

## **Lista de Exercício 07 - Tuplas e Dicionários**

- 1) Crie um programa que contenha uma lista de tuplas, onde cada tupla contém o nome de uma cor e seus valores RGB (Red, Green, Blue), informadas pelo usuário. Para inserir os valores, o sistema deverá solicitar quantas cores o usuário deseja informar. Após a inserção, solicite ao usuário que digite o nome de uma cor e, se a cor estiver na lista, exiba seus valores RGB.

Lista de Tupla = cores rgb[(nome da cor, (r,g,b)), (nome da cor, (r,g,b))]

Entrada	Saída

4	0
Vermelho 255	0
0	255
0	Amarelo 255
Verde	255
0	0
255	Azul
0	Valores RGB para a cor Azul: Red=0,
Azul	Green=0, Blue=255

Entrada	Saída
---------	-------

```
4 0
Vermelho 255 0
0 255
0 Amarelo 255
Verde 255
0 0
255 Ciano
0 A cor 'Ciano' não foi
    encontrada na lista.
```

- 2) Crie um programa que leia as coordenadas x, y e z de dois pontos no espaço 3D, representados como tuplas. Calcule a distância entre esses dois pontos usando a

fórmula da distância Euclidiana:  $d_{ij} = (x_{i2} - x_{j1})^2 + (x_{i3} - x_{j1})^2 + (x_{i4} - x_{j1})^2$  Exiba a distância calculada.

Entrada	Saída
1.0 2.0 3.0 4.0 5.0 6.0	A distância entre os dois pontos é: 5.20

- 3) No Hospital Todos Juntos, a escala de plantão dos técnicos em enfermagem, para o mês de novembro, está sendo definida. Há 30 técnicos. Faça um programa que armazene o nome e o dia de trabalho (1 a 30) de cada técnico. Caso o usuário digite um dia inválido, deverá permanecer no loop até digitar um dia entre 1 e 30. Por fim, separe em duas tuplas que terão os nomes e os dias de trabalhos dos funcionários. Uma tupla constará os funcionários que irão trabalhar em dia par e a outra os funcionários que irão trabalhar em dia ímpar. Atenção: o primeiro index de cada tupla será o nome e o segundo index será o dia de trabalho. As duas tuplas juntas devem conter informações sobre os 30 técnicos.

Entrada	Saída
<b>Exemplo para 5 técnicos</b>	<b>Exemplo Saída</b>
Maria 1 Marcos 2 Luana 0 3 Marta 4 Cássio 5	Dia inválido! Digite um dia entre 1 e 30  Pessoas no plantão (dias pares): (['Marcos', 'Marta'], [2, 4]) Pessoas no plantão (dias ímpares): (['Maria', 'Luana', 'Cássio'], [1, 3, 5])



**ENGENHARIA E ANÁLISE DE DADOS**  
**Imersão em Python**  
**Professora: Joyce Teixeira**

- 4) Faça um programa que crie um dicionário onde as chaves serão os números de 1 a 15 e os valores serão o quadrado desses números.

Entrada	Saída
	{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121, 12: 144, 13: 169, 14: 196, 15: 225}

- 5) Uma pista de Kart permite **3 voltas** para cada um dos **4 corredores**. Escreva um programa que leia todos os tempos em segundos e os guarde em um dicionário, onde a chave é o nome do corredor. Ao final diga quem foi o campeão, apresentando o nome (com a todas as letras maiúsculas) e sua média de tempo (com duas casas decimais). Lembre-se: o campeão tem a menor média.

Entrada	Saída
---------	-------

```
Alice 30.5          29.1
31.2              28.0
29.8              Diana 30.0
Bob              31.5
32.0             30.2
33.5             O campeão é CARLOS com média de tempo
32.2             de 28.67 segundos.
Carlos 28.8
```



## ENGENHARIA E ANÁLISE DE DADOS

### Imersão em Python

Professora: Joyce Teixeira

- 6) Escreva um programa que faça a atualização no dicionário `numeros_maria` de acordo com os valores de `numeros_sara`. Porém, ele não deve adicionar novos valores, só atualizar os existentes. Os dicionários já estão definidos, não é necessário requerer informações do usuário. São eles:

- `numeros_maria = {'a': 100, 'b': 200, 'c': 300}`
- `numeros_sara = {'a': 300, 'b': 200, 'd': 400, 'c': 500, 'e': 250}`

Entrada	Saída
	Os valores do dicionário <code>numeros_maria</code> são: <code>{'a': 300, 'b': 200, 'c': 500}</code>



## ENGENHARIA E ANÁLISE DE DADOS

### Imersão em Python

Professora: Joyce Teixeira

- 7) Crie um programa que represente uma agenda de contatos. O programa deve oferecer as seguintes opções ao usuário:

1. Incluir contato: Permite ao usuário adicionar um nome e um número de telefone à agenda.
2. Excluir contato: Permite ao usuário remover um contato da agenda usando o nome como referência.
3. Buscar contato: Permite ao usuário buscar um contato na agenda pelo nome e exibir o número de telefone correspondente.
4. Sair. Exibe todo o dicionário e a mensagem “Programa finalizado.”.

