LISTA DE EXERCICIOS 05 – UNIDADE 2

PROGRAMACAO IMPERATIVA E FUNCIONAL – TURMA A E B – 2021.2

- **Deadline: Quarta-feira, dia 01/12/2021, as 23:59.**
- **Conteúdo: Funções de alta ordem**
- **Linguagem: Haskell**
- Submeta todas as funções em um arquivo.hs único no formato L5_login.hs (por exemplo, minha submissão seria L5_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 resultara 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- Defina a função ***twice*** a qual recebe uma **função f** qualquer dos inteiros para os inteiros (*f :: Int->Int*) e uma entrada também do tipo inteiro (*a:: Int*) e retorna a função f aplicada duas vezes a entrada a. Por exemplo, se tivermos f = double (função que duplica um número) e a = 7, o retorno de twice será 28.

EX: twice double 7 => 28 (duplica 7 duas vezes)

EX: twice power 2 = 16 (eleva dois ao quadrado duas vezes)

2- Defina a função ***filterFirst :: (a->Bool) ->[a]->[a]*** onde filterFirst p xs remove o primeiro elemento de xs que não tem a propriedade p.

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

3- Defina uma função ***switchMap*** a qual mapeia duas funções ***f1*** e ***f2*** a longo dos elementos de uma lista, alternando quais funções são aplicadas a quais elementos: f1 eh aplicada aos elementos ímpares (primeiro, terceiro, quinto, etc) enquanto que f2 é aplicada aos elementos pares (segundo, quarto, sexto, etc).

EX: switchMap double addOne [1,2,3,4] => [2,3,6,5]

- Double = duplica os elementos = primeira entrada -> aplicada aos elementos de posições ímpares (i.e., primeiro, terceiro, quinto, ...)
- addOne = adiciona 1 ao elemento = segunda entrada -> aplicada aos elementos de posições pares (segundo, quarto, sexto, etc);