

# MC102 - Algoritmos e Programação de Computadores

MC102 Oferecimento anterior

## Tarefa de laboratório 05

---

### #ilovepython ;-)

Em textos disponíveis na Internet é muito comum o uso de [hashtags](#) e [emoticons](#). Um pesquisador da área de linguística está interessado em medir o quanto o uso destes elementos é importante para a interpretação destes textos. Para isso ele irá entregar os textos originais para um grupo de leitores e os mesmos textos sem emoticons e hashtags para outro grupo. Finalmente, ele irá comparar as impressões dos grupos sobre os textos.

Sua tarefa será escrever a versão inicial de um programa em Python para auxiliar o pesquisador a filtrar os textos. Adotaremos as seguintes regras simplificadas para a classificação dos elementos:

- **Palavras:** sequência de letras (sem acento).  
Exemplos: UNICAMP algoritmos programacao
- **Números:** sequência de dígitos precedidos ou não do sinal negativo (-). Não trataremos números reais, complexos ou com pontos indicando a separação em grupos de três dígitos.  
Exemplos: 2018 -100000
- **Hashtag:** caractere # seguido de letras.  
Exemplos: #python #unicamp #amoviajar
- **Emoticons:** grupo de dois ou mais caracteres que não se enquadram nas descrições anteriores. São compostos principalmente por caracteres de pontuação, mas podem conter letras, números ou serem precedidos por #.  
Exemplos: :-) #:-) :D <3

### Descrição da entrada

---

Nesta versão inicial, a entrada já virá pré-processada como uma sequência de N+1 linhas, com a primeira linha contendo o número N e as demais os elementos a serem classificados. Veja o exemplo abaixo:

```
5
#ilovepython
<3
happy
```

```
hacking  
:-)
```

Note que uma entrada contendo apenas o número 0 é uma entrada válida.

## Descrição da saída

---

A saída conterá os elementos que foram classificados como palavras simples ou números seguida de um pequeno relatório do número de hashtags ou emoticons removidos, quando for o caso.

- **Hashtags:**

- Caso `<h>` > 1 hashtags tenham sido removidas, a seguinte mensagem deve ser emitida:

```
<h> hashtags foram removidas.
```

- Caso apenas 1 hashtag tenha sido removida, a mensagem virá no singular:

```
1 hashtag foi removida.
```

- Não é necessário emitir mensagem caso nenhuma hashtag tenha sido removida.

- **Emoticons:**

- Caso `<e>` > 1 emoticons tenham sido removidos, a seguinte mensagem deve ser emitida:

```
<e> emoticons foram removidos.
```

- Caso apenas 1 emoticon tenha sido removido, a mensagem virá no singular:

```
1 emoticon foi removido.
```

- Não é necessário emitir mensagem caso nenhum emoticon tenha sido removido.

Para a entrada usada como exemplo a saída será:

```
happy  
hacking  
1 hashtag foi removida.  
2 emoticons foram removidos.
```

## Testes para o SuSy

---

Esta tarefa contém 6 testes abertos e 4 testes fechados. Todos os elementos a serem processados estão de acordo com as regras apresentadas na primeira seção.

## Dicas de Python 3 para esta tarefa:

---

- Para verificar o primeiro elemento de uma string `s` escreva `s[0]`.
- Para verificar todos os elementos de uma string `s` com exceção do primeiro elemento escreva `s[1:]`.
- Para verificar se uma string `s` contém apenas dígitos use a função `s.isdigit()`.

- Para verificar se uma string `s` contém apenas caracteres use a função `s.isalpha()`.

## Orientações para submissão

---

Veja [aqui](#) a página de submissão da tarefa. Lembre-se que o arquivo a ser submetido deve se chamar `lab05.py`. No link [Arquivos auxiliares](#) há um arquivo [aux-05.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 exhibe 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 2.

O prazo final para submissão é 30/09/2018. Recomenda-se a realização da tarefa antes da Prova 1.

---