

Guía de Trabajos Prácticos V – Programación Funcional

1. Crear una función que concatene L1 con el inverso de L2.
(concat-inv '(1 2 3) '(4 5 6)) → '(1 2 3 6 5 4)
2. Definir un predicado con tres argumentos: un átomo y dos listas. El predicado debe devolver T si el átomo pertenece a las dos listas.
(pertenece 'a '(a b c) '(r f g a)) → #t
(pertenece 'a '(a b c) '(r f g h)) → #f
3. Definir una función que devuelva el número de átomos que hay en una lista situados a la izquierda de un átomo determinado de dicha lista.
(cant-izq 'd '(a b c d e f)) → 3
4. Definir una función que añada un elemento a una lista en caso de que aquel no se encuentre en ésta.
(addif 'a '(b c d)) → '(b c d a)
(addif 'a '(a b c d)) → '(a b c d)
(addif 'c '(b c d)) → '(b c d)
5. Definir un predicado que tome tres argumentos: día, mes y año, y devuelva T si es una fecha válida.
(FECHAP 12 12 1986) => #t
(FECHAP 12 30 1987) => #f
(FECHAP 31 2 1986) => #f
(FECHAP 31 11 1876) => #t
6. Definir la función get-profundidad que obtiene el número máximo de listas anidadas que aparecen en una lista.
(get-profundidad '((1 (2)) (((5 7))) 4)) → 3
7. Definir una función APLANAR que reciba como argumento una expresión simbólica y elimine todos los paréntesis que aparezcan en esa expresión, devolviendo como resultado una lista con todos los átomos que aparezcan en el argumento.
(APLANAR '((1 2 3) (9 (2 3 4)) ((((3 4 (7))))))) → (1 2 3 9 2 3 4 3 4 7)
8. Definir la función rotar que rota los elementos de una lista hacia la derecha o hacia la izquierda.
(rotar 'derecha '(1 2 3 4)) → (4 1 2 3)
(rotar 'izquierda '(1 2 3 4)) → (2 3 4 1)
9. Definir la función calcula-cuadrados que recibe una lista de números y devuelve otra lista con el cuadrado de cada elemento.
(calcula-cuadrados '(1 2 3 4)) → (1 4 9 16)
10. Definir la función resta-listas que devuelve una lista con los elementos de la primera lista que no aparecen en la segunda.
(resta-listas '(1 2 3 4) '(2 3)) → (1 4)

11. Definir la función suma-listas que devuelve una lista con los elementos de la primera lista y los de la segunda que no aparecen en la primera.
(suma-listas '(1 2 3 4) '(2 3 5)) \rightarrow (1 2 3 4 5)
12. Definir una función que permita obtener la altura de un árbol que está representado por listas anidadas donde cada lista contiene los nodos correspondientes a su nivel. Ej: (((1 2) (3 4 5)) 6 (7 (8 (9 (10 11))))) representaría:

7

8

6

5

3

4

2

1

9

10

11