Parcial - tema2

Nota: 10.0 / 10.0 (APROBADO)

```
puntaje ej1: 2.5
puntaje ej2: 2
puntaje ej3: 2
puntaje ej4: 2
puntaje ej5: 0.75
puntaje ej6: 0.75
```

Datos del alumno

Nombre: Paloma Guevara

DNI: ***REDACTED***

LU: ***REDACTED***

Maquina: 45-03

Enunciado

Parcial Haskell - Tema 2

Importante

Template de funciones a implementar acá
Lista de funciones permitidas acá
Ejemplo de hunit acá

Enunciado

Resolver los siguientes ejercicios cuyas especificaciones en lenguaje semiformal figuran a continuación. Deben ser implementadas en Haskell utilizando los tipos requeridos y solamente las funciones que se ven en la materia Introducción a la Programación / Algoritmos y Estructuras de Datos I (FCEyN-UBA).

1. **Ejercicio 1 (2,5 puntos)**

   Se dice que n es un número abundante si la suma de sus divisores propios es mayor que n. Los divisores propios de un número son todos los divisores sin contar al número mismo. Por ejemplo, los divisores propios de 12 son 1, 2, 3, 4 y 6. La suma de los divisores propios de 12 es 1 + 2 + 3 + 4 + 6 = 16, que es mayor que 12. Por lo tanto, 12 es un número abundante.

   Se pide implementar cantidadNumerosAbundantes:

   problema cantidadNumerosAbundantes (*d*: Z,*h*: Z) : Z {
   requiere: {0 < d ≤ h}

asegura: {*res* es la cantidad de números abundantes en el rango [d..h]}
}

```
Ejemplo: cantidadNumerosAbundantes 12 24 debe devolver 4
```

2. **Ejercicio 2 (2 puntos)**
   Representaremos una *cursada aprobada* con una tupla String x Z x Z, donde:
   - La primera componente de la tupla contiene el nombre de una materia
   - La segunda componente de la tupla contiene el año de aprobación de la cursada
   - La tercera componente de la tupla contiene el cuatrimestre de aprobación de la cursada (el valor 0 representa un curso de verano)

   Se pide implementar cursadasVencidas, que dada una lista de cursadas devuelva aquellas materias cuya aprobación de la cursada ya venció, y por lo tanto ya no se puede rendir el final

   problema cursadasVencidas (*s*: seq⟨String x Z x Z⟩) :seq⟨String⟩ {
   requiere: { $s[i]_1 \geq 1993$ para todo i tal que $0 \leq i < |s|$}
   requiere: { $0 \leq s[i]_2 \leq 2$ para todo i tal que $0 \leq i < |s|$}
   asegura: { *res* no tiene elementos repetidos}
   asegura: { *res* contiene los nombres de todas las materias incluídas en *s* tales que la materia fue aprobada a más tardar en el primer cuatrimestre de 2021, inclusive}
   asegura: { *res* contiene solamente los nombres de las materias incluídas en *s* tales que la materia fue aprobada a más tardar en el primer cuatrimestre de 2021, inclusive}
   }

```
Ejemplo: cursadasVencidas [("Algoritmos y Estructuras de Datos I", 2020, 2), ("Algoritmos y Estructuras de Datos II", 2022, 1)] debe devolver ["Algoritmos y Estructuras de Datos I"]
```

3. **Ejercicio 3 (2 puntos)**

   problema saturarEnUmbralHastaNegativo (*s*: seq⟨Z⟩, *u*: Z) : seq⟨Z⟩ {
   requiere: {$u > 0$}
   asegura: { La longitud de *res* es igual a la cantidad de elementos no negativos consecutivos desde el inicio de *s* }
   asegura: {Para cualquier *i* en el rango $0 \leq i < |res|$ tal que $0 \leq s[i] \leq u$, se cumple que $res[i] = s[i]$}
   asegura: {Para cualquier *i* en el rango $0 \leq i < |res|$ tal que $s[i] > u$, se cumple que $res[i] = u$}
   }

```
Ejemplo: saturarEnUmbralHastaNegativo [3,8,5,0,7,-2,4] 5 debe devolver [3,5,5,0,5]
```

