

Lista de Exercício 08 - Funções

- 1) Crie uma função chamada **quadrado** que recebe um **número x como parâmetro**. A função deve **retornar um dicionário** onde as chaves são os números de 1 a x e os valores são o quadrado desses números. Teste a função no programa principal.

Entrada	Saída
15	{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}

- 2) Uma pista de Kart permite **3 voltas** para cada um dos **4 corredores**. No programa principal, implemente um código que leia o nome do corredor e todos os tempos em segundos e os guarde em um dicionário, onde a chave é o nome do corredor. Implemente duas funções:

1. Uma função chamada **calcular_media** que recebe uma lista de tempos como parâmetro e **retorna a média desses tempos**.
2. Uma função chamada **encontrar_campeao** que recebe o dicionário de tempos como parâmetro e usa a função **calcular_media** para calcular a média de tempo de cada corredor. A função deve identificar o campeão (corredor com a menor média de tempo) e **retornar o nome do campeão e sua média de tempo com duas casas decimais (use round(menor_media, 2))**.

No final do programa principal, exiba o nome do campeão em maiúsculas e sua média de tempo com duas casas decimais.

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 28.63 segundos.

- 3) 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. Finalizar programa.

Crie três funções para realizar cada uma das opções acima: **incluir_contato**, **excluir_contato** e **buscar_contato**.

- incluir_contato recebe como parâmetro o nome e o telefone
- excluir_contato e buscar_contato recebe como parâmetro o nome

No programa principal, um dicionário deve ser inicializado e o menu com as 4 opções deve ser exibido. Ao selecionar uma delas o usuário deve fornecer os parâmetros via input e a função equivalente à opção escolhida deve ser chamada. Ao finalizar o programa todo o dicionário deve ser exibido.

Entrada	Saída
---------	-------

```
1
Alice
1234567890
Contato adicionado com sucesso!

2
Alice
Número de telefone de Bob:

1
Bob
9876543210
9876543210.

3
Bob
Contato adicionado com sucesso!

1
Denis
999998888
Contato não encontrado na agenda.

2
Alice
Agenda:
{'Bob': 9876543210, 'Denis':
4
Contato adicionado com sucesso! 999998888} Programa finalizado.

Contato excluído com sucesso!
```

Questão Extra:

- 4) Escreva no programa principal um bloco de códigos que atualize o dicionário `numeros_maria` com base nos valores do dicionário `numeros_sara`. A atualização deve ser feita apenas nos valores já existentes no dicionário `numeros_maria`. Não é necessário adicionar novos valores. Os dicionários já estão definidos como segue:

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

Crie uma função chamada **atualizar_numeros** que recebe os **dois dicionários como parâmetros** e atualiza o dicionário `numeros_maria` de acordo com os valores do dicionário `numeros_sara`. Os valores no dicionário `numeros_maria` devem ser substituídos pelos valores correspondentes do dicionário `numeros_sara` se a chave existir em ambos os dicionários. A função não irá retornar nenhum valor. Irá imprimir o dicionário `numeros_maria` atualizado.

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