

Atenção:

- Para o exercício **1** deve apresentar:
 - a análise do problema, especificando os dados de entrada, os resultados pretendidos e o processamento requerido
 - o respetivo algoritmo em pseudocódigo
 - a implementação do programa em linguagem C
- Para os exercícios **2 a), 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
- **As perguntas 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. [5,0 valores] Desenvolva um programa que determine o maior valor negativo de uma sequência de valores inteiros lidos a partir do teclado. A leitura da sequência termina quando for introduzido o valor zero (0).

O programa deve também informar a posição em que o maior valor negativo ocorreu na sequência lida. Se esse valor existir mais do que uma vez na sequência de valores inteiros lidos, deve assinalar a posição da primeira ocorrência.

Exemplo de execução do programa:

```
Diga o número inteiro 1: 14
Diga o número inteiro 2: -11
Diga o número inteiro 3: 103
Diga o número inteiro 4: 120
Diga o número inteiro 5: -4
Diga o número inteiro 6: -18
Diga o número inteiro 7: -8
Diga o número inteiro 8: 14
Diga o número inteiro 9: -4
Diga o número inteiro 10: 2
Diga o número inteiro 11: 0
```

O maior valor negativo lido é o -4 e surgiu na posição 5

Nota: Esta questão deve ser resolvida sem utilizar "arrays". Resoluções que façam uso destas estruturas de dados, serão cotadas com zero valores.

2. a) [4,0 valores] Desenvolva uma função que devolva o produto dos "nd" dígitos menos significativos de um número inteiro passado como argumento.

Considere o seguinte protótipo para a função:

```
int produtoDigitos(int num, int nd);
```

Nota: Não é permitida a utilização de arrays.

2. b) [4,5 valores] Elabore um programa que obtenha do utilizador um número inteiro positivo com cinco dígitos. Caso o número não tenha exatamente cinco dígitos o programa deverá continuar a pedir números até que o utilizador introduza um número com cinco dígitos.

Obtido um número com cinco dígitos, o programa deverá, usando a função da alínea anterior, verificar se o 5 dígito desse número é maior menor ou igual ao produto dos 4 dígitos menos significativos

Nota: Não é permitida a utilização de arrays.

Exemplo de execução do programa:

```
/* 1º exemplo de execução */
Insira um numero com cinco digitos: 100
Insira um numero com cinco digitos: 1223331
Insira um numero com cinco digitos: 10009
O quinto digito e maior do que o produto dos 4 digitos menos
significativos!

/* 2º exemplo de execução */
Insira um numero com cinco digitos: 123456
Insira um numero com cinco digitos: 1234
Insira um numero com cinco digitos: 12345
O quinto digito e menor do que o produto dos 4 digitos menos
significativos!

/* 3º exemplo de execução */
Insira um numero com cinco digitos: 81214
O quinto digito e igual ao produto dos 4 digitos menos
significativos!
```

3. [6,5 valores] Desenvolva uma função que tenha como argumentos uma *string* “**origem**” e uma segunda string “**token**”.

A função deverá procurar, por ordem, todos os caracteres da string “**token**” na string “**origem**”, devolvendo o número de caracteres encontrados. Quando é encontrado um caracter, a procura pelo caracter seguinte inicia-se na posição da string “**origem**” em que se encontrou o caracter anterior. A função não deve fazer qualquer distinção entre letras maiúsculas ou minúsculas.

A função terá como cabeçalho:

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

A saída de um programa de teste para a função seria:

```
Introduza a frase inicial:
Como este e o 4o exame de Ip. E melhor que passe.
Introduza o token a pesquisar
ISEC
Foram encontrados 3 caracteres do token ISEC .
```

Justificação: Como este e o 4o exame de Ip. E melhor que passe.