

Programação - Exame da época normal

Eng^a Informática; Eng^a Informática – Pós-laboral; Eng^a Informática - Curso Europeu

Duração: 2h30m

20/06/2016

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

1. Desenvolva uma função em C que escreva na consola o conteúdo de um ficheiro de texto, colocando em maiúscula a primeira letra de todas as palavras. Todos os outros caracteres do ficheiro deverão permanecer inalterados. O nome do ficheiro é introduzido pelo utilizador.

Exemplo: Considerando que o ficheiro contém o seguinte texto:

Os jovens de agora parece que tem dificuldade em crescer. Nao sei porque. Se calhar as pessoas so crescem ao ritmo a que sao obrigadas.
Cormac McCarthy, in 'Este Pais Nao E para Velhos'

Na consola deverá surgir o seguinte conteúdo:

Os Jovens De Agora Parece Que Tem Dificuldade Em Crescer. Nao Sei Porque. Se Calhar As Pessoas So Crescem Ao Ritmo A Que Sao Obrigadas.
Cormac McCarthy, In 'Este Pais Nao E Para Velhos'

Nota: por questão de simplificação na apresentação dos exemplos, toda a acentuação foi removida

Sugestão: Pode utilizar esta função da biblioteca `<ctype.h>` para o ajudar na resolução

```
int toupper(int c);
```

Esta função recebe uma letra minúscula e devolve-a transformada em maiúscula. Se não receber uma letra minúscula, devolve o resultado inalterado.

2. Considere as seguintes definições:

```
struct pessoa {  
    char nome[50];  
    char telefone[20];  
    struct {  
        int dia, mes, ano;  
    } data_nasc;  
};
```

```
struct xpessoa {  
    char nome[50];  
    char telefone[20];  
    struct {  
        int dia, mes, ano;  
    } data_nasc;  
    char email[50];  
};
```

Um ficheiro binário contém informação sobre os contactos de várias pessoas. A informação existente nesse ficheiro está organizada e gravada usando a estrutura `struct pessoa`.

Pretende-se atualizar essa informação de modo a incluir um novo campo onde possa ser gravado o e-mail principal de cada pessoa, passando a ser representado através da estrutura `struct xpessoa`. Desenvolva uma função em C que receba o nome do ficheiro como argumento e efetue a sua atualização. Por motivos de segurança, a informação original deve ser mantida num ficheiro com extensão `“.bak”`. No ficheiro com as estruturas atualizadas, o novo campo relativo ao e-mail de cada pessoa deve ser iniciado com o valor `“nao definido”`.

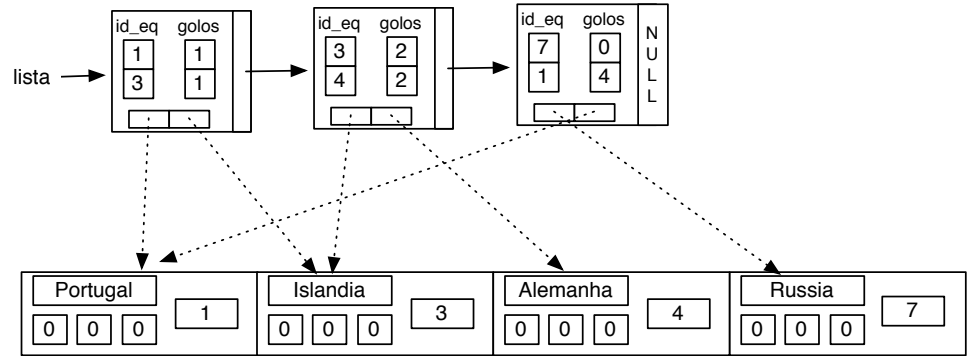
```

typedef struct res no, *pno;
typedef struct d eq,
*pequipa;

struct res{
    int id_eq[2], golos[2];
    equipa pont[2];
    pno prox;
};

struct d{
    char nome[50];
    int id;
    int v, e, d;
};

```



3. Uma estrutura dinâmica armazena informação sobre as equipas que estão a participar no campeonato de futebol Euro-2016 e sobre os resultados que já foram obtidos. A estrutura dinâmica é constituída por uma lista ligada de resultados (nós do tipo *no*) e por um vetor dinâmico de equipas (elementos do tipo *eq*). A figura ilustra um exemplo em que existem 4 equipas no vetor dinâmico e, para cada equipa, é armazenado o seu nome, o seu *id* numérico único e três contadores para o número de vitórias, empates e derrotas obtidas. O vetor dinâmico não está ordenado por nenhum critério.

Por sua vez, cada nó da lista ligada contém informação sobre um jogo: os identificadores numéricos das equipas envolvidas estão no campo *id_eq* e o número de golos marcados por cada equipa estão guardados no campo *golos*. O campo *pont* de cada nó da lista ligada armazena um conjunto de 2 ponteiros que devem referenciar as equipas envolvidas no jogo (no vetor de equipas). No exemplo da figura, o primeiro jogo realizou-se entre as equipas 1 e 3 (Portugal e Islândia) e o resultado foi 1-1.

a) A estrutura dinâmica descrita ainda não está completamente preenchida. Já existe um vetor de dinâmico parcialmente preenchido com o *nome* e *id* das equipas, mas os três contadores estão todos com o valor 0. A lista de resultados também já está completamente formada, com a identificação das equipas envolvidas em cada um dos jogos e o resultado obtido (campos *id_eq* e *golos*), mas os ponteiros do campo *pont* de todos os nós têm o valor NULL. Desenvolva uma função em C que complete a estrutura. Deve realizar as seguintes tarefas:

- Colocar os ponteiros de todos os campos *pont* existentes na lista ligada a apontar para a equipa correta no vetor de estruturas;
- Atualizar os 3 contadores das equipas no vetor dinâmico com os valores corretos, tendo em atenção os resultados armazenados na lista ligada.

A função recebe como argumentos um ponteiro para o início da lista de resultados, um ponteiro para o início do vetor de equipas e o número de elementos do vetor.

b) Considere agora que a estrutura dinâmica já está completamente preenchida e atualizada, mas uma das equipa foi condenada por corrupção. Desenvolva uma função em C que elimine toda a informação relativa a esta equipa da estrutura dinâmica. A sua estrutura deve ser retirada do vetor e os nós da lista que correspondam a jogos em que esta equipa esteja envolvida devem ser eliminados. Devem, além disso, ser atualizados os contadores das outras equipas envolvidas nos jogos que vão ser eliminados. A função tem o seguinte cabeçalho:

```

pequipa atualiza(int id, equipa v, int *total, pno* lista);

```

Recebe como argumentos o identificador único da equipa a ser eliminada, um ponteiro para o início do vetor de equipas, o endereço da variável contendo a dimensão deste vetor e o endereço do ponteiro de lista. Devolve um ponteiro para o início do vetor de equipas atualizado. O número de equipas no vetor e o ponteiro para o início da lista devem ser alterados diretamente através dos argumentos recebidos.