

### Atenção:

- Nos exercícios **1 e 2 a)** deve apresentar:
  - a análise do problema, especificando os dados de entrada, os resultados pretendidos e o processamento requerido
  - o respectivo algoritmo em pseudocódigo
  - a implementação do programa em linguagem C
- Para os exercícios **2 b) e 3** apenas é necessário apresentar a implementação do programa em linguagem C
- As funções desenvolvidas devem ser portáteis
- A prova é sem consulta
- Faça **uma pergunta em cada folha**
- Devem deixar **10-12 linhas em branco no início de cada folha** de prova

1. [5 valores] Desenvolva um **programa** que leia uma sequência de números inteiros positivos introduzidos de forma crescente. A introdução deverá terminar quando for introduzido um valor que não obedece a essa condição (a sequência ser crescente e o primeiro número ser maior do que zero). No final o programa deverá indicar o número de valores ímpares que foram introduzidos.

### Exemplos:

Sequência de valores introduzidos: 1, 10, 15, 19, 12	-> Resultado: 3
Sequência de valores introduzidos: 2, 4, 5, 7, 10, 9	-> Resultado: 2
Sequência de valores introduzidos: 0	-> Resultado: 0
Sequência de valores introduzidos: -3	-> Resultado: 0
Sequência de valores introduzidos: 2, 4, 5, 7, 7	-> Resultado: 2

2. As 2 alíneas desta pergunta devem ser resolvidas sem utilizar “arrays”. Resoluções que façam uso destas estruturas de dados, serão cotadas com zero valores.

- a) [4 valores] Faça uma função que receba como argumento um número inteiro **num**, e devolva um valor inteiro resultante da aplicação de uma das seguintes operações conforme o caso:

- Divisão por 2, se ele for par;
- Multiplicação por 3 e somar-lhe 1 se for ímpar.

Considere o seguinte protótipo para a função:

**int OpX (int num);**

### Por exemplo:

Se **num** for: **36** → a função deve devolver **18** (=36/2)

Se **num** for: **5** → a função deve devolver **16** (=5\*3+1)

- b) [4 valores] Desenvolva um programa que peça ao utilizador um número inteiro diferente de 13. De seguida, a função da alínea anterior deverá ser chamada repetidamente até retornar o número 1 ou o número 13. Em cada iteração o programa deverá mostrar o número devolvido pela função. Este processo deverá começar com o número introduzido pelo utilizador e continuado com o número que retorna da função. A saída do programa deverá ser estruturada como representado no exemplo seguinte.

Exemplo de execução do programa:

```
Introduza um número inteiro positivo diferente de 13: 6
Iteração 1: 3
Iteração 2: 10
Iteração 3: 5
Iteração 4: 16
Iteração 5: 8
Iteração 6: 4
Iteração 7: 2
Iteração 8: 1
```

**Nota:**

Utilize a função implementada na alínea a). Dado que o protótipo da função está bem definido, a resolução desta alínea é independente da anterior (podendo mesmo responder à alínea b) se não tiver respondido à anterior).

3. [7,0 valores] Faça uma **função** que simule o funcionamento de uma calculadora elementar para valores inteiros. A função recebe como argumentos um carácter (**c**) e duas *strings* (**num1**, **num2**). O carácter **c** deve corresponder a uma das 4 operações aritméticas elementares ('+', '-', '\*' e '/') e cada uma das *strings* é constituída por um conjunto de algarismos. A função deve calcular e devolver o **número inteiro** correspondente à **operação especificada** considerando como operandos os 2 números inteiros guardados em **num1** e **num2**.

Considere o seguinte protótipo para a função:

```
int calculadoraInteiros(char c, char num1[], char num2[]);
```

Assuma que quer o carácter **c** quer as strings **num1** e **num2** possuem valores válidos quando a função é chamada.

Por exemplo:

- a) se **c** = '+', **num1**="234" e **num2**="3218" o valor devolvido pela função deve ser o inteiro 3452  
b) se **c** = '\*', **num1**="234" e **num2**="20" o valor devolvido pela função deve ser o inteiro 4680

**Notas:**

Não pode usar a função `atoi()` nem a `sscanf()`

Caso necessite pode utilizar a função: `int strlen(char str[])`, a função `strlen` devolve o número de caracteres efectivos existentes na string "*str*" (sem contar com o '\0').