

Atenção:

- A prova é sem consulta.
- **As perguntas 2 a), 2 b) e 3 devem ser realizadas em três folhas independentes devidamente identificadas** (nome completo, número, unidade curricular, data).
- Deve **deixar 10-12 linhas em branco no início de cada folha de prova**.
- Para a resolução da pergunta **2a)** 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 da solução, em linguagem C.
- As funções desenvolvidas devem ser portáteis.
- Para os exercícios **2b) e 3** apenas é necessário apresentar a implementação do programa em linguagem C.

2. **a) [4 valores]** Elabore uma função que permita calcular o montante a pagar de estacionamento, em função do tempo decorrido, desde que o veículo entrou (hora e minutos da sua entrada) até que saiu (hora e minutos da sua saída) do parque de estacionamento. A função deverá ter o seguinte protótipo:

```
float a_pagar(int horaEntrada, int minEntrada, int horaSaida, int minSaida);
```

A função recebe, nos dois primeiros parâmetros, respetivamente, a hora e os minutos do momento da entrada no parque de estacionamento, e nos restantes dois parâmetros, a hora e os minutos da saída do parque. Considere que os valores recebidos pela função são sempre valores válidos, isto é, horas entre 0 e 23 e minutos entre 0 e 59, e que o tempo de saída do parque de estacionamento é sempre superior ao tempo de entrada no parque.

A função deverá retornar o montante a pagar, que deverá ser calculado dividindo o tempo de estacionamento por períodos de 15 minutos, e para cada período aplicar o tarifário indicado:

1º Período	Do 2º ao 8º período	Restantes períodos
0.35 €	0.25 €	0.20€
Montante máximo diário a pagar: 4.00€		

Para mais esclarecimento veja os seguintes exemplos de execuções da função:

Se o tempo de entrada for **22:50**, o tempo de saída que a função receberá nunca pode ser, por exemplo, de **01:00** (por ser inferior ao tempo de entrada). Se o tempo de saída for **23:21**, com o tempo de entrada acima referido, como o tempo total de estacionamento (31 minutos) equivale a 3 períodos, a função deverá retornar **0.85** (0.35€ do 1º período + 2 * 0.25€ do 2º e 3º períodos).

Se o tempo de entrada for **12:34** e o tempo de saída for **17:04**, como o tempo total de estacionamento (270 minutos) equivale a 18 períodos, a função deverá retornar **4.00** (pois o valor do cálculo dá $4.10 = 0.35€$ do 1º período + $7 * 0.25€$ do 2º ao 8º períodos + $10 * 0.20€$ dos períodos restantes, que é maior que 4.00).

b) [3,0 valores] Recorrendo à função desenvolvida na alínea **a)**, elabore um programa que peça ao utilizador a informação temporal da entrada e da saída de um veículo de um parque de estacionamento e mostre o montante a pagar.

Assim, o programa deve começar por solicitar a informação temporal da entrada, no formato hh:mm, solicitando que introduza novamente a informação enquanto esta não for válida. Posteriormente, deve ser solicitado ao utilizador a informação temporal referente à saída do parque, também no formato hh:mm, solicitando que introduza novamente a informação enquanto esta não for válida. A informação sobre a saída deve ser reintroduzida, enquanto esta não for temporalmente posterior à da entrada. Por fim, o programa deve mostrar o valor do montante a pagar, com duas casas decimais.

Exemplo de execução:

```
Entrada ( formato hh:mm ) 12:34
Saída  ( formato hh:mm ) 11:28
Saída  ( formato hh:mm ) 15:28
Total a pagar : 2.80 euros
```

Nota:

A correção das alíneas 2a) e 2b) será efetuada de forma independente. Assim, o facto de não responder à questão b) não inviabiliza a resposta à alínea a). Da mesma forma, poderá resolver a alínea 2b) sem resolver a 2a), considerando a chamada da função de acordo com o protótipo definido.

3. [5,0 valores] Considere o seguinte programa:

```
#include <stdio.h>
#include <string.h>

#define TAMFRASE 81
#define TAMPALAVRA 9

int UltimaPalavra(char frase[],char palavra[],int dmax);

void main()
{
    char frase[TAMFRASE],ultima[TAMPALAVRA];
    int p=0;

    printf("Indique uma frase: ");
    gets(frase);
    p= UltimaPalavra(frase,ultima,TAMPALAVRA-1);
    if (p>=0)
        printf("Ultima palavra: %s, que começa na posicao %d da frase\n",ultima,p);
    else
        printf("Nao foi possivel definir a ultima palavra\n");
}
```

Desenvolva apenas a função `UltimaPalavra` que recebe no primeiro parâmetro uma frase e devolve, através do segundo parâmetro, a última palavra dessa frase. O terceiro argumento da função define o número (máximo) de caracteres úteis que a última palavra pode conter. Assuma que as palavras (na frase) estão separadas por um ou mais espaços, podendo existir espaços no início e no fim da frase.

A função, para além da última palavra, deve devolver a posição (na frase) do primeiro carácter da última palavra ou, devolver, o valor -1 se não tiver a possibilidade de definir a última palavra, por esta não existir ou por ter mais caracteres que o número máximo de caracteres úteis definido no terceiro argumento da função.