

Taller: Introducción a GHCi

Ejercicios

1. Utilice `Ghci` para decidir si las expresiones $(2^{29})/(2^9)$ y 2^{20} son iguales. Recuerde que el operador de potenciación en `Hugs` es infijo y se escribe “`^`”.
2. Utilizando las funciones `head` y `tail`, y dada la lista “hola mundo”, obtenga el segundo elemento de la misma (la letra “o”).
3. Utilizando las funciones `head` y `reverse`, y dada la lista “hola mundo”, obtenga el último elemento de la misma (la letra “o”).
4. Podemos examinar los tipos de algunas expresiones a partir del comando `:t`, el cual, seguido de una expresión válida nos dice su tipo.
Ej:
`:t tail`
 $tail :: [a] \longrightarrow [a]$
Examina los tipos de las expresiones de los ejercicios que siguen.
5. Utilizando la función realizada en el ejercicio anterior y la función `mod` determine si un número, representado como la lista de sus dígitos (ej: $123 = [1,2,3]$) es par.
6. Utilizando la función `sum`¹, la función `mod` y un número representado de igual manera que en el [item 5] determine si un número es múltiplo de 3.
7. Utilizando las funciones de los [items 5, 6] determine si un número es múltiplo de 6.
8. Escriba una función que dado un número retorne la lista de sus dígitos.
9. Utilizando las funciones `reverse` y `==`² determine si una frase, representada como un string, es un palíndromo.
10. ¿Que arrojará como resultado la evaluación de la siguiente expresión en `Hugs`?

`(head.(drop 3)) "0123456"`

¹`sum` suma todos los elementos de una lista

²El operador de igualdad sobre listas

¿Que tipo tiene el valor resultante? ¿Que función sobre listas de las dadas en clase se podría implemntar de esta manera? ¿ Investigue qué comando se usa en ghci para saber el número de pasos realizados por el interprete.

11. Utilizando ghc compile alguna de sus funciones para obtener código ejecutable.
12. Utilice Cabal para crear un proyecto, el proyecto debe ejecutar la función **reverse** sobre una texto que se dé como entrada.