4. **Ejercicio 4 (2 puntos)**

   problema cantidadParesColumna (*matriz*: seq⟨seq⟨Z⟩⟩, *col*: Z) : Z{
   requiere: {Todos los elementos de la secuencia *matriz* tienen la misma longitud}
   requiere: {$|matriz| > 0$}
   requiere: {$|matriz[0]| > 0$}
   requiere: {$1 \leq col \leq |matriz[0]|$ }
   asegura: {*res* es la cantidad de números pares de los elementos *matriz*[i][col-1] para todo *i* tal que $0 \leq i < |matriz|$ }
   }

```
Ejemplo: cantidadParesColumna [[-9,8,2,3],[2,7,-5,3],[-1,0,5,6]] 2 debe devolver 2
```

5. **Ejercicio 5 (0,75 puntos)**

Conteste marcando la opción correcta.
¿Qué ocurre si una definición por pattern matching no contempla todos los casos posibles?

- ○ El programa no compila.
- ○ Haskell elige un valor por defecto automáticamente.
- ○ El programa puede lanzar un error en tiempo de ejecución si se invoca con un patrón no contemplado.

6. **Ejercicio 6 (0,75 puntos)**

Conteste marcando la opción correcta.
Dado un problema con parámetros *c* (de tipo Char) y *s* (de tipo String), cuya única precondición es *(esVocal(c)* ∨ *longitud(s)* > *3)*:

- ○ La precondición garantiza que siempre se trabajará con strings no vacíos.
- ○ Si *c* es una consonante y *s* tiene longitud igual a 2, no se garantiza el comportamiento correcto del programa.
- ○ Cualquier combinación de valores de *c* y *s* es válida, porque la precondición es una disyunción en vez de una conjunción.

Adjunta el archivo con tu solución:

Solo se puede adjuntar 1 archivo de extensión .hs. En caso de haber desarrollado tests propios, no deben ser entregados.

Seleccionar archivo a enviar: Browse... No file selected.

