

Problema 1 – Restaurante Inteligente

Um restaurante armazena os pedidos do dia em uma **lista de dicionários**, onde cada pedido tem: cliente, itens (lista de dicionários com prato e preco).

Tarefas:

1. Crie uma função que receba o nome de um cliente e retorne o **valor total gasto** (somando todos os itens pedidos).
2. Crie uma função que descubra qual **prato foi o mais vendido** no dia.
3. Mostre um ranking com os **3 clientes que mais gastaram**, em ordem decrescente.

Exemplo:

```
1  pedidos = [  
2      {  
3          "cliente": "Ana",  
4          "itens": [  
5              {"prato": "Lasanha", "preco": 30},  
6              {"prato": "Suco de Laranja", "preco": 8}  
7          ]  
8      },  
9      {  
10         "cliente": "Bruno",  
11         "itens": [  
12             {"prato": "Pizza", "preco": 40},  
13             {"prato": "Refrigerante", "preco": 6},  
14             {"prato": "Sobremesa", "preco": 12}  
15         ]  
16     },  
17     {  
18         "cliente": "Carla",  
19         "itens": [  
20             {"prato": "Pizza", "preco": 40},  
21             {"prato": "Suco de Laranja", "preco": 8}  
22         ]  
23     }  
24 ]
```

Problema 2 – Academia e Desempenho dos Atletas

A academia guarda os atletas em uma **lista de dicionários**, cada um com: nome, idade, modalidades (lista de esportes), treinos (dicionário com o nome do esporte como chave e a quantidade de treinos realizados como valor).

Tarefas:

1. Crie uma função que calcule a **média de idade** dos atletas que praticam um esporte específico.
2. Crie uma função que, dado um atleta, informe qual esporte ele **mais treinou**.
3. Monte uma lista com os **atletas que praticam mais de 2 modalidades** e exiba seus nomes.

Exemplo:

```
1  atletas = [  
2      {  
3          "nome": "Lucas",  
4          "idade": 20,  
5          "modalidades": ["Natação", "Corrida"],  
6          "treinos": {"Natação": 12, "Corrida": 8}  
7      },  
8      {  
9          "nome": "Mariana",  
10         "idade": 25,  
11         "modalidades": ["Musculação", "Yoga", "Pilates"],  
12         "treinos": {"Musculação": 15, "Yoga": 10, "Pilates": 5}  
13     },  
14     {  
15         "nome": "João",  
16         "idade": 22,  
17         "modalidades": ["Corrida", "Ciclismo"],  
18         "treinos": {"Corrida": 20, "Ciclismo": 18}  
19     }  
20 ]
```

Problema 3 – Loja de Música Online com Estatísticas

Uma loja virtual armazena músicas em uma **lista de dicionários**, cada música com: título, artista, downloads, avaliacoes (lista de notas de 1 a 5).

Tarefas:

1. Crie uma função que calcule a **nota média de avaliação** de cada música.
2. Crie uma função que mostre qual artista tem o **maior número total de downloads** somando todas as suas músicas.
3. Monte um **ranking das músicas mais bem avaliadas** (ordem decrescente da média das notas).

Exemplo:

```
1  musicas = [  
2      {  
3          "titulo": "Back in Black",  
4          "artista": "AC/DC",  
5          "downloads": 6800,  
6          "avaliacoes": [5, 4, 5, 5, 4, 5]  
7      },  
8      {  
9          "titulo": "Stairway to Heaven",  
10         "artista": "Led Zeppelin",  
11         "downloads": 8900,  
12         "avaliacoes": [5, 5, 4, 5, 5, 5]  
13     },  
14     {  
15         "titulo": "Enter Sandman",  
16         "artista": "Metallica",  
17         "downloads": 8100,  
18         "avaliacoes": [5, 5, 5, 4, 4, 5, 5]  
19     },  
20 ]
```

Problema 4 – Ranking de Filmes

Você recebeu uma lista de filmes (cada filme é um **dicionário**) com os campos:

- titulo → nome do filme
- diretor → nome do diretor
- bilheteria → valor em milhões de dólares
- avaliacaoes → lista de notas de 1 a 10

Tarefas:

1. **Top 3 maiores bilheterias**
 - Crie uma função top_bilheteria(filmes) que retorne os 3 filmes com **maior bilheteria**.
2. **Top 3 melhores avaliados**
 - Crie uma função top_avaliacao(filmes) que calcule a **média das avaliações** de cada filme e retorne os 3 melhores.
3. **Bilheteria por diretor**
 - Crie uma função bilheteria_por_diretor(filmes) que retorne um **dicionário** onde a chave é o diretor e o valor é o **total de bilheteria** de todos os seus filmes.
4. **Campeão absoluto**
 - Crie uma função campeao(filmes) que mostre qual filme é a **melhor combinação** de bilheteria alta e avaliação média alta.

Exemplo:

```
1 filmes = [  
2     {  
3         "titulo": "Inception",  
4         "diretor": "Christopher Nolan",  
5         "bilheteria": 830,  
6         "avaliacoes": [9, 10, 8, 9, 10]  
7     },  
8     {  
9         "titulo": "Avengers: Endgame",  
10        "diretor": "Anthony Russo",  
11        "bilheteria": 2797,  
12        "avaliacoes": [9, 9, 10, 10, 9]  
13    },  
14    {  
15        "titulo": "The Dark Knight",  
16        "diretor": "Christopher Nolan",  
17        "bilheteria": 1005,  
18        "avaliacoes": [10, 10, 9, 10, 10]  
19    },  
20    {  
21        "titulo": "Jurassic Park",  
22        "diretor": "Steven Spielberg",  
23        "bilheteria": 1029,  
24        "avaliacoes": [8, 9, 9, 8, 9]  
25    }  
26 ]
```