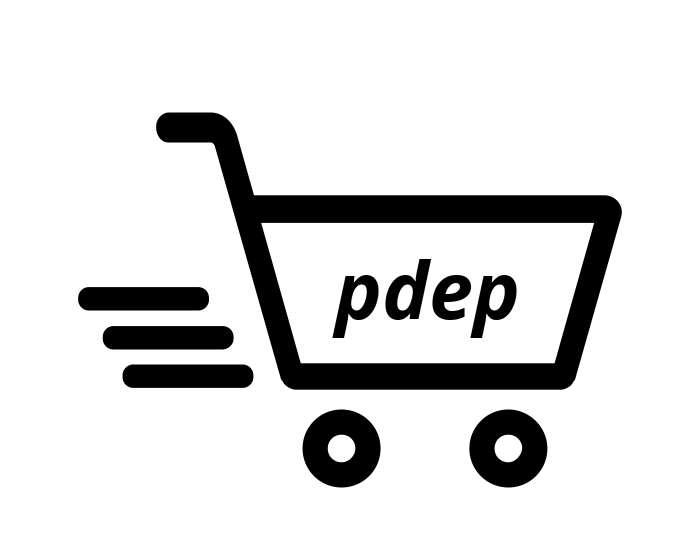
***PdeP - commerce***



En época de cuarentena, los sitios e-commerce son muy importantes para poder abastecer a la población sin que salga de sus hogares 🏠. Uno de los sitios más famosos de este estilo, nos pidió ayuda para finalizar algunas funcionalidades. Todo muy bien hasta acá, pero hay un pequeño detalle: el listado de funciones a desarrollar está **desordenado 😅**. Esto significa que algunas que aparecen al final son necesarias para realizar otras del principio. Vas a tener que pensar cuáles desarrollar primero 🤔. ¡Vamos!

**Todas las funciones pedidas deben estar tipadas.**

* precioTotal: Dado un precio unitario, una cantidad, un descuento y un costo de envío calcular el precio total. Para eso, hay que calcular el precio unitario con descuento y multiplicarlo por la cantidad. ¡No te olvides de agregar el precio del envío!
* productoDeElite: Un producto es de elite si es de lujo, codiciado y **no** es un producto corriente.
* aplicarDescuento: Dado un precio y un descuento, obtener el precio final con el descuento aplicado.
* entregaSencilla: Una entrega es sencilla, si se hace en un día sencillo. Los días sencillos son lo que tienen una cantidad de letras par en el nombre. Ejemplo de un día: “20 de Abril de 2020”.
* descodiciarProducto: Dado el nombre de un producto, generar uno que no sea codiciado. Para esto le vamos a sacar las últimas letras hasta que la cantidad de letras en el nombre quede igual a 10 (ó menor a 10 en productos con nombres cortos)
* productoDeLujo: Dado el nombre de un producto, saber si es de lujo. Un producto es de lujo cuando contiene una “x” o “z” en su nombre.
* aplicarCostoDeEnvio: Dado un precio y un costo de envío, obtener el precio final una vez sumado el costo de envío.
* productoCodiciado: Dado el nombre de un producto, saber si es un producto codiciado. Un producto es codiciado cuando la cantidad de letras en su nombre es mayor a 10.
* productoCorriente: Dado el nombre de un producto, saber si es un producto corriente. Un producto es corriente si la primera letra de su nombre es una vocal.
* productoXL: Dado un producto, conseguir su versión XL. Esta se consigue agregando ‘XL’ al final del nombre.
* versionBarata: Dado el nombre de un producto conseguir su versión barata. La misma es el producto descodiciado y con su nombre dado vuelta.

### Funciones que pueden utilizar (y tienen que **tipar**, las usen o no):

take :: Int -> String -> String

>> take 2 “Buenas!!”

“Bu”

>> take 5 “Saludos”

“Salud”

drop :: Int -> String -> String

>> drop 2 “Buenas!!”

“enas!!”

>> drop 3 “Saludos”

“udos”

head :: String -> Char

>> head “Buenas!!”

‘B’

>> head “Nos vemos”

‘N’

elem :: Char -> String -> Bool

>> elem ‘a’ “Buenas!!”

True

>> elem ‘y’ “Buenas!!”

False

reverse :: String -> String

>> reverse “Buenas!!”

"!!saneuB"

productoCorriente :: String -> Bool

productoCorriente nombreProducto = elem.head $ nombreProducto “AEIOU”

productoDeLujo :: String -> Bool

productoDeLujo nombreProducto = elem ‘x’ nombreProducto || elem ‘z’ nombreProducto

productoCodiciado :: String -> Bool

productoCodiciado nombreProducto = length nombreProducto > 10

descodiciarProducto :: String -> String

descodiciarProducto nombreProducto = reverse.drop.reverse $ nombreProducto (length nombreProducto - 10)

productoDeElite :: String -> Bool

productoDeElite nombreProducto = productoDeLujo nombreProducto && productoCodiciado nombreProducto && not (productoCorriente nombreProducto)

productoXL :: String -> String

productoXL nombreProducto = nombreProducto ++ “ XL”

versionBarata :: String -> String

versionBarata nombreProducto = reverse.descodiciarProducto $ nombreProducto

aplicarDescuento :: Num a => a -> a -> a

aplicarDescuento precio descuento = precio \* (descuento / 100)

entregaSencilla ::

entregaSencilla

precioTotal :: Num a => a -> a -> a -> a -> a

precioTotal precioUnitario cantidad descuento costoDeEnvio = (cantidad \* (aplicarDescuento precioUnitario descuento)) + costoDeEnvio