Programación Declarativa
2008-2009
Práctica 2

1. Escriba una **expresión** en Caml cuyo valor sea siempre el mismo que:

   - **`abs x`** (* sin utilizar la función **`abs`** *)
   - **`not x`** (* sin usar el operador **`not`** *)
   - **`max x y`** (* sin usar la función **`max`** *)
   - **`min x y`** (* sin usar la función **`min`** *)
   - **`string_of_bool b`** (* sin utilizar la función **`string_of_bool`** *)
   - **`x mod y`** (* sin utilizar el operador **`mod`** *)
   - **`x && (y || z)`** (* sin usar los operadores **`&&`** y **`||`** *)
   - **`Char.lowercase c`** (* sin usar la función **`Char.lowercase`** *)

2. Escriba en ocaml **definiciones** "alternativas" para las funciones **`abs`**, **`not`**, **`max`**, **`min`**, **`string_of_bool y lowercase`** , como si no estuviesen predefinidas.

3. 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 comprube 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.

```
"hola", "adios";;

0, 0.0;;

fst ('a',0);;

snd (false, true);;

(1,2,3);;

(1,2),3;;

fst ((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);;

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 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 > 0;;

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

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

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;;
```