

Programação 1  
Exercícios e Problemas

## Folha 4

(ciclos for)

### Exercícios

1. Descreva o que faz cada um dos seguintes programas:

- a) 

```
for i in range(10):  
    print(i)
```
- b) 

```
for i in range(2, 10):  
    print(i)
```
- c) 

```
for i in range(0, 10, 2):  
    print(i)
```
- d) 

```
for i in range(-10, -100, -30):  
    print(i)
```
- e) 

```
for i in range(0, 1, 2):  
    print(i)
```
- f) 

```
for i in range(10):  
    for j in range(10):  
        print(i*j)
```
- g) 

```
x = 10  
for i in range(x):  
    print(i)  
x = 5
```
- h) 

```
x = 10  
for i in range(x):  
    for j in range(x):  
        print(i)  
x = 2
```
- i) 

```
a = "Hello"  
for i in range(len(a)):  
    print(a[i])
```
- j) 

```
for i in "Hello":  
    print(i)
```

**2.** Indique como simplificaria o programa apresentado, alterando a forma de utilização da função `range`.

```
for i in range (50):  
    if i % 4 == 0:  
        print(i)
```

**3.** Considere o seguinte programa:

```
for i in range(10):  
    for j in range(i):  
        print(i, j)
```

**a)** O que faz este programa?

**4.** Considere este programa:

```
numero = int(input("Introduza um numero"))  
x = 0  
y = 0  
for n in range(numero):  
    if n % 2 == 0:  
        x += n  
    else:  
        y += n  
print(x)  
print(y)
```

**a)** O que faz este programa?

**b)** Altere o programa de forma a usar dois ciclos `for`, eliminando o uso de um `if`.

**c)** Altere o programa de forma a usar apenas um ciclo `for`, e não recorrendo ao uso de um `if`.

**d)** Altere o programa de forma a deixar de acumular valores quando o valor de `x` ou `y` forem iguais ou superiores a 20.

**5.** Considere o seguinte programa, que tem como objectivo escrever todos os tuplos  $(i,j)$  que podem ser formados com os algarismos 0 a 9 ( $i,j=0..9$ ):

```
i = 0  
while i <= 9:
```

```
j = 0
while j <= 9:
    print("(" + str(i) + "," + str(j) + ")")
    j = j + 1
print
i = i + 1
```

- a)** Reescreva o programa de forma a usar ciclos **for** em vez de ciclos **while**.
- b)** Altere o programa de forma a escrever apenas os tuplos onde *i* e *j* são pares.

**6.** Este programa tem como objectivo escrever a tabuada do número inteiro introduzido pelo utilizador. Reescreva o programa de forma a usar um ciclo **for** em vez de um ciclo **while**.

```
n = int(input("Escreve um número inteiro: "))
print ("Tabuada do", n, ":")
i = 1
while i <= 10:
    print (n, "x", i, "=", n * i)
    i = i + 1
```

**7.** Considere o seguinte programa que pede ao utilizador um número inteiro *k* e escreve no ecrã o número de vezes que se consegue efectuar a divisão inteira de *k* por 2 garantindo que cada uma dessas divisões obtém um quociente positivo.

```
i = int(input("Introduza um numero inteiro:"))
k = i
cont = 0
while k // 2 != 0:
    k = k // 2
    cont = cont + 1
print("O numero", i, "pode-se dividir", cont, "vezes por 2.")
```

- a)** Seria possível alterar o programa de forma a usar um ciclo **for** em substituição do ciclo **while**?
- b)** Comente o que caracteriza um ciclo **for** e identifique situações em que o seu uso é preferível em relação ao uso de um ciclo **while**.

**8. Considere o seguinte programa:**

```
texto = "programacao-1"
for c in texto:
    print(c)
```

**a)** O que faz este programa?

**b)** Altere o programa para imprimir tudo menos as vogais do texto dado, uma por linha.

**c)** Altere o programa de forma a imprimir as consoantes na mesma linha, como exemplificado abaixo:

```
prgrmc-1
```

**d)** Altere o programa de forma a executar a alínea anterior sobre uma linha de texto inserida pelo utilizador, como exemplificado abaixo:

```
Introduza o texto a processar: Ola, o meu nome e Manuel.
l, m nm Mnl.
```

**9. Considere o seguinte programa, igual ao do exercício anterior:**

```
texto = "programacao-1"
for c in texto:
    print(c)
```

Altere o programa de forma a recorrer ao *range* em substituição da iteração directa sobre o texto. Note que:

```
>>> texto = "ola"
>>> print(len(texto))
3
>>> print(texto[0])
o
>>> print(texto[2])
a
>>> print(texto[-1])
a
>>> print(texto[len(texto) - 1])
a
>>> print(texto[len(texto)])
```

```
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
```

```
print texto[len(texto)]  
IndexError: string index out of range
```

**10.** Considere o seguinte programa:

```
a = range(0, 10, 2)  
b = range(1, 10, 2)  
for i in a:  
    print(i)  
for i in b:  
    print(i)
```

**a)** O programa imprime os números pares entre 0 e 9 inclusive, e depois os números ímpares entre 1 e 9 inclusive. Altere o programa para que, usando os mesmos valores de intervalo (**range**), imprima a mistura intercalada das duas listas de números (*a* e *b*). Para o caso dado, o resultado é que sejam impressos todos os números inteiros de 0 a 9, inclusive. Note que **len(a) = 5**.

