

# Linguagens de Programação 2020/2021

## Exercício 2

Implemente as seguintes funções:

1. Uma função, chamada **Maximo**, que calcule o valor máximo existente num vetor de inteiros. Esta função deve estar preparada para lançar uma exceção caso o vetor esteja vazio.
2. Uma função, chamada **Impares**, que receba um número inteiro  $n$  e que devolva um vetor com os números ímpares no intervalo  $[1, n[$ .
3. Uma função, chamada **EscreveVetor**, que recebe um vetor de inteiros e que devolve uma **string** com a representação do vetor. **Nota:** Para converter um objeto do tipo **int** para um objeto do tipo **string** use a função `to_string(objeto int)`.
4. Uma função, chamada **Intervalo**, que receba um vetor de inteiros e um intervalo  $[a, b]$ , com  $a, b \in \mathbb{R}$  e  $a < b$ , e que calcule quantos elementos do vetor pertencem ao intervalo  $[a, b]$ . Esta função deve estar preparada para lançar uma exceção que faça sentido no seu contexto.

Na função **main**:

1. Peça um número inteiro positivo  $x$  ao utilizador. Caso o número introduzido seja um inteiro não positivo o programa deverá continuar a pedir números ao utilizador até ser introduzido um número inteiro positivo. **Nota:** Não é necessário lançar uma exceção para valores não numéricos.
2. Caso  $x$  seja um número par crie um vetor de inteiros de tamanho 5, caso contrário o tamanho do vetor deve ser 7. Preencha o vetor criado pedindo os valores ao utilizador e escreva-o para o ecrã.
3. Use a função **Impares** para determinar quais os números ímpares existentes até no intervalo  $[1, m[$  sendo  $m$  o valor máximo existente no vetor.
4. Use a função **EscreveVetor** para escrever o vetor com os números ímpares determinado no ponto anterior para o ecrã.
5. Peça um intervalo  $[a, b]$  ao utilizador, com  $a, b \in \mathbb{R}$ . Caso seja introduzido um valor não numérico, o programa deverá lançar uma exceção.
6. Use a função **Intervalo** para determinar quantos elementos do vetor inicial pertencem ao intervalo.
7. Use um bloco `try...catch` para lidar com as três exceções referidas anteriormente apresentando mensagens de erro específicas para cada tipo de erro.

Exemplo de execução do programa (instruções indicadas com uma seta (->) são o output esperado):

```
x: 3
v: 7 8 4 5 6 7 3
  -> v = (7, 8, 4, 5, 6, 7, 3)
  -> Os impares existentes no intervalo [1, 8[ sao: (1, 3, 5, 7)
a: 3
b: 5
  -> Dos valores que introduziu, 3 pertencem ao intervalo [3, 5]
```