

# Programación Declarativa

2007-2008

## Práctica 2

1. Intente deducir "a mano" cuál será el resultado de la compilación y ejecución de las siguientes expresiones. Escriba el resultado que espera y compruebe luego la respuesta real del compilador al introducir esas frases. Tome buena nota de los tipos de dato que vayan apareciendo, así como de todas las respuestas inesperadas y trate de explicar su razón. Si se produjera algún error, lea atentamente el mensaje del compilador y tome nota de la causa.

```
[1];;
```

```
[3;2;1];;
```

```
[1;"dos"];;
```

```
[1;2.0];;
```

```
2 :: [3;4];;
```

```
List.length [3;3;3;3];;
```

```
List.nth [5;4;3;2;1] 2;;
```

```
List.nth [5;4;3;2;1] 1 + List.nth [5;4] 0;;
```

```
[1;1+1;1+1+1];;
```

```
[1; 2] @ [3; 4];;
```

```
List.hd [1; 2; 3];;
```

```
List.tl [1; 2; 3];;
```

```
List.hd [0];;
```

```
List.tl [0];;
```

```
['a'; 'e'; 'i'; 'o'; 'u'];;
```

```
List.rev ['a'; 'b'; 'c'];;
```

```
[];;  
"hola" :: [];;  
2,5;;  
"hola", "adios";;  
0, 0.0;;  
fst ('a',0);;  
snd (false, true);;  
(1,2,3);;  
(1,2),3;;  
fst ((1,2),3);;  
[1,2,3];;  
fst;;  
snd;  
12, "octubre", 1492;;  
snd ((12, "octubre"), 1492);;  
snd (12, "octubre"), 1492;;  
snd ((12, octubre), 1492);;  
(1, 1+1, 1+1+1);;  
[1; 1+1; 1+1+1];;  
[1, 1+1, 1+1+1];;  
1, "dos", 3;;  
[1; "dos"; 3];;  
0 :: [1; 2; 3];;  
[];;
```

```
3 :: [];;

2 :: (3 :: []);;

1 :: (2 :: (3 :: []));;

(1 :: 2 :: 3 :: []) ;;

List.length ([1; 2] @ [3; 4]);;

List.append ['a'; 'e'; 'i'] (List.tl ['o'; 'u']);;

List.hd [];;

List.tl [];;

function x -> x;;

(function x -> x) (1,2,3);;

(function x -> x) int_of_char;;

function x -> x, x;;

(function x -> x, x) "hola";;

(function x -> x, x) 0;;

function x -> [x];;

(function x -> [x]) 0;;

function x -> List.hd x, List.tl x;;

(function x -> List.hd x, List.tl x) [3;2;1];;

(function x -> List.hd x, List.tl x) [[]];;

(function p -> fst p +. snd p);;

(function p -> fst p +. snd p) (1.5, 2.5);;

function p -> snd p, fst p;;

(function p -> snd p, fst p) (('a', true), "true");;

(function p -> snd p, fst p) ('a', true), "true";;

function x -> [x; [x]];;
```

```
function x -> x > 0;;  
  
(function x -> x > 0) 2;;  
  
(function x -> x > 0) (-3);;
```

2. Intente deducir “a mano” cuál será el resultado de la compilación y ejecución de las siguientes frases (expresiones y definiciones) si se introdujesen en ese orden en el compilador . Escriba el resultado como lo escribiría el compilador interactivo de ocaml y compruebe luego la respuesta real del compilador al introducir esas frases. Tome buena de todas las respuestas inesperadas y trate de explicar su razón. Si se produjera algún error, lea atentamente el mensaje del compilador y tome nota de por qué se ha producido.

```
let x,y,z = 1,2,3;;  
  
let x,y,z = z,x+z,x+y+z;;  
  
let doble x = 2 * x;;  
  
doble (x + y + z);;  
  
doble x + y + z;;  
  
let triple y = doble y + y;;  
  
triple x + doble x;;  
  
let f = function p -> doble (fst p) + triple (snd p);;  
  
f (x,y);;  
  
let swap p = snd p, fst p;;  
  
swap ("hola", "adios");;  
  
let dup = function x -> x, x;;  
  
dup 0;;
```

```
dup "hip";;
```

```
let pos x = x > 0;;
```

```
pos 2;;
```

```
pos (-3);;
```

```
pos - 3;;
```

```
pos 2.0;;
```

```
let absolut = function x -> if x < 0. then -.x else x;;
```

```
absolut (-10.) + absolut 10.;;
```

```
absolut -5.;;
```

```
absolut -.5;;
```

```
let longest p =  
  if String.length (fst p) > String.length (snd p)  
  then fst p else snd p;;
```

```
longest (longest ("uno", "dos"), "tres");;
```

```
let g x y = x + 2 * y;;
```

```
g 2 3;;
```

```
(g 2) 3;;
```

```
g (2 3);;
```

```
let h = g 0;;
```

```
h 5 + h 6;;
```