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

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

Entrada	Saída
4 Vermelho 255 0 0 Verde 0 255 0 Azul 0 0 255 Amarelo 255 255 0 Ciano	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: $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$ 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
Exemplo **para 5 técnicos**	Exemplo Saída
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])

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

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 numeros_maria são: {'a': 300, 'b': 200, 'c': 500}`

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

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	Produto "Camiseta" adicionado ao estoque.
3 Camiseta 40	Estoque de "Camiseta" atualizado para 40 unidades.
1 Calça Jeans 30 45.00	Produto "Calça Jeans" adicionado ao estoque.
4	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	Valor total do estoque: R$ 2350.00
6	Programa finalizado.

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

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