

<b>Departamento de Matemática da Universidade de Coimbra</b>		
<b>2008/2009</b>	<b>Programação Funcional</b>	<b>Folha 2 (v1.5)</b>

1. Como declararia, usando a construção para definição de tipos de dados `data`, o tipo primitivo `List`?
2. Considere a seguinte definição do tipo parametrizado `ArvBin a` (árvore binária):

```
data ArvBin a = Vazia | Folha a | Nodo (ArvBin a,ArvBin a)
```

- (a) Que informação nos podem dar os interpretadores de *Haskell* acerca desta definição (`info ...`)?
  - (b) Construa um valor `dic` deste tipo que represente uma árvore binária de palavras (e.g. índice de um dicionário);
  - (c) Defina uma função `contorno` que, dado um valor do tipo `ArvBin a`, devolva todos os elementos afixados nas folhas da árvore;
  - (d) Defina uma função `contornos_iguais` que, dadas duas árvores `ab1` e `ab2`, determine se os mesmos elementos ocorrem na mesma ordem, independentemente das estruturas internas de `ab1` e `ab2`.  
Note que uma solução correcta mas insatisfatória (porquê?) seria  
`contornos_iguais(x,y) = contorno(x) == contorno(y)`
3. Dado o tipo `ArvBin` definido na questão 2 construa uma função `maptree :: (a -> b) -> ArvBin a -> ArvBin b` que aplique uma dada função `a` a todos os elementos da árvore binária, dando como resultado uma nova árvore binária.
  4. Dado o tipo `ArvBin` definido na questão 2 construa funções *Haskell* para:
    - (a) percorrer a árvore em ordem;
    - (b) percorrer a árvore em pré-ordem;
    - (c) percorrer a árvore em pós-ordem.

em todos os casos a função deve devolver a lista dos elementos constitutivos da árvore.

5. Considere a seguinte definição do tipo parametrizado `ArvBinEA a` (árvore binária de expressões aritméticas):

```
data ArvBinEA a = Vazia | Folha a | Nodo (Char,ArvBinEA a,ArvBinEA a)
```

Considerando este tipo de dados podemos usá-lo para guardar expressões aritméticas assim como calcular o seu valor.

- (a) Dê exemplos de árvores deste tipo.
  - (b) Calcule o valor da expressão aritmética guardada na árvore usando uma das formas de percorrer a árvore vistos na pergunta 4.
  - (c) A forma de percorrer a árvore tem alguma influência no valor calculado? Porquê?
6. O tipo abstracto `Conj` (conjunto) pode ser implementado por:

```
module Conj(Conj,sing,união,inter,pertence,subconj,diff) where
data Conj a = Cc [a]
deriving (Eq,Show)
vazio :: Conj a
```

```

vazio = Cc []
sing :: a -> Conj a
sing x = ...
união :: Conj a -> Conj a -> Conj a
união (Cc c1) (Cc c2) = ...
inter :: Conj a -> Conj a -> Conj a
inter ...
pertence :: a -> Conj a -> Bool
pertence ...
...

```

(a) Complete a definição;

(b) Modifique a sua solução de modo a que, o conjunto seja representado por uma lista ordenada.

7. Escreva um tipo abstracto `Conj1` para representar conjuntos definidos pela sua função característica, ou seja sob a forma  $\{x \mid f(x)\}$ . A definição do tipo deve incluir as funções apresentadas no exercício anterior.
8. Construa um tipo abstracto polimórfico `Rel` que modele relações binárias com as operações usuais de adição de um par, projecção e inversão.
9. Pretende-se escrever uma especificação *Haskell* de polinómios de coeficientes reais usando, como modelo, sequências de pares (coeficiente, expoente), ordenados por expoentes. Defina o tipo, com duas operações que façam respectivamente, a soma e a multiplicação destes polinómios.
10. Defina uma implementação do Tipo Abstracto de Dados `Fila`, na forma de um módulo em *Haskell*.
11. Defina uma implementação do Tipo Abstracto de Dados `Pilha`, na forma de um módulo em *Haskell*.
12. Considere a seguinte declaração de uma árvore genealógica:

```

type Nome = String
type Sexo = Int
data Pessoa = Ps (Nome,Sexo)
data ArvGen = Ind Pessoa | Fam (Pessoa,Pessoa,[ArvGen])

```

- (a) Escreva as operações que achar necessário para gerir árvores deste tipo, por exemplo `criar`, `indivíduo`, `família`, `casamento`, `nascimento`, entre outras.
- (b) Escreva uma função que recolha numa lista todos os indivíduos pertencentes à árvore genealógica.
- (c) Escreva uma função que realize um teste de consistência a todas as famílias da árvore, verificando que os cônjuges têm sexo diferente.

13. Pretende-se criar um modelo de um “stock” de matérias primas, na forma de um tipo abstracto em *Haskell*. Cada objecto no “stock” tem associado informação respeitante a: código; preço; quantidade; lista de fornecedores e data da última encomenda. As operações pretendidas são as usuais, a saber, obtenção do registo de um objecto, actualização de um registo, listagem de todos os fornecedores potenciais da empresa, remoção e inserção de um objecto. Discuta o modelo adequado ao “stock” e implemente-o em *Haskell*.