

EXAME - 1ª Chamada

19·Janeiro·2005

Duração: 2:00 horas

Paradigmas de Programação I

LESI

Nº: _____ NOME: _____

I

Decidiu-se organizar a informação referente ao registo de temperaturas (mínima e máxima de cada dia) numa árvore binária de procura (ordenada pela data). Para isso declararam-se os seguintes tipos de dados:

```
type Data = (Int,Int,Int)
type TempMin = Int
type TempMax = Int
type Registo = (Data , TempMin, TempMax)
data Temperaturas = Vazia | Nodo Registo Temperaturas Temperaturas
```

1. Defina uma função

```
temps :: Temperaturas -> Data -> Maybe (TempMin,TempMax)
```

que permita saber as temperaturas de um determinado dia.

2. Defina uma função

```
maxTemp :: Temperaturas -> TempMax
```

que calcula a temperatura mais alta registada em toda a árvore.

3. Defina `Temperaturas` como instância da classe `Show` de forma a que visualização da árvore resulte numa listagem da informação ordenada por data (com um registo por linha).

Nº: _____ NOME: _____

II

Considere a seguinte função de I/O em Haskell:

```
g s = do { putStrLn s ;  
          x <- getLine ;  
          return ((read x):: Int)  
        }
```

1. Qual o tipo desta função?
2. Escreva uma função que utilize esta para ler dois números inteiros e retorne a sua soma ou subtração, consoante o valor de um parâmetro booleano que deverá receber.
3. Escreva uma função que funcione como um pequeno menu, e que utilize a função anterior para calcular somas e subtrações entre dois números inteiros, consoante a necessidade do utilizador.
4. O que seria necessário fazer para inserir este conjunto de funções num programa executável Haskell.

Nº: _____ NOME: _____

III

A função

```
split p l = foldr f ([],[]) l
  where   f x (s,n) = if (p x) then (x:s,n) else (s,x:n)
```

divide uma lista `l` em duas listas (`sim,nao`), sendo `sim` a lista dos elementos de `l` que satisfazem uma propriedade `p` (passada como argumento) e `nao` a lista dos que não satisfazem essa propriedade. Recordemos que `foldr` se define

```
foldr :: (a -> b -> b) -> b -> [a] -> b
foldr f z []      = z
foldr f z (x:xs) = f x (foldr f z xs)
```

1. Qual é o tipo mais genérico da função `split`?
2. Considere

```
sep n m = split (\x -> x < n) [0..m]
```

Exemplifique o funcionamento desta função apresentando a sequência de expressões intermédias que resultam do cálculo de `sep 3 7`.

3. Construa, usando `split`, uma função que ordena os elementos de uma lista de forma crescente.

