

Taller: Introducción a GHCI

Ejercicios

1. Instale una versión de `ghc` (*Glasgow Haskell Compiler*, el compilador e interprete para Haskell). Instale la herramienta *Stack*. Puede leer sobre Stack en <https://docs.haskellstack.org/en/stable/GUIDE/>
2. Utilice `ghci` para decidir si las expresiones $(2^{29})/(2^9)$ y 2^{20} son iguales. Recuerde que el operador de potenciación en `ghci` es infijo y se escribe “`^`”.
3. Utilizando las funciones `hd` y `tail`, y dada la lista “hola mundo”, obtenga el segundo elemento de la misma (la letra “o”).
4. Utilizando las funciones `hd` y `reverse`, y dada la lista “hola mundo”, obtenga el último elemento de la misma (la letra “o”).
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. Investigue las definiciones de las funciones `take` y `drop`, utilizando estas funciones implemente una función `cortar :: Int -> Int -> [Char] -> [Char]` que dados dos enteros `i` y `j` y un string `w`, devuelva el substring que se encuentra entre las posiciones `i` y `j`.
10. ¿Que arrojará como resultado la evaluación de la siguiente expresión en `ghci`?

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

¹`sum` suma todos los elementos de una lista

¿Qué tipo tiene el valor resultante? ¿Qué función sobre listas de las dadas en clase se podría implementar 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 Stack para crear un proyecto, el proyecto debe ejecutar la función **reverse** sobre una texto que se dé como entrada.