Use um dicionário onde as chaves são os nomes dos contatos e os valores são os números de telefone.

Entrada	Saída
1 Alice 1234567890	Contato adicionado com sucesso!
2 Alice	Contato excluído com sucesso!
1 Bob 9876543210	Contato adicionado com sucesso!
3 Bob	Número de telefone de Bob: 9876543210.
1 Denis 999998888	Contato adicionado com sucesso!
2 Alice	Contato não encontrado na agenda.
4	Agenda: {'Bob': 9876543210, 'Denis': 999998888}
	Programa finalizado.



## ENGENHARIA E ANÁLISE DE DADOS

### Imersão em Python

**Professora: Joyce Teixeira**

- 8) Desenvolva um programa para o controle de estoque de uma loja. O programa deve permitir as seguintes operações:
  1. Incluir produto: Permite ao usuário adicionar um produto ao estoque. O usuário deve fornecer o nome do produto, a quantidade em estoque e o preço unitário.
  2. Excluir produto: Permite ao usuário remover um produto do estoque usando o nome como referência.
  3. Atualizar estoque: Permite ao usuário atualizar a quantidade em estoque de um produto já existente.
  4. Exibir todo o estoque: Exibe a lista de todos os produtos no estoque,

incluindo seus nomes, quantidades em estoque e preços unitários.

5. Calcular valor total do estoque: Calcula e exibe o valor total do estoque com base nos preços dos produtos.

6. Sair: Encerra o programa.

Use um dicionário onde as chaves são os nomes dos produtos e os valores são listas contendo a quantidade em estoque e o preço unitário.

Entrada	Saída
---------	-------

```
1
Camiseta
50
25.00
estoque. Estoque de "Camiseta"

3
Camiseta
40
atualizado para 40 unidades.

1
Calça Jeans 30
45.00

4
Produto "Calça Jeans" adicionado ao
estoque.

Lista de Produtos em Estoque:
Nome: Camiseta
Quantidade em Estoque: 40 unidades
Preço Unitário: R$ 25.00
Nome: Calça Jeans
Quantidade em Estoque: 30 unidades
Preço Unitário: R$ 45.00

5
6
Produto "Camiseta" adicionado ao
Valor total do estoque: R$ 2350.00
Programa finalizado.
```



## ENGENHARIA E ANÁLISE DE DADOS

### Imersão em Python

**Professora: Joyce Teixeira**

9) Crie um dicionário chamado contas para armazenar informações das contas bancárias. As chaves serão os números das contas (por exemplo, "001", "002") e os valores serão outros dicionários contendo as informações da conta: nome do titular e saldo.

Implemente um menu com as seguintes opções:

- Digite 1 para criar uma nova conta. Solicite ao usuário o número da conta, nome do titular e saldo inicial.
- Digite 2 para consultar o saldo de uma conta específica. Solicite ao usuário o número da conta e exiba o saldo.
- Digite 3 para realizar um saque em uma conta. Solicite ao usuário o número da conta e o valor a ser sacado. Certifique-se de que a conta

exista e que o saldo seja suficiente.

- Digite 4 para realizar um depósito em uma conta. Solicite ao usuário o número da conta e o valor a ser depositado.
- Digite 5 para sair do programa.

Após cada operação, exiba uma mensagem apropriada informando o resultado da operação (por exemplo, "Conta criada com sucesso", "Saldo da conta: x.x", "Saque realizado com sucesso", "Depósito realizado com sucesso"). Garanta que o programa seja executado em um loop até que o usuário escolha a opção 5 para sair. Certifique-se de que o programa lide adequadamente com casos em que o usuário consulta, saca ou deposita em uma conta que não existe.

Entrada	Saída
---------	-------

1	
001	
João Silva	
1000.00	Saldo da conta: 1000.00
2	
001	
	Saque realizado com sucesso.
3	
001	
500.00	
	Depósito realizado com
4	
001	
300.00	sucesso. Conta não
2	
002	encontrada.
3	
001	
900.00 5	Saldo insuficiente.
Conta criada com sucesso.	Programa finalizado.



**ENGENHARIA E ANÁLISE DE  
DADOS Imersão em Python**  
**Professora: Joyce Teixeira**