

Ejercicios Prácticos de Programación Declarativa

Sesión de laboratorio 5

Curso 2019/20

- Subid un fichero `.hs` al Campus Virtual antes de que acabe la clase con los ejercicios de esta sesión. Es suficiente con que lo suba uno si lo hacéis entre dos.
- No olvidéis incluir comentarios y poner vuestros nombres en las primeras líneas del fichero.

Nota: Pueden ser útiles las funciones `words`, `unwords`, `lines`, `unlines` del `Prelude`. También puedes documentarte y emplear funciones predefinidas del tipo `IO a`, como `getLine`, `readFile`, `writeFile`, etc.

1. Escribe una función `adivina n` para jugar a adivinar un número. Debe pedir que el usuario introduzca un número hasta que acierte con el valor de `n`. Devuelve mensajes de ayuda indicando si el número introducido es menor o mayor que el número `n` a adivinar. Observa que el tipo de la función será `adivina :: Int -> IO ()`.
2. Escribe un programa que lea una línea introducida por teclado y muestre el número de palabras que contiene.
3. Programar funciones que realicen los siguientes procesos interactivos.

- a) `palabras :: String -> IO Int`
`palabras fileIn` obtiene como valor asociado el número de palabras de un fichero `fileIn`.
- b) `palabras' :: IO ()`
Como la anterior, pero leyendo el nombre del fichero de la entrada y mostrando el resultado con un mensaje de la forma: El fichero *Fichero* tiene *n* palabras.
- c) `promedia :: IO ()`
Proceso que va leyendo un entero de cada línea de la entrada y mostrando la suma y el promedio de los valores leídos hasta ese momento. El proceso se detiene al leer `-1` en la entrada.
- d) `formatea :: String -> String -> Int -> IO ()`
`formatea fileIn fileOut n` formatea a `n` columnas de ancho cada línea de `fileIn` y escribe el resultado en `fileOut`. Para formatear se meten espacios intermedios repartidos de manera uniforme entre palabras de modo que la línea quede justificada a izquierda y derecha. Puede suceder que, por su longitud, la línea quede con más de `n` columnas.
- e) *Si te animas:* `calculadora :: IO ()` (*Intérprete de expresiones aritméticas*)
Proceso que lee de la entrada una expresión aritmética (formada por números y operadores infijos `+,*,-,/`) y la evalúa.