Enviar

Solucion entregada por el alumno

```
module SolucionT2 where

suma :: [Integer] -> Integer
suma [] = 0
suma (x:xs) = x + suma xs

divisoresPropiosAux :: Integer -> Integer -> [Integer]
divisoresPropiosAux i n | i == n = []
                        | mod n i == 0 = i:divisores
                        | otherwise = divisores
    where divisores = divisoresPropiosAux (i+1) n

divisoresPropios :: Integer -> [Integer]
divisoresPropios 1 = []
divisoresPropios n = divisoresPropiosAux 1 n

esNumeroAbundante :: Integer -> Bool
esNumeroAbundante n = suma (divisoresPropios n) > n

-- Ejercicio 1
cantidadNumerosAbundantes :: Integer -> Integer -> Integer
cantidadNumerosAbundantes d h
    | d > h               = 0
    | esNumeroAbundante d = resultado + 1
    | otherwise           = resultado
    where resultado = cantidadNumerosAbundantes (d+1) h
```

```
-------------------------------------

filtrarCursadas :: [(String, Integer, Integer)] -> [(String, Integer, Integer)]
filtrarCursadas [(materia,año,cuatri)]
    | (año > 2021) || (año == 2021 && cuatri == 2) = []
    | (año < 2021) || (año == 2021 && cuatri < 2)  =  [(materia,año,cuatri)]

filtrarCursadas ((materia,año,cuatri):cursadas)
    | (año > 2021) || (año == 2021 && cuatri == 2) = cursadasFiltradas
    | (año < 2021) || (año == 2021 && cuatri < 2)  =  (materia,año,cuatri) : cursadasFiltradas
    where cursadasFiltradas = filtrarCursadas cursadas

primerElementoDeCada :: [(String, Integer, Integer)] -> [String]
primerElementoDeCada [] = []
primerElementoDeCada [(x,_,_)] = [x]
primerElementoDeCada ((x,_,_):xs) = x : primerElementoDeCada xs

pertenece :: String -> [String] -> Bool
pertenece x [y] = x == y
pertenece x (y:ys) = x == y || pertenece x ys

eliminarRepetidos :: [String] -> [String]
eliminarRepetidos [] = []
eliminarRepetidos [x] = [x]
eliminarRepetidos (x:xs)
    | pertenece x xs = eliminarRepetidos xs
    | otherwise      = x : eliminarRepetidos xs

-- Ejercicio 2
cursadasVencidas :: [(String, Integer, Integer)] -> [String]
cursadasVencidas [] = []
cursadasVencidas x = eliminarRepetidos (primerElementoDeCada (filtrarCursadas x))

-------------------------------------

-- Ejercicio 3
saturarEnUmbralHastaNegativo :: [Integer] -> Integer -> [Integer]
saturarEnUmbralHastaNegativo [] _ = []
saturarEnUmbralHastaNegativo (x:xs) u
    | x < 0     = []
    | x <= u    = x : resultado
    | otherwise = u : resultado
    where resultado = saturarEnUmbralHastaNegativo xs u

-------------------------------------

listaVacia :: [t] -> Bool
listaVacia [] = True
listaVacia (_:_) = False

todasVacias :: [[t]] -> Bool
todasVacias [] = False
todasVacias [[]] = True
todasVacias (x:xs) = (listaVacia x) && todasVacias xs

elementoEnIndice :: [t] -> Integer -> t
elementoEnIndice (x:xs) 1 = x
elementoEnIndice (x:xs) i = elementoEnIndice xs (i-1)

sacarPrimeraColumna :: [[Integer]] -> [Integer]
```

```
sacarPrimeraColumna []             = []
sacarPrimeraColumna [[x]]          = [x]
sacarPrimeraColumna ([x]:filas)    = x : sacarPrimeraColumna filas
sacarPrimeraColumna ((x:xs):filas) = x : sacarPrimeraColumna filas

decapitarFilas :: [[Integer]] -> [[Integer]]
decapitarFilas [[x]]          = [[]]
decapitarFilas [x:xs]         = [xs]
decapitarFilas ([x]:filas)    = [] : decapitarFilas filas
decapitarFilas ((x:xs):filas) = xs : decapitarFilas filas

transponer :: [[Integer]] -> [[Integer]]
transponer [] = []
transponer x
    | todasVacias x = []
    | otherwise = sacarPrimeraColumna x : transponer (decapitarFilas x)

valorAbsoluto :: Integer -> Integer
valorAbsoluto x
    | x >= 0    = x
    | otherwise = -x

contarPares :: [Integer] -> Integer
contarPares [x]
    | mod (valorAbsoluto x) 2 == 0 = 1
    | otherwise                    = 0
contarPares (x:xs)
    | mod (valorAbsoluto x) 2 == 0 = 1 + contarPares xs
    | otherwise                    = contarPares xs

-- Ejercicio 4
cantidadParesColumna :: [[Integer]] -> Integer -> Integer
cantidadParesColumna a i = contarPares (elementoEnIndice (transponer a) i)
```

Resultado de la compilacion

```
Loaded package environment from /home/jgodoy/.ghc/x86_64-linux-9.4.7/environments/default
[1 of 1] Compiling SolucionT2       ( correcciones/parcial-haskell-tm/guevara_paloma_XXXXXXXX_X_X/submission.hs.main.hs, correcciones/parcial-haskell-tm/guevara_paloma_XXXXXXXX_X_X/submission.hs.main.o )
```

Ejecucion de los tests

tema2-test-ej1.hs.compilacion.out

Puntaje del ej: 2.5 / 2.5

```
Cases: 13  Tried: 1  Errors: 0  Failures: 0

Cases: 13  Tried: 2  Errors: 0  Failures: 0

Cases: 13  Tried: 3  Errors: 0  Failures: 0

Cases: 13  Tried: 4  Errors: 0  Failures: 0
```

```
Cases: 13  Tried: 5  Errors: 0  Failures: 0

Cases: 13  Tried: 6  Errors: 0  Failures: 0

Cases: 13  Tried: 7  Errors: 0  Failures: 0

Cases: 13  Tried: 8  Errors: 0  Failures: 0

Cases: 13  Tried: 9  Errors: 0  Failures: 0

Cases: 13  Tried: 10  Errors: 0  Failures: 0

Cases: 13  Tried: 11  Errors: 0  Failures: 0

Cases: 13  Tried: 12  Errors: 0  Failures: 0

Cases: 13  Tried: 13  Errors: 0  Failures: 0

Cases: 13  Tried: 13  Errors: 0  Failures: 0
```

