

Programação II

Exercícios 6

Ficheiros de Valores Separados por Vírgulas (CSV)

Universidade de Lisboa
Faculdade de Ciências
Departamento de Informática
Licenciatura em Tecnologias da Informação

2020/2021

Nota: Para estes exercícios pode utilizar as funções de leitura e de escrita em ficheiros CSV desenvolvidas nas aulas teóricas: `ler_csv`, `escrever_csv`, `ler_csv_dicionario`, e `escrever_csv_dicionario`.

1. Escreva uma função `ler_primeiras_csv(nome_de_ficheiro, n)` que lê até n linhas de um ficheiro CSV, linha a linha e para uma lista de listas de *strings*. Caso o ficheiro tenha menos de n linhas, a função deverá ler o ficheiro todo.
2. A função `ler_csv(nome_ficheiro)` discutida nas aulas teóricas devolve uma lista de listas de strings. Muitas vezes precisamos de converter estas strings para valores de outros tipos, tais como inteiros ou números em vírgula flutuante. Escreva uma função `converter_todos` que, dado um iterável de iteráveis de valores e uma função de conversão, devolve um novo iterável de iteráveis onde todos os elementos do iterável original foram convertidos pela função parâmetro. Por exemplo:

```
>>> [list(it) for it in converter_todos([['0', '1', '2', '3', '4', '5'], ['0', '2', '4', '6', '8', '10']], int)]
```

`[[0, 1, 2, 3, 4, 5], [0, 2, 4, 6, 8, 10]]`
3. Muitas vezes, precisamos de aplicar uma função de conversão aos valores em cada coluna, que difere de coluna para coluna.
 - (a) Escreva uma função `converter`.

```
def converter(iteravel, iteravel_de_funcoes):
    """
    iterável cujos elementos são obtidos por
    aplicação de cada função a cada elemento

    Args:
        iteravel (iter): iterável de valores
        iteravel_de_funcoes (iter): iterável de fun
        ções

    Returns:
        (iter) novo iterável de valores
    """
```

Por exemplo:

```
>>> list(converter(['0', '3', '4', '5', '-1', '1'],
    [float] * 2 + [lambda x: x] * 2 + [lambda x:
        -int(x) if int(x)<0 else int(x)] * 2))
[0.0, 3.0, '4', '5', 1, 1]
```

- (b) Utilizando a função da alínea anterior, escreva agora a função converter_iteraveis.

```
def converter_iteraveis(iter_de_iteraveis,
    iteravel_de_funcoes):
    """
    converte cada 'coluna' do iter_de_iteraveis
    pela função respetiva no
    iteravel_de_funcoes

    Args:
        iter_de_iteraveis (iter): iterável de iterá
        veis (de valores)
        iteravel_de_funcoes (iter): iterável de fun
        ções

    Returns:
        iter : iterável de iteráveis de valores
    """
```

Por exemplo,

```
>>> [list(it) for it in converter_listas([
    ['Joao', '2001', '17.35'], ['Ana', '1999', '14.50'],
    ['Adao', '2000', '18.71']],
    [lambda x: x, int, float])]
```

```
[['Joao', 2001, 17.35], ['Ana', 1999, 14.5],  
 ['Adao', 2000, 18.71]]
```

- (c) Finalmente, utilizando a função `ler_csv(nome_ficheiro)` e a função da alínea anterior, escreva uma função `ler_csv_com_conversao(nome_ficheiro, iteravel_de_funcoes)` que lê um ficheiro e devolve um iterável de iteráveis onde as 'colunas' se apresentam convertidas de acordo com as funções no `iteravel_de_funcoes`.
4. Escreva uma função `escrever_primeiros_csv(nome_ficheiro, iter_de_iteraveis, n)` que escreva num ficheiro em formato CSV os primeiros `n` elementos do `iter_de_iteraveis`. Se o iterável tiver menos de `n` elementos a função deverá escrever o iterável completo.
5. Escreva uma função que calcule a média dos vários alunos constantes numa pauta. A pauta é dada por um ficheiro CSV com um número variável de colunas. Nas duas primeiras aparece o nome do aluno e o número de estudante. Nas restantes colunas aparecem um número variável de notas (números inteiros). Assuma que todas as linhas têm o mesmo comprimento.

O resultado deve ser também escrito num ficheiro CSV onde as primeiras colunas são as do ficheiro de entrada e a última contém a média. A média de cada aluno deve ser um número com duas casas decimais.

Sugestão. Organize o seu código deste modo: ler, converter, escrever. Para ler utilize a função `ler_csv` das aulas teóricas, para converter escreva uma função para o efeito, e para escrever utilize a função `escrever_csv` também das aulas teóricas. Finalmente a função principal não tem mais do que chamar as outras três.

Para verificar o seu código, pode aceder ao ficheiro `pauta.csv` na pasta Documentos do Moodle.

