

## Exercícios de Lista: Do Intermediário ao Avançado

Aqui estão mais 10 exercícios para você aprimorar suas habilidades em Python, organizados por nível de complexidade crescente.

### Nível Intermediário

#### 1. Verificando Palíndromos

Um **palíndromo** é uma sequência que lê o mesmo de trás para frente. Crie um programa que verifique se uma lista é um palíndromo.

- **Entrada:** Uma lista de elementos.
- **Saída:** True se a lista for um palíndromo, False caso contrário.

**Exemplo:** lista1 = ["a", "b", "c", "b", "a"] -> True lista2 = [1, 2, 3] -> False

#### 2. Removendo Todas as Ocorrências

Crie um programa que remova **todas** as ocorrências de um item específico de uma lista. Se o item não existir, nada acontece.

- **Entrada:** Uma lista de elementos e o item a ser removido.
- **Saída:** A lista original, mas sem o item especificado.

**Exemplo:** numeros = [1, 5, 2, 5, 3, 5] item\_a\_remover = 5

**Resultado Esperado:** [1, 2, 3]

#### 3. Encontrando o Segundo Maior Valor

Encontre o segundo maior número em uma lista de números. O programa não pode usar o método `.sort()` da lista.

- **Entrada:** Uma lista de números inteiros.
- **Saída:** O segundo maior número.

**Exemplo:** numeros = [10, 5, 20, 15]

**Resultado Esperado:** 15

#### 4. Rotacionando uma Lista

Crie um programa que rotacione uma lista para a direita por um determinado número de posições. Os elementos que saem do final da lista devem ir para o início.

- **Entrada:** Uma lista e um número de posições para rotacionar.
- **Saída:** A lista rotacionada.

**Exemplo:** lista = [1, 2, 3, 4, 5] posicoes = 2

**Resultado Esperado:** [4, 5, 1, 2, 3]

#### 5. Média das Sublistas

Você tem uma lista de listas. Calcule a média de cada sublista e armazene os resultados em uma nova lista.

- **Entrada:** Uma lista de listas, onde cada sublista contém números.

- **Saída:** Uma nova lista contendo as médias de cada sublista.

**Exemplo:** matriz = [[10, 20, 30], [5, 10, 15], [1, 2, 3]]

**Resultado Esperado:** [20.0, 10.0, 2.0]

## Nível Avançado

### 6. Combinando Listas Ordenadas

Dadas duas listas que já estão em ordem crescente, combine-as em uma única lista também em ordem crescente, sem usar o método `.sort()`.

- **Entrada:** Duas listas ordenadas.
- **Saída:** Uma única lista, ordenada, contendo todos os elementos das duas listas.

**Exemplo:** lista1 = [1, 3, 5] lista2 = [2, 4, 6]

**Resultado Esperado:** [1, 2, 3, 4, 5, 6]

### 7. Encontrando uma Sublista

Verifique se uma lista (`sub_lista`) está contida dentro de outra (`lista_principal`). Se a sublista for encontrada, o programa deve imprimir a posição (índice inicial) onde ela começa.

- **Entrada:** Duas listas, `lista_principal` e `sub_lista`.
- **Saída:** O índice inicial da sublista ou uma mensagem indicando que ela não foi encontrada.

**Exemplo:** principal = [10, 20, 30, 40, 50] sub = [30, 40]

**Resultado Esperado:** A sublista foi encontrada na posição 2.

### 8. Organizando um Texto

Dada uma lista de palavras, organize-as em uma única string de texto. O texto deve ter uma palavra por linha.

- **Entrada:** Uma lista de strings.
- **Saída:** Uma string de texto onde cada palavra da lista está em uma nova linha.

**Exemplo:** palavras = ["Olá", "mundo", "Python"]

**Resultado Esperado:** Olá mundo Python

### 9. Codificador Simples

Crie um codificador simples. Para cada letra em uma lista de strings, mova-a uma posição para a frente no alfabeto (por exemplo, 'a' vira 'b', 'b' vira 'c').

- **Entrada:** Uma lista de letras (strings de um único caractere).
- **Saída:** Uma nova lista com as letras codificadas.

**Exemplo:** letras = ["a", "b", "c"]

**Resultado Esperado:** ["b", "c", "d"]

### 10. Encontrando Pares com uma Soma Específica

Dada uma lista de números e um valor alvo, encontre todos os pares de números na lista que somam o valor alvo.

- **Entrada:** Uma lista de números e um valor alvo (inteiro).

- **Saída:** Uma nova lista de listas, onde cada lista interna contém um par que soma o valor alvo.

**Exemplo:** numeros = [1, 2, 3, 4, 5] alvo = 5

**Resultado Esperado:** [[1, 4], [2, 3]]