

# Computação I - Python

## Laboratório 8

Seguindo com nossas boas práticas, para cada um dos exercícios a seguir:

- antes de começar a escrever código, faça o estudo do problema e o planejamento de sua solução.
- lembre de botar a **documentação**:

```
'''Calcula e retorna a divisão de a por b;  
int, int -> float'''
```

- escolha **nomes elucidativos** para suas funções e parâmetros;
- pense em **dados de teste** relevantes para testar sua função.
- recorra ao **teste de mesa** quando estiver com dificuldade para entender algum erro de funcionamento ou resultado inadequado de sua função.
- **para fazer a entrega desta atividade prática, escreva suas funções no IDLE.**

Vamos lá!

---

1. Defina uma função que dado um valor (iteravel) de tipo string, list ou tuple, e um potencial elemento (elem), devolva a quantidade de vezes em que elem aparece em iteravel.  
**Importante!** Não é para usar as funções pré-definidas para esses tipos, como list.count, por exemplo. Use o comando for para percorrer a estrutura de entrada e calcular o resultado.
2. Defina uma função que dado um valor (iteravel) de tipo string, list ou tuple, um potencial elemento (elem), e dois índices não negativos (ini e fim), devolva a quantidade de vezes em que elem aparece no trecho de iteravel limitado por ini (inclusive) e fim (exclusive).  
**Atenção1:** elem pode não ocorrer em iteravel. Lide com isso.  
**Atenção2:** ini e fim podem não ser índices existentes no iteravel. Nesse caso considere apenas a parte de iteravel que possua índices na faixa [ini,fim[.  
**Dica:** fica bem fácil de fazer se você usar a função do exercício anterior.  
**Importante!** Não é para usar as funções pré-definidas para esses tipos, como list.count, por exemplo. Use o comando for para percorrer a estrutura de entrada e calcular o resultado.
3. Em programação, usamos o conceito *máscara* para indicar informações que podem ser mostradas ao usuário e informações que devem ser "mascaradas", isto é, escondidas. Numa implementação de um jogo da forca, usamos uma máscara para indicar que posições da palavra secreta o jogador já acertou, e cujo conteúdo portanto deve ser mostrado ao jogador, e que posições devem continuar escondidas. Cada vez que o jogador acerta uma letra da palavra oculta, a máscara é atualizada de forma a ficar com tracinhos nos lugares de letras que o jogador ainda não acertou e mostrar, no seu devido lugar, as letras que ele já acertou.

Por exemplo, se a palavra for "carta", a máscara atual for: - a - - a e o jogador adivinhar a letra "t", a máscara deveria ser atualizada para: - a - t a.

Escreva uma função chamada *atualiza\_mascara* que recebe como entrada uma string contendo a palavra secreta (no exemplo acima, essa palavra seria "carta"), uma lista cujos elementos são os caracteres da máscara atual (no exemplo acima ['-', 'a', '-', '-', 'a']), e uma string com a letra que ele acabou de escolher. Sua função deve atualizar a máscara, colocando a letra escolhida no seu devido lugar (caso esteja na palavra). Se a letra não estiver na palavra, a máscara deve ser retornada sem modificações. A função não terá valor de retorno.

4. Considere a série

$$\sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots$$

cuja soma converge para  $\frac{\pi}{4}$ , ou seja,

$$\lim_{n \rightarrow \infty} s_n = \frac{\pi}{4}$$

- (a) Escreva uma função que, dado  $n$ , calcule a soma da série até o termo  $n$ .
- (b) Escreva uma função que calcule a soma da série com erro inferior a 0,01, ou seja, calcule a soma desta série de modo que  $|s_n - \frac{\pi}{4}| < 0,01$ . Sua função deve retornar também o número de termos necessários para chegar a este erro, ou seja, sua função retornará uma tupla com os dois valores calculados. Importante: Utilize obrigatoriamente a função definida no item anterior. *Dica: `math.fabs(numero)` retorna o valor absoluto (módulo) de um número.*