ELEMENT is in the top-level of LIST, then return LIST otherwise return NIL. Use EQ to test for equivalence.

(xmember element list)

2. Define function XREVERSE, that works like Common Lisp's REVERSE function. Return a list with every element of LIST in reverse order.

(xreverse list)

3. Define function XLENGTH, that works like Common Lisp's LENGTH function. Return the number of elements in the top-level of LIST.

(xlength list)

4. Define function XFLATTEN. Return a list with all sublists recursively flattened.

(xflatten list)

Examples.

```
(xflatten '(a b c))           => (a b c)
(xflatten '(a (b c)))         => (a b c)
(xflatten '((((a b)) c)))    => (a b c)
```

5. Define function FFTE. Like MEMBER, FFTE returns whether an atom is in a list. However, if ELEMENT occurs in LIST then as opposed to merely returning the LIST, FFTE should return a list containing the atom occurring after ELEMENT in the original list and the number of elements between ELEMENT and the end of the list. Also, to make things a little more interesting if ELEMENT occurs multiple times in LIST then the return value should be for the instance of ELEMENT closest to the end of the list. If ELEMENT does not occur in the list then return NIL.

(ffte element list)

Examples.

; This is the simple case where A does not occur in the list, thus we return NIL.

```
(ffte 'a '(d e f))
=> NIL
```

; In this case A occurs twice in the list so we return the last occurrence

; of A which is followed by B and is two elements from the end of the list.

```
(ffte 'a '(a a b c))
=> (b 2)
```

; In this case G occurs at the end of the list, thus the element following G is NIL and there are 0 elements between G and the end of the list.

```
(ffte 'g '(e f g))
=> (nil 0)
```