Ficha 6

Programação Funcional

LEI 1º ano

1. Defina a função `ePrimo :: Integer -> Bool` que testa se um número é primo.

2. Defina a função `primos :: Integer -> [Integer]` que calcula a lista de números primos até um dado valor limite.

3. O *Crivo de Eratóstenes* é um método simples e prático para encontrar números primos até um certo valor limite. Pedemos descreve-lo assim:

   - começamos com a lista de inteiros entre 2 (o primeiro primo) e o valor limite;
   - destacamos o primeiro elemento da lista (que irá ficar na lista de saida), retiramos da cauda da lista todos os multiplos dele, e continuamos a processar a caula da lista (já filtrada) pelo mesmo método.

   Defina uma função que implemente este algoritmo

4. O *Teorema Fundamental da Aritmética* (enunciado pela primeira vez por Euclides) diz que qualquer número inteiro (maior do que 1) pode ser decomposto num produto de números primos. Esta decomposição é além disso única a menos de uma permutação. Por exemplo,

   $$212121 = 3 \times 3 \times 7 \times 7 \times 13 \times 37$$

   $$222222 = 2 \times 3 \times 7 \times 11 \times 13 \times 37$$

   (a) Defina uma função `factoriza :: Integer -> [Integer]` que, dado um número (maior do que 1) calcula a lista dos seus factores primos (por exemplo, `factoriza 212121` deve calcular a lista `[3,3,7,7,13,37]`).

   (b) Dadas as factorizações de dois números é fácil calcular (as factorizações de) o máximo divisor comum (`mdc`) e o mínimo múltiplo comum (`mmc`).

   - o máximo divisor comum obtém-se com *os factores comuns elevados à menor potência*. Assim,

     $$\begin{aligned} \texttt{mdc}\ 212121\ 222222 &= \texttt{mdc}\ (3^2 \times 7^2 \times 13^1 \times 37^1)(2^1 \times 3^1 \times 7^1 \times 11^1 \times 13^1 \times 37^1) \\ &= 3^1 \times 7^1 \times 13^1 \times 37^1 \\ &= 10101 \end{aligned}$$

   - o mínimo múltiplo comum obtém-se com *os factores comuns e não comuns elevados à maior potência*. Assim,

     $$\begin{aligned} \texttt{mmc}\ 212121\ 222222 &= \texttt{mmc}\ (3^2 \times 7^2 \times 13^1 \times 37^1)(2^1 \times 3^1 \times 7^1 \times 11^1 \times 13^1 \times 37^1) \\ &= 2^1 \times 3^2 \times 7^2 \times 11^1 \times 13^1 \times 37^1 \\ &= 4666662 \end{aligned}$$

   Defina as funções `mdcF, mmcF :: Integer -> Integer -> Integer` que calculam o máximo divisor comum e mínimo múltiplo comum usando as factorizações dos números em causa.

(c) Uma outra forma (muito mais eficaz) de calcular o máximo divisor comum entre dois números baseia-se na seguinte propriedade (também atribuída a Euclides):

$$\texttt{mdc}\ x\ y = \texttt{mdc}\ (x+y)\ y = \texttt{mdc}\ x\ (y+x)$$

Apresente uma definição da função `mdc` usando esta propriedade. Note ainda que a função `mmc` pode ser definida usando `mdc`:

```
mmc :: Integer -> Integer -> Integer
mmc x y = (x * y) `div` (mdc x y)
```

5. Uma representação possível de polimómios (alternativa à que vimos na aula teórica) é pela sequência dos coeficientes - vamos ter de armazenar também os coeficientes nulos pois será a posição do coeficiente na lista que dará o grau do monómio. Teremos então

```
type Polinomio = [Coeficiente]
type Coeficiente = Float
```

A representação do polinómio $2\,x^5 - 5\,x^3$ referido acima será então

```
[0,0,0,-5,0,2]
```

que corresponde ao polinómio $0\,x^0 + 0\,x^1 + 0\,x^2 - 5\,x^3 + 0\,x^4 + 2\,x^5$.

(a) Defina a operação que calcula o valor do polonómio para um dado x.

(b) Defina a operação que calcula a derivada de um polinómio.

(c) Defina a operação de adição de polinómios.

(d) Defina a operação de multiplicação de polinómios.

6. Defina a função `subLists :: [a] -> [[a]]` que calcula todas as sublistas de uma lista; por exemplo, `subLists [1,2,3] = [[1,2,3],[1,2],[1,3],[1],[2,3],[2],[3],[]]`.