

Prova Final/2ª Chamada de Programação 1 (IF968)

Fernando Castor
Centro de Informática
Universidade Federal de Pernambuco

4 de julho de 2016

Obs₁: ao terminar a prova, levante a mão, espere que o professor dê o “ok” e envie sua prova via email, em um arquivo chamado `seulogin.py` para `castor@cin.ufpe.br`. O assunto do email deve ser `P1_FINAL_2016_1`.

Obs₂: suas funções devem funcionar com quaisquer valores para seus parâmetros e não apenas com os que aparecem nos exemplos.

1. **(6,0 ptos.)** Construa uma função chamada `tokenizar` que recebe um string e o quebra em uma lista de palavras (strings). Por exemplo:

```
>>> tokenizar("Duas palavras")
['Duas', 'palavras']
```

```
>>> tokenizar("Duas  palavras 42")
['Duas', 'palavras', '42']
```

```
>>> tokenizar("Pala.vragi.gante")
['Pala', 'vragi', 'gante']
```

```
>>> tokenizar("Ignore: virgulas, dois pontos, pontos de exclamacao, etc.")
['Ignore', 'vírgulas', 'dois', 'pontos', 'etc']
```

Nos dois últimos exemplos acima os pontos finais, as vírgulas e o dois pontos são tratados como se fossem espaços em branco. Isso significa que eles também são separadores de palavras, como pode ser visto no terceiro exemplo.

A função `tokenizar` e todas as funções que ela chamar (dentro do seu programa) devem funcionar de maneira recursiva, sem `while`, `for` e sem compreensão de listas. É fortemente recomendável que você crie várias funções auxiliares para pegar uma palavra, descartar espaços em branco no início de um string, etc. O uso de funções que tornam trivial a questão como `split`, `splitlines`, `rstrip`, etc., é terminantemente proibido e acarretará em nota 0 (ZERO). Isso também vale para a próxima questão.

2. **(5,0 ptos.)** Construa uma função `formatar` que tem dois parâmetros: um string correspondendo a um texto a ser formatado e um número correspondente ao limite de caracteres em uma linha. A função produz como resultado uma lista de listas de palavras. O exemplo a seguir ilustra o funcionamento dessa função:

```
>>> formatar("Duas palavras", 9)
[['Duas'], ['palavras']]

>>> formatar("Palavragigante", 5)
[['Palavragigante']]

>>> formatar("I must not fear. Fear is the mind-killer...", 21)
[['I', 'must', 'not', 'fear', 'Fear', 'is', 'the'], ['mind', 'killer']]
```

No primeiro exemplo, a função é chamada com um string contendo duas palavras e com um limite de 9 caracteres por linha. Como "Duas" tem 4 caracteres, "palavras" tem 8, no total essa linha tem 12 caracteres enquanto o limite é de 9. Por causa disso, o resultado é uma lista com duas listas como elementos: a primeira corresponde à primeira linha, que contém apenas a palavra "Duas", e a segunda lista corresponde à segunda linha, que contém apenas a palavra "palavras". O resultado seria esse mesmo que houvessem diversos espaços em branco entre as duas palavras, antes ou depois delas.

No segundo exemplo, como há uma palavra que, sozinha, tem mais caracteres que o limite da linha, essa palavra é colocada em uma linha só dela. Por fim, o terceiro exemplo apresenta um texto mais complicado, que inclui pontos finais ("."). Cada sinal de pontuação é tratado como se fosse um espaço em branco, assim como na questão anterior.