Práctica 2- Definición de funciones (II)

1. Crea una función que dados un elemento y una lista (que puede contener sublistas), busque el elemento recursivamente. Si el elemento se encuentra en la lista, la función debe devolver el nivel de anidamiento de la sublista en la que aparece. Si no lo encuentra, debe devolver 0. Si el elemento aparece varias veces, es suficiente con encontrarlo una vez.

   Ej: > (busca-rec '9 '(2 f h (j k (j u 9) j u y) g j u) '1)
   > 3

2. Crea una función que dados un elemento y una lista (que puede contener sublistas), busque el elemento recursivamente. Si el elemento se encuentra en la lista, la función debe escribir en pantalla "El elemento X está en el nivel Y, siendo X el elemento buscado e Y el nivel de anidamiento. Si el elemento no aparece en la lista, el programa debe escribir "Elemento no encontrado". Utilizar la función del ejercicio anterior.

   Ej: > (busca-rec '9 '(2 f h (j k (j u 9) j u y) g j u) '1)
   > El elemento 9 está en el nivel 3.

3. Crea una función que dada una lista que puede contener números, strings y símbolos anidados a cualquier nivel, devuelve una nueva lista donde cada elemento es intercambiado por su tipo de dato. Realiza dos versiones de la función, una recursiva y otra con *MAPCAR*.

   Ej: > (sustituye-tipo '(hola ("mundo" 5)))
   > (símbolo (string número)

4. Crea una función que dada una matriz de números de dimensión MxN, devuelva la suma de todos los elementos de la matriz. Realiza dos versiones de la función, una recursiva y otra con *MAPCAR.*

   Ej: > (suma-matriz '((1 5 6) (4 2 3) (1 0 1))
   > 23