

Atenção:

- Para o exercício **1 e 2a)** 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 **2b) e 3** apenas é necessário apresentar a implementação do programa em linguagem C
 - As funções desenvolvidas devem ser portáteis
 - A prova é sem consulta
 - As **perguntas/alíneas** devem ser **realizadas em folhas independentes devidamente identificadas** (nome, número, unidade curricular, data)
 - Devem **deixar 5-6 linhas em branco no início de cada folha de prova**.
1. **[5,5 valores]** Elabore um programa que solicite ao utilizador uma sequência de números inteiros positivos. A sequência deve terminar quando for introduzido o número zero. O programa deve mostrar o número de valores da maior sequência de elementos que estejam sequencialmente ordenados (estritamente crescente).

Nota: Não é permitida a utilização de arrays.

Exemplo de execução do programa:

Introduza uma sequência de números inteiros positivos:

3
1
88
7
13
23
4

No. valores da maior sequência sequencialmente ordenada: 3

No exemplo o resultado é 3, porque a maior sequência de valores é 7, 13, 23.

2. a) **[4,5 valores]** Desenvolva uma função que devolva a soma dos 3 dígitos menos significativos de um número inteiro passado como argumento.

Considere o seguinte protótipo para a função:

```
int soma3Digitos(int numero);
```

Nota: Não é permitida a utilização de arrays.

b) [4,0 valores] Elabore um programa que, considerando uma sequência de 100 valores inteiros positivos inseridos pelo utilizador e recorrendo à função anterior, determine se a soma dos 3 dígitos menos significativos é maior do que a soma dos 3 dígitos mais significativos. O programa deverá garantir que os números inseridos são sempre positivos.

Nota: Não é permitida a utilização de arrays.

Exemplo de execução do programa:

```
Insira o 1º numero: 123456
Maior soma: menos significativos.
Insira o 2º numero: -34
Erro! Insira o 2º numero: 222111
Maior soma: mais significativos.
... (continuação de inserção de números)
Insira o 99º numero: 745655
Maior soma: Iguais.
Insira o 100º numero: 111234
Maior soma: menos significativos.
Fim do programa. Obrigado.
```

3. [6 valores] Desenvolva uma função que tenha como argumentos uma *string* “*st*”, assim como um vector de caracteres “*vecCar*” e um vector de inteiros “*vecInt*” ambos de tamanho “*tam*”.

A função deve guardar em cada um dos elementos de “*vecInt*” quantas vezes surge na *string* “*st*” o carácter guardado na posição correspondente de “*vecCar*”.

A função deve devolver o número de caracteres presentes na *string* “*st*”.

O protótipo da função deverá ser:

```
int NCarac (char st[], char vecCar[], int vecInt[], int tam);
```

Exemplos de utilização da função:

```
1- Se inicialmente
   char st[50]="hoje esta a chover!";
   char vecCar[5]= {'a', 'e', 'i', 'o', 'u'};

   Após a seguinte chamada da função
   nrcarac = NCarac (st, vecCar, vecInt, 5);

   vecInt ficaria com 

|   |   |   |   |   |
|---|---|---|---|---|
| 2 | 3 | 0 | 2 | 0 |
|---|---|---|---|---|

;
   e a função retorna 19 (guardado em nrcarac)

2- Se inicialmente
   char st[50]="vamos passear ao sol no parque";
   char vecCar[4]= {'p', 'q', 'r', 's'};

   Após a seguinte chamada da função
   nrcarac = NCarac (st, vecCar, vecInt, 4);

   vecInt ficaria com 

|   |   |   |   |
|---|---|---|---|
| 2 | 1 | 2 | 4 |
|---|---|---|---|

;
   e a função retorna 30 (guardado em nrcarac)
```