

**PROGRAMACIÓN DECLARATIVA**  
13 DE SEPTIEMBRE DE 2006

NOMBRE: \_\_\_\_\_

I.I.

I.T.I.G

1. (6 puntos) Escriba el resultado de la compilación y ejecución de las siguientes frases, con tipos y valores, como lo indicaría el *toplevel* de *ocaml*:

```
let x, y = 2, 5;;
```

```
let f y = x + y;;
```

```
f (let f = function x -> x * x in f x);;
```

```
let v x f = f x;;
```

```
let cero x = v 0 x and dos x = v 2 x;;
```

```
let g = cero (-) in g 1;;
```

```
let h = dos (+);;
```

```
dos h;;
```

```
g 0, h 0;;
```

```
let doble = let vacia = "" in function  
  vacia -> vacia  
  | s -> s ^ s;;
```

```
doble "hola";;
```

```
let rec ap = function  
  [] -> 0 | h::t -> h (ap t);;
```

**2.** (2 puntos) Realice una nueva definición para la función  $f$ , de forma que sólo se utilice recursividad terminal.

```
let rec f = function
  [] -> 0
  | h::t -> h + 2 * f t;;
```

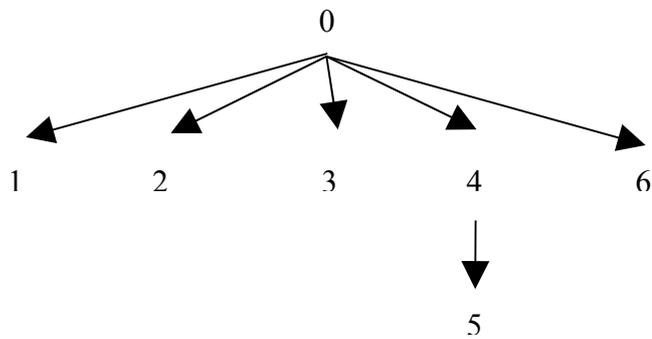
**3.** (2 puntos) Considere la siguiente definición en *ocaml* para el tipo de dato '*arbolgen*' (que sirve para representar árboles con nodos etiquetados con valores de tipo '*a*', en los que de cada nodo puede colgar cualquier número de ramas)

```
type 'a arbolgen = Nodo of 'a * 'a arbolgen list;;
```

de modo que, por ejemplo, el valor *ag1* definido por

```
let ag1 = Nodo (0, [Nodo (1, []); Nodo (2, []);  
                  Nodo (3, []); Nodo (4, [Nodo (5, [])]);  
                  Nodo (6, [])])
```

correspondería al árbol



Defina una función *nnodos* : 'a arbolgen -> int que devuelva el número de nodos de un árbol, de forma que, por ejemplo, *nnodos ag1 = 7*