

---

# Desenvolvimento para a Internet e Aplicações Móveis

2022/23

---

**Python**

**Exercícios**



ISCTE  Instituto Universitário de Lisboa  
Gabinete de Comunicação e Imagem

Gabinete de Comunicação e Imagem  
ISCTE  Instituto Universitário de Lisboa

---

## 3.1 Strings

Considere o poema “Corrente”, de Chico Buarque:

```
poema = 'Eu hoje fiz um samba bem pra frente / Dizendo realmente o que é que eu  
acho / Eu acho que o meu samba é uma corrente / E coerentemente assino embaixo  
/ Hoje é preciso refletir um pouco / E ver que o samba está tomando jeito / Só  
mesmo embriagado ou muito louco / Pra contestar e pra botar defeito / Precisa  
ser muito sincero e claro / Pra confessar que andei sambando errado / Talvez  
precise até tomar na cara / Pra ver que o samba está bem melhorado / Tem mais é  
que ser bem cara de tacho / Não ver a multidão sambar contente / Isso me deixa  
triste e cabisbaixo / Por isso eu fiz um samba bem pra frente / Dizendo realmente  
o que é que eu acho / Eu acho que o meu samba é uma corrente / E coerentemente  
assino embaixo / Hoje é preciso refletir um pouco / E ver que o samba está  
tomando jeito / Só mesmo embriagado ou muito louco / Pra contestar e pra botar  
defeito / Precisa ser muito sincero e claro / Pra confessar que andei sambando  
errado / Talvez precise até tomar na cara / Pra ver que o samba está bem melhorado  
/ Tem mais é que ser bem cara de tacho / Não ver a multidão sambar contente /  
Isso me deixa triste e cabisbaixo'
```

- a) Imprima na consola os 5º e 6º versos do poema.
- b) Imprima na consola o poema formatado da seguinte forma:

```
Eu hoje fiz um samba bem pra frente  
Dizendo realmente o que é que eu acho  
Eu acho que o meu samba é uma corrente  
  
...
```

Sugestão: procure um método na biblioteca Python3 para substituir um caractere numa string.

- c) Junte a estrofe seguinte ao final do poema:

```
Por isso eu fiz um samba bem pra frente  
Dizendo realmente o que é que eu acho  
Isso me deixa triste e cabisbaixo
```

- d) Imprima na consola os dois últimos versos do poema na sua forma depois das alíneas 1, 2 e 3.

---

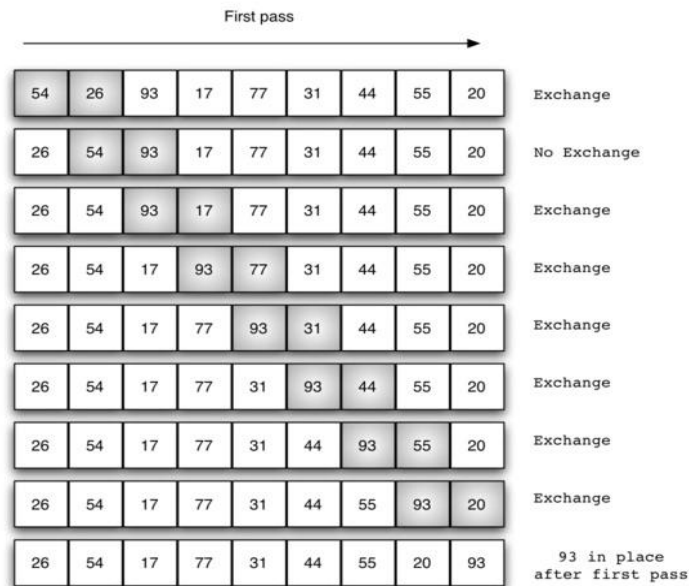
## 3.2 Strings e estruturas de controlo

Considere novamente o início do poema “Corrente” de Chico Buarque.

- a) Escreva e execute um programa que:
  - conta o número de vogais existentes no texto;
  - imprime na consola as ocorrências da cada vogal;
  - identifica a vogal mais utilizada;
  - imprime na consola a vogal mais utilizada;
  - se existirem várias vogais empatadas com o maior número de ocorrências, deve imprimir “Há vários vencedores.”.
- b) Melhore o seu programa de forma identificar quem são os vários vencedores em caso de empate.

