Departamento de Matemática da Universidade de Coimbra		
2002/2003	**Especificação e Verificação**	**Folha 2** (v1.4)

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. 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.

4. 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.

5. Construa um tipo abstracto polimórfico `Rel` que modele relações binárias com as operações usuais de adição de um par, projecção e inversão.

6. 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.

7. Defina uma implementação do Tipo Abstracto de Dados `Fila`, na forma de um módulo em Haskell.

8. Defina uma implementação do Tipo Abstracto de Dados `Pilha`, na forma de um módulo em Haskell.

9. 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.

10. 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.