

Funções recursivas

- Atividade 1
- Atividade 2
- Atividade 3
- Atividade 4

Atividades

- Nas atividades seguintes, quando for solicitado para definir variáveis e funções, elas devem ser definidas em um arquivo fonte e testadas no GHCi.
- Coloque todas as funções em somente 1 arquivo e chame de Lab06.hs

Atividade 1

- Defina uma função recursiva em Haskell que resolva a seguinte equação recorrente:
- $R(x) = R(x-1) - 2$ para $x > 0$ e $R(x) = 1$ para $x=0$

Atividade 2

- Defina uma função recursiva em Haskell para calcular o máximo divisor comum de dois números inteiros não negativos x e y , usando o algoritmo de Euclides, abaixo. Este algoritmo é expresso da seguinte forma:
- $$\begin{aligned} \text{mdc}(x,y) &= y && \text{se } x \bmod y = 0 \\ &= \text{mdc}(y, x \bmod y) && \text{se } x \bmod y \neq 0 \end{aligned}$$
- Nota: o prelúdio (biblioteca interna) já tem a função `gcd` que calcula o máximo divisor comum de dois números inteiros. Use essa função para testar seu código no GHCi.

Atividade 3

- Defina uma função recursiva em Haskell para calcular a potência de um número, considerando que o expoente é um número natural.
- Utilize o método das multiplicações sucessivas
- $y^n = y \times y \times y \dots \times y$
n vezes

Atividade 4

- Defina uma função recursiva em Haskell para calcular a multiplicação (por meio da adição) de dois números inteiros e positivos. Considere que a multiplicação por meio de adições pode ser expressa como uma função recorrente:
 - $x * y = x + x * (y - 1)$ para $n > 0$
 - $x * 0 = 0$ caso base