

## Folha 3

**1. Sem usar o computador**, siga o funcionamento de cada um dos programas seguintes anotando a variação dos valores das variáveis, e indique **o que escreve no *standard output***. Diga ainda qual é o **objetivo do programa** (i.e., escreva o enunciado de um problema que pode ser resolvido pelo programa).

a) Considere que o valor dado no input é 12.

```
n = int(input())
s = 0
i = 1
while i <= n:
    s = s-i*i
    i += 1
print(s)
```

Depois de resolver o problema, guarde o programa em `progl_a.py` e execute-o com [re-direcionamento do standard output](#) para passar a ser um ficheiro `res_a`, assim: `python progl_a.py > res_a`. Pode executar `cat res_a` na *shell* de comandos para ver o conteúdo do ficheiro `res_a`.

b) Considere que os valores dados no input são 13 -4 18 4 5 16 -10 32 7 18 33 17 -4 20 -12, sendo dado um inteiro em cada linha.

```
n = int(input("Valor de n (positivo)? "))          # input com output de mensagem
x = int(input("Valor de x (nao nulo)? "))
c = 0
cm = 0
while c < n:
    c += 1
    y = int(input("Valor de y? "))
    if y%x == 0:
        cm += 1
print("%d em %d" %(cm,n))
```

Depois de resolver o problema, escreva o programa num ficheiro `progl_b.py` e os dados num ficheiro `dados_b`, colocando um inteiro em cada linha. Execute `progl_b.py` na *shell* com [re-direcionamento do standard input](#) para ser `dados_b`, assim:

```
python progl_b.py < dados_b
```

**2.** Em cada alínea, escreva um programa que analise uma **sequência de inteiros que termina por 0** e resolve o problema indicado. A sequência é lida do *standard input* (entrada padrão) e será dado um inteiro em cada linha.

a) **Verificar se todos os valores são divisíveis pelo primeiro valor dado.** O output é [Todos multiplos de d](#) ou [Nem todos multiplos de d](#), com `d` substituído por esse valor. O programa pode não ler a sequência até ao fim se for dado um número que não é múltiplo do primeiro valor.

b) **Contar o número de ocorrências do primeiro elemento.** O resultado será zero se a sequência não tiver nenhum valor além do terminador 0.

c) **Contar quantas vezes ocorre o segundo elemento**, se forem dados pelo menos dois valores além de 0. Note que o primeiro valor também conta se for igual ao segundo.

O comportamento deve ser como se ilustra a seguir.

\$ python prog2c.py	\$ python prog2c.py	\$ python prog2c.py	\$ python prog2c.py
5	5	5	5
-8	5	0	3
4	6	Nao aplicavel	0
6	4		Ocorrencias = 1
4	8		
8	-10		
-10	2		
5	5		
5	9		
9	-8		
-8	0		
0	Ocorrencias = 3		
Ocorrencias = 2			

3. Pretendemos determinar a **soma dos múltiplos de quatro de 1 até 1000 (inclusivé)**. Bastaria escrever `print(250*(4+1000)//2)`, se recordarmos que se trata da *soma dos primeiros 250 termos de uma progressão aritmética de razão 4* e primeiro termo igual a 4. Os programas seguintes implementam abordagens alternativas (menos eficientes) e que têm erros. **Identifique os erros em cada caso e corrija o programa.**

a) Programa 1:

```
k = 4
while k != 1000:
    s = 0
    s = s+k
    k = k+4
    print("Soma = " + str(s))
```

b) Programa 2

```
for valor in range(0,1000,4):
    soma += valor
print("Soma = %d" % soma)
```

4. Escreva um programa para determinar os **dois maiores inteiros positivos distintos** numa sequência de  $n$  inteiros lida do *standard input*. O valor de  $n$  é o primeiro valor dado. Admita que  $n \geq 1$ . Se não existirem inteiros positivos, deve escrever `Nao existem inteiros positivos, sem acentos`. Se os inteiros positivos forem iguais, deve escrever `Todos os inteiros positivos iguais a k`, com  $k$  substituído pelo valor correspondente. Caso contrário, escreva apenas o valor maior e o segundo maior, na mesma linha, separados por um espaço.