

-----Exercícios FP-----

DICIONÁRIOS

1. O Peter gosta de fazer bolos. Ele tem algumas receitas e ingredientes. Infelizmente, ele não é bom em matemática. Podes ajudá-lo a descobrir quantos bolos ele pode fazer, considerando as suas receitas? Escreve uma função **cakes()**, que recebe a receita (dicionário) e os ingredientes disponíveis (também um dicionário) e retorna o número máximo de bolos que o Peter pode fazer (inteiro). Para simplificar, não há unidades para as quantidades (por exemplo, 1 kg de farinha ou 200 g de açúcar são simplesmente 1 ou 200). Ingredientes que não estão presentes nos objetos, podem ser considerados como 0.

SETS

2. Crie uma função chamada **elementos_unicos()** que receba uma lista de listas e retorne um conjunto que contém todos os elementos que aparecem exatamente uma vez em todas as listas combinadas. Se um elemento aparecer mais de uma vez ao longo de todas as listas, ele não deve ser incluído no conjunto retornado. Por exemplo, dado **[[1, 2, 3], [3, 4, 5], [5, 6, 7]]**, a função deve retornar **{1, 2, 4, 6, 7}**, pois os números 3 e 5 aparecem mais de uma vez.

LAMBDA

3. Escreva uma única expressão que use uma função lambda para ordenar uma lista de tuplas. Cada tupla contém dois elementos: um nome (string) e uma idade (inteiro). A lista deve ser ordenada pela idade, do mais jovem para o mais velho.
Por exemplo, dada a lista **[("João", 25), ("Ana", 20), ("Miguel", 30)]**, a expressão deve retornar **[("Ana", 20), ("João", 25), ("Miguel", 30)]**.
4. Escreva uma expressão que utilize uma função lambda para realizar a seguinte tarefa: Dada uma lista de strings, retorne uma nova lista com cada string revertida e convertida para maiúsculas, mas apenas para aquelas strings que tenham um número ímpar de caracteres. As strings com um número par de caracteres devem permanecer inalteradas.

RECURSIVIDADE

5. Escreva uma função recursiva em Python chamada **soma_recursiva()** que calcule a soma de todos os números inteiros de 1 até n, onde n é um parâmetro da função. Se n for menor ou igual a 0, a função deve retornar 0.

6. Desenvolva uma função recursiva chamada **contar_digitos_recursivo()** que receba um número inteiro positivo e retorne a quantidade de dígitos que esse número possui. Por exemplo, para o número 12345, a função deve retornar 5. Se o número for negativo, a função deve tratar como se fosse positivo (ignorando o sinal).

7. Crie uma função recursiva chamada **permutacoes_recursivas** que receba uma string e retorne todas as permutações possíveis dessa string. Uma permutação é um rearranjo das letras da string.
Por exemplo, as permutações da string "abc" seriam "**abc**", "**acb**", "**bac**", "**bca**", "**cab**", "**cba**". A função deve retornar uma estrutura que **não contenha** elementos repetidos.
Se for testada a string "aab" apenas deve ser retornado "aab", "aba", "baa"