

Programação – Exame Especial

6 de setembro de 2023 – Duração: 120 minutos

LEI, LEI-PL, LEI-CE

1. Uma empresa farmacêutica armazena toda a informação relativa aos medicamentos por ela produzidos num ficheiro de texto. Quando guardados neste ficheiro, cada fármaco é armazenado em 4 linhas consecutivas, como ilustrado no exemplo abaixo. Produtos consecutivos estão separados por uma linha em branco.

Designação: Associação de ácido clavolamico com amoxicilina

Nome comercial: Clavomox

Id: 9100903

Em produção: sim

Designação: digoxina

Nome comercial: Lanoxin

Id: 2245031

Em produção: nao

Designação: Azitromixina

Nome comercial: Zithromax

Id: 1635422

Em produção: nao

Ao longo do tempo, alguns medicamentos deixam de ser produzidos. Nesta altura, a informação relativa à produção no ficheiro de texto (linha 4 para cada produto) passa de *sim* a *nao*, indicando que a produção deste medicamento foi interrompida. Periodicamente, estes medicamentos devem sair do ficheiro de texto e passar a ser armazenados num ficheiro binário.

Escreva uma função em C que transfira a informação dos fármacos que deixaram de ser produzidos, do ficheiro de texto para um ficheiro binário. Cada registo a ser guardado no ficheiro binário deve ser do tipo *struct farmaco*. Repare que, nesta estrutura, o campo *terminouProducao* diz respeito à data em que o fármaco foi transferido para o ficheiro binário. A data atual é passada à função.

```
typedef struct {
    int d, m, a;
} data;

struct farmaco{
    int id;
    char designacao[100];
    char nome[100];
    data terminouProducao;
};
```

Considerando o ficheiro de texto exemplificado anteriormente, e a data atual 6/9/2023, a função deverá criar um ficheiro binário com 2 estruturas. Estes medicamentos (digoxina e Azitromixina) deverão ser retirados do ficheiro de texto.

2245031	digoxina	Lanomix	6	9	2023
1635422	Azitromixina	Zithromax	6	9	2023

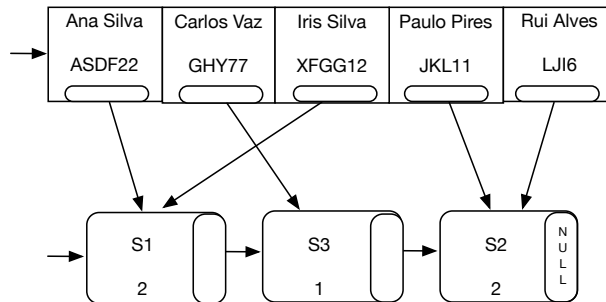
A função recebe como parâmetros os nomes dos ficheiros e a data atual. Devolve o número de produtos transferidos para o ficheiro binário, ou -1, em caso de erro.

[Cotação: 30%]

Considere as seguintes definições:

```
typedef struct sala no, *pno;
struct sala{
    char id[10];
    int total;
    pno prox;
};
```

```
typedef struct pessoa{
    char nome[100], num[20];
    pno p;
}aluno;
```



A informação dos alunos que vão realizar um teste está armazenada num array dinâmico, constituído por estruturas do tipo *aluno*. Este array está ordenado alfabeticamente pelo nome do aluno, não existindo 2 alunos com o mesmo nome. Existe ainda uma lista ligada simples, constituída por nós do tipo *no*, contendo informação sobre as salas onde vai ser realizada a prova. Cada nó da lista contém a identificação da sala e um contador indicando quantos alunos farão a prova nesse local. A lista não está ordenada de acordo com nenhum critério e não existem salas sem alunos associados. Cada aluno armazenado no array tem um campo *p*, referenciando a sala a que está associado. No exemplo da figura pode verificar-se que existem 3 salas e 5 alunos, estando o aluno Carlos Vaz associado à sala S3.

2. Escreva uma função em C que troque as salas atribuídas a 2 alunos. A função recebe, como parâmetros, os nomes dos alunos, o endereço inicial da lista de salas, o endereço inicial e a dimensão do array de alunos. A função devolve 1 se a troca for bem sucedida, ou -1, se algum dos alunos não pertencer ao array.

[Cotação: 20%]

3. Escreva uma função em C que adicione um aluno à estrutura dinâmica e o associe a uma das salas. A função tem o seguinte protótipo:

```
aluno* adicionaAl(aluno* v, int *tot, pno p, char *nome, char *num);
```

A função recebe o endereço inicial do array de alunos, o endereço de uma variável inteira onde se encontra a dimensão do array, um ponteiro para o início da lista de salas e os dados do novo aluno. Deve adicionar o aluno ao array dinâmico (mantendo a ordem alfabética) e associá-lo à sala que tenha menos alunos nessa altura. Caso existam várias salas com um número mínimo de alunos, a função deve escolher uma delas. A função devolve um ponteiro para o início do array depois da atualização. Ao responder a esta pergunta, pode assumir que a lista ligada tem, no mínimo, 1 sala.

[Cotação: 40%]

Programação – Exame Especial

6 de setembro de 2023 – Duração: 120 minutos

LEI, LEI-PL, LEI-CE

Nome: _____ Número: _____

4. Funções Recursivas

Considere as seguintes definições:

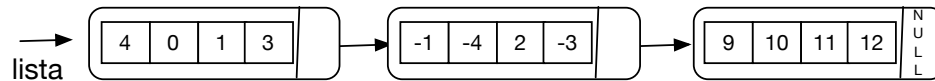
```
#define TAM 4

typedef struct dados no, *pno;
struct dados{
    int tab[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 < p->tab[x] || x >= TAM)
        return;
    else{
        if(x == p->tab[x])
            printf("%d\t", (p->tab[x]));
        printRec(p->prox, x+1);
        if(x >= p->tab[x])
            printf("%d\t", x);
    }
}
```

Considerando que foi criada a seguinte lista ligada contendo 3 nós do tipo *no*:



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

```
printRec(lista, 1);
```

Resposta