

Trabalho 02 - Classificação do Brasileirão

No Campeonato Brasileiro de Futebol, também chamado de Brasileirão, diversos times disputam o título de melhor time do Brasil.

Os times jogam todos contra todos em dois turnos, ida e volta, e vão acumulando pontos. Cada vitória gera três pontos, cada empate um ponto e a derrota não gera pontos. Os times são classificados de acordo com os pontos, quem tem mais pontos fica na frente de quem tem menos pontos. Se dois ou mais times têm o mesmo número de pontos, então, os seguintes critérios são utilizados para o desempate: número de vitórias, saldo de gols (número de gols feitos menos o número de gols sofridos) e ordem alfabética (Sim, é injusto! Mas essa é uma simplificação...).

O programa deve funcionar de forma genérica, para qualquer quantidade de resultados e para quaisquer times que apareçam nos resultados.

Projete um programa que leia os resultados dos jogos de um arquivo texto, usando o código em anexo nesta especificação. Cada linha do arquivo texto descreve o resultado de um jogo, escrito da forma “Anfitrião Gols Visitante Gols” (os nomes dos times anfitrião e visitante não têm espaço). Por exemplo, a linha “Sao-Paulo 3 Flamengo 1”, descreve o jogo em que Sao-Paulo era anfitrião e marcou 3 gols e o Flamengo era visitante e marcou 1 gol.

Supondo que um arquivo `resultados.txt` tenha o seguinte conteúdo

```
Sao-Paulo 1 Atletico-MG 2
Flamengo 2 Palmeiras 1
Palmeiras 0 Sao-Paulo 0
Atletico-MG 1 Flamengo 2
```

O seu programa deve produzir respostas para as seguintes perguntas:

- 1) Qual a classificação dos times no campeonato?
- 2) Qual o time com melhor aproveitamento jogando como anfitrião?
- 3) Qual o time com a defesa menos vazada?

Para a pergunta 1, o seu programa deve produzir a seguinte saída, considerando que a largura das colunas é determinada pelo conteúdo (por exemplo, a largura da coluna dos nomes dos times deve levar em conta a quantidade de caracteres do time com nome mais extenso).

```
Flamengo      6 2  2
Atletico-MG  3 1  0
Palmeiras    1 0 -1
Sao-Paulo    1 0 -1
```

Para a pergunta 2, você deve identificar o(s) time(s) que obtiveram o maior número de pontos em relação à quantidade de pontos possíveis jogando como anfitrião. Por exemplo, um time que realizou 5 jogos como anfitrião (15 pontos possíveis) e conquistou três vitórias e um empate, obteve 10 pontos, portanto, 66% de aproveitamento.

Para a pergunta 3, você deve identificar o time que sofreu o menor número de gols, considerando todos os jogos daquele time.

Resolver este problema “de uma” vez é complicado, então você deve decompô-lo em subproblemas menores. Para isso você pode planejar o uso de funções auxiliares que resolvam esses subproblemas e implementar essas funções depois.

Por exemplo, para a pergunta 1, você pode decompor o problema da seguinte maneira:

- Descobrir os nomes dos times
- Calcular os pontos, número de vitórias e saldo de gols de um time por vez
- Classificar os times

Requisitos para implementação:

- Usar tipos adequados para representar as informações
- Usar recursividade adequadamente na solução de pelo menos duas perguntas
- Usar funções que modificam seus argumentos quando adequado
- Não usar tuplas, dicionários ou outras funções não discutidas em sala de aula.

Anexo

```
import sys

def main():
    if len(sys.argv) < 2:
        print('Nenhum nome de arquivo informado.')
        sys.exit(1)

    if len(sys.argv) > 2:
        print('Muitos parâmetro. Informe apenas um nome de arquivo.')
        sys.exit(1)

    jogos = le_arquivo(sys.argv[1])

    # TODO: solução da pergunta 1
    # TODO: solução da pergunta 2
    # TODO: solução da pergunta 3

def le_arquivo(nome: str) -> list[str]:
    """
    Lê o conteúdo do arquivo *nome* e devolve uma lista onde cada elemento
    representa uma linha.

    Por exemplo, se o conteúdo do arquivo for
    Sao-Paulo 1 Atletico-MG 2
    Flamengo 2 Palmeiras 1

    a resposta produzida é
    ['Sao-Paulo 1 Atletico-MG 2', 'Flamengo 2 Palmeiras 1']
    """
    try:
        with open(nome) as f:
            return f.readlines()
    except IOError as e:
        print(f'Erro na leitura do arquivo "{nome}": {e.errno} - {e.strerror}.');
        sys.exit(1)

if __name__ == '__main__':
    main()
```