

- O exame deve ser entregue como um arquivo texto com a extensão .hs.
- Todas questões devem estar no arquivo.
- O número de cada questão deve ser informado imediatamente antes da implementação como um comentário.

- ✓ 1. (1,5 ponto) Declare uma função que receba como parâmetros uma lista e um dado  $d$ , a função deve retornar uma dupla de listas, a primeira lista deve conter todos os elementos menores que  $d$  e a segunda todos os elementos maiores.

Assinatura:  $[a] \rightarrow a \rightarrow ([a], [a])$

Exemplo: maioresMenores [1, 4, 7, 10, 8, 9, 3, 2] 7 => ([1, 4, 3, 2], [10, 8, 9])

- ✓ 2. (1,5 ponto) Declare uma função que receba como parâmetros uma *string* e um caractere  $c$ , a função deve retornar a *string* recebida substituindo todas ocorrências do caractere  $c$  pelo caractere '\*'.

Assinatura:  $[\text{Char}] \rightarrow \text{Char} \rightarrow [\text{Char}]$

Exemplo: marcar "teste" 'e' => "t\*st\*"

- ✓ 3. (1,5 ponto) Declare uma função que receba como parâmetros um elemento e uma lista previamente ordenada, a função deve inserir esse elemento na lista tendo como resultado uma lista ordenada contendo o novo elemento. A lista não deve ser reordenada, a função deve percorrer a lista até achar a posição correta.

Assinatura:  $a \rightarrow [a] \rightarrow [a]$

Exemplo: inserir 10 [1, 4, 7, 9, 12, 14] => [1, 4, 7, 9, 10, 12, 14]

- ✓ 4. (2,0 pontos) Declare uma função que receba duas listas previamente ordenadas e faça a intercalação (merge) dos elementos tendo como resultado a junção das duas listas em uma lista também ordenada. A lista não deve ser reordenada, deve ser feita a intercalação dos elementos.

Assinatura:  $[a] \rightarrow [a] \rightarrow [a]$

Exemplo: intercalacao [5, 15, 17, 20] [1, 2, 13, 15, 22] => [1, 2, 5, 13, 15, 17, 20, 22]

- ✓ 5. (1,5 ponto) Declare uma função que retorne o número de palavras em uma *string*.

Assinatura:  $(\text{Num } a) \Rightarrow [\text{Char}] \rightarrow a$

Exemplo: contaPalavras "Curso de Ciencia da Computacao" => 5

6. (2,0 pontos) Considerando o tipo algébrico de dados *Arvore*, declare uma função similar a função *map* para esse tipo de dado. A função deve aplicar uma função passada como parâmetro em todos os elementos armazenados na árvore.

data Arvore a = Galho a (Arvore a) (Arvore a) | Folha deriving Show

Assinatura:  $(a \rightarrow b) \rightarrow \text{Arvore } a \rightarrow \text{Arvore } b$

Exemplo: `mapArv (+1) (Galho 10 (Galho 8 Folha Folha) (Galho 18 Folha Folha))`

`=> (Galho 11 (Galho 9 Folha Folha) (Galho 19 Folha Folha))`