

Atenção

- O exame é sem consulta.
- As resoluções das perguntas 2-a), 2-b) e 3 devem ser realizadas em três folhas independentes, devidamente identificadas (nome completo, número de aluno - do Inforestudante, unidade curricular e a data). A primeira folha (com o canto descartável) é a **folha de prova** e as outras (duas) folhas são **folhas de continuação**.
- Deixe 10 linhas em branco no início de cada folha de continuação.
- Nas **folhas de rascunho** coloque o nome completo e o número de aluno (do Inforestudante).
- Para a resolução de cada uma das perguntas deve apresentar apenas a implementação da solução em linguagem C. Sempre que possível, *comente* o código apresentado, para tornar a sua solução mais perceptível. Use, para o nome das variáveis, designações que representem o domínio do problema que está a resolver.
- Nota:** a **correção** das questões 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) [3,0 valores] Considere que pretende gerir um conjunto de terminais de levantamento de dinheiro (terminais do tipo ATM). Os terminais estão preparados para fornecerem notas de **50**, de **20**, de **10** e de **5** euros. No entanto, nos pedidos de levantamento de dinheiro, os terminais devem fornecer o menor número de notas.

Desenvolva uma função que recebe, por parâmetros, a quantidade de dinheiro a levantar, **valor**, um **array**, **notas**, com os diferentes tipos de notas existentes nos terminais (50, 20, 10 e 5) e um **array**, **existentes**, com o número de notas de cada tipo, existentes no terminal. A função devolve um **array**, **entrega**, com o número de notas de cada um dos diferentes tipos de notas que, no total, corresponde à quantidade de dinheiro a levantar. Considere que **n** é o número de elementos existentes em cada um dos três **arrays**. Por exemplo, se no terminal existirem 2 notas de 50, 15 notas de 20 e uma nota de 5, o **array existentes** deve ter os seguintes elementos: {2, 15, 0, 1}.

```
int EntregaDoPedido(int valor, int notas[], int existentes[], int entrega[], int n)
```

A função devolve **1** se for possível cumprir o pedido de levantamento de dinheiro. A função devolve **0** se não for possível cumprir o pedido.

2b) [4,0 valores] Desenvolva um programa que peça ao utilizador o valor que pretende levantar, garantindo que é um valor inteiro pertencente ao intervalo [5, 400]. Se o pedido de levantamento for possível, o programa mostra o número de notas de cada tipo que são entregues pelo terminal. Se o pedido de levantamento não for possível, o programa deve pedir ao utilizador para inserir outro valor (Listagem 1). Se o utilizador introduzir o valor zero, o programa termina (Listagem 2). O programa deve mostrar, no início e no fim do processo de levantamento de dinheiro, as notas, de cada tipo, existentes no terminal (Listagem 1 e Listagem 2).

```
Terminal -> estado inicial:
10 nota(s) de 50; 20 nota(s) de 20; 30 nota(s) de 10;
-----
Qual o valor que pretende [5, 400] -> 5
Entrega indisponível

Qual o valor que pretende [5, 400] -> 44
Entrega indisponível

Qual o valor que pretende [5, 400] -> 60
1 nota(s) de 50; 1 nota(s) de 5;
-----
Terminal -> estado final:
9 nota(s) de 50; 20 nota(s) de 20; 29 nota(s) de 10;
```

Listagem 1 – Primeiro exemplo da utilização do programa.

```
Terminal -> estado inicial:
10 nota(s) de 50; 20 nota(s) de 20; 30 nota(s) de 10; 40 nota(s) de 5;
-----
Qual o valor que pretende [5, 400] -> 1

Qual o valor que pretende [5, 400] -> 500

Qual o valor que pretende [5, 400] -> 0
-----
Terminal -> estado final:
10 nota(s) de 50; 20 nota(s) de 20; 30 nota(s) de 10; 40 nota(s) de 5;
```

Listagem 2 – Segundo exemplo da utilização do programa.

3. [3,0 valores] Implemente a função **substituiPalavras** que troca, na *string* **frase** (parâmetro da função), todas as palavras que tenham o mesmo tamanho da palavra contida na *string* **palavra** (parâmetro da função), por esta palavra.

A função devolve o número palavra substituídas e, por parâmetro, a **frase** alterada. Assuma que na **frase** existem pelo menos duas palavras e que todas as palavras estão separadas por, pelo menos, um espaço. Assuma que a *string* **palavra** não está vazia e que nesta *string* não existem caracteres espaço.

int substituiPalavras (char frase[], char palavra[])

Exemplo da utilização da função:

```
void main()
{
    char palavra[MAX]="XXXXX";
    char fr[MAX]= "Era uma linda coisa  sim pois era, pois!";

    printf("\n***\nForam substituidas %d palavras na frase: %s\n", substituiPalavras(fr, palavra), fr);
}
```

Exemplo da utilização do programa:

```
***
Foram substituidas 3 palavras na frase: Era uma XXXXX XXXXX  sim pois era, XXXXX
```