

Programação – Exame de Recurso

19 de julho de 2023 – Duração: 120 minutos

LEI, LEI-PL, LEI-CE

1. Considere a seguinte definição:

```
struct artigo{  
    char ref[10];  
    int qtd;  
    float preco;  
};
```

Um armazém guarda num ficheiro binário informação sobre os produtos que disponibiliza, incluindo as quantidades em stock e o preço unitário.

O ficheiro tem a seguinte organização:

- i. No início surge um valor real de precisão simples (*float*) contendo o preço médio dos artigos armazenados no ficheiro.
- ii. A seguir surgem várias estruturas do tipo *struct artigo*, uma para cada um dos produtos existentes.

Ao lado pode consultar um exemplo de um ficheiro com 4 produtos.

| |
|-----------------------|
| 168.6 |
| X12DF 12 23.5 |
| VG56 7 341.9 |
| LP34 1 12.0 |
| KLM13 102 297.2 |

Escreva uma função em C que efetue as seguintes operações:

- i) Atualize o preço de todos os produtos do ficheiro binário. Todos os preços devem ser aumentados 10%. O preço médio que surge no início do ficheiro também deve ser atualizado.
- ii) Crie um ficheiro de texto, para onde deve copiar informação sobre todos os produtos que tenham um stock inferior a um determinado limite. No ficheiro de texto que for criado, deve surgir em cada linha a referência e quantidade em stock dos produtos relevantes (informação separada por um ou mais espaços em branco). Considerando o ficheiro exemplificado, e assumindo que o valor limite é 10, deveria ser criado um ficheiro de texto com a seguinte organização:

| |
|--------|
| VG56 7 |
| LP34 1 |

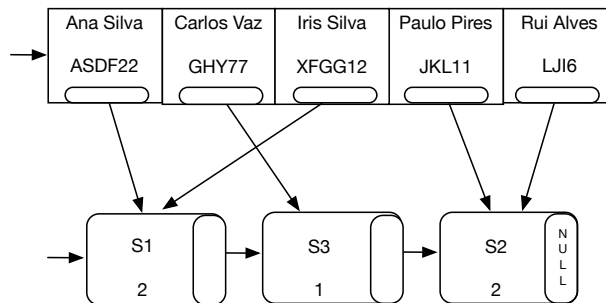
A função recebe como parâmetros os nomes dos ficheiros e o valor limite para a quantidade em stock a considerar. Devolve o número de produtos colocados no ficheiro de texto. Em caso de erro, devolve -1.

[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 verifique se 2 alunos vão realizar a prova na mesma sala. 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 os 2 alunos estiverem na mesma sala, 0 se estiverem em salas diferentes, ou -1, se algum dos alunos não pertencer ao array.

[Cotação: 20%]

3. Escreva uma função em C que transfira um aluno de uma sala para outra. A função tem o seguinte protótipo:

```
pno transfereAl(aluno* v, int tot, pno p, char *nome, char *nSala);
```

A função recebe o endereço inicial e a dimensão do array de alunos, um ponteiro para o início da lista de salas, o nome do aluno a transferir e o id da sala para onde o aluno deve ser transferido. Se a sala de onde o aluno sair ficar vazia, a função deve retirar esse nó da lista ligada e eliminar o espaço. A função devolve um ponteiro para o início da lista de salas depois da atualização.

[Cotação: 40%]