

IF686 - 2023.2
Lista de Exercícios 1
Programação Funcional (Haskell)

1. (0,2 points) Defina uma função recursiva para calcular o máximo divisor comum de dois números inteiros não negativos `a` e `b`, usando o algoritmo de Euclides

$$mdc(a, b) = \begin{cases} a & \text{se } b = 0 \\ mdc(b, a \bmod b) & \text{se } b > 0 \\ mcd(a, -b) & \text{se } b < 0 \end{cases}$$

2. (0,2 points) Defina função `numDiv`: `Integral a => a -> a -> a` que recebe dois valores inteiros e retorna o número de vezes que uma divisão exata pode ser realizada, neste processo de substituir o dividendo pelo quociente. O processo de divisões sucessivas deve terminar quando uma divisão exata não existe.

Exemplos:

```
numDiv 8 2 ==> 3
numDiv 9 2 ==> 0
```

Defina `numDiv` recursivamente, usando guardas. Dica: use as funções `mod` e `div`

3. (0,2 points) Defina função `unicos` :: `[Integer] -> [Integer]`, que receba uma lista `x` e retorne os elementos de `x` que ocorrem uma única vez nela.

Exemplos

```
unicos [1,1,2,3,4,4,5,5,6] ==> [2,3,6]
unicos [1,1,2,3,4,5,5,6] ==> [2,3,4,6]
unicos [9,9,23,65,87,12,12,15,11,11,13,12] ==> [23,65,87,15,13]
```

Não utilizar a função `filter` da biblioteca.

4. (0,2 points) Defina função `halve` que, ao receber uma lista qualquer como argumento, divide os elementos dessa lista em duas listas `x` e `y`, de modo que os comprimentos de `x` e `y` diferem em no máximo 1. Por exemplo, `halve [2,5,6,1,3]` deve retornar `([2,6,3],[5,1])`. Observar que os elementos da primeira lista são aqueles dos índices ímpares da lista dada como argumento, enquanto a segunda lista são os dos índices pares.

Exemplos:

```
halve [1..5] ==> ([1,3,5],[2,4])
halve [1..6] ==> ([1,3,5],[2,4,6])
halve [9,9,23,65,87,12,12,15,11,11,13,12] ==>
  ([9,23,87,12,11,13],[9,65,12,15,11,12])
```

5. (0,2 points) Defina função `remDiv:: Int -> [a] -> ([a],[a])` tal que `remDiv n x` retorna o par `(x1, x2)` em que `x1` tem os `n-1` primeiros valores de `x` e `x2` tem os `n+1` valores de `x`.

Exemplos

```
remDiv 3 [1..6] ==> ([1,2],[4,5,6])
remDiv 5 [1..6] ==> ([1,2,3,4],[6])
remDiv 7 [1..6] ==> ([1,2,3,4,5,6],[])
```