

## LISTA 08

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

- Podem aparecer na especificação do retorno desejado nos enunciados o nome de uma variável ou parâmetro entre os sinais `|` e `;`, como por exemplo `|var|;`. Isso significa que espera-se o valor da variável neste lugar, e não seu identificador.
- Antes de começar a escrever o código faça um estudo do problema e o planejamento da sua solução
- Não se esqueça de documentar todas as suas funções.
- Escolha nomes **elucidativos** para suas funções e parâmetros;
- Pense em **valores** de teste relevantes para testar sua função. Ela tem alguma resposta esperada para valores negativos? Valores fracionários? Que tal testar também com valores no extremo do conjunto de dados de interesse da função (maiores valores esperados, menores valores esperados)?
- Quando estiver com dificuldade para entender algum erro de funcionamento ou resultado inadequado de sua função, não fique paralisado olhando para a tela! Pegue lápis e papel e recorra ao **teste de mesa**.
- Sua solução não deve usar de conceitos e bibliotecas que não foram vistos ao longo das últimas semanas.
- **para fazer a entrega desta atividade prática, escreva suas funções no editor do IDLE, salvando todas em um único arquivo.**
- **as questões que tiverem matéria que ainda não foi apresentada, serão desconsideradas**

- (1) 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).

**Atenção** elem pode não ocorrer em iterável. 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[`.

**Importante!** Não é para usar as funções pré definidas como `list.count`, por exemplo Use o comando `for` para percorrer a estrutura de entrada e calcular o resultado.

- (2) 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 não deve ser modificada. A função não terá valor de retorno.