

# Programação – Exame da Época Especial

8 de setembro de 2021 – Duração: 90 minutos

## LEI, LEI-PL, LEI-CE

### Parte I

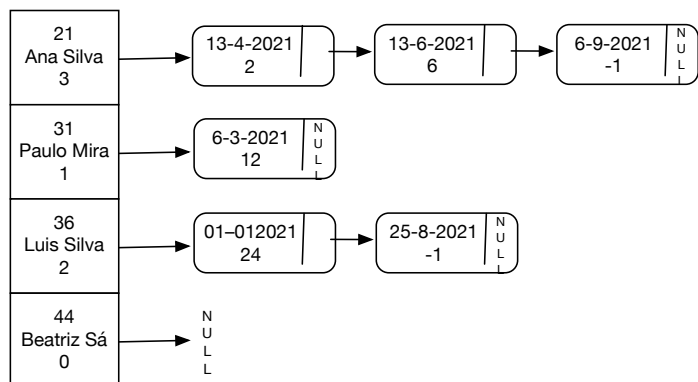
No contexto da gestão dos internamentos numa clínica, assuma a existência das seguintes definições:

```
typedef struct paciente Paciente, *pPaciente;
typedef struct internamento Inter, *pInter;
typedef struct {int dia, mes;} data;

struct paciente{
    int id;                // Identificador único
    char nome[100];        // Nome do paciente
    int ninternamentos;    // Número de internamentos nesse ano
    pInter lista;          // Ponteiro para a lista de internamentos
};

struct internamento{
    data din;              // Data de entrada
    int nDias;             // Número de dias internado
    pInter prox;
};
```

A informação completa dos pacientes registados e dos respetivos internamentos está armazenada numa estrutura dinâmica com as seguintes características: existe uma tabela dinâmica constituída por estruturas do tipo *Paciente*. Cada paciente registado na clínica tem os seus dados armazenados numa posição da tabela (Id numérico único, nome e número de internamentos registados no ano atual).



Os pacientes na tabela estão ordenados pelo valor do identificador. A cada paciente está associada uma lista ligada simples (através do campo *lista*), constituída por nós do tipo *Inter*, contendo informação dos seus internamentos. Cada nó da lista contém a data de internamento e o número de dias passado na clínica. Caso o paciente ainda esteja internado, o campo *nDias* tem o valor -1. Cada uma destas listas está ordenada por data. A figura em cima exemplifica um exemplo com 4 pacientes.

1. Escreva uma função em C que indique qual o paciente que já passou mais dias na clínica no ano atual. Na contabilização não devem ser considerados internamentos que ainda estejam a decorrer. A função recebe, como parâmetros, o endereço do vetor de clientes e a sua dimensão. Devolve o identificador do paciente com mais dias de internamento. Caso não existam pacientes com internamentos completos, a função devolve -1. Em caso de empate, a função devolve o identificador numérico mais baixo.

[Cotação 1.: 20%]

2. Escreva uma função em C que elimine da estrutura dinâmica um paciente com um determinado identificador numérico. Deve eliminar a estrutura do paciente do vetor dinâmico e a lista ligada de internamentos. Todo o espaço deve ser libertado. O cabeçalho da função é o seguinte:

```
pPaciente elimina(pPaciente v, int *nPaciente, int id);
```

A função recebe, como parâmetros, o endereço do vetor de clientes, o endereço de uma variável inteira onde está armazenada a dimensão do vetor dinâmico e o identificador do paciente a eliminar. Devolve o endereço do vetor dinâmico depois da atualização. Caso não exista nenhum paciente com o identificador indicado, a função não efetua nenhuma alteração.

[Cotação 2.: 40%]

## Parte II

Um ficheiro binário contém informação sobre os postos de carregamento de carros elétricos (PCCE) de um determinado país. Os dados relativos aos PCCE são armazenados numa estrutura do tipo *struct pcce*, contendo um identificador único, coordenadas geográficas (latitude e longitude), minutos de uso, potência de carregamento suportada e estado do equipamento. O estado pode possuir os seguintes valores: 0 (Disponível), 1 (Avariado) ou 2 (Em uso). O ficheiro binário é constituído por várias estruturas desse tipo, não estando ordenado por nenhum critério em particular.

```
typedef struct pcce{
    int id;
    float lat, lon;
    int minutos;
    float potencia;
    int estado;
} PCCE, *pPCCE;
```

Periodicamente é recebido um ficheiro de texto contendo informação sobre o estado de funcionamento dos PCCE. A informação de cada PCCE é descrita numa linha com o seguinte formato: *<id> <estado> <minutos\_de\_carga>*. O estado poderá ser qualquer um dos valores referidos anteriormente e os minutos de carga são os minutos acumulados desde a última atualização.

A informação no ficheiro de texto não está ordenada por nenhum critério em particular. Ao responder a esta questão, pode assumir que o mesmo *id* não surge em mais do que uma linha. A seguir ilustra-se um exemplo de um ficheiro de texto com este formato:

34	1	123
12	2	223
7	2	1103
77	0	111
...		

3. Escreva uma função em C que atualize os dados do ficheiro binário, a partir da informação disponível num ficheiro de texto. Para cada PCCE que surja no ficheiro de texto, deve ser atualizado o seu estado e os minutos de uso no ficheiro binário. Poderão eventualmente surgir identificadores no ficheiro de texto que não estejam no ficheiro binário. Neste caso, essa informação deve ser ignorada. A função recebe os nomes dos 2 ficheiros como parâmetro.

[Cotação 3.: 40%]