

Guía de ejercicios N° 1

Fecha de entrega: 3er clase del cuatrimestre

1. Indique cual es el valor que devuelve la llamada de las siguientes funciones en Lisp:

1• (CADR '(a b c))

2• (CADR '(a (b c)))

3• (CADR '((a b) c))

4• (CDADDR '((a) i (l i n)))

5• (CDADDR '((a) i (li n)))

6• (CDADDR '((a) i (lin)))

7• (CONS '(a b) '(c d e))

8• (CONS '(a b) '((c d e)))

9• (CONS '((a b)) '(c d e))

10• (CONS '((a b)) '((c d e)))

11• (APPEND '(a b) '(c d e))

12• (APPEND '(a b) '((c d e)))

13• (APPEND'((a b)) '(c d e))

14• (APPEND '((a b)) '((c d e)))

15• (LIST '(a b) '(c d e))

16• (ATOM '(a b c))

19• (LISTP '(a b c))

17• (NUMBERP 34)

20• (NUMBERP 'ma)

18• (ATOM 34)

21 • (SYMBOLP 34)

2. Escriba la combinación de selectores o constructores necesarios para obtener la salida a partir de las listas dadas:

- 1-• ((a b) c (m p)) ----- > m
- 2-• ((a (b c)) (m p)) ----- > b
- 3-• (a (d (h) i j) f (g y)) ----- > h
- 4-• (a (d (h) i j) f (g y)) ----- > j
- 5-• '(a (d) e) '(f g (h (i) j) k) ----- > ((A (D) E) F G (H (I) J) K)
- 6-• '(a b (c)) '(d (e f (g)) h) ----- > ((A B (C)) (D (E F (G)) H))
- 7-• '((f) g h) '(a (b c d) ((e))) ----- > ((F) G H A (B C D) ((E)))

3. Utilizando los átomos a, b y/o NIL construir las siguientes listas usando los siguientes constructores de Lisp:

- 1-• Con CONS ----- > ((a) (b))
- 2-• Con LIST ----- > ((a) (b))
- 3-• Con APPEND y LIST ----- > ((a) (b))
- 4-• Con CONS ----- > ((a) b (a b))
- 5-• Con LIST ----- > ((a) b (a b))
- 6-• Con CONS ----- > ((a b) a (b))
- 7-• Con CONS ----- > ((a b) a b (a (b)))

4. Escribir ejemplos de listas para que la siguiente combinación de constructores devuelva el átomo b

- 1• (CADR '(...)) 5• (CDAR '(...))
- 2• (CADAR '(...)) 6• (CAADR '(...))
- 3• (CDAAR '(...)) 7• (CDADR '(...))
- 4• (CADDR '(...)) 8• (CDDAR '(...))

5. Dado un número entero, definir una función que devuelva el número siguiente del número dado.

Ej: (siguiente 8) -----> 9

6. Definir una función que calcule el factorial de un número N, siendo N >= 0.

Ej: (fact 5) -----> 120

7. Definir una función booleana que determine si un número es positivo.

Ej.1: (espositivo 5) -----> T Ej.2: (espositivo -3) -----> NIL

8. Construir una función que aplicada a 2 listas de la misma cantidad de elementos, devuelva otra lista que contenga los elementos de ambas en forma intercalada.

Ej.: (inter '(A B C) '(D E F)) -----> (A D B E C F)

9. Una función que aplicada a una lista devuelva otra lista con los elementos de las posiciones pares de la lista original.

Ej: (pares '(A B C D E) -----> (B D)

10. Escribir una función que aplicada a 2 listas numéricas de igual longitud, devuelva una nueva lista donde cada componente es la suma de las componentes respectivas de las listas dadas.

Ej: (suma '(5 8 3) '(1 5 4)) -----> (6 13 7)

11. Construir una función que aplicada a una lista la devuelva invertida, a primer nivel

Ej.1: (invertir '(5 8 3 6)) -----> (6 3 8 5)

Ej.2: (invertir '(5 (8 3) 6)) -----> (6 (8 3) 5)

12. Construir una función que aplicada a un átomo y una lista de átomos unicamente, devuelva T si dicho átomo pertenece a la lista y NIL en caso contrario.

Ej.1: (pertenece 'm '(a p m w)) -----> T

Ej.2: (pertenece 'b '(a p m w)) -----> NIL

13. Escribir una función que recibe una lista con el nombre de una variable seguido del operador ++ o bien -- por ej. (a ++) y devuelva la lista que contiene la variable, seguido del símbolo igual, seguido de la variable, seguido del operador + o -, seguido del número 1.

Ej.1: (convertir '(a ++)) -----> (a = a + 1).

Ej.2: (convertir '(sum --)) -----> (sum = sum - 1).

14. Escribir una función que recibe una lista con el nombre de una variable seguido de alguno de los siguientes operadores de asignación: +=, -=, *=, /=, %= seguido de una expresión aritmética y devuelva la lista que contiene la variable, seguido del símbolo igual, seguido de la variable, seguido del operador +, −, *, /, %, según corresponda, seguido de una sublista que contiene la expresión aritmética recibida

Ej.1: (convertir '(a += b * c)) -----> (a = a + (b * c))

Ej.2: (convertir '(tot *= b - c * 10)) -----> (tot = tot * (b - c * 10))