### 3.3 Listas e estruturas de controlo

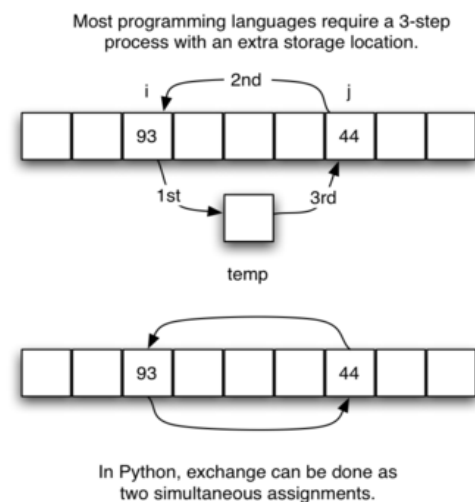
Considere o algoritmo de ordenação de listas “Bubble Sort”. Este algoritmo percorre cada elemento de uma lista, compara-o com o elemento adjacente seguinte e troca os dois elementos se o primeiro for maior que o segundo. O procedimento é repetido até ordenar a lista.



- a) Programe o algoritmo “Bubble Sort”. Teste o seu programa.
- b) O seu programa pode ser otimizado com recurso às propriedades do Python. De facto, a maior parte das linguagens de programação exigem o processamento de três passos para a troca de valores entre dois elementos da lista (ver figura adjacente).

Em Python essa troca pode ser feita numa única instrução. Por exemplo, a troca entre **a** e **b** pode ser feita com a instrução **a, b = b, a**.

Otimize o seu programa para ordenar a lista com o mínimo possível de passos. Conte o tempo gasto para ordenar uma lista através de cada uma das suas versões do “Bubble Sort”. Qual é a mais rápida?

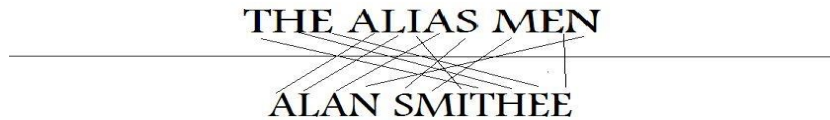


---

## 3.4 Estruturas de controlo

Considere a operação de transposição de letras através da qual uma palavra ou uma frase pode ser composta com as letras de outra palavra ou de outra frase.

Por exemplo, a palavra “amor” resulta da transposição de letras da palavra “roma”. Ou um outro exemplo, com mais de uma palavra:



Conceba várias variantes de um programa que recebe duas strings e devolve True se as strings são transponíveis ou devolve False se as strings não são transponíveis. Considere que as duas strings têm igual número de caracteres alfabéticos. Todos os caracteres são minúsculos.

As variantes para a transposição de letras são as seguintes:

- Verifique se cada caracter da primeira string existe também na segunda string. Se existir, retire o caracter na segunda string e substitua-o por “None”. Se no final do processo a segunda string for composta apenas por “None”, então as strings podem ser transpostas. Conte o número de passos necessários até obter a solução. Considere que um passo é uma iteração numa cadeia de caracteres. Conte também o tempo necessário até obter a solução.
- Ordene os caracteres das duas strings. Se o resultado da ordenação forem duas strings iguais, então as strings são transponíveis. Conte o número de passos necessários e o tempo até obter a solução.
- Programe a variante “força bruta”, isto é, que testa todas as possibilidades. Para isso considere as letras da primeira string e faça uma lista com strings correspondentes a todas as combinações possíveis com essas letras. Finalmente, verifique se a segunda string se encontra na lista. Conte o número de passos necessários e o tempo até obter a solução.
- Considere as duas strings. Conte o número de cada caractere que existe em cada string. Por exemplo, o número de “a” em cada string, o número de “b” em cada string, etc. Se o número de cada caractere em cada string for igual, então as duas strings são transponíveis. Conte o número de passos necessários e o tempo até obter a solução.
- Qual destas variantes executa menos passos até à solução? E qual é a mais rápida?