

Homework 2:

Due Tuesday September 6, 2011 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.

Problems 3-1, 3.3, 3.4 and 3.5 in Chapter 3 of Winston and Horn's *LISP*

Problems 4-1, 4-5 and 4-6 in Chapter 4 of Winston and Horn's *LISP*

3.1 Define EXCHANGE, a procedure that returns the elements of a two-element in reverse order.

```
Lisp-> (exchange '(adam eve))  
(eve adam)
```

3.3 Define ROTATE-LEFT, a procedure that takes a list as its argument and returns a new list in which the former first element becomes the last.

```
Lisp-> (rotate-left '(a b c))  
(b c a)  
Lisp-> (rotate-left (rotate-left '(a b c)))  
(c a b)
```

3.4 Define ROTATE-RIGHT which is like ROTATE-LEFT except in the opposite direction. You might want to use BUTLAST.

3.5 A palindrome is a list that has the same sequence of elements when read from right-to-left that it does from left-to-right. Define PALINDROME such that it takes a list as its argument and returns a plindrome that is twice as long.

4.1 Define DIVISIBLE-BY-THREE, a predicate that determines if an integer is divisible by three. You may want to use REM.

```
Lisp-> (divisible-by-three 3)  
T  
Lisp-> (divisible-by-three 4)  
Nil
```

4.5 Write IF forms that are equivalent to (ABS X), (MIN A B), and (MAX A B), where X, A, and B are numbers.

4.6 Compose COND forms that are equivalent to (NOT U), (OR X Y Z), and (AND A B C).

Outline

I. Example of a Classical AI Production System

Irrevocable Control Strategy
Tentative Control Strategy

II. Architectures for the Implementation of Action Functions

A. Production Systems (Rule-Based Systems)
B. State Machines
C. Artificial Neural Networks
D. Subsumption Architecture

III. Final Thoughts on Stimulus-Response (SR) Agents - Chapter 2