

1. Escreva uma função

```
conta_c(s,v)
```

que retorna o número de caracteres da string `s` (com possível repetição) que estão na string `v`. Por exemplo,

```
conta_c("verdes anos","aes") -> 5 (caracteres indicados)
      ^  ^  ^  ^  ^
```

Nota. Pode usar o predicado `in` que permite testar se um caracter existe numa string. Por exemplo, o predicado “`c` é vogal?” pode ser implementado com

```
c in "aeiou"
```

2. Escreva uma função

```
sub_x(s)
```

que retorna a lista que resulta de `s`, substituindo todas as consoantes por “`p`”. Por exemplo,

```
sub_x("Pera Banana") -> "pepa papapa"
```

Admite-se que `s` só contém letras (maiúsculas ou minúsculas) e espaços.

3. Escreva uma "função"

```
pares(n)
```

que calcula a lista de todos os tuplos (a,b) com $1 \leq a < b \leq n$. Exemplo:

```
>>> pares(4)
[(1,2), (1,3), (1,4), (2,3), (2,4), (3,4)]
```

4. Escreva uma função

```
unica(li)
```

que calcula a lista ordenada de todos os elementos de `li`, de modo que cada elemento de `li` ocorra uma e uma só vez no resultado. Por exemplo,

```
unica([2,3,5,2,5,1,2,2]) -> [1,2,3,5]
```

Para além das funções usuais (como `len`, `range`,...) é permitida a utilização da função `sorted` (de ordenação) exemplificada em seguida.

```

>>> a=[2,3,2,3,1]
>>> sorted(a)
[1,2,2,3,3]
>>> a
[2,3,2,3,1]

```

5. A seguinte sequência de comandos faz com que **a** e **b** referenciem o mesmo objecto

```

>>> a=[4,1,2]
>>> b=a
>>> a[1]="112910"
>>> a
[4, 112910, 2]
>>> b
[4, 112910, 2]

```

Suponha que o parâmetro **a** da função **cp(a)** que vai implementar é um inteiro ou uma lista de 0 ou mais elementos que podem ser inteiros ou listas do mesmo tipo (definição indutiva); por exemplo **[2, [3,4], [[5]]]**. A função deve construir e retornar uma cópia da lista **a**; assim, deve verificar-se por exemplo o seguinte

```

>>> a=[2, [3, 4], [[5]]]
>>> a
[2, [3, 4], [[5]]]
>>> b=cp(a)
>>> b
[2, [3, 4], [[5]]]
>>> a[2][0]=1000
>>> a
[2, [3, 4], [1000]]
>>> b
[2, [3, 4], [[5]]]

```

Notas. Pode averiguar se **x** é uma lista com a condição **type(x)==list**; analogamente, **x** é um inteiro sse **type(x)==int**.