Departamento de Computación
FCEFQyN, Universidad Nacional de Río Cuarto
Asignatura: Programación Avanzada
Primer Cuatrimestre de 2024

Práctico 5: Programación Funcional: listas infinitas - funciones de alto orden - listas por comprensión

NOTAs: Les recomendamos, antes de comenzar a resolver los ejercicios, repasar la teoría.
Los ejercicios marcados con * son para resolver en su casa.

1. Generar una lista infinita de unos.

2. Generar una lista infinita de naturales comenzando desde un número dado.

3. Generar una lista con los primeros **n** naturales.

4 *. Retornar los primeros 5 elementos de una lista infinita de enteros positivos.

Utilizando funciones de alto orden resolver:

5. Dada una lista de enteros, retornar sus cuadrados, es decir, dado $[x_0, x_1, \ldots, x_n]$ deberia retornar $[x_0^2, x_1^2, \ldots, x_n^2]$

6. Dado un entero positivo, retornar la lista de sus divisores.

7. Dada una lista de naturales, obtener la lista que contenga solo los números primos de la lista original.

8 *. Dada una lista de naturales, retornar la suma de los cuadrados de la lista.

9. Dada una lista de naturales, retornar la lista con sus sucesores.

10. Dada una lista de enteros, sumar todos sus elementos.

11 *. Definir el factorial usando **fold**.

12 *. Redefinir la función `and` tal que `and xs` se verifica si todos los elementos de xs son verdaderos. Por ejemplo: `and [1<2, 2<3, 1/=0] = True`, `and [1<2, 2<3, 1 == 0] = False`.

13. Usando `foldl` o `foldr` definir una función `tam::[a]->Int` que devuelve la cantidad de elementos de una lista dada. Dar un ejemplo en los cuales `foldr` y `foldl` evaluen diferente con los mismos parametros.

Utilizando listas por comprensión resolver:

14. Dada una lista de enteros, retornar sus sucesores.

15 *. Dada una lista de naturales, retornar sus cuadrados.

16. Dada una lista de enteros, retornar los elementos pares que sean mayores a 10.

17. Dado un entero, retornar sus divisores.

18 *. Definir la función `todosOcurrenEn :: Eq a => [a] -> [a] -> Bool` tal que `todosOcurrenEn xs ys` se verifica si todos los elementos de xs son elementos de ys. Por ejemplo: `todosOcurrenEn [1,5,2,5] [5,1,2,4] = True`, `todosOcurrenEn [1,5,2,5] [5,2,4] = False`

19. Dado un natural **n**, retornar los números primos comprendidos entre 2 y **n**.

20. Dadas dos listas de naturales, retornar su producto cartesiano.

21 *. Dadas una lista y un elemento retornar el número de ocurrencias del elemento x en la lista ys.

22. Escribir la función `split2 :: [a] - > [([a],[a])]`, que dada una lista `xs`, devuelve la lista con todas las formas de partir `xs` en dos. Por ejemplo: `split2 [1,2,3] = [([],[1,2,3]), ([1],[2,3]), ([1,2],[3]),([1,2,3],[])]`.

23 *. Definir una función que, dada una lista de enteros, devuelva la suma de la suma de todos los segmentos iniciales.
Por ejemplo: `sumaSeg [1,2,3] = 0 + 1 + 3 + 6 = 10`.

24. Definir la lista infinita de los números pares.