

Programação – Exame de Recurso

22 de julho de 2022 – Duração: 120 minutos

LEI, LEI-PL, LEI-CE

Uma pequena mercearia de bairro possui um ficheiro de texto onde armazena informação sobre os seus clientes: sequência alfanumérica descrevendo o id do cliente, número do cartão de cidadão e valor gasto em compras desde o início do mês. Os dados relativos a cada cliente estão armazenados em linhas consecutivas (1 linha por cliente), estando a informação separada por um ou mais espaços em branco. A seguir ilustra-se um excerto de um ficheiro com esta organização:

X123K4	12345	124.6
10PSA	94432	1005.67
QQ12AA	195431	798.0
...		

1. Escreva uma função em C que crie um ficheiro binário onde deverão ser armazenadas estruturas do tipo *struct cliente*, cada uma delas contendo informação sobre um cliente que surja no ficheiro de texto.

```
struct cliente{
    char id[20];           // id alfanumérico do cliente
    int cc;                // Número do cartão de cidadão
    float valor;           // Valor gasto em compras desde o início do mês
    char cVIP;             // Cliente VIP (S/N)
};
```

O ficheiro binário a criar deve conter apenas estruturas deste tipo (1 para cada cliente) e a ordem pela qual os clientes surgem no ficheiro binário é a mesma pela qual estão armazenados no ficheiro de texto. O campo *cVIP* de cada cliente deve ser preenchido com o carácter 'S' ou 'N', indicando se ele é ou não um cliente VIP. Considera-se que um cliente é VIP se o valor gasto em compras é superior à média dos gastos mensais de todos os clientes. A função recebe o nome dos dois ficheiros como parâmetro. Devolve 1 se tudo correr bem, ou 0, caso contrário.

[Cotação: 30%]

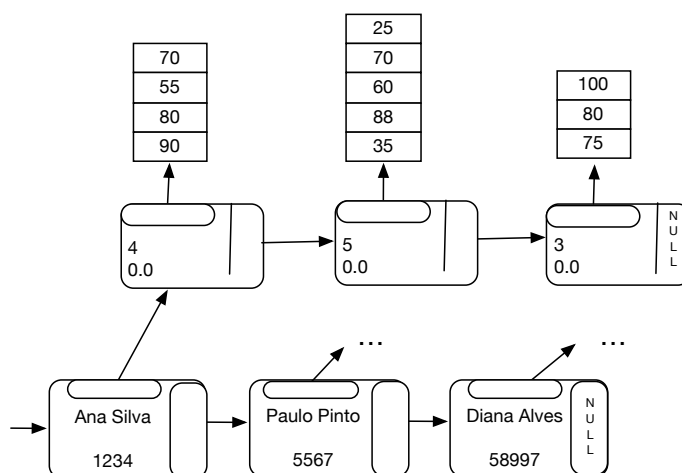
Considere as seguintes definições:

```
typedef struct prova teste, *pteste;
typedef struct pessoa aluno, *paluno;

struct prova{
    int *alíneas;          //Ponteiro para o vetor com as notas obtidas no teste
    int totA;              // Número de alíneas do teste
    float nota;            // Nota obtida no teste
    pteste prox;
};

struct pessoa{
    char nome[100];        // Nome do aluno
    int num;               // Número do aluno
    pteste lista;          // Ponteiro para a lista de testes
    paluno prox;
};
```

Uma estrutura dinâmica armazena informação sobre os alunos de uma escola, nomeadamente dados pessoais e notas obtidas nos testes realizados. 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 contém o nome e número de um aluno. A partir de cada nó desta lista, é possível aceder a uma lista ligada simples constituída por nós do tipo *teste*. Cada nó desta lista tem informação sobre um teste realizado por esse aluno. O campo *totA* indica o número de alíneas desse teste e o ponteiro *alíneas*



referencia um vetor dinâmico onde estão armazenadas as notas obtidas nessas alíneas (em percentagem). O campo *nota* guarda a classificação obtida no teste. As notas dos testes ainda não foram contabilizadas, para nenhum dos alunos presentes na estrutura dinâmica (todos os campos *nota* têm o valor 0.0).

2. Escreva uma função em C que calcule as notas obtidas pelos alunos em cada um dos testes que realizaram. A nota é calculada como uma média aritmética simples das pontuações obtidas em cada alínea. Os campos *nota* de todos os testes presentes na estrutura dinâmica devem ser atualizados com os valores contabilizados. Pode assumir que todos os testes têm, pelo menos, uma alínea. A função recebe como parâmetro um ponteiro para o início da lista de alunos. Devolve o número total de testes existente na estrutura dinâmica.

[Cotação: 20%]

3. Escreva uma função em C que adicione um novo teste a um aluno. Esta operação implica acrescentar um novo nó à lista de testes do aluno e o respetivo vetor com as pontuações. A nota do teste deve ser contabilizada e armazenada no campo *nota*. A função tem o seguinte protótipo:

```
paluno adicionaTeste(paluno p, char *nome, int num, int al, int notas[]);
```

Recebe um ponteiro para o início da lista de alunos, o nome e número do aluno ao qual deverá ser adicionado o teste, o número de alíneas do teste e as notas obtidas em cada alínea. Caso o aluno indicado ainda não se encontre na estrutura dinâmica, a função deve igualmente adicionar um novo nó à lista ligada de alunos. Ao responder a esta questão pode assumir que as listas não estão ordenadas por nenhum critério em particular. A função devolve um ponteiro para o início da lista de alunos atualizada.

[Cotação: 40%]

Programação – Exame de Recurso

22 de julho de 2022 – Duração: 120 minutos

LEI, LEI-PL, LEI-CE

Nome: _____ Número: _____

Funções Recursivas

4. Considere as seguintes definições:

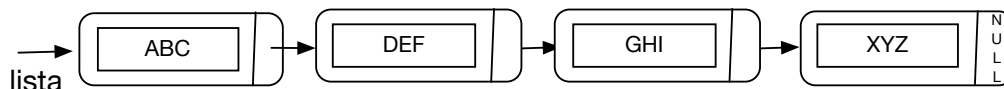
```
#define TAM 4

typedef struct dados no, *pno;
struct dados{
    char txt[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 || strlen(p->txt)<=x)
        return;
    else{
        printf("%s\t", (p->txt)+x);
        printf("%c\t", p->txt[x]);
        printRec(p->prox, x+1);
    }
}
```

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%]