

Programação – Exame Especial

18 de julho de 2024 – Duração: 120 minutos

LEI, LEI-PL, LEI-CE

1. Considere as seguintes definições:

```
typedef struct Artigo artigo;
typedef struct Data data;

struct Data{ int hora, minuto, dia, mes, ano;};

struct Artigo{
    int cp;           // Código numérico único do artigo
    int np;           // Quantidade do artigo existente em stock
    float preco;      // Preço unitário (em euros)
    data venda;       // Data em que foi realizada a última venda
};
```

Um entreposto comercial armazena num ficheiro binário os dados sobre os artigos que comercializa. A organização do ficheiro binário é a seguinte:

- No início encontra-se um inteiro M que indica quantos artigos comercializa.
- Seguem-se M estruturas do tipo artigo, uma para cada um dos artigos comercializados.

A figura mostra um exemplo com cinco artigos. A informação no ficheiro binário não está ordenada por nenhum critério em particular.

5	1234501	1234522	1234507	1234580	1234570
	2000	1050	1850	34	3350
	1.18	1.50	1.25	0.15	0.25
	08 05 17 06 2024	09 25 15 06 2024	11 08 15 06 2024	11 40 11 06 2024	16 20 11 06 2024

Quando o entreposto faz uma venda de um determinado artigo, a um cliente, o ficheiro é alterado. As alterações são efetuadas nos campos que armazenam a quantidade que permanece em stock e a data da última venda.

Um ficheiro de texto contém uma ordem de encomenda, consistindo no pedido de compra de vários artigos. Cada linha contém 2 inteiros, especificando o código do artigo e a quantidade a comprar. A seguir pode consultar um exemplo de uma ordem de encomenda para 2 artigos. Os pedidos no ficheiro de texto não estão ordenados por nenhum critério em particular.

1234580	80
1234501	30

Escreva uma função em C que processe uma ordem de encomenda e atualize o ficheiro binário. Para cada artigo encomendado, a função deve atualizar a quantidade que fica em stock depois da encomenda e a data em que foi realizada a operação. No caso de ser encomendada uma quantidade superior ao que existe em stock, é apenas vendido o que existe no entreposto. No exemplo em cima, apenas serão vendidas 34 unidades do artigo 1234580.

A função recebe como parâmetros os nomes dos ficheiros binário e de texto e a data em que está a ser efetuada a encomenda. Devolve o valor total da encomenda, ou -1, caso exista algum problema no acesso aos ficheiros.

[Cotação: 30%]

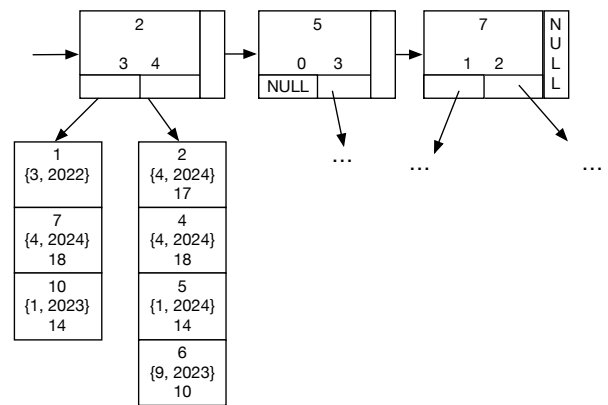
2. Considere as seguintes definições:

```
typedef struct {int mes, ano;} data;  
typedef struct gestor no, *pno;
```

```
struct gestor{  
    int idG;  
    struct emCurso *v1;  
    struct completo *v2;  
    int totV1, totV2;  
    pno prox;  
};
```

```
struct emCurso{  
    int id;  
    data inicio;  
};
```

```
struct completo{  
    int id;  
    data final;  
    int duracao;  
};
```



Um gabinete de gestão de projetos armazena informação numa estrutura dinâmica. Uma lista ligada de estruturas do tipo *no* armazena informação sobre os gestores (1 nó da lista por gestor). Um gestor é identificado pelo seu id numérico único e a lista está ordenada por este valor. Cada gestor mantém um registo dos seus projetos em curso e projetos terminados:

- Os projetos em curso estão armazenados num vetor dinâmico de estruturas do tipo *struct emCurso*. Cada projeto em curso é identificado pelo seu identificador numérico único e pela data de início do projeto (mês, ano). Na estrutura *struct gestor*, o campo *v1* permite aceder ao vetor de projetos em curso e o campo *totV1* indica a dimensão deste vetor.
- Os projetos terminados estão armazenados num vetor dinâmico de estruturas do tipo *struct completo*. Cada projeto completo é identificado pelo seu identificador numérico único, pela data em que terminou (mês, ano) e pela duração em meses. Na estrutura *struct gestor*, o campo *v2* permite aceder ao vetor de projetos completos e o campo *totV2* indica a dimensão deste vetor.

Os vetores de projetos em curso e os vetores de projetos terminados estão **ordenados pelo identificador de projeto**. Os identificadores são únicos para todo o gabinete de projetos, ou seja, não há repetição de identificador. Na figura em cima exemplifica-se uma situação com 3 gestores. O primeiro gestor, com id2, tem 3 projetos em curso e 4 projetos terminados.

2a) Escreva uma função em C que imprima na consola o estado de um projeto (em curso ou concluído) e o identificador do gestor responsável. A função recebe, como parâmetros, o endereço do início da lista de gestores e o identificador do projeto a pesquisar. Caso o projeto não exista, a função escreve “Inexistente” na consola.

[Cotação: 20%]

2b) Escreva uma função em C que elimine da estrutura dinâmica todos os gestores que não tenham projetos em curso. Todo o espaço retirado da estrutura dinâmica deve ser libertado. A função recebe, como parâmetro, o endereço inicial da lista ligada de gestores. Devolve o endereço inicial da lista depois da atualização.

[Cotação: 40%]

Programação – Exame Especial

18 de julho de 2024 – Duração: 120 minutos

LEI, LEI-PL, LEI-CE

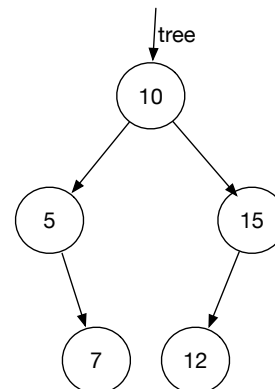
Nome: _____ Número: _____

4. Funções Recursivas

Considere a definição do tipo de dados seguinte, usado para criar uma árvore binária constituída por elementos do tipo *no*:

```
typedef struct elemento no, *pno;  
struct elemento{  
    int valor;  
    pno esq, dir;  
};
```

A função $f()$ recebe como parâmetros um ponteiro para a raiz de uma árvore binária ordenada e um valor inteiro.



```
void f(pno tree, int val){  
    if(tree == NULL || val <= 0)  
        return;  
    else{  
        printf("%d\t", tree->valor);  
        if(val%2 == 0) {  
            f(tree->esq, val / 10);  
            f(tree->dir, val / 10);  
        }  
        else {  
            f(tree->dir, val / 10);  
            f(tree->esq, val / 10);  
        }  
    }  
}
```

Considerando que o ponteiro *tree* aponta para a raiz da árvore ilustrada em cima, qual é o output na consola quando é efetuada a seguinte chamada da função $f()$?

`f(tree, 21);`

Resposta:

a) 10

b) 10 15 5

c) 10 5 15

d) 5 15 10

e) 5 10 15

f) 10 5

[Cotação: 10%] (uma resposta errada não desconta)