

Lista de Problemas – Funções

1. Par ou Ímpar

Crie uma função chamada ``par_ou_imp(ar(numero))`` que receba um número inteiro e retorne a string "par" se o número for par ou "ímpar" se for ímpar.

2. Contador de Vogais

Crie uma função ``contar_vogais(texto)`` que receba uma string e retorne a quantidade de vogais (a, e, i, o, u) presentes nela.

3. Contar Ocorrências de Letra

Crie uma função ``contar_letra(texto, letra)`` que receba uma string e uma letra, e retorne quantas vezes essa letra aparece na string.

4. Verificar Tipo Numérico

Crie uma função ``verificar_tipo(valor)`` que receba um valor e retorne se ele é do tipo ``int``, ``float`` ou outro.

5. Verificação de Login

Crie uma função ``verificar_login(usuario, senha)`` que receba um nome de usuário e uma senha, e verifique se eles existem em uma lista de usuários cadastrados.

Exemplo de estrutura:

```
usuarios = [  
    {"usuario": "ana", "senha": "1234"},  
    {"usuario": "carlos", "senha": "abcd"},  
    {"usuario": "joao", "senha": "senha123"}  
]
```

6. Cadastro de Novo Usuário

Crie uma função ``cadastr(ar_usuario(lista, novo_usuario, nova_senha))`` que adicione um novo dicionário à lista de usuários, desde que o nome de usuário ainda não exista. Use a estrutura do exercício 5.

7. Atualizar Senha

Crie uma função `atualizar_senha(lista, usuario, nova_senha)` que atualize a senha de um usuário existente. Se o usuário não for encontrado, a função deve informar isso. Use a estrutura do exercício 5.

8. Encontrar Repetidos

Crie uma função `nomes_repetidos(lista_nomes)` que receba uma lista de nomes e retorne uma nova lista contendo apenas os nomes que aparecem mais de uma vez.

9. Consulta Parcial de Produto

Dada uma lista de listas com peças automotivas no formato `[id, nome_produto, valor]`, crie uma função `buscar_produto(lista, termo)` que retorne todos os produtos cujo nome contenha o termo buscado (parcialmente).

Exemplo de dados:

```
pecas = [  
    [1, "Filtro de óleo", 35.90],  
    [2, "Pastilha de freio", 120.00],  
    [3, "Óleo de motor", 89.90],  
    [4, "Filtro de ar", 45.00],  
    [5, "Correia dentada", 150.00]  
]
```

10. Soma por Categoria

Crie uma função `soma_por_categoria(lista, categoria)` que receba uma lista de listas com peças automotivas no formato [id, nome_produto, valor, categoria] e retorne a soma dos valores dos produtos dessa categoria. Caso a categoria não exista, a função deve informar isso.

Exemplo de dados:

```
pecas = [  
    [1, "Filtro de óleo", 35.90, "Motor"],  
    [2, "Pastilha de freio", 120.00, "Freio"],  
    [3, "Óleo de motor", 89.90, "Motor"],  
    [4, "Filtro de ar", 45.00, "Motor"],  
    [5, "Correia dentada", 150.00, "Motor"]  
]
```

Bibliografia

FORBELLONE, A. L. V., EBERSPACHER, H. F., Lógica de Programação: A construção de algoritmos e estruturas de dados, São Paulo: Pearson, 2013.

MANZANO, J. A. N. G., OLIVEIRA, J. F., Algoritmos: Lógica para desenvolvimento de programação de computadores, São Paulo: érica, 2019.

MENEZES, N. N. C., Introdução a Programação com Python: algoritmos e lógica de programação para iniciantes. Editora Novatec, 3º edição, 2019.