

Práctica 9.- Recursividad II.

- **Objetivo de la práctica:** Profundizar en el concepto de recursividad, aplicando los conocimientos adquiridos al recorrido del árbol construido mediante el *generador de expresiones* de la práctica III. Con los ejercicios de las próximas prácticas, completaremos el código fuente necesario para realizar por completo la práctica solicitada.

Ejercicio 1.- Pasar de expresión a cadena de texto.

Éste primer ejercicio servirá para completar el código de la unidad *ExprTree*. Tal y como se detalló en la explicación de la práctica, en la unidad *ExprParser* tenéis disponible una función que se encarga de transformar una expresión cadena (*tipo string*) introducida en el analizador de expresiones, a una expresión en memoria con forma de árbol (*tipo Expr*).

La primera función que debéis programar, se encarga de realizar la tarea inversa: recibe como parámetro una expresión de *tipo Expr*, y devuelve como resultado su representación como cadena (*string*). Para ello, y mediante la utilización de recursividad y de las funciones disponibles en la unit *ExprTree*, debéis ir recorriendo el árbol de expresión, sustituyendo cada nodo del árbol por su representación como cadena. *Recordad que:*

Nodos lista:	Head='List'	Terminal=""	→ { }
Nodos función:	Head='NombreFunción'	Terminal=""	→ NombreFunción[]
Nodos símbolo:	Head='Symbol'	Terminal='Símbolo'	→ Símbolo

La función solicitada, deberá ser declarada de la siguiente forma:

Function ExprToStr(X : Expr) : String;

Ejercicio 2.- Función TreeForm.

En este segundo ejercicio, vamos a programar una nueva función de la *unit ExprTree*, que nos permitirá visualizar en nuestro analizador de expresiones el árbol asociado a una expresión (*tipo Expr*), de forma análoga al resultado mostrado por la función "Tree" de Linux o ms-dos.

Debéis realizar el recorrido del árbol de expresión de forma recursiva "por niveles", desde el nivel 1 (raíz) hasta el nivel n (nodos terminales).

El procedimiento deberá ser declarado de la siguiente forma:

Procedure TreeForm (X : Expr);