

Desafios.**Instruções:**

- Desafios devem ser entregues na Plataforma Odette até **dia 04/11/2024 às 14h**.
- Você deve enviar um único arquivo **.zip** com a resposta de todos os 5 desafios. Envie apenas o arquivo compactado. **Não** envie os arquivos **.py**.

Conteúdo:

- Coleções: Dicionários.

-
1. Crie um programa para encontrar todas as chaves que estão associadas a um determinado valor em um dicionário. Para isso, crie uma função que receberá 2 parâmetros, o primeiro parâmetro será um dicionário a ser analisado, e como segundo parâmetro um valor a ser buscado. A função retornará uma lista com as chaves encontradas, essa lista pode ter diversos valores, apenas um ou nenhum.

Exemplo:

```
Cientes= {  
    "Nome": "Ana Maria Braga",  
    "Endereco": "Av. Maria Augusta, s/n",  
    "OperadoraCelular": "Vivo",  
    . . .  
}
```

Resultado esperado da função para localizar o valor "Maria": Retornar as chaves NOME, ENDEREÇO

2. A Picpay tem um sistema para controlar as compras de um cartão de crédito, esse sistema pode ser modelado por um dicionário com os campos **saldo**, com o saldo restante do cartão, **transações**, com o número de transações que foram realizadas, e **média**, com a média de gastos por transação. Escreva um programa para simular um uso desse cartão de crédito, para isso, crie uma função chamada **compra()** que recebe como parâmetros o dicionário com a conta e o valor da compra e retorna um novo dicionário para aquela conta, com o saldo devedor, número de transações e média de gastos atualizados.

Faça o teste com o seguinte dicionário:

```
conta = {'transações':0, 'saldo':1000, 'media':0}
```

Depois de serem realizadas 2 compras com as seguintes chamadas da função

compra(conta,100) e **compra(conta,200)** o dicionário deve ter os seguintes valores:


```
{'transações': 2, 'saldo': 700, 'media': 150.0}
```

-
3. Crie uma função que recebe uma palavra, essa função deve retornar as letras únicas para a construção desta palavra e a quantidade de cada letra utilizada.

Exemplo de entradas e saídas:

letras ("abracadabra")

```
Letra a aparece 5 vezes
Letra b aparece 2 vezes
Letra r aparece 2 vezes
Letra c aparece 1 vez
Letra d aparece 1 vez
```

	Semana 10		
	Admissão 2025	Curso Preparatório Python	29/10/2024

4. Escreva um programa para armazenar uma agenda de telefones em um dicionário. Cada pessoa pode ter um ou mais telefones e a chave do dicionário é o nome da pessoa. Seu programa deve ter as seguintes funções:

`incluirNovoTel` – essa função acrescenta um novo nome na agenda, com um ou mais telefones. Ela deve receber como argumentos o nome e os telefones.

`incluirTelefone` – essa função acrescenta um telefone em um nome existente na agenda. Caso o nome não exista na agenda, você deve incluí-lo, use a função anterior para incluir o novo nome.

`excluirTelefone` – essa função exclui um telefone de uma pessoa que já está na agenda. Se a pessoa tiver apenas um telefone, a pessoa deve ser excluída da agenda.

`excluirNome` – essa função exclui uma pessoa da agenda.

5. Você está trabalhando em um sistema de gerenciamento de estoque para uma grande empresa de varejo. A empresa possui um extenso catálogo de produtos e deseja otimizar a pesquisa de produtos por código. Para isso, você precisa criar uma função que inverta o dicionário que mapeia códigos de produtos para seus nomes.

Instruções:

Você recebe um dicionário chamado **estoque** onde as chaves são códigos de produtos (números inteiros) e os valores são os nomes dos produtos (strings). Sua tarefa é criar uma função chamada **`inverter_estoque(estoque)`** que aceite o dicionário **estoque** como argumento e retorne um novo dicionário, **estoque_invertido**, onde as chaves são os nomes dos produtos e os valores são os códigos correspondentes.

Desafio Adicional:

Além de inverter o dicionário, seu código deve lidar com possíveis colisões de nome, ou seja, situações em que dois produtos têm o mesmo nome. Nesse caso, você deve garantir que o dicionário **estoque_invertido** armazene todos os códigos associados a um nome de produto em uma lista. Seu código precisa considerar a possibilidade de múltiplos códigos para um único nome de produto. Por exemplo, se dois produtos tiverem o mesmo nome "Camiseta" com códigos diferentes, o dicionário **estoque_invertido** deve conter uma lista de códigos para a chave "Camiseta".