

Paradigmas de Programación

Práctica 1

Nota Importante:

Cuando se solicite la entrega de esta práctica, cada alumno deberá subir a su repositorio de prácticas (del cual se indicará su ubicación más adelante) un directorio **p1** cuyo contenido debe ser únicamente los ficheros **nombre.ml**, **expr1.ml** y **expr2.ml**.

Sea muy cuidadoso a la hora de crear el directorio y los ficheros, y **respete los nombres indicados**. En particular, fíjese que todos estos nombres sólo contienen letras en minúsculas, números y puntos.

Además, **todos los ficheros deben compilar sin errores** con las siguientes órdenes:

```
ocamlc -o nombre nombre.ml
ocamlc -c expr1.ml
ocamlc -c expr2.ml
```

Ejercicios:

1. Construya un programa ejecutable **nombre** que escriba en la salida estándar dos líneas de texto: la primera con su nombre completo y la segunda con su dirección de correo electrónico en la UDC. El código fuente del programa, escrito en OCaml, debe guardarse en el fichero de texto **nombre.ml**.
2. Se trata de analizar la serie de expresiones OCaml incluidas en el fichero **expresiones.pdf**. Para ello, abriremos el compilador interactivo de OCaml y, con cada expresión del fichero, haremos lo siguiente:
 - La escribimos en el fichero de texto **expr1.ml** utilizando un editor (por ejemplo, **gedit**).
 - Debajo, y usando comentarios (***...***), intentamos predecir el resultado que dará OCaml sobre su compilación y ejecución, procurando usar la misma notación.
 - Copiamos la expresión en el terminal en el que tengamos abierto el compilador interactivo de OCaml y comprobamos el resultado. Si no es el previsto, lo corregimos e intentamos razonar por qué y en qué nos hemos equivocado.
 - Para toda expresión que produzca un error:
 - La escribiremos en el fichero de texto entre comentarios.
 - Indicaremos, también entre comentarios, el tipo de error (léxico, sintáctico, de tipo o de ejecución) y la causa del mismo.
 - Usaremos el manual del lenguaje para averiguar el significado de los operadores y funciones que aparecen en cada expresión.
 - Es importante poner entre comentarios todo aquello que se pide explícitamente que se escriba así, porque el fichero **expr1.ml** debe compilar (aunque obviamente si se genera el correspondiente programa ejecutable, éste no tendrá ningún efecto “visible”, porque el fichero **expresiones.pdf** no contiene instrucciones de entrada/salida).

3. Escriba en un fichero de texto `expr2.ml` un programa OCaml que defina (en este orden) los siguientes valores:
- Un valor `u` de tipo `int` a partir de una expresión que contenga, al menos, 4 operadores infijos.
 - Un valor `v` de tipo `float` a partir de una expresión que incluya una función predefinida.
 - Un valor `w` de tipo `char` a partir de una expresión que incluya una sub-expresión de tipo `int`.
 - Un valor `x` de tipo `bool` a partir de una expresión que incluya una o más funciones u operadores.
 - Un valor `y` de tipo `string` a partir de una expresión que contenga una frase `if-then-else`.