

Entregue todos os métodos em um arquivo chamado `lab6.py`. Critérios de avaliação: Principalmente (50%) o código funciona, os métodos fazem o que foi pedido, parâmetros de entrada e valores de retorno são corretos. Adicionalmente (50%) o código é **bem legível**, **o mais simples possível** e eficaz.

1. (2 pontos) Crie a função chamada `escreverNoArquivo` com dois parâmetros de entrada: um nome de arquivo e uma lista de objetos para "escrever" nesse arquivo, um objeto por linha. Os objetos da lista podem ser números, strings ou listas (listas com elementos quaisquer). O valor de retorno deve ser o número total de caracteres escritos no arquivo. Veja um exemplo abaixo:

```
>>> escreverNoArquivo("arq.txt", [-3.5, 9, [12.0, '@900#', 9.8], 'abece', 'k'])
36
```

Se você abrir o arquivo `arq.txt` em notepad, você deveria simplesmente ver estas linhas:

```
-3.5
9
[12.0, '@900#', 9.8]
abece
k
```

2. (4 pontos) Implemente a função chamada `retornarLista` cujo parâmetro de entrada é uma string que representa uma lista (lista de elementos quaisquer) e o valor de retorno é a lista. Veja o exemplo abaixo. É proibido importar bibliotecas de Python, você deve conseguir resolver a questão com *built-in* métodos e funções (dica: método `float` para transformar uma string numérica em um número, `strip` e `split` da classe `str` para trabalhar com strings e eventualmente vai ter que tratar de exceções).

```
>>> lista = retornarLista("[12.0, '@900#', 'a', 9.8]")
>>> type(lista)
<class 'list'>
>>> lista
[12.0, '@900#', 'a', 9.8]
```

3. (4 pontos) Crie a função `lerArquivo` para ler arquivos que contém um número ou uma string ou uma lista por linha. A função recebe um parâmetro: o nome do arquivo a ser lido. O valor de retorno deve ser uma lista com dados lidos em cada linha transformados para formato da respectiva classe (`float`, `list`, `str`). A função deve aproveitar a função `retornarLista` para transformar uma string de lista em um objeto da classe `list`. Veja o exemplo para o arquivo `exemplo.txt` que contém as seguintes linhas:

```
-3.5
4.0
-4.5
[12, '@900#', 9.8, -2, 'a', 'ABC']
bl
cm
dn
eo
fp
aba EEE bo
```

```
>>> lerArquivo("exemplo.txt")
[-3.5, 4.0, -4.5, [12.0, '@900#', 9.8, -2.0, 'a', 'ABC'], 'bl', 'cm', 'dn', 'eo', 'fp', 'aba EEE bo']
>>> lista = lerArquivo("exemplo.txt")
>>> type(lista[0])
<class 'float'>
>>> type(lista[3])
<class 'list'>
>>> type(lista[4])
<class 'str'>
```