6. Escreva uma função que calcule a média anual das temperaturas, bem como a precipitação total anual, de cada estação meteorológica, a partir de um registo de temperaturas. Este registo é dado por um ficheiro CSV com cinco colunas, que indicam, por ordem: nome da estação meteorológica, latitude, longitude, data, precipitação, e temperatura média diária.

O resultado deve ser também escrito num ficheiro CSV onde a primeira coluna é o nome da estação, e as duas colunas seguintes são a média das temperaturas e a precipitação total. Estes valores deverão ser números com duas casas decimais.

Sugestão. Organize o seu código deste modo: ler, converter, escrever. Para ler utilize a função `ler_csv` das aulas teóricas, para converter

escreva uma função para o efeito, e para escrever utilize a função `escrever_csv` também das aulas teóricas. Finalmente a função principal não tem mais do que chamar as outras três.

Para verificar o seu código, pode aceder ao ficheiro `temperaturas.csv` na pasta Documentos do Moodle. Nota: Este ficheiro tem cabeçalho, que deverá ser ignorado.

7. Escreva uma função `grafico_para_csv` que receba o nome de um ficheiro e um gráfico (um par de listas), e escreva o gráfico no ficheiro organizado por colunas: abcissas na primeira coluna, ordenadas na segunda. Tente duas soluções.

- (a) Uma solução “direta” em que faz a transposta à medida que vai escrevendo no ficheiro, e
- (b) Uma solução em que transpõe primeiro o par de listas e depois utiliza a função `escrever_csv` (aulas teóricas) para escrever o resultado. Neste caso comece por escrever uma função `transposta` que transpõe uma lista de listas, isto é, que troca as linhas pelas colunas. Assuma que a dimensão das linhas e das colunas é qualquer. Por exemplo:

```
>>> transposta([[1, 2, 3], [4, 5, 6], [7, 8, 9],
                [10, 11, 12], [13, 14, 15]])
[[1, 4, 7, 10, 13], [2, 5, 8, 11, 14], [3, 6, 9,
                12, 15]]
```

8. Pretende-se neste exercício construir uma função `csv_para_grafico` que devolva o gráfico de uma função, isto é um par de listas (abcissas, ordenadas). Assuma que o ficheiro CSV contém as abcissas na primeira coluna e as ordenadas na segunda. Sugestão: utilize as funções `ler_csv` das aulas teóricas, `transposta` (exercício 7), `converter_iteraveis` (exercício 3), e **`tuple`**, por esta ordem. Poderia alterar a ordem das funções `transposta` e `converter`? E das funções `converter_iteraveis` e **`tuple`**?

Para verificar o seu código, pode aceder ao ficheiro `grafico2.csv` na pasta Documentos do Moodle.

9. Escreva uma função `graficos_para_csv` que, dado o nome de um ficheiro e uma lista de gráficos (uma lista de listas de números), escreva os gráficos num ficheiro CSV. Organize a informação no ficheiro do seguinte modo: a primeira coluna contém as abcissas, a segunda coluna contém as ordenadas da primeira função, a terceira coluna contém as ordenadas da segunda função, e por aí em diante. Assuma que todas as funções têm listas de abcissas iguais.
10. Suponha que precisamos de processar ficheiros contendo a seguinte informação sobre jogadores de voleibol, referentes a uma dada época

desportiva: na primeira coluna o nome do jogador, na segunda o clube a que pertence, na terceira o número de pontos que marcou, na quarta o número de ases (número de serviços que resultaram diretamente em ponto), na quinta a eficiência de ataque (rácio entre o número de passes que recebe e o número de pontos que marca), na sexta o número de blocos efetuados, e na sétima o número de cartões amarelos.

Vamos usar a função `ler_csv_dicionario` (aulas teóricas) para ler o ficheiro CSV. A informação aparecerá como uma lista de dicionários. Eis uma possível entrada na lista resultante:

```
{ 'clube': 'SC_Espinho', 'ataque': '0.64', 'nome': 'Luis', 'ases': '0', 'pontos': '47', 'amarelos': '2', 'blocos': '86' }
```

Para podermos processar esta informação temos de fazer algumas conversões nos dados. Por exemplo o ataque deverá ser um **float** e os ases um número inteiro. Escreva uma função `ler_volei` que leia um ficheiro CSV e faça as conversões que achar necessárias. Assuma que o ficheiro não tem cabeçalho.

Para verificar o seu código, pode aceder ao ficheiro `volei.csv` na pasta Documentos do Moodle.

11. Utilizando a função `escrever_csv_dicionario` (aulas teóricas) desenvolva uma função que, dadas uma lista de dicionários com informação volei e o nome de um ficheiro, escreva os dados num ficheiro CSV. O ficheiro deverá ter um cabeçalho e as colunas devem apresentar-se pela ordem descrita no exercício anterior.