Programação - Exame de Recurso

Engª Informática; Engª Informática – Pós-laboral; Engª Informática – Curso Europeu

Duração: 2h30m 10/07/2015

Atenção: É obrigatório apresentar uma estratégia genérica para cada um dos exercícios.

---

1. Uma loja mantém os dados sobre os seus fornecedores guardados em estruturas do tipo *Fornecedor*. Cada estrutura deste tipo contém o nome da empresa e o seu telefone.

```
typedef struct f Fornecedor;
struct f{
      char empresa[100];
      int telefone;
};
```

Estes dados estão guardados em dois ficheiros binários mas, no formato atual, podem existir repetições, *i.e.*, a mesma empresa pode surgir nos 2 ficheiros. Desenvolva uma função em C que elimine a informação redundante dos ficheiros, ou seja, fornecedores que estejam presentes em ambos os ficheiros devem passar a constar em apenas um deles, sendo removidos do outro.

Exemplo:

Antes

Ficheiro A
Limoes Gelados 938463894
Adaptiz 234654786
Astorion 254765871
Vale da pimenta 243619650

Ficheiro B
Astorion 254765871
Arranque 564214671
Limoes Gelados 938463894
Lua de papel 234123123

→

Depois

Ficheiro A
Ficheiro B 938463894
Adaptiz 234654786
Astorion 254765871
Vale da pimenta 243619650

Ficheiro B
Arranque 564214671
Lua de papel 234123123

A remoção de uma empresa que surge em ambos os ficheiros pode ser efectuada em qualquer um deles. No entanto, as empresas que surjam apenas uma vez não devem ser retiradas do ficheiro em que se encontram. A função recebe os nomes dos ficheiros como argumento, devendo ser implementada de modo a suportar qualquer número de fornecedores.

2. Desenvolva uma **função recursiva em C** que encontre o maior valor armazenado numa **árvore binária ordenada de forma crescente**, constituída por elementos do tipo *struct no*. A função recebe um ponteiro para a raiz e devolve o maior valor armazenado nessa árvore.

```
struct no{
      int val;
      struct no *esq, *dir;
};
```

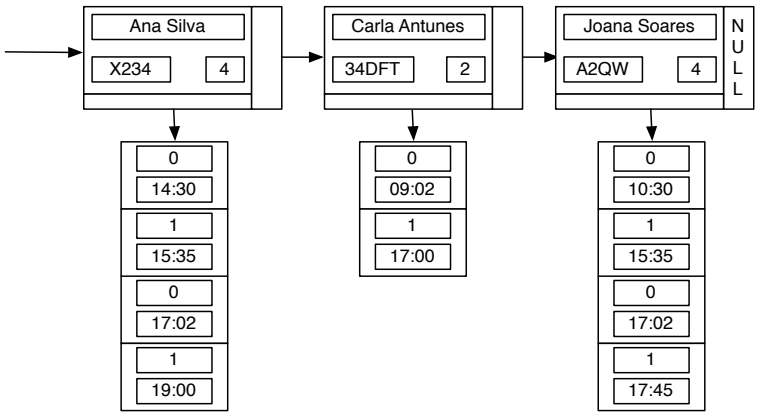
Ao responder a esta questão pode assumir que a árvore recebida não está vazia.

3. Considere as seguintes definições:

```
typedef struct tipoA cliente, *pcliente;
typedef struct tipoB acesso, *pacesso;

struct hora{int h, m;};

struct tipoA{
      char nome[100];          // Nome
      char id[10];             // Identificador alfanumérico
      int total;               // Nº de elementos do vetor de acessos
      pacesso v;               // Ponteiro para o vetor de acessos
      pcliente prox;           // Ponteiro para o próximo cliente
};

struct tipoB{
      int in_out;              // (0:Entrada; 1:Saída)
      struct hora registo;     // Registo da hora do acesso
};
```

Um parque de estacionamento armazena numa estrutura dinâmica informação sobre a sua utilização por parte dos clientes registados. **Esta informação diz respeito a um único dia.**

Existe uma lista ligada constituída por estruturas do tipo *cliente* contendo informação sobre os clientes habituais do parque. **A lista ligada está ordenada alfabeticamente pelo nome do cliente.** Cada nó da lista referencia um vetor dinâmico, através do campo *v*, contendo estruturas do tipo *acesso* onde está armazenada informação sobre os acessos desse cliente ao longo do dia. Cada elemento do vetor regista a hora em que se processou a entrada ou saída do parque.



No exemplo da figura pode ver-se que a cliente Ana Silva, com identificador X234, tem 4 acessos registados: entrou no parque às 14.30, saiu às 15.35, voltou a entrar às 17.02 e saiu às 19.00. Ao responder a esta questão pode assumir que os vetores de acessos estão corretos e completos: têm um número par de elementos, sendo que, para cada par de elementos consecutivos, o primeiro corresponde a uma entrada e o seguinte à respetiva saída (a hora registada no segundo elemento é superior à do anterior).

a) Desenvolva uma função em C que verifique quantos clientes estão no parque a uma determinada hora. A função recebe como argumentos um ponteiro para o início da lista de clientes e uma estrutura do tipo *struct hora* contendo a hora a verificar. Devolve o número de clientes no parque a essa hora.
Exemplo: Considerando a situação ilustrada na figura, às 12.05 estão 2 clientes no parque.

b) Desenvolva uma função em C que adicione uma nova pessoa e respetivos acessos à estrutura dinâmica. A informação a adicionar está armazenada num ficheiro de texto com o formato exemplificado na figura ao lado: a primeira linha tem o nome, a segunda tem o identificador e o número de acessos e as seguintes as horas de acesso (uma hora de acesso por linha, com entradas e saídas consecutivas).
A função deve alocar todo o espaço necessário para o novo cliente e fazer a atualização da estrutura dinâmica, mantendo a ordem alfabética dos clientes. Recebe como argumentos um ponteiro para o início da lista de clientes e o nome do ficheiro de texto. Devolve como resultado um ponteiro para o início da lista atualizada.

```
Elsa Dinis
AS342   4
12:30
13:40
15:00
17:00
```