

# Programação – Exame da Época Normal

12 de julho de 2022 – Duração: 120 minutos

## LEI, LEI-PL, LEI-CE

---

### Parte I - Ficheiros

1. Considere a seguinte definição:

```
struct cliente{
    int id;           // id numérico do cliente
    char cc[10];      // Número do cartão de cidadão
    float valor;      // Valor gasto em compras desde o início do mês
    float v_divida;    // Valor em dívida desde o início do mês
    char cCert;       // Cliente certificado (S/N)
};
```

Uma pequena mercearia de bairro possui um ficheiro binário onde armazena informação sobre os seus clientes. A informação de cada cliente está armazenada em estruturas do tipo *struct cliente* e a organização do ficheiro é a seguinte: no início estão armazenados 2 inteiros A e B, o primeiro (A) indicando qual o número de clientes armazenados no ficheiro e o segundo (B) indicando quantos destes são clientes certificados. A seguir estão armazenadas A estruturas do tipo *struct cliente*. O ficheiro não está ordenado por nenhum critério em particular. Nesta altura ainda não foram contabilizados os clientes certificados. Todos os campos *cCert* das estruturas *struct cliente* têm o valor 'X' e o inteiro B tem o valor 0.

Escreva uma função em C que atualize a informação que se encontra no ficheiro binário. Devem ser efetuadas 3 atualizações:

- i. Devido à lei de proteção de dados é ilegal guardar o número de cartão de cidadão de clientes. Desta forma, todos os campos *cc* dos clientes devem passar a armazenar a string "NA".
- ii. Um cliente é considerado certificado se tiver um valor em dívida inferior a 10% do valor gasto em compras desde o início do mês. Com base nesta regra, atualize o campo *cCert* de todos os clientes armazenados no ficheiro (S/N).
- iii. Atualize o valor B que se encontra no início do ficheiro, com a indicação de quantos clientes estão certificados.

A função recebe o nome do ficheiro como parâmetro.

[Cotação: 30%]

## Parte II – Estruturas Dinâmicas

Considere as seguintes definições:

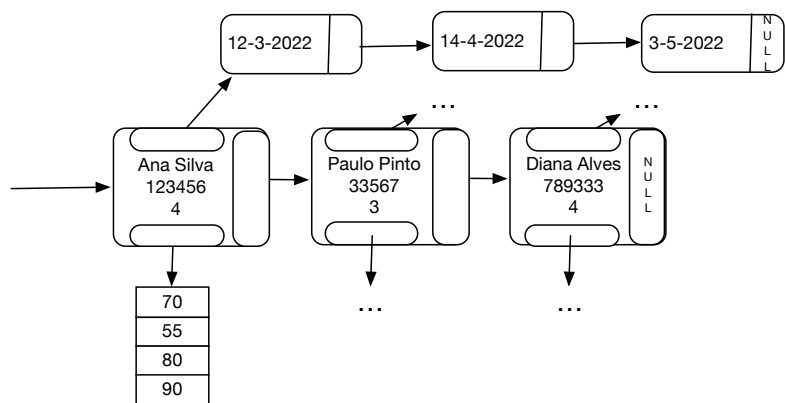
```
typedef struct data{
    int dia, mes, ano;
    struct data *prox;} *pdata;

typedef struct pessoa aluno, *paluno;
struct pessoa{
    char nome[100];    // Nome do aluno
    int num;           // Número do aluno
    int nprovas;        // Provas realizadas
    int* notas;         // Ponteiro para a tabela com as notas
    pdata faltas;       // Ponteiro para a lista com registo de faltas
    paluno prox;
};
```

Uma escola mantém uma estrutura dinâmica com informação sobre os seus alunos. A informação consiste em dados pessoais, notas obtidas nos testes já realizados e registo de faltas. Ao lado é mostrado um exemplo de uma estrutura dinâmica deste tipo.

Existe uma lista ligada principal constituída por nós do tipo *aluno*. Cada nó tem informação pessoal de um aluno: nome, número e número de provas realizadas.

A partir de cada nó desta lista, é possível aceder a informação adicional sobre o aluno. Um vetor dinâmico de inteiros, referenciado pelo campo *notas*, contém as notas obtidas nas provas realizadas por esse aluno. Além disso, o campo *faltas* é um ponteiro para uma lista ligada de nós do tipo *struct data* onde estão registados os dias em que esse aluno faltou.



2. Escreva uma função em C que calcule e apresente a nota média de todos os alunos presentes na estrutura dinâmica. A função recebe como parâmetro um ponteiro para o início da lista de alunos. Deve escrever na consola o nome, o número de provas e a média obtida por cada aluno (1 linha por aluno). O valor da média deve surgir com uma casa decimal. Se o aluno ainda não tiver realizado nenhuma prova deve surgir o valor 0.0.

```
Ana Silva, 4 provas: 73.7
...
```

[Cotação: 20%]

3. Escreva uma função em C que elimine da estrutura dinâmica o aluno que tiver mais faltas. Toda a memória dinâmica relativa a esse aluno deve ser eliminada (nó da lista principal, vetor com notas e lista de faltas). Caso existam vários alunos empatados nesta situação (maior número de faltas), a função não faz nenhuma alteração na estrutura dinâmica. A função recebe como parâmetro um ponteiro para o início da lista de alunos e devolve esse ponteiro depois da atualização.

[Cotação: 40%]

# Programação – Exame da Época Normal

12 de julho de 2022 – Duração: 120 minutos

## LEI, LEI-PL, LEI-CE

Nome: \_\_\_\_\_ Número: \_\_\_\_\_

### Parte III – Funções Recursivas

Considere as seguintes definições:

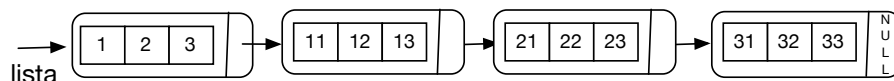
```
#define TAM 3

typedef struct dados no, *pno;
struct dados{
    int a[TAM];
    pno prox;
};
```

A função recursiva *printRec()* recebe como parâmetros um ponteiro do tipo *pno* e um valor inteiro:

```
void printRec(pno p, int x){
    if(p == NULL || x >= TAM)
        return;
    else{
        printf("%d\t", p->a[x]);
        printRec(p->prox, x+1);
        printf("%d\t", p->a[x]);
    }
}
```

Considerando que foi criada a seguinte lista ligada contendo 4 nós do tipo no:



Qual o output na consola se for feita a seguinte chamada:

```
printRec(lista, 0);
```

Resposta

[Cotação: 10%]