

4.1 Escreva uma definição da função `conta_letras(txt)` que retorna o número de letras (maiúsculas ou minúsculas) sem acentos da cadeia de caracteres `txt`.

Exemplo:

```
>>> conta_letras('Ola, mundo!')
8
```

4.2 Escreva uma definição da função `apenas_letras(txt)` que testa se uma cadeia de caracteres contém apenas letras maiúsculas ou minúsculas (sem acentos). O resultado deve ser `True` ou `False`.

Exemplos:

```
>>> apenas_letras("Abracadabra")
True
>>> apenas_letras("Ola, mundo!")
False
```

4.3 Escreva uma definição da função `filtra_letras(txt)` que, dada uma cadeia de caracteres `txt`, retorna uma cadeia de caracteres com apenas as letras maiúsculas ou minúsculas da cadeia `txt`.

Exemplo:

```
>>> filtra_letras('Ola!, -- disse ele...')
'Oladisseele'
```

4.4 Escreva uma função `inversa(txt)` que retorne a cadeia de caracteres dada por ordem inversa.

Exemplo:

```
>>> inversa('Ola Mundo!')
'!odnuM alO'
```

4.5 Uma cadeia de caracteres é um *palíndromo* se as sequências obtidas lida da esquerda para a direita e vice-versa são iguais, independentemente das letras serem maiúsculas ou minúsculas. Exemplo: "reviveR" é um palíndromo.

Escreva uma definição da função `palindromo(txt)` que verifica se uma cadeia de caracteres é um palíndromo; o resultado deve ser `True` ou `False`.

4.6 Mais geralmente, uma cadeia é um palíndromo se se lê da mesma forma nos dois sentidos, ignorando os espaços entre letras, sinais de pontuação e/ou a troca de maiúsculas e minúsculas. Assim, os exemplos seguintes são palíndromos:

```
"Amora me tem aroma."
"Madam, I'm Adam."
"A man, a plan, a canal: Panama"
```

Escreva uma função `palindromo_geral(txt)` para testar se uma cadeia de caracteres `txt` é um palíndromo neste sentido mais geral.

Sugestão: pode resolver este problema combinando o método `lower()` de cadeias de caracteres e as soluções dos exercícios 4.3 e 4.5.

4.7 Escreva uma função `rem_espacos(txt)` que remova dois ou mais espaços seguidos numa cadeia de caracteres `txt`, substituindo-os por um único espaço; outros caracteres devem ficar inalterados. Exemplo:

```
>>> rem_espacos(' Ola,      Mundo  !')
' Ola, Mundo !'
```

4.8 Defina uma função `forte(passwd)` que verifica se uma palavra-passe, dada pela cadeia de caracteres `passwd`, é forte. Isto é, tem 8 caracteres ou mais e tem, pelo menos, uma letra maiúscula, uma letra minúscula e um algarismo. O resultado deve ser um valor lógico (`True` ou `False`).

```
>>> forte('9EwL56')
False
>>> forte('HXKW1393')
False
>>> forte('ffu4G7Fghjk')
True
```

4.9 Defina uma função `rem_vogais(txt)` que, dada uma cadeia de caracteres `txt`, devolve uma nova cadeia de caracteres removendo as vogais (minúsculas ou maiúsculas e sem acentos) da cadeia original. Exemplo:

```
>>> rem_vogais('Abracadabra!')
'brcdbr!'
```

4.10 A cifra de *Vigenère* é uma variação um pouco mais segura da cifra de César que usa uma palavra-chave em vez de um deslocamento único ¹. Começamos por repetir a palavra-chave (e.g., “LUAR”) ao longo do texto da mensagem; cada letra da chave de ‘A’ a ‘Z’ fazemos corresponder um índice de deslocamento de 0 a 25 (e.g., “LUAR” corresponde aos deslocamentos 11, 20, 0 e 17). Assim, o texto “ATAQUEDEMADRUGADA” seria cifrado em “LNAHFYDVXUDIFAAUL”:

A	T	A	Q	U	E	D	E	M	A	D	R	U	G	A	D	A
L	U	A	R	L	U	A	R	L	U	A	R	L	U	A	R	L
L	N	A	H	F	Y	D	V	X	U	D	I	F	A	A	U	L

Escreva uma função `vigenere(chave,txt)` que implemente esta cifra.

¹Ver https://pt.wikipedia.org/wiki/Cifra_de_Vigenère