

Exercícios de Recursão em Haskell

* Testado em Java

1) Faça uma rotina recursiva para calcular a somatória de todos os número de 1 a N (N será lido do teclado).

Caso base: $n = 1$ soma = 1

Recursão: $n + \text{soma}(n-1)$

2) Crie um programa em Haskell que conte os dígitos de um determinado número usando recursão. O número (n) deve digitado pelo usuário.

Caso base: se $n=0$ numDigito = 1

recursão: $\text{numDigito}(n/10)$

Dica: para saber a quantidade dígitos basta contar quantas vezes a divisão vai ser feita antes do resultado ser 0.

3) Crie um programa que calcula o máximo divisor comum entre dois números usando a recursão. Obs.: Dois números naturais sempre têm divisores comuns. Assim, o máximo divisor comum entre os dois é o maior de seus divisores.

Exemplo: os divisores comuns de 12 e 18 são 1,2,3 e 6. Dentre eles, 6 é o maior. Então, chamamos o 6 de máximo divisor comum de 12 e 18 e indicamos $\text{m.d.c.}(12,18) = 6$.

Caso base: $n1 = 0$ mdc= $n2$ OU $n2 = 0$ mdc = $n1$

Recursão: $\text{mdc}(n2, n1 \% n2)$;

4) Escrever um programa em Haskell que calcule o valor de a elevado a b (potencia), onde a é base e b expoente.

Caso base: $b = 0$ potencia = 1.

Recursão:

$(b > 0) = \text{base} * \text{potencia}(\text{base}, \text{expoente} - 1)$

$(b < 0) = 1/\text{potencia}(\text{base}, -\text{expoente})$

DESAFIO: Implemente uma aplicação recursiva para o problema da Torre de Hanói. O problema da Torre de Hanói consiste de três pinos, A, B e C, denominados : origem, destino e trabalho, respectivamente, e n discos de diâmetros diferentes. Inicialmente, todos os discos se encontram empilhados no pino origem, em ordem decrescente de tamanho, de baixo para cima. O objetivo é empilhar todos os discos no pino destino, atendendo às seguintes restrições:

1. Apenas um disco pode ser removido de cada vez.

2. Qualquer disco não pode ser jamais colocado sobre outro de tamanho menor.

* Já tem várias implementações prontas na Internet, o exercício apenas como desafio para o programador(a).