PROGRAMAÇÃO FUNCIONAL
EXERCÍCIOS – GENERALIZAÇÃO

1. Implemente três funções que recebam uma lista e retorne outra lista contendo o dobro de cada elemento da lista original, usando três definições diferentes: compreensão de lista, recursão e map.

2. Defina uma função que calcule o tamanho de uma lista usando as funções map e sum.

3. Dada a função:
   ```
   incrementaLista :: [Int] -> [Int]
   incrementaLista lista = filter maiorQueUm (map incrementa lista)
      where
            maiorQueUm n = n > 1
            incrementa n = n + 1
   ```

   como você a redefiniria usando filter antes de map, como em:

   ```
   incrementaLista lista = map f1 (filter f2 lista)
   ```

4. Forneça o tipo e defina a função iter de forma que:
   Iter n f x = f (f (f ... (f x)))
   Onde a função f ocorre n vezes no lado direito da equação.
   Exemplo:
   ```
      iter 3 f x = f (f (f x))
      iter 0 f x = x
   ```

5. Usando a função iter, defina uma função que receba n como entrada e retorne $2^n$.

6. Defina uma função que receba n como entrada e retorne a soma dos quadrados dos números naturais de 1 até n, usando map e foldr.

7. Defina uma função que retorne a soma dos quadrados dos números inteiros positivos em uma lista de inteiros. Por exemplo, a lista [-3, 2, -9, 4, 8, -6] terá como resultado 84, pois $2^2 + 4^2 + 8^2 = 4 + 16 + 64 = 84$.

8. Defina a função da questão anterior usando composição de funções.

9. Defina a função alternaMap que mapeia duas funções ao longo de uma lista, alternando a aplicação.
   Exemplo:
   ```
   > alternaMap (+1) (+10) [1, 2, 3, 4]
   > [2, 12, 4, 14]
   ```

10. Defina as seguintes funções:

```
split :: [a] -> ([a], [a])
merge :: ([a], [a]) -> [a]
```

A função split recebe uma lista e retorna uma tupla com duas lista, distribuindo de forma alternada os elementos da lista original. A função merge recebe uma tupla com duas lista e as intercala para formar um única lista.

Exemplo:

```
> split [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
([1,3,5,7,9],[2,4,6,8,10])
> merge ([1,3,5,7,9,11],[2,4,6,8,10])
[1,2,3,4,5,6,7,8,9,10,11]
```