Paquete #2 de ejercicios (estructuras algorítmicas)

1) **[***sin usar ELT ni POSITION***]** Defina una función ***ElemInPos*** que reciba tres argumentos: *elem, lista* y *pos.* La función debe devolver *T* si *elem* está en la posición *pos* de *lista, NIL* si no lo está.

2) Escriba la función ***Inicio-en*** que recibe como argumentos una lista y un elemento cualquiera. La función debe entregar como respuesta una copia de la lista original pero comenzando con la primera ocurrencia del elemento dado en la lista original.
3) Modifique la función del ejercicio anterior para que se llame ***Termina-en*** y entregue como respuesta una copia de la lista original pero que termina en la última ocurrencia del elemento dado

4) Construya una función ***Primer-impar*** que reciba como argumento una lista y como respuesta entregue otra lista conteniendo el **primer elemento** de la lista original que sea un número impar y la posición (índice) donde se encuentra. Observe que los elementos de la lista pueden ser de cualquier tipo de datos.

5) Modifique la función del inciso anterior para que entregue en la lista de respuesta el **último elemento** de la lista que sea un número real mayor o igual que cero y el número de veces que dicho elemento se repite en toda la lista.
6) Escriba la función ***Conteo*** que recibe como argumento una lista cualquiera y, como respuesta, entregue una celda de construcción cuya primera parte contiene el conteo de elementos numéricos de la lista original y cuya segunda parte contiene el conteo de sublistas contenidas en la lista original.

7) Defina una función ***Aplana*** que reciba como argumento una lista con elementos anidados a cualquier nivel de profundidad y, como respuesta, entregue una lista conteniendo los mismos elementos pero todos ellos al nivel principal de profundidad.

8) Escriba la función ***Diagonal*** que recibe como argumento una lista conteniendo *m* sub-listas de *m* elementos cada una de ellas y que representa una matriz de *m* x *m* elementos. Como respuesta, esta función debe devolver una lista conteniendo los elementos en la diagonal principal de dicha matriz. Observe que los elementos de la matriz pueden ser de cualquier tipo de datos, no forzosamente numéricos.

9) Construya una función que reciba como argumento una lista cualquiera y, como respuesta, entregue una lista, con el mismo número de elementos de primer nivel, pero que contiene un símbolo **A** si el elemento en la posición correspondiente es un átomo, un símbolo **L** si el elemento correspondiente es una lista y un símbolo **N** si el elemento en la posición correspondiente es una lista vacía.

10) Defina la función ***Suma-numérica*** que recibe como argumento una lista cualquiera (no anidada), y como respuesta entrega la suma de exclusivamente aquellos elementos de la lista que son numéricos.

11) Escriba una función ***Filtra-vocales*** que reciba como argumento una lista (con elementos de cualquier tipo y anidada a cualquier nivel de profundidad) y, como respuesta entregue una copia de la lista argumento en la cual se han removido las letras vocales (tanto mayúsculas como minúsculas).

12) Construya una función ***Filtra-múltiplos*** que reciba como argumentos una lista y un número entero. Como respuesta debe entregar una copia de la lista argumento en la cual se han removido todos los múltiplos del entero recibido.

13) Defina la función ***Celdas*** que recibe como argumento una lista (con elementos de cualquier tipo y anidada a cualquier nivel de profundidad) y, como respuesta entrega el número de celdas de construcción que contiene la representación interna de la lista argumento.

14) Construya una función ***Implica*** con aridad indeterminada, que implemente el operador lógico de la implicación.

15) Escriba una función ***Mult*** que recibe como argumento dos listas, conteniendo sub-listas numéricas, representando matrices. La función debe regresar la multiplicación de las dos matrices si es que éstas son compatibles, en caso de no serlo debe regresar NIL.