**5.1** Escreva uma função `media_arit(xs)` que, dada uma lista de $n$ valores numéricos `xs`, retorna a média aritmética dos seus valores, isto é,

$$\frac{1}{n}\sum_{i=1}^{n} x_i = \frac{x_1 + x_2 + \cdots + x_n}{n}$$

**5.2** Escreva uma função `media_geom(xs)` que, dada uma lista de $n$ valores numéricos `xs`, retorna a média geométrica dos seus valores, isto é,

$$\left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{n}} = \sqrt[n]{x_1 x_2 \cdots x_n}$$

**5.3** Escreva uma função `desvio_padrao(xs)` cujo resultado é o desvio padrão amostral de uma lista de $n$ valores, isto é,

$$\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2}$$

onde $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ é a média aritmética dos valores. Pode assumir que $n > 1$.

**5.4** Escreva uma função `intervalo(xs,a,b)` cujo resultado é a contagem dos valores da lista $xs$ que estão entre $a$ e $b$ inclusivé; pode assumir que $a \leq b$.

**5.5** Escreva uma função `segundoMaior(l)` que, dada uma lista de inteiros `l`, retorna o segundo maior elemento da lista. Exemplos:

```
>>> segundoMaior([1,10])
1
>>> segundoMaior([-1,20,12,-10])
12
```

Nota: pode assumir que a lista `l` tem, pelo menos, dois elementos.

**5.6** Escreva uma função `ocorrencias(txt,c)` que retorna uma lista com os índices das ocorrências dum caracter `c` na cadeia `txt`. Por exemplo:

```
>>> ocorrencias('banana', 'a')
[1, 3, 5]
```

**5.7** Recorde que um número inteiro $d$ é *divisor próprio* de $n$ se e só se o resto da divisão de $n$ por $d$ for zero e $d$ for inferior a $n$.

(a) Escreva uma função `divisores(n)` que calcula a lista dos divisores de próprios de $n$, por ordem crescente.
Exemplo: `divisores(12)` dá `[1, 2, 3, 4, 6]`

(b) Um número inteiro é *perfeito* se for igual à soma dos seus divisores próprios. Exemplo: 6 é perfeito porque $6 = 1 + 2 + 3$ mas 10 não é porque $10 \neq 1+2+5$. Escreva uma função `perfeito(n)` que testa se $n$ é perfeito ou não; o resultado deve ser um valor lógico.

**5.8** Escreva uma função `repetidos(lista)` que testa se há elementos repetidos numa lista; o resultado deve ser um valor lógico. A sua função deve funcionar com listas de vários tipos (e.g. de números ou de cadeias de carateres). Exemplos:

```
>>> repetidos(['ola', 'ole', 'abba', 'ole'])
True
>>> repetidos([3, 2, -5, 0, 1])
False
```

**5.9** Escreva uma função `palavras(txt)` que retorna a lista das palavras na cadeia de caracteres `txt`. As palavras devem incluir apenas letras maiúsculas ou minúsculas; assuma ainda que a cadeia não tem letras acentuadas. Exemplo:

```
>>> palavras("---A Maria tinha um cordeirinho?")
['A', 'Maria', 'tinha', 'um', 'cordeirinho']
```

**5.10** O *triângulo de Pascal* é constituido pelos valores $\binom{n}{k}$ das combinações de $n$ em $k$ em que $n$ é a linha e $k$ é a coluna. As primeiras cinco linhas do triângulo são:

$$\begin{array}{ccccccc} & & & 1 & & & (n=0) \\ & & 1 & & 1 & & (n=1) \\ & 1 & & 2 & & 1 & (n=2) \\ 1 & & 3 & & 3 & & 1 \quad (n=3) \\ 1 & 4 & & 6 & & 4 & 1 \quad (n=4) \end{array}$$

Para obter os valores que compõem a $n$-ésima linha podemos usar as seguintes igualdades:

$$\binom{n}{0} = \binom{n}{n} = 1$$

$$\binom{n}{k} = \binom{n}{k-1} \times \frac{n+1-k}{k}, \quad \text{se } 0 < k \leq n$$

Escreva uma função `pascal(n)` cujo resultado é uma lista com os coeficientes da $n$-ésima linha do triângulo de Pascal.

Por exemplo:

```
>>> pascaln(4)
[1, 4, 6, 4, 1]
```

Nota: Os valores devem estar no formato inteiro.

**5.11** Duas palavras ou frases são *anagramas* se se escrevem com as mesmas letras usadas o mesmo número de vezes mas eventualmente em posições diferentes. Por exemplo: "Marcelo Sousa" e "salmão escuro" são anagramas bem

como “Aníbal Cavaco” e “cabala nociva” (ignorando a distinção entre maísculas e minúsculas e os acentos).

Escreva uma função `anagramas(txt1,txt2)` que verifique se duas cadeias são anagramas; o resultado deve ser `True` ou `False`. Deve considerar equivalentes as letras maiúsculas e minúsculas e ignorar todos os caracteres não-letras (espaços, sinais de pontuação, etc.); pode assumir que as letras não têm acentos.