



## LÓGICA PARA CIENCIAS DE LA COMPUTACIÓN

### Trabajo Práctico "H" Programación Funcional en LISP

Primer Cuatrimestre de 2008

## Ejercicios

1. Para cada una de las siguientes secuencias de símbolos, indicar si la misma constituye o no una s-expresión, y en caso afirmativo indicar si es un átomo numérico, un átomo simbólico o una lista.

- a) atomo
- b) (esto es un atomo)
- c) lista
- d) ((a b) (a b))
- e) 2
- f) (2)
- g) +(2 2)
- h) (+ 2 2)
- i) (/ (+ 2 2) (- 3 1))
- j) ()

2. Evaluar las siguientes expresiones:

- a) (/ 9 (+ 2 3 (- 7 3)))
- b) (MIN (MAX 3 1 4) (MAX 2 7 (+ 3 7)))
- c) (CAR (CDR (CAR '((a b) (c d))))))
- d) (CDR (CDR (CAR '((a b) (c d))))))
- e) (CDR (CAR (CDR '((a b) (c d))))))
- f) (CDR '(CAR (CDR (CDR ((a b) (c d))))))

3. Definir una secuencia de operaciones CAR y CDR para extraer el átomo *pera* de las siguientes listas:

- a) (manzana naranja pera uva)
- b) ((manzana naranja) (pera uva))
- c) (((manzana) (naranja) (pera) (uva)))
- d) (manzana ((naranja) (pera)) (((uva))))

- e) (((((manzana))) ((naranja)) (pera) uva)
  - f) (manzana (naranja (pera uva)))
  - g) (((((manzana) naranja) pera) uva)
4. Considerando las operaciones sobre listas APPEND, CONS y LIST, reformular cada una de las siguientes expresiones en base a las otras operaciones.
- a) (APPEND '(1 2 3) '(a b c))
  - b) (LIST '(1 2 3) '(a b c))
  - c) (CONS '(1 2 3) '(a b c))
  - d) (APPEND '((1) (2 3)) '(a (b c)))
  - e) (LIST '((1) (2 3)) '(a (b c)))
  - f) (CONS '((1) (2 3)) '(a (b c)))
  - g) (APPEND '(1) '(a b c))
  - h) (LIST '(1) '(a b c))
  - i) (CONS '(1) '(a b c))
  - j) (APPEND '2 '(a b c))
  - k) (LIST '2 '(a b c))
  - l) (CONS '2 '(a b c))
5. Evaluar las siguientes expresiones en el orden en que se listan a continuación:
- > (SETQ op1 '+)
  - > (SETQ op2 '-)
  - > (SETQ op op1)
  - > op
  - > (EVAL (LIST op '1 '2 '3))
  - > (SETQ op 'op1)
  - > op
  - > (EVAL op)
  - > (EVAL (EVAL '(QUOTE op)))
6. Implementar una función que dados tres números enteros representando las longitudes de los lados de un hipotético triángulo, determine si es posible construir dicho triángulo.
7. Definir en LISP sendas funciones para:
- a) Calcular la longitud de una lista.
  - b) Eliminar el primer elemento de una lista.
  - c) Eliminar el último elemento de una lista.
  - d) Invertir una lista.
  - e) Rotar una lista a izquierda.

8. Implementar en LISP una función que calcule las raíces de una ecuación cuadrática, considerando cuidadosamente todos los casos posibles.
9. Definir en LISP a las siguientes funciones:
  - a) La función factorial.
  - b) La función de Fibonacci.
  - c) La función de Ackerman.
10. Adoptando una representación de conjuntos a través de listas de átomos, implementar en LISP funciones para:
  - a) Determinar si un elemento pertenece a un conjunto.
  - b) Comprobar si conjunto está contenido en otro conjunto.
  - c) Comprobar si dos conjuntos son iguales.
  - d) Hallar la intersección de dos conjuntos.
  - e) Hallar la unión de dos conjuntos.
  - f) Calcular la diferencia entre dos conjuntos.
  - g) Calcular el producto cartesiano entre dos conjuntos.
11. Definir una función que dada una lista anidada, retorne otra lista sin anidamiento alguno que contenga a todos sus átomos. Por ejemplo:
 

```
> (aplanar '(1 2 3 (4 (5 6)) ((7 (8 (9)))) 10))
(1 2 3 4 5 6 7 8 9 10)
```
12. Definir una función que dada una lista anidada, retorne otra lista formada por dos sublistas, las que a su vez contengan los elementos pares e impares respectivamente. Por ejemplo:
 

```
> (parimpar '(0 1 2 (3 (4 5)) ((6 (7 (8)))) 9))
((0 2 4 6 8) (1 3 5 7 9))
```
13. En base a la siguiente definición, ¿cuál es el propósito de la función misterio?
 

```
(DEFUN misterio (S)
  (COND
    ((ATOM S) 0)
    ((NULL S) 1)
    (T (MAX (1+ (Misterio (CAR S))) (Misterio (CDR S))))
  )
)
```
14. Definir en LISP una función `mas` que calcule la suma de dos números apelando a incrementos sucesivos.
15. Formular en LISP una función `masmas` que calcule la suma de  $n$  argumentos, donde  $n \geq 1$ . A tal efecto, adoptar la convención de que los  $n$  argumentos son suministrados en una lista.