**Laboratorio 04**

**Realizar los siguientes programas**

1. Cree una función que ingrese dos elementos, 1 o 2 (par o impar ) y una lista, y retorna la lista de los pares/impares según corresponda (no usar función odd ni even).

listita :: Int -> [Int] -> [Int]

listita choice n

| choice == 1 = filter (\x -> x `mod` 2 == 0) n

| choice == 2 = filter (\x -> x `mod` 2 /= 0) n

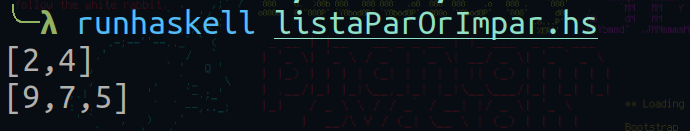
| otherwise = error "error"

main :: IO()

main = do

print( listita 1 [1,2,3,4,5])

print( listita 2 [9,8,7,6,5])



1. Cree una función que reciba una lista y evalúe si esta está ordenada (devuelve verdadero o falso).

order :: [Int] -> Bool

order [] = True

order[x] = True

order (x:y:o) = x <= y && order(y:o)

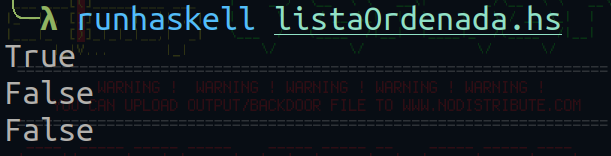
main :: IO()

main = do

print( order([1,2,3,4,5,6] ))

print( order([1,2,3,9,5,6] ))

print( order([6,5,4,3,2,1] ))



1. Cree una función que reciba dos listas ordenadas y devuelva una lista con los elementos de ambas listas (lista final ordenada).

listita :: [Int] -> [Int] -> [Int]

listita [] y = y

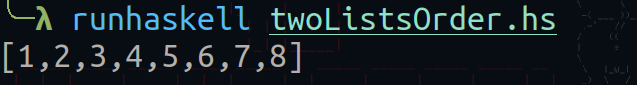
listita x [] = x

listita (k:x) (m:y) | k <= m = k : listita x (m:y) | otherwise = m : listita (k:x) y

main :: IO()

main = do

print $ listita [1,2,3,4] [5,6,7,8]



1. Definir la función que reciba un número y una lista de números, y devuelve verdadero si el número se encuentra en la lista.

listita :: Int -> [Int] -> Bool

listita \_ [] = False

listita x (y:o) | x == y = True | otherwise = listita x o

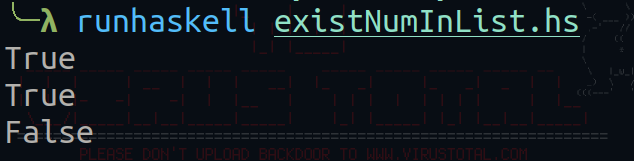
main :: IO()

main = do

print $ listita 2 [10,2,50,4]

print $ listita 50 [10,2,50,4]

print $ listita 9 [10,2,50,4]



1. Definir la función longitud\_camino, de tal forma que : longitud\_camino [(1,2),(4,6),(7,10)] es 10.0

distance :: (Double, Double) -> (Double, Double) -> Double

distance (x1, y1) (x2, y2) = sqrt((x2-x1)^2 + (y2-y1)^2)

longitud\_camino :: [(Double, Double)] -> Double

longitud\_camino [] = 0

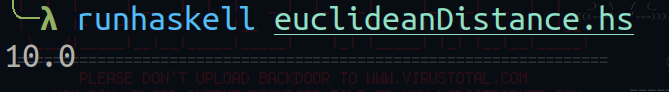
longitud\_camino [\_] = 0

longitud\_camino (point1:point2:ps) = distance point1 point2 + longitud\_camino (point2:ps)

main :: IO()

main = do

print $ longitud\_camino [(1,2),(4,6),(7,10)]



*\*Copiar el código y una imagen de la compilación correspondiente*