1. Utilizando uma lista por compreensão:

   1.1. Dado uma lista e um valor retorna todos os elementos da lista que seja superior ou igual ao valor.

   1.2. Dado uma lista retorna a quantidade de números pares existente na lista

   1.3. Dado uma lista retorna a quantidade de letras existentes na lista.

2. Utilizando recursão

   2.1. Escreva uma que calcule a soma de todos os números de 1 a n, onde n é dado como parâmetro.

   2.2. Escreva uma função que encontre e retorne o elemento mínimo em uma lista, onde a lista é dada como parâmetro.

   2.3. Escreva uma função que dado uma lista de números inteiros retorna uma sequência de números de 3 a 9 com limites exclusivos.

3. Utilizando Funções de Ordem superior

   3.1. Defina uma função altMap :: (a -> b) -> (a -> b) -> [a] -> [b] que alternadamente aplique suas duas funções de argumento a elementos sucessivos em uma lista, por sua vez sobre a ordem. Por exemplo:

   ```
   > altMap (+10) (+100) [0,1,2,3,4]

   [10,101,12,103,14]
   ```

   3.2. Crie uma função que recebe como parâmetro uma função isApproved e uma lista contendo o nome e a nota do aluno e deverá retornar uma lista só com alunos aprovados.

4. Utilizando declaração de tipos e classes:

   4.1. Crie um tipo que representa o nome de uma pessoa e outro que represente o número de telefone.

   4.2. Crie um tipo phonebook que será um lista contendo o nome de uma pessoa e o seu número de telefone, utilizando os tipo declarados anteriormente.

4.3. Crie uma função *inPhoneBook* que recebe o tipo número de telefone e retorna o nome da pessoa a qual pertence o número caso não existir retorna uma mensagem.

Bom trabalho.
Frederico Soares