tema2-test-ej2.hs.compilacion.out

Puntaje del ej: 2 / 2

```
Cases: 10  Tried: 1  Errors: 0  Failures: 0

Cases: 10  Tried: 2  Errors: 0  Failures: 0

Cases: 10  Tried: 3  Errors: 0  Failures: 0

Cases: 10  Tried: 4  Errors: 0  Failures: 0

Cases: 10  Tried: 5  Errors: 0  Failures: 0

Cases: 10  Tried: 6  Errors: 0  Failures: 0

Cases: 10  Tried: 7  Errors: 0  Failures: 0

Cases: 10  Tried: 8  Errors: 0  Failures: 0

Cases: 10  Tried: 9  Errors: 0  Failures: 0

Cases: 10  Tried: 10  Errors: 0  Failures: 0

Cases: 10  Tried: 10  Errors: 0  Failures: 0
```

tema2-test-ej3.hs.compilacion.out

Puntaje del ej: 2 / 2

```
Cases: 15  Tried: 1  Errors: 0  Failures: 0

Cases: 15  Tried: 2  Errors: 0  Failures: 0

Cases: 15  Tried: 3  Errors: 0  Failures: 0

Cases: 15  Tried: 4  Errors: 0  Failures: 0

Cases: 15  Tried: 5  Errors: 0  Failures: 0

Cases: 15  Tried: 6  Errors: 0  Failures: 0

Cases: 15  Tried: 7  Errors: 0  Failures: 0

Cases: 15  Tried: 8  Errors: 0  Failures: 0

Cases: 15  Tried: 9  Errors: 0  Failures: 0

Cases: 15  Tried: 10  Errors: 0  Failures: 0

Cases: 15  Tried: 11  Errors: 0  Failures: 0

Cases: 15  Tried: 12  Errors: 0  Failures: 0

Cases: 15  Tried: 13  Errors: 0  Failures: 0

Cases: 15  Tried: 14  Errors: 0  Failures: 0

Cases: 15  Tried: 15  Errors: 0  Failures: 0

Cases: 15  Tried: 15  Errors: 0  Failures: 0
```

tema2-test-ej4.hs.compilacion.out

Puntaje del ej: 2 / 2

```
Cases: 15  Tried: 1  Errors: 0  Failures: 0

Cases: 15  Tried: 2  Errors: 0  Failures: 0

Cases: 15  Tried: 3  Errors: 0  Failures: 0

Cases: 15  Tried: 4  Errors: 0  Failures: 0

Cases: 15  Tried: 5  Errors: 0  Failures: 0

Cases: 15  Tried: 6  Errors: 0  Failures: 0

Cases: 15  Tried: 7  Errors: 0  Failures: 0

Cases: 15  Tried: 8  Errors: 0  Failures: 0
```

```
Cases: 15  Tried: 9  Errors: 0  Failures: 0

Cases: 15  Tried: 10  Errors: 0  Failures: 0

Cases: 15  Tried: 11  Errors: 0  Failures: 0

Cases: 15  Tried: 12  Errors: 0  Failures: 0

Cases: 15  Tried: 13  Errors: 0  Failures: 0

Cases: 15  Tried: 14  Errors: 0  Failures: 0

Cases: 15  Tried: 15  Errors: 0  Failures: 0

Cases: 15  Tried: 15  Errors: 0  Failures: 0
```

mchoice.json-ej5.compilacion.out

Puntaje del ej: 0.75 / 0.75

```
mchoice ej5: respuesta del alumno=3, respuesta correcta=3

Ran 1 test in 0 seconds

OK
```

mchoice.json-ej6.compilacion.out

Puntaje del ej: 0.75 / 0.75

```
mchoice ej6: respuesta del alumno=2, respuesta correcta=2

Ran 1 test in 0 seconds

OK
```

FIN