

LISTA DE EXERCICIOS 04 – UNIDADE 2

PROGRAMACAO IMPERATIVA E FUNCIONAL – TURMA A E B – 2021.1

-
- **Deadline:** Quarta-feira, dia 17/11/2021, as 23:59.
 - **Conteúdo:** Listas, Tuplas e tipos algébricos
 - **Linguagem:** Haskell
 - Submeta todas as funções em um arquivo.hs único no formato L4_login.hs (por exemplo, minha submissão seria L4_ptlb.hs). Indique a qual questão cada função pertence através de comentários no código. **Qualquer arquivo enviado fora desse padrão será automaticamente descartado;**
 - Não use funções padrões do Haskell (como length, tail, sum, reverse, etc), exceto quando exigido claramente pela questão. **O uso indevido de funções padrões resultará na anulação da questão.**
 - O nome das funções implementadas **deve ser igual ao indicado pela questão**, caso contrário o script de correção não conseguira identificar sua questão e **ela será automaticamente zerada.**
 - Se o tipo da função não está claramente definido na descrição da questão, considere uma função que recebe uma lista de inteiros e retorna um inteiro ou lista de inteiros;
 - É permitida a criação de funções auxiliares. Também é permitida a criação de variáveis para testar suas funções.
-

- 1- Crie uma função que recebe uma lista de listas de valores do tipo Int e devolve uma lista de pares (Int, Int) em que o primeiro elemento é um valor Int que existe em pelo menos uma das sub-listas da entrada e o segundo é o número de ocorrências desse valor nas sub-listas:

EX: `group [[4,2,4,3,4,4,4,5,4,6], [1,2,3,4,5], [2]] => [(4,7), (2,3),(3,2),(5,2),(6,1),(1,1)]`

EX: `group [[1,7,8],[9],[7,7,1],[1]] => [(1,3),(7,3),(8,1),(9,1)]`

EX: `group [[1]] => [(1,1)]`

EX: `group [] => []`

- 2- Defina uma tupla *type Aluno* que contenha as seguintes informações: (1) código do aluno (String 4 caracteres – exemplo “B1S3”), (2) nome completo do aluno (String), (3) idade (Int), (4) genero (‘F’ para feminino, ‘M’ para masculino, ‘NB’ para não-binário) e (5) nota do enem (float de 0 a 10).

- a. Defina uma função *mediaEnem :: [Aluno] -> Float* que, dado uma lista de alunos cadastrados, retorna a média das notas do Enem destes alunos.

EX: *mediaEnem* [(“BB12”, “Pamela Bezera”, 19, ‘F’, 5.6), (“C1F4”, “Victor Hazin”, 21, ‘M’, 7.9)] => 6.75

EX: *mediaEnem* [(“GM07”, “Sanji Vinsmoke”, 22, ‘M’, 8.0)] => 8.0

- b. Defina uma função *minMaxIdade :: [Aluno] -> (String, Int, String, Int)* que, dado uma lista de alunos, retorne uma tupla onde os dois primeiros elementos é o nome e a idade do aluno mais novo e os dois ultimos elementos é o nome e a idade do aluno mais velho.

EX: *minMaxIdade* [(“BB12”, “Pamela Bezera”, 19, ‘F’, 5.6), (“C1F4”, “Victor Hazin”, 21, ‘M’, 7.9), (“GM07”, “Sanji Vinsmoke”, 22, ‘M’, 8.0), (“O95T”, “Shanks”, 30, ‘NB’, 4.5)] => (“Pamela Bezerra”, 19, “Shanks”, 30)

Defina um novo tipo *data Disciplina* que contém as seguintes informações sobre uma disciplina: (1) código da disciplina (String de 3 caracteres – exemplo “PIF”) e (2) nome da disciplina (String).

- c. Crie uma função *cadastro :: Disciplina -> Aluno -> [(String, String, Float)] -> [(String, String, Float)]* que, dado uma disciplina e um aluno, insere uma tupla contendo o codigo da disciplina, o nome do aluno, e a nota do aluno do enem (nesta ordem) em uma lista de cadastro.

EX: *cadastro* (Disc “PIF3” “Programação Imperativa e Funcional”) (“BB12”, “Pamela Bezera”, 19, ‘F’, 5.6) [(“AED4”, “Victor Hazin”, 7.9), (“LG87”, “Sanji Vinsmoke”, 8.0)] => [(“AED4”, “Victor Hazin”, 7.9), (“LG87”, “Sanji Vinsmoke”, 8.0), (“PIF3”, “Pamela Bezerra”, 5.6)]