

Paradigmas de Programación

Práctica 4

Ejercicios:

1. Realice las siguientes tareas en un fichero de texto `ej41.ml`:

- Defina (recursivamente) una función `sum_cifras: int -> int` que, para cualquier valor `n:int`, `n >= 0`, obtenga la suma de todas las cifras de la representación decimal del número `n`. Por ejemplo, debe cumplirse: `sum_cifras 1024 = 7`.
- Considere la siguiente función:

```
let num_cifras n = String.length (string_of_int (abs n))
```

Redefínala (recursivamente) sin utilizar el tipo de dato `string`.

- Defina (recursivamente) una función `exp10: int -> int` tal que, para cualquier valor `n:int`, `n >= 0`, `exp10 n` sea 10^n .
- Sin utilizar el tipo de dato `string`, defina (recursivamente) una función `reverse: int -> int` tal que, para cualquier valor `n:int`, `n >= 0`, `reverse n` sea el entero que se obtiene al invertir el orden de las cifras de la representación decimal de `n`. Por ejemplo, debe cumplirse:

```
reverse 3 = 3
reverse 12 = 21
reverse 20 = 2
reverse 10247 = 74201
reverse 0b101 = 5
reverse 0b100000 = 23
```

- Dado `s`, un valor de tipo `string`, la expresión `s.[i]` representa el `char` que ocupa la posición `i` dentro de `s` (el primer `char` ocupa la posición 0 y el último ocupa la posición `String.length s - 1`).

Defina (recursivamente) una función `palindromo: string -> bool` que indique si el `string` sobre el que se aplica se lee igual de izquierda a derecha que de derecha a izquierda. Por ejemplo, debe cumplirse:

```
palindromo "" = true
palindromo "a" = true
palindromo "elle" = true
palindromo "radar" = true
palindromo "reconocer" = true
palindromo "recoger" = false
palindromo "Ana" = false
```

El fichero `ej41.ml` debe compilar sin errores con la orden `ocamlc -c ej41.mli ej41.ml`.

2. Si x es un número entero cualquiera e y es un número entero mayor que 0, se cumple la siguiente propiedad:

$$x^y = x \times x^{y-1}$$

Utilizando directamente esta propiedad, defina (recursivamente) una función `power: int -> int -> int` tal que, para cualesquiera `x:int`, `y:int`, `y >= 0`, `power x y` tenga el valor de x^y . Por ejemplo, debe cumplirse: `power 2 10 = 1024`.

También son ciertas las siguientes propiedades (razónelas desde el punto de vista matemático):

$$\begin{aligned}x^y &= (x \times x)^{y/2}, & \text{si } y \text{ es par} \\x^y &= x \times (x \times x)^{y/2}, & \text{si } y \text{ es impar}\end{aligned}$$

Utilice directamente estas dos propiedades para definir (recursivamente) una función `power': int -> int -> int`, que sea una versión mejorada (en términos de eficiencia) de la definición anterior.

Explique por qué `power'` debería ser mejor que `power` en términos de eficiencia y razone si realmente merece la pena la ganancia obtenida al estar operando en `int` (y no en \mathbb{Z}).

Todo lo anterior sería igualmente válido para potencias de base real y exponente natural. Defina (recursivamente) una función `powerf: float -> int -> float`, tal que, para cualesquiera `x:float`, `n:int`, `n >= 0`, `powerf x n` tenga el valor de x^n ,

Realice todas las implementaciones de este ejercicio en un fichero de texto `power.ml`. Las explicaciones que se piden inclúyalas dentro de este mismo fichero entre comentarios. Este fichero debe compilar sin errores con la orden `ocamlc -c power.mli power.ml`.

3. Implemente en un fichero de texto `mcd.ml` una función `mcd: int * int -> int`, tal que, para cualesquiera `x:int`, `y:int`, `x >= 0`, `y >= 0`, (`x <> 0 || y <> 0`), `mcd (x,y)` sea el máximo común divisor de `x` e `y`.

Defina (recursivamente) esta función basándose en las siguientes propiedades:

$$\begin{aligned}mcd(x, y) &= mcd(y, x) \\mcd(x, y) &= mcd(x \bmod y, y), \text{ si } y > 0\end{aligned}$$

El fichero `mcd.ml` debe compilar sin errores con la orden `ocamlc -c mcd.mli mcd.ml`.

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 `p4` cuyo contenido debe ser únicamente los ficheros `ej41.ml`, `power.ml`, y `mcd.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 órdenes anteriormente citadas.