Desenvolvimento para a Internet e Aplicações Móveis

2023/24

Python

Exercícios

ISCTE Instituto Universitário de Lisboa
Gabinete de Comunicação e Imagem

3.1 Strings e listas

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) Transforme a *string* "poema" numa lista de versos e 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
...
```

Dica: repare que os vários versos do poema estão separados sempre da mesma forma...

b) Complete o poema juntando-lhe os seguintes versos ao final:

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

c) Imprima na consola apenas os dois últimos versos do poema na sua forma final, depois das alíneas a) e b).

d) Desenvolva código para calcular e mostrar na consola o número de versos em que está presente a palavra "samba".

e) Desenvolva código para calcular e mostrar o número de vezes em que aparece cada uma das vogais ao longo do poema completo.

3.2 Dicionários

Programe um script em Python que:

a) Cria dois dicionários: o primeiro deve conter as disciplinas do semestre passado e as notas que obteve; o segundo deve conter as disciplinas deste semestre e as notas que prevê obter, por exemplo, `'DIAM': 20`. Imprima na consola os dois dicionários.

b) Obtenha um terceiro dicionário que resulta da concatenação de ambos os dicionários da alínea anterior. A partir deste ponto trabalhe apenas com este terceiro dicionário que terá todas as disciplinas do ano letivo. Imprima na consola este dicionário.

c) Mostre na consola uma lista que contém apenas as chaves (disciplinas) do dicionário.

d) Mostre na consola uma lista que contém apenas os valores (notas) do dicionário.

e) Ordene o dicionário por ordem alfabética das disciplinas e mostre o resultado na consola.

Para cada uma das seguintes alíneas pretende-se que desenvolva uma função (ou procedimento) e que teste o código desenvolvido mostrando os resultados na consola.

f) Verificar se uma determinada disciplina exista ou não no dicionário. Por exemplo, se `'DIAM'` existir, seria mostrada a mensagem `'DIAM existe no dicionario'`.

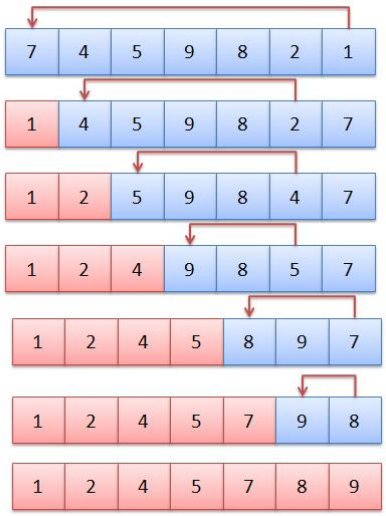
g) Obter a lista de disciplinas cuja nota é superior a um determinado valor.

h) Calcular a média das notas das disciplinas presentes no dicionário

i) Obter a lista das três disciplinas com as melhores notas.

3.3 Listas e estruturas de controlo

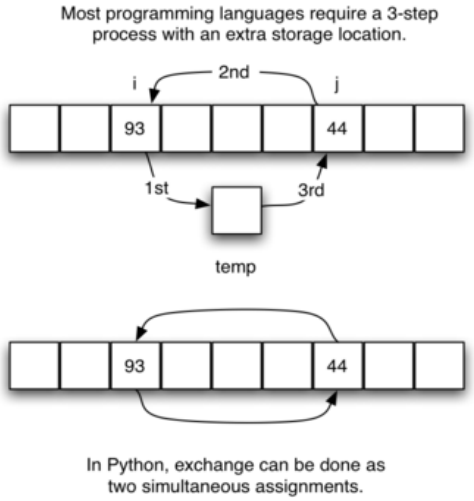
Considere o algoritmo de ordenação "Selection Sort". A ideia por detrás do funcionamento deste algoritmo é a seguinte: em cada passo, o algoritmo procura o valor mínimo numa parte não ordenada da lista (ou array) e coloca esse valor no final da parte da lista que já estiver ordenada. Inicialmente assume-se que todo a lista está desordenada e os passos são repetidos sucessivamente (tantas vezes quanto o número de elementos da lista menos 1). Após cada passo, a parte ordenada da lista cresce e a parte desordenada decresce como ilustrado na seguinte figura:



a) Programe o algoritmo "Selection Sort" e teste-o utilizando uma lista com pelo menos 10 valores diferentes.

b) O número de instruções no seu programa pode ser minimizado recorrendo às propriedades do Python. De facto, a maior parte das linguagens de programação exigem o processamento de três passos para uma troca de valores entre dois elementos da lista (ver figura adjacente).

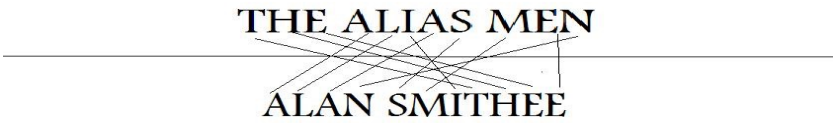
   Em Python essa troca pode ser feita com uma única instrução. Por exemplo, a troca entre **`a`** e **`b`** pode ser feita com uma única instrução **`a,b = b,a`**. Aplique esta particularidade do Python à sua implementação do algoritmo "Selection Sort".



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:



Existem várias maneiras de verificar se duas strings são transponíveis, por exemplo as que são descritas nas seguintes variantes:

i) Verificar se cada letra da primeira string existe também na segunda string. Se existir, retira-se essa letra da segunda string. Se no final do processo a segunda string ficar sem nenhuma letra, então as strings são transponíveis.

ii) Ordenar as letras das duas strings por ordem alfabética. Se o resultado da ordenação das letras for igual em ambas as strings, então as strings são transponíveis.

iii) Testar todas as possibilidades – variante “força bruta”. Para isso constrói-se uma lista com todas as combinações possíveis que se podem fazer com as letras da primeira string. Se alguma dessas combinações for igual à segunda string (descartando os espaços em branco), então as strings são transponíveis.

iv) Contar o número de ocorrências de cada letra em cada uma das strings (o número de “a”s em cada string, o número de “b”s em cada string, etc.). Se o número de ocorrências de cada letra em ambas as strings forem todas iguais, então as duas strings são transponíveis.

a) Conceba pelo menos duas variantes de uma função que verifica se duas strings são transponíveis, devolvendo True ou False consoante o caso. Considere que as ambas strings têm um número igual de letras e que as letras são sempre minúsculas.

b) Compare as variantes implementadas quanto ao número de passos necessários e quanto ao tempo de execução. Considere que cada passo corresponde a percorrer os caracteres numa das strings. Para medir tempos de execução pode utilizar, por exemplo, uma das abordagens descritas em: https://www.geeksforgeeks.org/how-to-check-the-execution-time-of-python-script/.