

# Programação I

## Dicionários (ficha 10)

*v1.0*

1. Considere o seguinte dicionário para representar uma lista telefónica:

```
lista_tel= {'joao':961111, 'ana':932222, 'carla':913333, 'manuel':964444}
```

Implemente a função `encontra_telefone(nome,ltel)` para devolver o número de telefone associado ao `nome`, no dicionário `ltel`. Se o nome não existir a função deve devolver zero.

```
print( encontra_telefone('ana',lista_tel) )
932222
print( encontra_telefone('joana',lista_tel) )
0
```

2. Desenvolva a função `novo_tel(nome,tel,dic)`. A função regista no dicionário `dic` a informação sobre o telefone `tel` associado ao `nome`. Não há retorno nem output. Exemplo:

```
novo('joana', 917777, lista_tel)
print( encontra_telefone('joana', lista_tel) )
917777
```

3. Implemente uma função `cliente(tel, ltel)` para a funcionalidade de pesquisa inversa. Com um número de telefone e os dados num dicionário, saber o nome da pessoa.

```
print( cliente(913333,lista_tel) )
carla
print( cliente(919999,lista_tel) )
DESCONHECIDO
```

4. Para facilitar a leitura da lista, implemente uma função `mostra_lista(ltel)` que imprime a lista por ordem alfabética e apresenta o resultado como no exemplo. Use o método `sort()` sobre as chaves do dicionário. Use também a formatação de strings.

```
mostra_lista( lista_tel )
ana: 932222
carla: 913333
joao: 961111
manuel: 964444
```

5. Considere um dicionário em Python cujas entradas são a tradução de algumas palavras entre o Inglês (chaves) e o Português (valores):

```
dicEnPt= {'white':'branco', 'the':'o', 'cat':'gato', 'mouse':'rato',
'chases':'persegue', 'black':'preto' }
```

6. Implemente uma função `traduz(pal,dic)`. Esta função deve devolver a tradução encontrada para a palavra `pal` no dicionário `dic`. Se não existir uma tradução naquele dicionário, deve devolver a palavra original. Teste a sua função e compare com os exemplos seguintes:

```
print( traduz('cat', dicEnPt) )
gato
print( traduz('bird', dicEnPt) )
bird
```

7. Tratando `dicEnPt` como variável global, implemente a função `palavra_para_portugues(pal)` que utiliza a função anterior para encontrar a tradução para palavras em inglês.

```
print( palavra_para_portugues('cat') )
gato
```

8. Implemente a função `lista_para_portugues(lista_pals)` traduz cada uma das palavras em `lista_pals` utilizando a função anterior.

```
print( lista_para_portugues(['mouse', 'white']) )
['rato', 'branco']
```

9. Implemente uma nova função `frase_para_portugues(frase)` que utiliza a função do exercício anterior para traduzir a string `frase`. Assuma que não há sinais de pontuação e que a tradução é puramente literal, palavra a palavra.

```
print( frase_para_portugues('the white cat chases the black mouse') )
o branco gato persegue o preto rato
```

10. Imagine que a associação entre adeptos e clubes é registada num dicionário como:

```
adeptos={'benfica':['joao','ana','carla'], 'sporting':['hugo','patricia'],
'porto':['jose']}
```

Implemente as funções:

1. `consulta_clube(nome, d)`: devolve o clube do adepto `nome`, de acordo com o dicionário `d`.
2. `mais_adeptos(d)`: devolve o nome do clube com mais adeptos de acordo com o dicionário `d`.

```
print( mais_adeptos(adeptos) )
benfica
clube= consulta_clube('patricia', adeptos)
print( clube )
sporting
print( consulta_clube('vitor', adeptos) )
None
```