

**Aula prática nº 9 – Manipulação, leitura e escrita de ficheiros de texto****Exercícios**

- 1) Crie um programa que calcule a soma de uma lista de valores gravados num ficheiro. Considere que o ficheiro contém apenas um valor por linha, como no exemplo abaixo. O nome do ficheiro deve ser pedido ao utilizador.

| nums.txt |
|----------|
| 12.39    |
| 1.93     |
| 7.85     |
| ...      |

- 2) Escreva um programa que determine a frequência de ocorrência de todas as letras que ocorrem num ficheiro de texto. O nome do ficheiro deve ser passado como argumento na linha de comando (use `sys.argv`). Descarregue "Os Lusíadas" ([documento 3333 do Projeto Gutenberg](#)) e faça a contagem. Ajuste o programa para não distinguir maiúsculas de minúsculas. (Sugestão: use `str.isalpha()` para detetar letras e `str.lower()` para converter para minúsculas.)

| \$ python3 countLetters.py pg3333.txt |
|---------------------------------------|
| n 14470                               |
| a 32088                               |
| v 4371                                |
| w 268                                 |
| j 1111                                |
| ...                                   |

- 3) Retome o programa 4 da aula prática 08 e acrescente duas opções, uma para ler a lista de contactos de um ficheiro e outra para gravar num ficheiro. O nome do ficheiro a ler ou a gravar deve ser pedido ao utilizador. Em ambos os casos, use `try...except` para detetar erros e permitir que o programa regresse ao menu sem terminar.
- 4) Crie uma função que devolva numa lista os nomes dos ficheiros de um certo diretório que têm uma certa extensão. A procura deve percorrer todos os diretórios recursivamente. Use a função `os.walk`, como no exemplo descrito nos slides da aula. O nome do diretório e a extensão a procurar devem ser argumentos passados na linha de comandos.

- 5) O ficheiro `/etc/dictionaries-common/words` contém uma lista de palavras (da língua inglesa), uma por linha. Algumas dessas palavras têm letras consecutivas iguais, tais como "accent", "access", "pool", etc.
- a) Faça um programa para listar e contar todas essas palavras. Sugestão: comece por criar uma função `repeats(w)`, que teste se a string `w` tem letras consecutivas iguais.
  - b) Conte também todas as ocorrências (de pares de letras consecutivas iguais) por palavra, e o total no ficheiro. Por exemplo, "accent" tem 1 par, "access" tem 2 pares de letras consecutivas iguais. Sugestão: pode alterar a função `repeats` para devolver uma contagem em vez de `True/False`.
- 6) Crie um programa que leia o ficheiro `stocks.csv` e que determine:
- a) Empresa mais transacionada (com maior volume total).
  - b) Dia e valor em que cada ação atingiu o valor mais elevado.
  - c) Empresa com maior valorização diária.
  - d) Empresa com maior valorização durante o período a que se refere o ficheiro.
  - e) Crie uma função que calcule a valorização de um dado portefólio entre duas datas dadas. O portefólio deve ser um dicionário com o número de ações de cada título, e.g.: `{'NFLX': 100, 'CSCO': 80}`.
- Cada linha do ficheiro tem o formato seguinte:
- ```
Nome, Data, PreçoAbertura, PreçoMaximo, PreçoMinimo, PreçoFecho, Volume
```
- 7) Escreva uma função `compareFiles` que verifica se dois ficheiros são iguais. Para poupar tempo e memória, leia e compare blocos de 1 KiB de cada vez, e termine logo que descubra diferenças. Use a função `read`. (1 KiB lê-se "1 kibibyte" e corresponde a 1024 bytes.)
- 8) Para saber o tamanho em bytes de um ficheiro, pode usar a função `os.stat("ficheiro").st_size`. Crie uma função que percorra um diretório e todos os seus subdiretórios recursivamente, e mostre o tamanho de cada ficheiro.
- 9) Adapte a função anterior para devolver um dicionário que a cada tamanho (usado como chave) associe uma lista de ficheiros que têm esse tamanho.
- 10) Por vezes temos várias cópias de um mesmo ficheiro espalhados no disco com diferentes nomes. Faça um programa para descobrir esses ficheiros duplicados. Sugestão: use as funções que desenvolveu atrás para detetar ficheiros com tamanho igual e compare esses dois a dois.