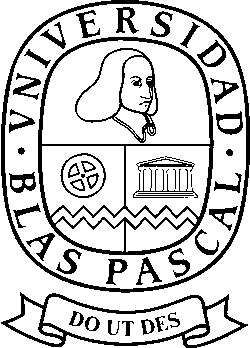
**Examen Parcial**

Carrera **:**  ING. INFORMÁTICA

Materia **:** PD

Alumno : Gino Ventura Turno : Div.: “ “

Tema**: Haskell Tema I** Legajo: Fecha : 16/09/2022



PUNTAJE: ­­\_\_\_\_\_\_\_\_\_\_\_\_\_ CALIFICACIÓN DEFINITIVA: \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Criterios:**

Escribir el código

Pegar un print de la ejecución

**Desarrollo**

1. Definir la función conc\_lista tal que conc l1 l2 es la concatenación de l1 y l2.

Por ejemplo, conc\_lista [2,3] [3,2,4,1] -> [2,3,3,2,4,1]

Nota: conc\_lista es equivalente a (++).

concatenarListas:: [Int] -> [Int] -> [Int]

concatenarListas [] ys = ys

concatenarListas (x:xs) ys = x : (concatenarListas xs ys)



1. Redefinir la función init tal que init\_list l es la lista l sin el último elemento.

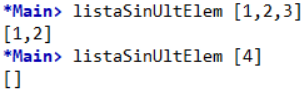
Por ejemplo, init\_list [1,2,3] -> [1,2]

Init\_list [4] -> []

listaSinUltElem :: [Int] -> [Int]

listaSinUltElem [x] = []

listaSinUltElem (x:xs) = x : listaSinUltElem xs



1. Redefinir la función length tal que length\_list l es el número de elementos de l.

Por ejemplo, length\_list [1,3,6] -> 3

longitudLista :: [Int] -> Int

longitudLista [] = 0

longitudLista (\_:xs) = 1 + longitudLista xs



1. Definir la función palíndromo tal que (palíndromo xs) se -- verifica si xs es un palíndromo; es decir, es lo mismo leer xs de -- izquierda a derecha que de derecha a izquierda.

Por ejemplo, palindromo [3,2,5,2,3] -> True

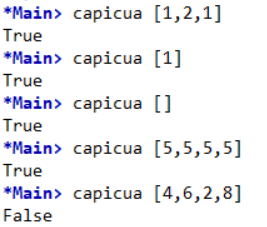
palindromo [3,2,5,6,2,3] -> False

capicua :: [Int] -> Bool

capicua [] = True

capicua [x] = True

capicua (x:xs) = reverse (x:xs) == (x:xs)



1. Implementar una función que devuelva una lista con los n elementos mayores de la lista original.

Por ejemplo, list\_may 3 [1,3,4,2,6] -> [3,4,6]

list\_may 2 [1.3.4.2.6] -> [4,6]

listaMayores :: Int -> [Int] -> [Int]

listaMayores n x = take n (x)