

# Universidade da Beira Interior

## Departamento de Informática

### Inteligência Artificial

#### Ficha prática 1

Ano letivo 2023-24

### Exercícios

Nesta ficha vamos fazer um breve apanhado sobre a linguagem Python e olhar para a implementação de um grafo, que nos será útil mais à frente.

Sugerimos a consulta dos seguintes links: [LearnPython.org](https://www.learnpython.org/) tutorial, [Official Python Tutorial](https://www.python.org/doc/official-python-tutorial/) e [tutorialspoint.com/python](https://www.tutorialspoint.com/python/).

1. Escreva um programa que mostre no ecrã os números ímpares entre 10 e 34 (tire partido da função `range()` e verá que só usa 2 linhas de código). Modifique o programa para que os números ímpares apareçam todos na mesma linha.
2. Escreva um programa que peça ao utilizador um número inteiro positivo menor que 100 e mostre no ecrã quantas dezenas e quantas unidades tem o número lido.
3. Escreva um programa que contenha as seguintes funções:
  - `ler`: lê uma frase lida do teclado e devolve-a;
  - `gravar`: grava para um ficheiro, cujo nome é passado como parâmetro, uma frase também passada como parâmetro à função;
  - `contaVogais`: recebe uma frase e devolve o número de vogais.

Com as funções anteriores construa um programa que lê duas frases do teclado e grava para um ficheiro de texto a que tiver mais vogais. Confirme abrindo o ficheiro criado num editor de texto.

4. Veja como lidar com listas aqui.  
Escreva um programa que tenha uma função que leia uma lista de inteiros positivos do teclado (termine a leitura com a introdução de um número negativo); outra função que receba 2 listas e mostre no ecrã os números que

aparecem em ambas as listas. Agora escreva o programa para ler duas listas do teclado e mostrar os elementos que são membros de ambas as listas (a interseção das listas).

5. A biblioteca `numpy` é muito útil para processamento matemático de dados (tipo `matlab`). Para a podermos usar devemos fazer o seguinte `import`:

```
import numpy as np
```

Agora podemos criar, por exemplo, um array 7x3 (7 linhas e 3 colunas) inicializado a zero com:

```
a = np.zeros([7,3])
```

Escreva um programa um programa que peça ao utilizador duas matrizes quadradas 2x2,  $A$  e  $B$ , e mostre no ecrã:

- (a) o produto elemento a elemento  $A.B$
  - (b) o produto matricial  $A * B$
  - (c) a diferença entre matrizes  $A - B$
  - (d) o logaritmo dos elementos de  $A$  (se existirem elementos negativos, use o seu valor absoluto)
  - (e) o maior valor da segunda linha de  $A$  vezes o menor valor da primeira coluna de  $B$ .
6. O Python tem vários tipos de dados estruturados standard, entre os quais, as listas, os dicionários e os conjuntos. Vejamos um exercício com dicionários: escreva um programa que permita guardar a nota que obteve em várias UCs, usando um dicionário. Deve ter um menu que permita: inserir um novo par UC + nota, alterar o valor da nota duma UC, mostrar todos os pares UC + nota e mostrar a nota média.
7. Os conjuntos são coleções de elementos, sem repetição (não há elementos duplicados). Use dois conjuntos para guardar duas frases lidas do teclado. Depois mostre o seguinte (tire partidos das operações sobre conjuntos):
- (a) Todas as letras que aparecem em ambas as frases (se aparecer a mesma letra mais que uma vez, só deve ser mostrada uma vez).
  - (b) As letras que aparecem na primeira frase mas não na segunda.
  - (c) As letras que aparecem simultaneamente em ambas as frases.
  - (d) As letras que só aparecem na primeira frase ou só aparecem na segunda frase.
8. Para trabalharmos com grafos vamos importar a seguinte biblioteca:
- ```
import networkx as nx
```
- Agora escreva um programa que use a referida biblioteca para representar as principais ligações rodoviárias entre as 18 capitais de distrito de Portugal continental tal como apresentadas na Figura 1.

O seu programa deverá listar para cada cidade do mapa o conjunto das suas cidades vizinhas (aquelas que se podem alcançar sem passar por nenhuma outra cidade).

Sugestões:

- Comece por ver o simples tutorial que aparece no site da `networkx` em <https://networkx.github.io/documentation/stable>.

- Represente num ficheiro de texto cada ligação (aresta do grafo) numa linha com o seguinte formato: ORIGEM,DESTINO,DISTÂNCIA. Não coloque espaços entre as vírgulas. Exemplo: AV,VI,100.

- Use a biblioteca CSV para ler do ficheiro: <https://docs.python.org/3/library/csv.html>

- Carregue, a partir do ficheiro de texto, cada aresta e adicione a um grafo.

