

# Programación Declarativa

## 2007-2008

### Práctica 9

1. Revise las definiciones de las funciones **suma\_hasta**, **from\_to**, **hd**, **tl**, **length**, **nth** y **rev** (de las prácticas 4 y 5) para que se activen las excepciones correspondientes cuando sea oportuno y de modo que sólo se utilice recursividad terminal.
2. Defina las funciones **for\_all**, **exists**, **mem**, **find**, **filter**, **partition** y **merge** de forma que se comporten como las homónimas predefinidas en el módulo *List* (no pueden utilizarse las funciones de ese módulo).
3. Redefina las funciones **fold\_left** y **fold\_right** del módulo *List*, sin utilizar ese módulo.
4. Considere las siguientes definiciones efectuadas en ocaml

```

type 'a bintree = Empty | Node of 'a * 'a bintree * 'a bintree;;

let raiz = function
  Node (r,_,_) -> r
  | _ -> raise (Failure "raiz");;

let ramaizda = function
  Node (_,i,_) -> i
  | _ -> raise (Failure "ramaizda");;

let ramadcha = function
  Node (_,_,d) -> d
  | _ -> raise (Failure "ramadcha");;

```

Partiendo sólo de las funciones **raiz**, **ramaizda** y **ramadcha** (es decir, sin utilizar en ningún momento los constructores **Empty** y **Node**) defina en ocaml las siguientes funciones:

**num\_nodos**: 'a bintree -> int  
el número de nodos del árbol

**altura**: 'a bintree -> int  
el número de niveles del árbol (atención: altura del árbol vacío = 0)

**t\_sum**: int bintree -> int  
la suma de todos los nodos

**hojas**: 'a bintree -> 'a list  
la lista de hojas del árbol, de izda a derecha

**preorden**: 'a bintree -> 'a list  
todos los nodos del árbol en pre-orden

`t_max: 'a bintree -> 'a`  
el valor máximo de los nodos