

## Expressões condicionais, guardas e padrões

**2.1** Escreva duas definições, respectivamente usando expressões condicionais e guardas, da função `classifica :: Int -> String` que faz corresponder uma classificação qualitativa a uma nota de 0 a 20:

$\leq 9$	“reprovado”
10–12	“suficiente”
13–15	“bom”
16–18	“muito bom”
19–20	“muito bom com distinção”

**2.2** O *índice de massa corporal* (IMC) é uma medida simples para classificar o peso de adultos.<sup>1</sup> O IMC de um indivíduo é calculado como o valor do peso (em quilogramas) a dividir pelo quadrado da altura (em metros):

$$\text{IMC} = \text{peso} / \text{altura}^2$$

Por exemplo: um indivíduo com 70Kg e 1.70m de altura tem IMC igual a  $70/1.70^2 \approx 24.22$ . Classificamos o resultado nos seguinte intervalos:

	IMC	< 18.5	“baixo peso”
$18.5 \leq$	IMC	< 25	“peso normal”
$25 \leq$	IMC	< 30	“excesso de peso”
$30 \leq$	IMC		“obesidade”

Escreva uma definição da função `classifica :: Float -> Float -> String` que determina a classificação acima; os dois argumentos da função são, respectivamente, o peso em quilogramas e a altura em metros.

**2.3** Considere duas possíveis definições das funções `max` e `min` do prelúdio-padrão que calculam, respectivamente, o máximo e o mínimo de dois valores:

```
max, min :: Ord a => a -> a -> a
max x y = if x>y then x else y
min x y = if x<y then x else y
```

- (a) Escreva definições deste género para duas funções `max3` e `min3` para calcular, respectivamente, o máximo e o mínimo de três números.
- (b) Observe que as operação de máximo e mínimo são *associativas*. Por exemplo, para calcular o máximo de três valores podemos determinar o máximo entre dois deles e depois o máximo do resultado com o terceiro. Re-escreva as funções `max3` `min3` usando esta ideia e as funções de `max` e `min` do prelúdio-padrão.

---

<sup>1</sup><https://www.euro.who.int/en/health-topics/disease-prevention/nutrition/a-healthy-lifestyle/body-mass-index-bmi>

## 2.4 Escreva uma definição da função lógica ou-exclusivo

```
xor :: Bool -> Bool -> Bool
```

usando múltiplas equações com padrões.

**2.5** Pretende-se implementar uma função `safetail :: [a] -> [a]` que estende a função `tail` do prelúdio de forma a dar a lista vazia quando o argumento é a lista vazia (em vez de um erro). Escreva três definições diferentes usando condicionais, equações com guardas e padrões.

**2.6** Escreva duas definições da função `curta :: [a] -> Bool` para testar se uma lista tem zero, um ou dois elementos, usando:

- (a) a função `length` do prelúdio-padrão;
- (b) múltiplas equações e padrões.

**2.7** A mediana de três valores é o valor “no meio” quando os colocamos por ordem crescente. Por exemplo: `mediana 2 3 (-1) == 2`.

- (a) Escreva uma definição da função `mediana` para determinar a mediana de 3 valores quaisquer. Qual será o seu tipo mais geral? Note que podemos determinar a mediana usando apenas comparações de ordem.
- (b) Em vez de definir a mediana diretamente usando comparações (que provavelmente terá sido a sua primeira ideia), pode usar o seguinte método: somamos os 3 valores e subtraímos o maior e o menor. Re-defina a função `mediana` desta forma. Qual será agora o tipo mais geral?

## 2.8 (T) Defina uma função

```
converte :: Int -> String
```

para converter um inteiro positivo inferior a 1 milhão para texto em português. Alguns exemplos:

```
converte 21 = "vinte e um"
converte 1234 = "mil duzentos e trinta e quatro"
converte 123456
  = "cento e vinte e três mil quatrocentos e cinquenta e seis"
```

*Ideia:* Vamos começar por definir funções auxiliares para converter para texto os números inferiores a 100 e 1000. *Este exercício será resolvido numa aula teórica.*