

MC102 – Algoritmos e Programação de Computadores

Detetives e Assassinos

Nesta tarefa, vamos ler uma sequência de crimes no seguinte formato:

<assassino> <vítima> <detetive>

Estas informações não estão ordenadas nem por tempo nem por ordem alfabética. Sua tarefa será ao final da leitura gerar um relatório ordenado informando para cada pessoa:

- Se ela ainda está viva ou se foi assassinada. Caso tenha sido assassinada o comentário (*in memoriam*) deve ser colocado após o nome.
- Se a pessoa não atuou como detetive e não matou ninguém uma indicação de que foi uma **vítima inocente**.
- Se a pessoa matou uma ou mais pessoas e não atuou como detetive, uma indicação de **assassino(a)**. Logo após uma lista de quantos detetives, assassinos(as) ou vítimas inocentes ele matou.
- Se a pessoa atuou como detetive, uma indicação de **detetive** e o número de casos que ela resolveu. Um detetive também pode ter matado pessoas. Os crimes podem ter ocorrido em serviço ou o detetive pode ter matado de maneira premeditada. Não é necessário entrar neste mérito, mas apenas apresentar uma lista quantos detetives, assassinos ou vítimas inocentes foram mortos por este detetive.

Entrada

A entrada consiste de um inteiro N ($1 \leq N \leq 100$) seguido de N linhas no formato <assassino> <vítima> <detetive> .

Caso seja fornecido um número menor do que 1 ou maior do que 100, a mensagem de erro "Valor inválido na entrada." deve ser emitida.

Saída

A saída é um relatório por pessoa, em ordem alfabética, separado por linhas contendo 60 hífens.

- Caso a pessoa seja uma vítima, reportar:

```
-----
nome_vitima (in memoriam): vítima inocente.
-----
```

- Caso a pessoa seja um assassino, reportar:

```
-----
nome_assassino: assassino(a).
Matou D detetive(s).
Matou A assassino(s).
Matou I inocente(s).
-----
```

Se o assassino também tiver sido morto, colocar a observação (*in memoriam*) como no item anterior. Escrever uma linha relatando o número de mortes apenas se o número de mortes na categoria for maior do que 0.

- Caso a pessoa seja um detetive, reportar:

```
-----
nome_detetive: detetive.
Resolveu X caso(s).
Matou D detetive(s).
Matou A assassino(s).
Matou I inocente(s).
-----
```

Se o detetive também tiver sido morto, colocar a observação (*in memoriam*) como nos itens anteriores. Escrever uma linha relatando o número de mortes apenas se o número de mortes na categoria for maior do que 0.

Testes para o SuSy

Veja, como exemplo, os quatro primeiros testes. Para verificar o conteúdo dos outros testes, favor consultar diretamente os arquivos.

arq1.in	1 Francisco Gilberto Karen	----- Francisco: assassino(a). Matou 1 inocente(s). ----- Gilberto (in memoriam): vítima inocente. ----- Karen: detetive. Resolveu 1 caso(s). -----	arq1.res
---------	-------------------------------	---	----------

arq2.in	2 Caio Estela Marcos Caio Mateus Marcos	----- Caio: assassino(a). Matou 2 inocente(s). ----- Estela (in memoriam): vítima inocente. ----- Marcos: detetive. Resolveu 2 caso(s). ----- Mateus (in memoriam): vítima inocente. -----	arq2.res
arq3.in	2 Maria Cícero Bianca Maria Alan Cícero	----- Alan (in memoriam): vítima inocente. ----- Bianca: detetive. Resolveu 1 caso(s). ----- Cícero (in memoriam): detetive. Resolveu 1 caso(s). ----- Maria: assassino(a). Matou 1 detetive(s). Matou 1 inocente(s). -----	arq3.res
arq4.in	4 Otávio Alexandre Nicole Fernanda Gustavo André Otávio Fernanda André André Otávio Nicole	----- Alexandre (in memoriam): vítima inocente. ----- André: detetive. Resolveu 2 caso(s). Matou 1 assassinos(s). ----- Fernanda (in memoriam): assassino(a). Matou 1 inocente(s). ----- Gustavo (in memoriam): vítima inocente. ----- Nicole: detetive. Resolveu 2 caso(s). ----- Otávio (in memoriam): assassino(a). Matou 1 assassinos(s). Matou 1 inocente(s). -----	arq4.res

Esta tarefa inclui mais três testes abertos e outros três testes fechados.

Dicas de Python 3 para esta tarefa:

- Para ler os elementos de uma linha e armazená-los nas variáveis assassino, vitima e detetive utilize:

```
assassino, vitima, detetive = input().split()
```

- Para escrever o separador do relatório utilize:

```
print(60 * "-")
```

- Dicionários permitem incluir e recuperar facilmente as informações associadas a uma chave. Como exemplo, observe um dicionário que organiza uma lista de elementos por cor:

```
>>> objetos_por_cor = {} # cria dicionário vazio
>>> objetos_por_cor["azul"] = [] # associa uma lista vazia à cor azul
>>> objetos_por_cor["branca"] = ["bola"] # associa uma lista contendo um elemento à cor branca
>>> objetos_por_cor["branca"].append("urso") # adiciona um elemento à lista da cor branca
>>> print(objetos_por_cor["branca"]) # imprime lista associada à cor branca
['bola', 'urso']
>>> print("preta" in objetos_por_cor) # imprime valor da expressão booleana que indica presença ou ausência da cor preta na lista
False
```

Lembre-se que:

- Para cada pessoa, é necessário saber se está viva ou morta.
- Para cada detetive, é necessário saber o número de casos que ele resolveu.
- Para cada assassino ou detetive que tenha matado pessoas pode-se utilizar a lista de vítimas para a classificação final.

- Para percorrer todos os elementos de um dicionário utilize:

```
for i in nome_dicionario :
    # código por elemento i
```

- Para percorrer todos os elementos de um dicionário de maneira ordenada por chave utilize:

```
for i in sorted(nome_dicionario) :
    # código por elemento i
```

- Se você ainda não usou dicionários, faça testes simples antes de tentar resolver esta tarefa. Por exemplo, leia a entrada da tarefa, crie um dicionário de pessoas e

imprima os elementos em ordem alfabética. Em uma segunda etapa, associe um valor a cada pessoa indicando se está viva ou morta.

Orientações para submissão

Veja [aqui](#) a página de submissão da tarefa. Lembre-se que o arquivo a ser submetido deve se chamar `main.py`. No link [Arquivos auxiliares](#) há um arquivo [args-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 e que a nota final é proporcional ao número de testes que executaram corretamente.

O peso desta tarefa é 3.

O prazo final para submissão é 06/05/2018.
