

# Programação – Exame da Época de Recurso

8 de julho de 2021 – Duração: 90 minutos

## LEI, LEI-PL, LEI-CE

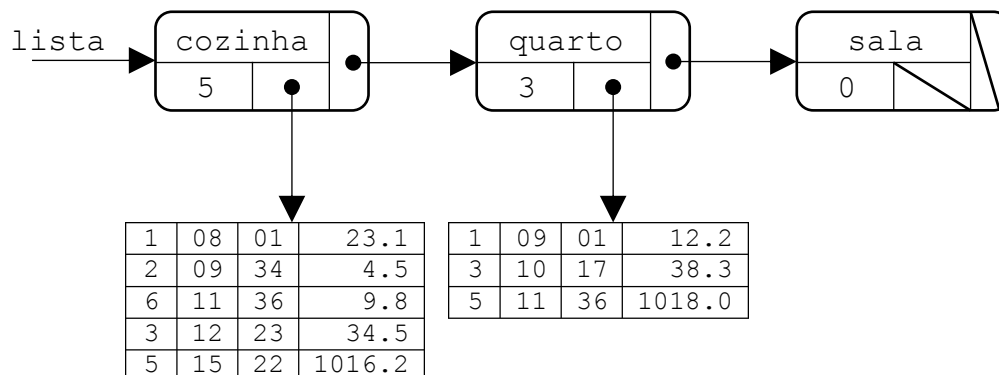
### Parte I

No contexto do desenvolvimento de um sistema para gerir os sensores distribuídos por várias divisões de uma casa, assuma a existência das seguintes definições:

```
typedef struct divisao Divisao, *pDivisao;
typedef struct leitura Leitura, *pLeitura;
struct divisao {
    char nome[100];           // nome da divisão
    int nr_leituras;           // número de leituras guardadas
    pLeitura leituras;         // tabela de valores dos sensores
    pDivisao prox;
};

struct leitura {
    int tipoSensor;            // tipo de sensor
    struct {int h, m;} hora;    // hora de recolha do valor
    double valor;
};
```

Cada divisão é caracterizada por um nome único (*string*), que a identifica e possui um conjunto de sensores (termómetros, barómetros, nível de CO<sub>2</sub>, ...), a partir dos quais são obtidas leituras/valores ao longo de um dia. A informação completa da habitação está armazenada numa estrutura dinâmica, constituída por uma lista ligada com nós para cada uma das divisões. A partir de cada um destes nós é possível aceder a uma tabela dinâmica do tipo *Leitura*, em que cada elemento corresponde a um sensor existente nessa divisão. Para cada sensor, a última leitura feita é sempre a que está registada na tabela. Quando surge uma nova, substitui a anterior (valor lido e hora a que foi feita a leitura).



1. Escreva uma função em C que imprima os valores de um determinado tipo de sensor, para todas as divisões da casa. A função recebe como parâmetros um ponteiro para a lista de divisões e o identificador do sensor a pesquisar. Devolve como resultado o maior valor encontrado. Caso o sensor não exista, a função devolve o valor -1.0. Para o exemplo da figura, caso se pedissem os valores para o sensor do tipo 3, a função deveria apresentar o seguinte no ecrã e retornar o valor 38.3:

```
cozinha: 34.5 [12:23]
quarto: 38.3 [10:17]
```

[Cotação 1.: 20%]

2. Escreva uma função em C que registre uma nova leitura de um sensor na estrutura dinâmica. A função recebe, como parâmetros, um ponteiro para o início da lista, o nome da divisão onde foi registado o novo valor, o identificador numérico do sensor, a hora (h – horas; m – minutos) e o valor registado. O cabeçalho da função é o seguinte:

```
pDivisao adiciona(pDivisao lista, char *div, int idS, int h, int m,
                  double val);
```

Se a divisão e o respetivo sensor existirem, deverá apenas ser atualizada a informação (valor e hora de leitura). Se existir a divisão, mas não esse sensor nessa divisão, deverá ser adicionado à tabela. Se a divisão ainda não existir, então deverá ser adicionado um novo nó à lista. As divisões na lista ligada estão ordenadas alfabeticamente. Os sensores nas tabelas não estão ordenados de acordo com nenhum critério em particular.

**[Cotação 2.:**

**10% - Atualizar sensor existente; 15% - Adicionar sensor a divisão existente; 15% - Adicionar divisão]**

## Parte II

Considere as seguintes definições:

```
struct venda{                                struct cc{
    struct{int dia, mes;} data;                int nif;
    int nif;                                   int id;
    char mat[15];                             };
};
```

Um ficheiro de texto contém informação sobre as cartas de condução de um determinado concelho. Cada carta é identificada pelo *nif* do titular (valor inteiro com 9 dígitos) e pela identificação numérica (valor inteiro positivo) da carta. A informação sobre as várias cartas surge em linhas consecutivas, não estando ordenada por nenhum critério em particular. A seguir pode consultar a forma como a informação surge ao longo das linhas, num exemplo de um ficheiro de texto com 3 cartas:

```
NIF: 123456789 ; ID: 334455
NIF: 123123123 ; ID: 20945
NIF: 987654321 ; ID: 1023
```

Um ficheiro binário armazena dados sobre as vendas de veículos automóveis nesse concelho, num determinado ano civil. Os dados sobre cada venda são armazenados numa estrutura do tipo `struct venda`, contendo a data da venda (dia e mês), o *nif* do comprador e a matrícula do automóvel. Este ficheiro binário é constituído por várias estruturas deste tipo, não estando ordenado por nenhum critério em particular. Podem existir várias vendas associadas ao mesmo comprador ao longo do ano.

3. Escreva uma função em C que obtenha a identificação das pessoas que compraram **2 ou mais carros** ao longo do ano. Esta informação deve ser guardada num ficheiro binário, a criar pela função. Os dados de cada um destes compradores devem ficar armazenados numa estrutura do tipo `struct cc`. Esta estrutura tem 2 campos, o *nif* do comprador e o identificador da sua carta de condução. Caso um destes compradores não exista no ficheiro de texto, então a identificação da carta deve ficar com o valor -1. A única informação a gravar no ficheiro binário são as estruturas dos compradores de 2 ou mais carros. A ordem pela qual as estruturas são colocadas no ficheiro é irrelevante. Cada comprador de 2 ou mais carros deve surgir uma única vez no ficheiro binário. A função recebe os nomes dos 3 ficheiros como parâmetro.

**[Cotação 3.: 40%]**