

Teste 7 Arquivos

De 20 a 27 de Junho de 2022

Instruções: Este teste possui 3 partes.

- A primeira, composta pelas questões de 1 a 3, é para ser entregue ao final da aula.
- A segunda, composta pelas questões de 4 a 7, deve ser entregue até dia 20 de Junho.
- A terceira, o desafio, é opcional, e sua entrega vai também até 20 de Junho.

Parte 1 – Em sala

Questão 1. Escreva uma função `linhas_grandes(arquivo)` que conta o número de linhas com menos de 10 caracteres em um arquivo. Você pode assumir que este arquivo pode ser corretamente acessado apenas abrindo-o como um arquivo de texto para leitura.

Questão 2. Escreva uma função `gerar_estatísticas(arquivo)` que lê um arquivo de texto em `ascii`, e cujas linhas contém um único número, escrito em base 10. Esta função deve *retornar* uma tupla contendo, nesta ordem:

- a quantidade de números lidos;
- a média dos números;
- o maior dos números;
- o menor dos números.

Dê uma documentação para sua função.

Questão 3. Testes. Crie (no próprio Spyder) 3 arquivos de teste para cada uma das funções acima, com os nomes `T7_qN_arqK.txt`, substituindo N e K apropriadamente.

Parte 2 – Obrigatória

Questão 4. Lendo sem ler tudo de uma única vez. Modifique, se necessário, a função da questão 2 para não armazenar todos os números lidos na memória (numa lista, tupla, ...).

Coletando estatísticas

Um arquivo de texto em `utf-8` tem a seguinte estrutura: cada linha tem um nome de uma pessoa, seguido de um número inteiro, separado por um espaço. Por exemplo, uma linha poderia ser

Bernardo Freitas Paulo da Costa 2222

Questão 5. Coletando dados. Escreva e documente a função `ler_nomes_números(arquivo)`, que retorna uma lista de tuplas, onde cada tupla contém uma `string` com o nome da pessoa, e em seguida um `int` com o número correspondente.

Questão 6. Trabalhando os dados - © uma cientista de dados. Suponha que os números sejam os anos de nascimento das pessoas correspondentes. Escreva a função `maiores_de_18(arquivo)` que retorna a lista dos nomes das pessoas que têm mais de 18 anos. (Para evitar questões de dia e mês, vamos considerar neste teste que estas são as que nasceram em 2003 ou antes)

Generalize esta função para receber a idade mínima como argumento, e documente.

Questão 7. “Tabela”. Escreva uma função `alfabetizar(arquivo, novo_arquivo)` que cria um arquivo de nome dado pela `string` `novo_arquivo`, que contém as mesmas informações do arquivo original, mas com os nomes em ordem alfabética.

Dica: use a função de ler da questão 5, e dê uma olhada na ajuda para a *keyword* `key` da função `sort()`.

Parte 3 — Desafio opcional

Questão 8. Detectando erros de digitação. Um arquivo contém muitos nomes de pessoas, com seus DREs, num formato como acima. Só que, ao contrário do arquivo acima, as linhas podem ter repetição, e, até, erros de digitação. Vamos aproveitar a repetição dos dados para tentar achar o nome correto de uma pessoa.

Assim, escreva a função `identificar(nome, dre, arquivo)` que retorna o nome e o DRE mais provável de uma pessoa com o `nome` e `dre` dados, consultando todas as entradas do `arquivo`, e testando a proximidade dos dados com a função `difflib.get_close_matches()`, com índice de proximidade (argumento `cutoff`) 0.9.

Todos os argumentos são `strings`.