

Primeira lista de exercícios

Prof.: Carlos Camarão

23 de Março de 2017

1. Defina função *numerosEmPalavras*, que converte um número inteiro, que representa uma quantia em dinheiro usado no Brasil, isto é, em reais, em palavras.

Exemplos:

numerosEmPalavras 1000 retorna "mil reais".

numerosEmPalavras 1024 retorna "mil e vinte e quatro reais".

numerosEmPalavras 1324 retorna "mil, cento e vinte e quatro reais"

2. Defina função *tabFib*, que recebe inteiro *n* e retorna tabela (cadeia de caracteres) com *n* linhas, cada linha contendo *i* e (*fib i*), para *i* de 1 a *n*.
A complexidade da função da função *fib* definida deve ser linear.

3. Escreva função *wc* que recebe cadeia de caracteres (valor de tipo *String*) e retorne uma tripla contendo o número de caracteres, o número de palavras e o número de linhas da cadeia.

Use a função *isSpace*, definida em *Data.Char*, para testar se um caractere é um espaço (separador de palavras).

Veja a definição da função *words*, no prelúdio, mas como exercício no uso da linguagem Haskell calcule o número de palavras, caracteres e linhas em um único passo sobre o texto (cadeia de caracteres).

4. Usando a função *reverse*, defina no prelúdio, escreva função *palin*, que testa se uma cadeia de caracteres é ou não um palíndromo, considerando apenas caracteres que são letras ou números, e desconsiderando a caixa de letras. Por exemplo, "A cara rajada da jararaca" e "Madam I'm Adam" devem ser consideradas como palíndromos. Você pode usar *isLetter* e *isDigit*, definidas em *Data.Char*.