

**Atenção:**

1. A FUNÇÃO\_DE\_GENERALIZAÇÃO deverá ser substituída por map, filter ou foldr1.
2. Usem somente as estruturas praticadas em sala de aula.

**Questão 1**

Faça uma função que, dado um inteiro, retorne um booleano indicando se ele é perfeito (True) ou não (False). Um número perfeito é aquele em que a soma de seus divisores (exceto ele mesmo) é igual ao próprio número.

Hugs > ehPerfeito 6                    -- Também funciona com 28, 496 e 8128

True

Hugs > ehPerfeito 8

False

**Questão 2**

Faça uma função que receba como parâmetro uma String e retorne uma tupla com dois elementos. O primeiro é um caractere qualquer da String e o outro é o número de vezes que ele apareceu a partir do momento que ele foi descoberto.

Hugs > contaString “felliipe rey”

[('f', 1), ('e', 3), ('l', 2), ('i', 1), ('p', 1), ('e', 2), (' ', 1), ('r', 1), ('y', 1)]

**Questão 3**

Faça uma função que inverte uma String.

Hugs > inverte “Fellipe Rey”

“yeR epilleF”

**Questão 4**

Usando generalização (map, filter ou foldr1), faça uma função que calcule o quadrado de cada elemento uma lista de entradas

Hugs > FUNÇÃO\_DE\_GENERALIZAÇÃO squares [1, -2, 3]

[1, 4, 9]

### Questão 5

Faça uma função que simule um produto cartesiano entre duas listas de Int A e B e retorne uma lista de tuplas Int, cada tupla com dois parâmetros.

```
Hugs > [1,2] [7,8]
```

```
[(1,7), (1,8), (2,7), (2,8), (7,1), (7,2), (8,1), (8,2)]
```

### Questão 6

Usando generalização (map, filter ou foldr1), faça uma função que pegue os números positivos de uma lista

```
Hugs > FUNÇÃO_DE_GENERALIZAÇÃO positives [1, -2, 3]
```

```
[1, 3]
```

### Questão 7

Usando generalização (map, filter ou foldr1), faça uma função que some o dobro de todos os elementos de uma lista

```
Hugs > FUNÇÃO_DE_GENERALIZAÇÃO sumDouble [1, -2, 3]
```

```
4
```

```
-- Exemplo do cálculo:  $(1*2) + (-2 * 2) + (3 * 2) = 4$ 
```

### Questão 8

Usando generalização (map, filter ou foldr1), faça uma função que concatene todas as Strings de uma lista.

```
Hugs > FUNÇÃO_DE_GENERALIZAÇÃO concatena [“con”, “ca”, “te”, ‘na’]
```

```
concatena
```