

Lista 5 - Módulo

Submissão: Crie um dois arquivos.py:
<matrícula> _main.py e modulo.py. Depois compacte eles em um arquivo .zip(só aceitarei essa forma de compactação) e submeta ao sigaa.

Exemplo:

```
modulo.py
#implementação da função
...
#implementação de outra função
...

000000_main.py
#importa o módulo

if __name__ == "__main__"
    #chama as funções do módulo
    #para testar sua impementação
```

O aluno 000000 criou os códigos fonte acima nos arquivos 000000 _main.py e modulo.py. Daí ele compactou usando .zip. Não é .rar, .7zip ou etc. É .zip. Por quê tem que ser .zip? Porque o .rar, usado por alguns, não funciona em qualquer sistema operacional.

1. (0,05) Escreva uma função chamada insert que tenha 3 parâmetros, listaA, listaB e um índice, e

então retorne uma cópia de listaA com os elementos de listaB inseridos no índice. Seu código deve funcionar com strings ou listas.

Exemplo:

```
>>> insert([1, 2, 3], ['a', 'b', 'c'], 2)
[1, 2, 'a', 'b', 'c', 3]
>>> insert("123","abc",2)
'12abc3'
```

2. (0,1) Escreva uma função chamada `ate_o_primeiro` que leva dois parâmetros, uma lista (ou string) e um objeto, e retorna uma cópia da lista até (mas não incluindo) a primeira ocorrência desse objeto, ou todos os elementos se esse objeto não estiver presente.

Exemplo:

```
>>> ate_o_primeiro([1, 2, 3, 4], 3)
[1, 2]
>>> ate_o_primeiro([1, 2, 3, 4], 9)
[1, 2, 3, 4]
>>> ate_o_primeiro('abcdef', 'd')
'abc'
```

3. (0,1) Escreva uma função chamada `corta_lista` que “corta” uma lista. Ou seja, dada uma lista e um índice, retorna uma cópia da lista, mas com os itens antes e depois do índice trocado. Como sempre, seu código também deve funcionar com strings. Exemplo:

```
>>> corta_lista([0,1,2,3,4,5,6,7,8,9], 3)
[4, 5, 6, 7, 8, 9, 3, 0, 1, 2]
>>> corta_lista("ABCDEFGX1234",7)
'1234XABCDEFG'
```