

Ejercicios Prácticos de Programación Declarativa

Sesión de laboratorio 1

Curso 2019/20

- Realizad los siguientes ejercicios en un mismo fichero .hs.
- Escribid vuestro nombre al comienzo del fichero como líneas comentadas.
- incluid comentarios significativos.
- Subid el fichero al Campus Virtual antes de que acabe la clase. Es suficiente con que lo suba uno si lo hacéis entre dos.

1. Definid expresiones Haskell para los siguientes apartados:

- a) Calculad cuántos años hay en 10^{10} segundos (supón que todos los años tienen 365 días; en otro momento puedes hacerlo teniendo en cuenta bisiestos).
- b) Calculad cuántos años enteros, días restantes enteros, horas restantes enteras, minutos restantes enteros y segundos restantes hay en 10^{10} segundos.
- c) Generalizad los dos apartados anteriores, convirtiendo el número de segundos en argumento de una función. Indicad en el programa los tipos de las funciones.

2. Pensad cuáles de las siguientes expresiones tardarán poco (digamos centésimas o milésimas de segundos), regular (digamos décimas o segundos) o mucho (digamos toda una vida) en ser evaluadas. Probadlo en el intérprete, pero no esperéis toda la vida, interrumpid el cómputo cuando sea necesario.

```
last [1..105]  
last [1..107]  
last [1..1020]  
head [1..1020]  
last [1020..1]  
head (tail [1..1020])  
length [1..1020]  
last (take (107) [1..1020])  
head (take (107) ([1..100] ++ [1..1020]))  
last (take 100 ([1..1020] ++ [1..100]))  
last (drop 100 ([1..1020] ++ [1..100]))  
head (drop (107) ([1..1020] ++ [1..100]))  
[1..107] == [1..107]  
[1..1020] == [1..1020]  
[1..1020] == [1..1020+1]  
[1..1020] == [2..1020]  
head (reverse [1..107])  
last (reverse [1..107])  
reverse [1..1020] == reverse [1..1020+1]
```

Escribid los resultados obtenido como comentarios en el fichero que entreguéis.

3. Programad la función `media` que calcula la media aritmética de una lista de números, usando para ello la función `length` que calcula el número de elementos de una lista. ¿Surge algún problema con los tipos? Utilizad la función de conversión `fromIntegral` para resolverlo.
4. Programad las siguientes funciones, declarando sus tipos:
 - `digitos x` = número de dígitos del número entero `x`.
 - `reduccion x` = resultado del proceso de sumar los dígitos del entero `x`, sumar los dígitos del resultado obtenido, y así sucesivamente hasta obtener un número menor que 10. La reducción de un entero negativo es la de su valor absoluto.
 - `perm n` = número de permutaciones de `n` elementos
 - `var n m` = número de variaciones de `n` elementos tomados de `m` en `m`
 - `comb n m` = número de combinaciones de `n` elementos tomados `m` en `m`
5. Definid la conjunción booleana por ajuste de patrones, pero de cuatro o cinco formas diferentes, cambiando el número de ecuaciones, o las combinaciones de patrones `True`, `False`, `x`, ... en cada ecuación, o el orden de ecuaciones, etc. y de manera que al menos haya una versión estricta en el primer argumento y otra estricta en el segundo, pero no en el primero. Para que coexistan todas definiciones en el mismo programa, dadles nombres (o usa operadores) diferentes.