

#### Atenção:

- Esta prova é sem consulta.
- As resoluções das perguntas 2-a), 2-b) e 3 devem ser realizadas em três folhas independentes, devidamente identificadas (nome, número, unidade curricular, data). A primeira é a **folha de prova** e as outras são **folhas de continuação**.
- Deixe 10 linhas em branco no início de cada folha de continuação.
- Para a resolução da pergunta 2-a) deve apresentar:
  - A **análise do problema** (da função), especificando os dados de entrada, os resultados pretendidos, o conhecimento requerido e a estratégia.
  - A implementação da solução em linguagem C.
- Para a resolução da pergunta 2-b) e 3) apenas é necessário apresentar a implementação em linguagem C.
- **Nota:** a correção das alíneas 2a) e 2b) será efetuada de forma independente. O facto de não responder à questão 2b) não inviabiliza a resposta à questão 2a). Da mesma forma, *poderá resolver a questão 2b) sem resolver a questão 2a)*, considerando a *chamada da função* de acordo com o *protótipo definido*.

**2. a) [4 valores]** Desenvolva uma função que receba como argumentos um vetor de números inteiros, a sua dimensão e um número inteiro, de acordo com o seguinte protótipo:

`int EliminaValor(int v[ ], int tam, int num);`

A função deve procurar se o número *num* existe dentro do vetor *v* e, caso exista, deve eliminá-lo deslocando os elementos à sua direita uma posição para a esquerda, decrementando o tamanho *tam* em uma unidade, e preenchendo com zero o último elemento do vetor. Se houver mais ocorrências de *num* no vetor, o procedimento anterior deve ser repetido o nº de vezes que for necessário. No final a função deve devolver o valor de *tam* atualizado.

#### Exemplo de execução:

Considerando o vetor **v** inicialmente constituído pelos seguintes elementos:

**v**

102	23	51	48	76	120	51	114
-----	----	----	----	----	-----	----	-----

após a seguinte chamada à função: **EliminaValor(v, 8, 51);**

o vetor **v** será alterado conforme é mostrado abaixo e será devolvido o valor **6**.

**v**

102	23	48	76	120	114	0	0
-----	----	----	----	-----	-----	---	---

Considerando o mesmo vetor inicial, após a seguinte chamada à função: **EliminaValor(v,8,76);**

o vetor **v** será alterado conforme é mostrado abaixo e será devolvido o valor **7**.

**v**

102	23	51	48	120	51	114	0
-----	----	----	----	-----	----	-----	---

Finalmente, considerando o mesmo vetor inicial, após a seguinte chamada à função: **EliminaValor(v,8,95);**

o vetor **v** ficará inalterado (porque o número não existe no vetor) e será devolvido o valor **8**.

**2. b) [4 valores]** Imagine que um grupo de alunos finalistas do DEIS decidiu fazer **N** rifas para angariar dinheiro para a viagem final de curso. Para tal, foi criado um vetor de números inteiros cujo índice dos elementos corresponde ao nº de série das rifas -1. Ou seja, a rifa nº 1 corresponde à posição 0 do vetor, a rifa nº 2 corresponde à posição 1 do vetor, etc... Por outro lado, cada elemento do vetor contém o nº do cartão de cidadão da pessoa que comprou a respetiva rifa. As rifas são vendidas por ordem crescente do nº de série existindo a possibilidade da mesma pessoa comprar várias rifas em momentos diferentes, o que significa que podem existir repetições do mesmo nº de cartão de cidadão dispersas pelo vetor.

Desenvolva um programa que peça ao utilizador os números de identificação das pessoas que forem comprando as rifas, considerando que o período estipulado para a venda é suficiente para que todas sejam vendidas. Terminado esse período, o utilizador deve indicar o nº da rifa premiada (que tem que pertencer ao intervalo [1..N]). Em seguida, o programa deve anunciar o nº do cartão de cidadão da pessoa que a comprou e depois, chamando a função definida em 2a), eliminar todas as ocorrências do cartão de cidadão dessa pessoa no vetor. Finalmente, o programa deve indicar o nº total de rifas compradas por essa pessoa.

Para validar o nº de cartão de cidadão a ser inserido pelo utilizador, considere que já existe uma função definida cujo protótipo é **int VerificaCC(int id)**; a qual recebe como argumento um número inteiro: se este corresponder a um nº de cartão de cidadão válido, devolve 1. Caso contrário devolve 0.

**Exemplo de execução do programa** (assumindo **N** igual a **8**):

```
Rifa nº1 => indique nº do cartão de cidadão do comprador: 17846543
Rifa nº2 => indique nº do cartão de cidadão do comprador: 23419821
Rifa nº3 => indique nº do cartão de cidadão do comprador: 19856542
Nº de cartão inválido!
Rifa nº3 => indique nº do cartão de cidadão do comprador: 19856543
Rifa nº4 => indique nº do cartão de cidadão do comprador: 23419821
Rifa nº5 => indique nº do cartão de cidadão do comprador: 81726513
Rifa nº6 => indique nº do cartão de cidadão do comprador: 98204315
Rifa nº7 => indique nº do cartão de cidadão do comprador: 19856543
Rifa nº8 => indique nº do cartão de cidadão do comprador: 23419821

Indique nº da rifa sorteada: 4
-----
O vencedor tem o cartão de cidadão nº 23419821 e comprou um total de 3 rifas.
```

**3. [4 valores]** Desenvolva uma função que recebe a string “frase” e devolva a quantidade de palavras nessa string. Considere que as palavras estão separadas por um só espaço em branco, não existindo espaços em branco no início nem no fim da frase. Para além disso, a função deve ainda copiar a maior palavra (em número de caracteres) para a string “maior”. A função terá como cabeçalho:

```
int funcao(char frase[], char maior[]);
```

Considere o seguinte programa main:

O resultado será:

```
int main(void){
    char frase[]="Como eu gosto de IP";
    char maior[20];
    int n_palavras;

    n_palavras=funcao(frase, maior);
    printf("%d palavras\n", n_palavras);
    printf("maior palavra = %s\n", maior);
    return 0; }
```

```
5 palavras
maior palavra = gosto
```