

Paradigmas de Programación

Práctica 1

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`.

El fichero `nombre.ml` **debe compilar sin errores** con la siguiente orden:

```
ocamlc -o nombre nombre.ml
```

Ejemplo de ejecución:

```
$ ./nombre
Jorge Graña Gil
jorge.grana@udc.es
$
```

2. Construya un programa ejecutable `pi` que calcule una buena aproximación del número π y muestre el resultado por la salida estándar seguido de un salto de línea. El código fuente del programa, escrito en OCaml, debe guardarse en el fichero de texto `pi.ml`.

El fichero `pi.ml` **debe compilar sin errores** con la siguiente orden:

```
ocamlc -o pi pi.ml
```

Ejemplo de ejecución:

```
$ ./pi
3.14159265359
$
```

3. Se trata ahora de analizar la serie de expresiones OCaml incluidas en el fichero `expresiones.pdf` que se proporciona junto con este mismo enunciado. Para ello, abriremos el compilador interactivo de OCaml y, con cada expresión del fichero, haremos lo siguiente:

- La escribimos en el fichero de texto `expr.ml` utilizando un editor de textos.
- Seguidamente, intentaremos predecir el resultado que proporcionaría OCaml sobre su compilación y ejecución, y lo escribimos como comentario en la línea siguiente, procurando usar la misma notación. En OCaml, los comentarios comienzan con `(*` y terminan con `*)`.
- Por último, 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 `expr.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).

El fichero `expr.ml` **debe compilar sin errores** con la siguiente orden:

```
ocamlc -c expr.ml
```

4. Construya un programa ejecutable `e` que calcule una buena aproximación del número e y muestre el resultado por la salida estándar seguido de un salto de línea. El código fuente del programa, escrito en OCaml, debe guardarse en el fichero de `e.ml`. En este ejercicio, el código fuente no puede contener ningún punto y coma.

El fichero `e.ml` **debe compilar sin errores** con la siguiente orden:

```
ocamlc -o e e.ml
```

Ejemplo de ejecución:

```
$ ./e
2.71828182846
$
```

Nota Importante:

Cuando se solicite la entrega de esta práctica, cada alumno deberá enviar únicamente los ficheros `nombre.ml`, `pi.ml`, `expr.ml` y `e.ml`.

Más adelante se darán indicaciones más precisas sobre la tarea de entrega y sobre el mecanismo de envío de los ficheros.

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

Además, **todos los ficheros deben compilar sin errores** con las órdenes anteriormente citadas.