

# Linguagens de Programação – Prof. Vander Alves

## Lista de exercícios sobre Haskell

1) Defina uma função que retorne o maior entre quatro inteiros.

```
maior4 :: Int -> Int -> Int -> Int -> Int
```

2) Defina uma função que receba uma nota e retorne a menção do aluno. A nota é um valor do tipo **Float** entre 0.0 (inclusive) e 10.0 (inclusive) e a menção é uma string. Considere a seguinte tabela para tradução da menção:

Intervalo da nota	Menção
De 9 a 10	"SS"
De 7 a 8.9	"MS"
De 5 a 6.9	"MM"
De 3 a 4.9	"MI"

```
converterNotaParaMencao :: Float -> String
```

3) Implemente funções que satisfaçam a cada um dos requisitos abaixo:

- Retorna a diferença entre duas listas. O resultado é uma lista.
- Retorna a interseção entre duas listas. O resultado é uma lista.
- Retorna a união entre duas listas (pode haver repetição de elementos). O resultado é uma lista.
- Retorna a união entre duas listas (não há repetição de elementos). O resultado é uma lista.
- Retorna o último elemento de uma lista.
- Retorna o n-ésimo elemento de uma lista.
- Inverte uma lista.
- Ordena uma lista em ordem decrescente, removendo as eventuais repetições de elementos.
- Retorna um booleano indicando se uma lista de inteiros é decrescente ou não. Proponha 3 soluções: usando **sort**; usando apenas recursão; usando **fold**, **map** e **zip**.

4) Defina uma função que recebe uma lista de strings como entrada e computa uma lista de pares de (**String**, **Int**) representando o histograma (o número de ocorrência) de seus elementos:

```
histograma :: [String] -> [(String, Int)]
```

5) Defina a função **myZipWith**, que tem como parâmetros uma função binária (que tem dois parâmetros) e duas listas, retornando uma lista de valores resultantes da aplicação dessa função nos elementos correspondentes dessas listas:

```
myZipWith :: (a -> b -> c) -> [a] -> [b] -> [c]
```

6) Resolva em Haskell o seguinte problema: a partir de duas notas das provas de cada aluno, determinar a lista dos alunos aprovados, com suas respectivas médias. O resultado deve estar ordenado crescentemente pela média aritmética das notas. A aprovação ocorre se, e somente se, tal média é maior ou igual a 5.0.

```
aprovadosOrdemDeMedia :: [(String,Float,Float)] -> [(String,Float)]
```

7) Considere a representação de matrizes como lista de listas em que cada elemento da lista é uma lista que representa uma linha da matriz. Com base nisso, determine as seguintes funções:

a) some duas matrizes de inteiros

```
somaMatricial :: [[Int]] -> [[[Int]] -> [[[Int]]]
```

b) compute a transposta de duas matrizes de inteiros

```
matrizTransposta :: [[Int]] -> [[Int]]
```

c) compute a multiplicação de duas matrizes de inteiros

```
multiplicacaoMatricial :: [[Int]] -> [[Int]] -> [[Int]]
```

8) Com relação aos slides de *Tipos Algébricos*, estenda o tipo **Expr** para poder também representar multiplicação. Altere também a definição da função de avaliação **eval**

9) Crie a função **foldTree**, que recebe uma função e uma árvore polimórfica binária como parâmetros, e retorna o valor resultante de acumular a aplicação dessa função por todos os nós da árvore.

10) Defina uma função que some os elementos de uma árvore binária que armazena inteiros em seus nós. Resolva o exercício de duas formas diferentes: a) usando a função **foldTree** definida acima; b) sem usar a função **foldTree**

11) Refaça o Exercício 2, usando um tipo algébrico para modelar a menção. Discuta, em no máximo quatro linhas, vantagens e desvantagens da soluções.