

Segundo Maior

Parte 1

Considere o código a seguir, projetado para encontrar e retornar o maior número em uma lista de números arbitrários (`ints` e/ou `floats`), ou `None` se a lista recebida for vazia:

```
def largest_number(input_list):
    best_so_far = 0
    for i in range(len(x)):
        if i > best_so_far:
            best_so_far = i
    print(best_so_far)
```

Contudo, esse código não funciona escrito como acima. Na verdade, há *vários* problemas com esse trecho.

Tente agora:

O objetivo dessa função é sempre retornar o maior número em `input_list`.

Antes de continuar, tente identificar e explicar os bugs no código. Você pode assumir que todos os elementos são `ints` ou `floats`.

Após identificar os erros no código acima, corrija esses problemas e defina uma função `largest_number` que funcione propriamente. Não use funções integradas como `max`.

Parte 2

Agora, escreva e teste uma função chamada `second_largest_number` que recebe uma única lista contendo 0 ou mais números. Se há 0 ou 1 números na lista, sua função deve retornar `None`. Do contrário, ela deve retornar o *segundo maior* número na lista.

Você pode querer usar seu código da parte 1 como um ponto de partida.

Sua função não deve usar funções como `max`.

Você não precisa cuidar do caso em que elementos na lista de entrada não são números.

Por exemplo, executar o seguinte código:

```
print(second_largest_number([]))
print(second_largest_number([2]))
print(second_largest_number([94, 87, 20, 35]))
print(second_largest_number([1, 2, 3, 3, 3]))
print(second_largest_number([20, -1, -10]))
```

deve imprimir:

None

None

87

3

-1

(Note que no penúltimo exemplo, o maior, segundo maior e terceiro maior são todos 3 s. Seu código deve ser capaz de funcionar com números repetidos desse jeito, ao invés de procurar o segundo maior valor único).

Submissão

Comece fazendo um plano e testando alguns casos manualmente no papel. Então traduza para Python e teste usando o interpretador. Quando estiver pronto, faça upload do seu arquivo Python no **Problema 2.4** no