



DESCRIÇÃO

A atividade consiste em resolver os exercícios abaixo utilizando o paradigma funcional, na linguagem Haskell. Lembre-se: sempre que for necessário, crie funções auxiliares para resolver a tarefa.

1. Descubra o que faz a função `show`.
2. Declare uma função que retorne os n últimos elementos de uma lista.
3. Declare uma função que retorne a interseção entre duas listas sem os elementos repetidos.
4. Declare uma função que receba um inteiro e converta para binário. Utilize `String` para representar o binário.
5. Declare uma função que receba como argumento uma lista de duplas e retorne uma lista contendo as duplas cujo primeiro elemento é um número par.
6. Utilizando `filter`, declare um comando que receba uma `String` e elimine os caracteres que representam dígitos.
7. Declare uma função que receba uma lista de `String` e concatene cada `String` com o próprio tamanho. Ex: entrada `["Joao", "Maria", "oi"]` → saída `["Joao4", "Maria5", "oi2"]`. **(EXTRA)** Resolver utilizando `zipWith`.
8. Utilizando `zipWith`, declare uma função que retorne uma lista de `String` dobrada. Ex: entrada `["Joao", "Maria", "oi"]` → saída `["JoaoJoao", "MariaMaria", "oi oi"]`.
9. **(EXTRA)** Declare uma função que liste infinitamente os números primos. Interrompa a execução (CTRL+C) quando encontrar o primeiro número primo com seis dígitos (100003).
10. Pesquise e crie um exemplo que utilize `$` substituindo o uso de parênteses.
11. Revise suas respostas! Faça este exercício por último.

PESO DA AVALIAÇÃO

Notas de aula.

OBSERVAÇÕES



INSTITUTO FEDERAL

Catarinense

Campus Blumenau

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

PARADIGMAS DE PROGRAMAÇÃO

LISTA DE EXERCÍCIOS

- Plágio = **ZERO** (inclui cópia ou simples alteração de trabalho de colegas)