**b)** Altere o programa para que o limite superior da lista de números pares e ímpares seja inserido pelo utilizador. Os números dados podem ser diferentes dando origem a listas de números pares e ímpares de tamanhos diferentes. Intercale os números enquanto tal for possível e imprima os números restantes a seguir. Segue-se um exemplo de execução:

```
>>> Introduza o limite superior para os números pares:  
20  
>>> Introduza o limite superior para os números ímpares:  
10  
0 1 2 3 4 5 6 7 8 9 10 12 14 16 18
```

## Problemas

1. Escreva um programa que leia um inteiro  $n$  e escreva os números pares existentes entre 0 e  $n$ , inclusive, um por linha. Utilize um ciclo **for** para o efeito.
2. Escreva um programa que leia um inteiro  $n$  e um inteiro  $m$  e escreva os números entre 0 e  $n$ , inclusive, de  $m$  em  $m$ . Utilize um ciclo **for** para o efeito.
3. Escreva um programa que leia um inteiro  $n$  e escreva a soma dos números inteiros entre 0 e  $n$ , inclusive. Utilize um ciclo **for** para o efeito.
4. Escreva um programa que leia um inteiro  $n$  e escreva quantos números pares existem, entre 0 e  $n$ , inclusive. Utilize um ciclo **for** para o efeito.
5. Escreva um programa que peça ao utilizador um número inteiro  $k$  e escreva no ecrã o número de múltiplos de 3 que não sejam múltiplos de 2, no intervalo de 0 a  $k$ , inclusive.
6. Escreva um programa que leia um inteiro  $n$  e escreva a soma dos números inteiros ímpares entre 0 e  $n$ , inclusive. Utilize um ciclo **for** para o efeito.
7. Escreva um programa que leia um inteiro  $n$  e escreva a soma dos seus divisores. Utilize um ciclo **for** para o efeito.
8. Escreva um programa que leia um inteiro  $n$  e escreva no ecrã o seu factorial. Utilize um ciclo **for** para o efeito.
9. Escreva um programa que leia um inteiro positivo  $n$  e escreva no ecrã a soma dos números positivos ímpares até ao número, inclusive. Para tal utilize um ciclo **for** e não recorra a comandos alternativos.
10. Escreva um programa que leia dois inteiros positivos  $n$  e  $m$  e escreva no ecrã os números entre 0 e  $n$ , inclusive, enquanto a soma destes for menor que  $m$ . Use um ciclo **for** para o efeito.

11. Escreva um programa que recebe um número  $k$  e escreva no ecrã uma tabela com  $k$  linhas e  $k$  colunas, em que cada posição apresenta a multiplicação da linha pela coluna da tabela. Para tal recorra a ciclos **for**. Segue um exemplo

```
>>> Introduza um número inteiro: 10
0 0 0 0 0 0 0 0 0 0
0 1 2 3 4 5 6 7 8 9
0 2 4 6 8 10 12 14 16 18
0 3 6 9 12 15 18 21 24 27
0 4 8 12 16 20 24 28 32 36
0 5 10 15 20 25 30 35 40 45
0 6 12 18 24 30 36 42 48 54
0 7 14 21 28 35 42 49 56 63
0 8 16 24 32 40 48 56 64 72
0 9 18 27 36 45 54 63 72 81
```

12. Escreva um programa que leia um inteiro positivo  $n$  e crie um *desenho* igual ao apresentado abaixo, em que largura e comprimento são iguais a  $n$ . Note que, nas posições em que a linha e a coluna são pares, o símbolo apresentado é um **#** e, nas restantes, o símbolo apresentado é um **+**. Para tal recorra a ciclos **for**.

```
# + # + #
+ + + + +
# + # + #
+ + + + +
# + # + #
```

13. Escreva um programa que leia um inteiro positivo  $n$  e crie um *desenho* igual ao apresentado abaixo, em que largura e comprimento são iguais a  $n$ . Para tal recorra a ciclos **for**.

```
+ + + + +
# + + + +
# # + + +
# # # + +
# # # # +
```

14. Escreva um programa que leia a linha de texto inserida pelo utilizador e escreva um carácter desse texto por linha. Para tal recorra a ciclos **for**. Segue-se um exemplo de interacção:

```
>>> Insira o texto: Hello World!  
H  
e  
l  
l  
o  
  
W  
o  
r  
l  
d  
!
```

15. Escreva um programa que receba uma sequência de caracteres do utilizador e imprima os caracteres por ordem inversa, um por linha, sem recorrer a comandos alternativos. Recorra a ciclos **for**.