

# Programação I

## 5. Função range, ciclos for

v1.0

1. Indique o valor (e respetivo tipo) resultante da avaliação (sem interpretador Python) das seguintes expressões. Confirme os resultados avaliando as expressões com o interpretador de Python em modo interativo.
  - (a) `range(4)`
  - (b) `range(3,7)`
  - (c) `range(7,3)`
  - (d) `range(7,9,3)`
  - (e) `range(0,9,3)`

2. Interprete mentalmente o código abaixo, indicando o output resultante. Confirme a sua resposta implementando um script com aquele código.

```
x=1
for y in [5,6,7]:
    x=y-x
print( x )          # qual o valor de x neste momento?
```

3. Elabore um programa que imprima as potências da base 2, desde  $2^0$  até  $2^n$ . O programa começa por ler o maior expoente. Apresente uma solução com `while` e outra solução com o ciclo `for`.

```
introduza a maior potencia de 2 que pretende consultar: 2
2**0== 1
2**1== 2
2**2== 4
```

4. Implemente a função fatorial utilizando um ciclo *for*.
5. Implemente a função `max_lista(lista)`. A função recebe um argumento com uma lista de valores numéricos e retorna o maior desses valores, ou zero se a lista estiver vazia. Apresente uma solução com o ciclo `while` e outra com o ciclo `for`.
6. Implemente a função `e_primo(n)` que indica se o número  $n$  é primo ou não. Para verificar se um  $n^\circ$  é primo pode guiar-se pelo exemplo apresentado na aula teórica (slide 37, 08-iteracao.pdf)
7. Utilizando a função implementada no exercício anterior, implemente uma nova função `mostra_primos(n)` que imprime os números primos entre 2 e  $n$  (valor inserido pelo utilizador).
8. Utilizando ainda a função `e_primo(n)`, implemente uma nova função `conta_primos(n)` que conta o número de números primos entre 2 e  $n$ .
9. Implemente um programa que pede uma lista com valores inteiros e que invoca a função `media(lista)` para efetuar o cálculo da média desses valores. Utilize a função `eval()` para converter a *string* para uma lista.

```
insira uma lista com valores inteiros: [1,3,3]
media: 2.33333333333
```

10. Implemente uma nova função `media_acima(lista,valor)` que calcula a média dos considerando apenas os valores da lista maiores ou iguais a `valor`.

```
>>> media_acima([1,2,5],4)
5.0
```

11. Implemente a função `converte_lst(lista)`, que aceita uma lista de algarismos (de 0 a 9) e devolve um inteiro que corresponde ao número composto por aqueles algarismos na base 10.

```
>>> converte_lst([1,2,5])
125
```

12. Implemente um programa que pede 3 listas ao utilizador (uma de cada vez), relativas a nome do produto, quantidade respetiva e o preço unitário (admita que o utilizador nunca se engana e insere sempre listas do mesmo tamanho e com o tipo de elementos correto). O objetivo é calcular os custos parciais de cada produto e o total.

```
lista com produtos: ['iogurte', 'pao']
lista com as quantidades: [3, 2]
lista com preco unitario: [0.55, 1.60]
iogurte 1.65 eur
pao 3.2 eur
TOTAL: 4.85 eur
```

13. Os primeiros dois valores da sequência de Fibonacci são 1. A partir do 3º, os valores correspondem à soma dos dois anteriores. Implemente a função `fibonacci(n)` que devolve numa lista os  $n$  primeiros números da sequência de Fibonacci.

```
>>> fibonacci(8)
[1,1,2,3,5,8,13,21]
```

14. Implemente a função `fibonacci_ate(v)` que devolve numa lista a sequência de Fibonacci até ao número  $v$ .

```
>>> fibonacci_ate(50)
[1,1,2,3,5,8,13,21,34]
```