

Atenção:

- Para os exercícios **1** e **2a)** deve apresentar:
 - A análise do problema, especificando os dados de entrada, os resultados pretendidos e o processamento requerido;
 - O respetivo algoritmo em pseudo-código;
 - 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 10-12 linhas em branco no início de cada folha de prova**.

1. **[5,5 valores]** Desenvolva um programa que leia uma sequência de números inteiros positivos, terminada por zero (0). O programa deverá ir somando a multiplicação entre cada par de números consecutivos. Se sobrar algum número, o programa deverá somá-lo ao resultado obtido. O número zero (0) não deverá ser usado nos cálculos. No final, deverá ser mostrado o resultado.

Exemplos de execução:

```
Introduza uma sequencia de numeros terminados por zero
1 2 3 4 5 0
Resultado: 19
```

Nota: O resultado deste exemplo é devido a $1*2+3*4+5=19$

```
Introduza uma sequencia de numeros terminados por zero
1 2 3 4 0
Resultado: 14
```

Nota: O resultado deste exemplo é devido a $1*2+3*4=14$

2. Esta pergunta deve ser resolvida sem utilizar "arrays". As resoluções que façam uso destas estruturas de dados, serão cotadas com zero valores.

- a) **[3,5 valores]** Desenvolva uma função que receba como argumento um valor inteiro positivo e um dígito (valor inteiro entre 0 e 9), conforme o seguinte cabeçalho: **int ContaDigito(int valor, int digito);**

A função deve retornar o número de vezes que o dígito surge no valor inteiro passado como primeiro argumento.

Se a função for chamada com o valor 23432 e for pedida a contagem do dígito 3 então a função deve retornar 2, uma vez que o dígito 3 surge 2 vezes em 23432. Se para o mesmo número for pedida a contagem do dígito 8 então a função deve retornar 0.

- b) **[5,0 valores]** Desenvolva um programa que peça números inteiros positivos ao utilizador e que, recorrendo à função anterior, contabilize o número de dígitos par que surgiram no total de todos os números. O programa deve terminar a solicitação de números inteiros quando for introduzido um valor inteiro sem qualquer dígito par (pode considerar o dígito 0 como sendo par). No final do programa deverá ser apresentado o número total de dígitos par introduzidos.

Exemplo de execução:

```
Introduza valores inteiros positivos (para terminar
indique um valor que não tenha dígitos par):
1234
22344
1200
-4
ERRO: Deve introduzir valores inteiros positivos
2468
1357
Número de dígitos par introduzidos: 13
```

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.

3. **[6,0 valores]** Desenvolva uma função que tenha como argumentos uma *string* “**origem**”, assim como uma segunda *string* “**token**” e um inteiro “**pos**”. A função deverá, partindo da posição “**pos**”, procurar todas as instâncias da *string* “**token**” na *string* “**origem**”, devolvendo o número de instâncias encontradas. A função não deve fazer qualquer distinção entre letras maiúsculas ou minúsculas ou se o “**token**” se encontra no início, meio ou fim de uma palavra. O protótipo da função deverá ser:

int ProcuraToken(char origem[], char token[], int pos);

Exemplo de utilização da função:

```
Se considerarmos:
char sr[250]=” O meu ultimo exame. Melhor que passe!”;
char tk[3]= ”me”;
int nvezes1, nvezes2;

Após a seguinte chamada da função
nvezes1 = ProcuraToken(st, tk, 0);
a função retorna 3 (guardado em nvezes1)

E após a seguinte chamada da função
nvezes2 = ProcuraToken(st, tk, 10);
a função retorna 2 (guardado em nvezes2)
```