

Introducción a la programación: Práctica de recursión

(Ejercicios tomados del blog del profesor: Juan Carlos Gómez)

A continuación encontrará un conjunto de ejercicios cortos para los cuales se le solicita una versión recursiva. Pretendemos que estos ejercicios sean un excelente entrenamiento para su dominio de la recursividad.

Recuerde, debe utilizar versiones recursivas y no usar las funciones especializadas de Python.

Ejercicios

1. Largo de una lista L.
Ejemplo:
`largo([1,2,'4','6',87])` -> retorna 5
2. Elemento en la posición k de una lista L.
Ejemplo:
`sacar_elemento(1, ['5', [1,2,3], 43, 'Mi profe es muy inteligente'])` -> retorna [1,2,3]
`sacar_elemento(3, ['5', [1,2,3], 43, 'Mi profe es muy tuanis'])` -> retorna 'Mi profe es muy tuanis'
3. Retornar el Último elemento en una lista L.
Ejemplo:
`ultimo([1, 45, 'p', (1,2), 7])` -> retorna 7
4. Recibir una lista de enteros y retornar una lista con los números pares.
Ejemplo:
`lista_pares([1,4,7,8,2,4])` -> retorna [4,8,2,4]
5. Concatenar dos listas L y M.
Ejemplo:
`concatenar([1,2,3], [4,5,6])` -> retorna [1,2,3,4,5,6]
6. Invertir una lista L.
Ejemplo:
`invertir([1,2,3,4,5,6,7])` -> retorna [7,6,5,4,3,2,1]
7. Encontrar el mayor de una lista.
Ejemplo:
`mayor([34,65,21,87,14,69, 87])` -> retorna 87
8. Encontrar el menor de una lista.
Ejemplo:
`mayor([34,15,21,87,14,69, 87])` -> retorna 15
9. Quitar la primera ocurrencia de x en L. Se debe retornar una lista que no contenga la primera ocurrencia de x.
Ejemplo:
`nombre([1,3,4, 3, 3], 3)` -> retorna [1,4,3,3]
10. Ordenar una lista siguiendo la siguiente estrategia:
A. Si la lista es vacía se retorna la lista.
B. Sino, encontrar el menor de la lista y retornar la lista con ese elemento como primero concatenado con la lista ordenada que no contenga a ese elemento.
Nota: la función resultante será muy ineficiente pero demuestra el poder de la recursividad.
Ejemplo:
`ordenar([3,7, 2, 4, 1, 8])` -> retorna [1,2,3,4,7,8]

11. Insertar un elemento x en una lista ordenada tal que retorne una lista ordenada.
Ejemplo:
`insertar(5, [1,4,6,8,9]) -> retorna [1,4,5,6,8,9]`
12. Retornar una lista en la cual cada uno de los elementos de la lista de entrada fueron sustituidos por un valor x.
Ejemplos:
`convertir([1,2,3], 4) -> retorna [4, 4, 4]`
`convertir([]) -> retorna []`
`convertir([100, 2], 1) -> retorna [1, 1]`
13. Retornar una lista con tantos elementos x según se indique.
Ejemplos:
`repetir(4, 2) -> retorna [4, 4]`
`repetir(5, 0) -> retorna []`
`repetir('hola', 2) -> retorna ['hola', 'hola']`.
14. Una función que retorne true si todos los elementos de una lista son números enteros.
Ejemplos:
`son_numeros([]) -> retorna True`
`son_numeros([2,4]) -> retorna True`
`son_numeros([5,'hola']) -> retorna False`.
15. Una función que retorne la suma de todos los elementos que son números enteros en una lista. La lista puede contener otros elementos que no son lista.
Ejemplo:
`sumar_solo_numeros(['hola', 2, 4, 'casa']) -> retorna 6`.
16. Una función que retorne True si una lista de números está ordenada ascendentemente.
17. Una función que retorne True si es una lista en donde todos los elementos son 1 y 0 pero que aparecen en forma alternada.
Ejemplos:
`alternada([1,0]) -> retorna True`
`alternada([0,1,0]) -> retorna True`
`alternada([1]) -> retornan True`
`alternada([0]) -> retornan True`
`alternada([1,1,0]) -> retorna False`
`alternada([9]) -> retorna False`
18. Una función que duplique cualquier elemento x en una lista L.
Ejemplo:
`duplo(2,[4,2,6,2]) -> retorna [4,2,2,6,2,2]`.
19. Una función que repita cualquier elemento x, N veces, en una lista L.
Ejemplos:
`repetir_varios(2,3,[1,2,4,1,2,0]) -> retorna [1, 2, 2, 2, 4, 1, 2, 2, 2, 0]`
`repetir_varios(2,0,[1,2,4,1,2,0]) -> retorna [1, 4, 1, 0]`