

Haskell – Exercício Treino – Funções de Alta Ordem

Obs: Algumas funções podem funcionar com qualquer tipo. Nesses casos, não defina o cabeçalho dessas funções.

1. Implemente uma função que equivale à função `foldl1`.
2. Implemente uma função que equivale à função `foldr1`.
3. Implemente uma função que recebe uma função `a->Bool` e uma lista e retorna `True` se algum elemento dessa lista obtém `True` com a função de entrada, ou `False`, caso todos os elementos obtenham `False`.
4. Implemente uma função que recebe uma função `a->Bool` e uma lista e retorna o primeiro elemento dessa lista que obtém `True` com a função recebida. Se nenhum elemento obter `True` com a função de entrada, o programa deve acusar erro.
5. Implemente uma função que recebe um valor e uma lista de funções unárias e aplica cada função sobre o resultado da anterior, começando do valor inicial. No final é retornado o valor obtido pela última função da lista.
6. Implemente uma função que recebe uma lista de funções unárias e uma lista de valores e retorna uma lista em que cada elemento é o resultado da aplicação da função da primeira lista com o elemento da segunda que se encontra na mesma posição.
7. Implemente uma função que recebe uma função `Int->Bool`, outra função `Int->Int` e uma lista de inteiros e retorna uma lista cujos elementos são o resultado da segunda função com os elementos da lista que obtém `True` na primeira função.
8. Implemente uma função obtém a cabeça de uma lista usando a composição das funções `last` e `reverse`.