

UCF

School of Computer Science

COP 4020: Programming Languages

Summer 2006

Homework 1 (LISP)

Due on 06/12/06 by 11:59pm

Instructions

You must implement the following 5 functions in Common Lisp.

For this homework you are restricted to only using the following Common Lisp commands.

- CAR
- CDR
- COND
- NULL
- CONS
- DEFUN
- LIST
- ATOM
- LISTP
- EQ
- EQL
- EQUAL
- and the arithmetic expressions and comparators.

You may also use functions you create from a problem for another problem.

Make sure all the functions are named as specified in the assignment bellow. Put them all in a file named with your first initial concatenated with your last name, with a .lisp extension. So if you name is John Smith, then you will turn in a file called jsmith.lisp.

You will lose a letter grade if you do not follow instructions.

Email this file to acook@cs.ucf.edu no later than Monday, June 12 at 11:59pm. The assignment will be accepted late for up to a day at a loss of 30%. Then after 11:59pm on Tuesday 06/13/06, the assignment will no longer be accepted and you will receive no

credit.

Plagiarism will not be tolerated. Feel free to discuss amongst yourselves, but do your own work. If you are not getting it, please make arrangements to see the TA, he can help explain things. The point of the homework is to learn to think recursively in Lisp.

Functions to implement

1. Write a function called LAST-ELEMENT that returns the last element in a list

```
(LAST-ELEMENT '(A B C)) returns C
(LAST-ELEMENT '(A (B C))) returns (B C)
(LAST-ELEMENT '((A B) C (D (E F)))) returns (D (E F))
```

2. Write a function called MY-APPEND that simulates the actual *append* in LISP.

```
(MY-APPEND '(A B) '(C D)) returns (A B C D)
(MY-APPEND '(A (B C)) '((C D) E)) returns (A (B C) (C D) E)
```

3. Write a function called SWAP that takes a list and swaps every two elements in it.

```
(SWAP '(1 2 3 4 5)) returns (2 1 4 3 5)
(SWAP '("hi" 2 "bye" 3)) returns (2 "hi" 3 "bye")
```

4. Write a function called MY-REVERSE that simulates the reverse function in LISP. The function should take a list and return a list with the elements from the input list in reverse order. You might consider using a helper function on this.

```
(MY-REVERSE '(A B C)) returns (C B A)
(MY-REVERSE '(A (1 2) (B (C D)) 3)) returns (3 (B (C D)) (1 2) A)
```

5. Write a function called ROTATE-LIST that takes a list and rotates the elements by the specified number of times. If the rotate count is positive it should rotate the list to the left, and if the count is negative it should rotate it to the right. Therefore the function will take 2 arguments: the list and the rotate count. For this function you might consider using a helper function, and maybe a couple of functions from the previous problems...

```
(ROTATE '(1 2 3 4 5) 0) returns (1 2 3 4 5)
(ROTATE '(1 2 3 4 5) 4) returns (5 1 2 3 4)
(ROTATE '(1 2 3 4 5) -2) returns (4 5 1 2 3)
```