

Tarefa de laboratório 06



Em um campeonato de futebol, todos os times inscritos se enfrentam duas vezes: uma em seu estádio e outra no estádio adversário. O principal elemento para classificação é a **pontuação** acumulada: cada time recebe três pontos por vitória, um ponto por empate e zero ponto por derrota.

Em caso de empate, o desempate é feito por **número de vitórias**, **saldo de gols** (diferença entre o número de gols pró marcados e os gols sofridos por cada time) e **número de gols pró**. Caso os times permaneçam empatados em todos estes itens, ocupariam a mesma posição no campeonato. Nesta

tarefa, no entanto, todos os testes levarão a um **único vencedor**.

Descrição da entrada

A primeira linha da entrada indicará o número de times do campeonato e as linhas seguintes irão descrever os resultados das partidas. Uma partida será descrita em uma linha pelo nome e o número de gols pró do primeiro time, pelo caractere "x" e pelo nome e o número de gols pró do segundo time.

```
<numero_de_times>
<time_A> <gols_A> x <time_B> <gols_B>
<time_B> <gols_B> x <time_A> <gols_A>
<time_A> <gols_A> x <time_C> <gols_C>
<time_C> <gols_C> x <time_A> <gols_A>
...
<time_B> <gols_B> x <time_C> <gols_C>
<time_C> <gols_C> x <time_B> <gols_B>
...
```

Exemplo:

```
3
Ibis 0 x Cascavel 3
Cascavel 2 x Ibis 0
Ibis 0 x Dourados 1
Dourados 2 x Ibis 0
Cascavel 3 x Dourados 2
Dourados 3 x Cascavel 1
```

Descrição da saída

A saída conterá os resultados obtidos pelos times, apresentados em ordem alfabética de acordo com o padrão abaixo:

```
<nome_do_time> <pontuacao> <num_vitorias> <saldo_gols> <num_gols_pro>
```

A última linha conterá a string **Vencedor:** seguida do nome do vencedor.

Vencedor: <vencedor>

Para o exemplo da seção anterior, a saída será:

```
Cascavel 9 3 4 9
Dourados 9 3 4 8
Ibis 0 0 -8 0
Vencedor: Cascavel
```

Testes para o SuSy

Esta tarefa contém 7 testes abertos e 3 testes fechados. Como mencionado acima, todos os testes levam a um único vencedor.

Dicas de Python 3 para esta tarefa:

- Para separar os dados da partida, você pode utilizar a função `split`. Faça, por exemplo, o seguinte teste:

```
linha_entrada = input()
dados_partida = linha_entrada.split()
print("0 time da casa", dados_partida[0], "marcou", dados_partida[1], "gols.")
print("0 time visitante", dados_partida[3], "marcou", dados_partida[4], "gols.")
```

- Para armazenar os dados dos times, você pode usar dicionários. Veja exemplos de algumas ações:

- Inicia dicionário vazio

```
dic_times = {}
```

- Verifica se um time está em um dicionário:

```
if "Ibis" in dic_times:
    print("Ibis faz parte dos times inscritos.")
```

- Atribui dados a um elemento do dicionário. Suponha que a variável `dados_time` contém os dados que queremos armazenar:

```
dic_times["Araguaia"] = dados_time
```

- Percorre os elementos de um dicionário em ordem alfabética:

```
for t in sorted(dic_times):
    print(t, ":", dic_times[t])
```

Orientações para submissão

Veja [aqui](#) a página de submissão da tarefa. Lembre-se que o arquivo a ser submetido deve se chamar `lab06.py`. No link [Arquivos auxiliares](#) há um arquivo `aux-06.zip` que contém todos os arquivos de testes abertos e seus respectivos resultados compactados. Os arquivos `executa-testes.py` e `executa-testes-windows.py` também estão neste pacote.

Observe o limite máximo de 20 submissões.

A nota final é proporcional ao número de testes que executaram corretamente, desde que o código esteja coerente com o enunciado. **A submissão de um código que não implementa o algoritmo requisitado, mas que exibe as saídas esperadas dos testes abertos a partir da comparação de trechos da entrada será considerada fraude e acarretará a atribuição de nota zero à média final da disciplina.**

O peso desta tarefa é 3.

O prazo final para submissão é 14/10/2018.