

PARADIGMAS Y LENGUAJES DE PROGRAMACIÓN - LENGUAJE LISP -

PROCESOS RECURSIVOS

NOTA:

Cada ejercicio se debe desarrollar usando procesos recursivos.

Las funciones definidas en un ejercicio se pueden volver a usar en otros ejercicios

RECORDAR:

- Las variables nuevas que se definan deben ser locales, tener mensajes descriptivos para su ingreso y las validaciones correspondientes.
- Las funciones deben ser genéricas. Se deben poder ejecutar para cualquier valor que se ingresen en las variables.
- Se recomienda leer el documento REGLAS DE ESTILO LISP que se ha dejado en el Aula

Ejercicio Nº 1

A partir de una lista heterogénea que es ingresada como parámetro, definir la función **suma** que permita obtener la suma de todos los elementos de una lista.

Ejercicio Nº 2

A partir de una lista que es ingresada como parámetro y que contiene únicamente elementos atómicos, defina una función que cuente la cantidad de números impares que posee la misma.

Ejercicio Nº 3

A partir de una lista heterogénea que es ingresada como parámetro, definir la función **promedio** que permita obtener el promedio de todos los elementos numéricos de una lista.

Ejercicio Nº 4

A partir de una lista heterogénea que es ingresada como parámetro, definir una función que devuelva una lista cuyos elementos son el resultado de evaluar uno a uno si cada uno de los elementos de la lista ingresada como parámetro es una sublista.

Ejercicio Nº 5

Definir una función predicado, la que a partir de una lista solicitada al operador, verifique si **todos** los elementos de la lista son valores numéricos

Ejercicio Nº 6

A partir de una lista con las Notas los parciales de Programación de un alumno, la que es ingresada por el operador, definir una función predicado llamada **Aprobado**. La función debe evaluar si la materia está aprobada. Una materia está aprobada si todas las notas son 6 o valores mayores a 6

Ejercicio Nº 7

Definir la función **soloListas**, la que a partir de una lista heterogénea ingresada como parámetro, devuelva una lista cuyos elementos serán sublistas. Cada sublista estará conformada por el elemento que sea una sublista de la lista original junto con su longitud.

.

Ejercicio Nº 8

Definir una función predicado que determine si una lista numérica que es recibida como parámetro está ordenando en forma ascendente.

PARADIGMAS Y LENGUAJES DE PROGRAMACIÓN - LENGUAJE LISP -

Ejercicio Nº 9

A partir de dos listas ingresadas como parámetros, definir una función que devuelva una nueva lista donde cada elemento sea el resultado de la diferencia de los elementos de la lista 1 con los elementos de la lista 2 que se encuentren en la misma posición

Ejercicio Nº 10

A partir de una lista que es ingresada por el operador, definir una función que devuelva una nueva lista conteniendo dos sublistas. La primera sublista estará formada por los elementos de la lista original que son números enteros. La segunda sublista estará formada por los elementos de la lista original que no son números enteros.

Ejercicio Nº 11

A partir de una lista heterogénea que contiene únicamente elementos atómicos, definir una función que devuelva una lista sin los elementos repetidos de la lista original. Si el elemento está repetido debe aparecer una sola vez

Ejercicio Nº 12

En la compañía telefónica *FunTel* modelan la historia de llamadas del usuario mediante una lista conteniendo dos sublistas.

- la primera sublista corresponde al tiempo de duraciones de llamadas (en minutos) en el horario **normal**
- la segunda sublista corresponde al tiempo de duraciones de llamadas (en minutos) en el horario **reducido**

Se necesita que desarrollen las siguientes funciones:

- a. `cuandoHabloMas`, que determine en que horario hablo mas. Si en los dos hablo la misma cantidad, responder IGUAL
- b. `LLamadaMasLarga`: que determine cuál fue la llamada más larga y en que horario
- c. `LLamadaMasCorta`: que determine cuál fue la llamada más corta y en que horario