

Programação - Exame de Recurso

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

Duração: 2h30m

07/07/2016

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

1. Os ginastas de um clube desportivo têm o seu cadastro guardado num ficheiro de texto. Cada linha do ficheiro diz respeito a um ginasta e inclui a seguinte informação: ano de nascimento, sigla da área gímnica a que pertencem e nome completo. O ficheiro “ginastica.txt” (ver abaixo) representa um exemplo deste tipo. Desenvolva uma função em C que crie um novo ficheiro de texto com os dados de todos os atletas pertencentes a uma dada área. A função recebe como argumentos o nome do ficheiro original e a sigla pretendida. O ficheiro a criar deve ter como nome a área gímnica e a extensão “.txt”. Por exemplo, “TRI.txt” representa o resultado da aplicação da função sobre o ficheiro exemplificado e para a especialidade “TRI”.

ginastica.txt

2009	ACRO	Maria Rita Goncalves
2005	TRI	Leonard Barata Murys
2006	ACRO	Eliana Portela
2006	TUM	Francisca Ferreira Almeida
2005	TRI	Ana Raquel Silveirinha
2006	ACRO	Sara Janeiro
2008	TUM	Constanca Rocha

TRI.txt

2005	TRI	Leonard Barata Murys
2005	TRI	Ana Raquel Silveirinha

2. Num encontro de natação, a ordem de passagem está organizada usando estruturas do tipo *struct grupo*. Cada grupo passa a uma determinada hora, num dado estilo, e inclui um vetor de atletas. A posição dos atletas dentro do vetor (campo *a*) identifica a pista onde irão nadar, ou seja a 1^a posição do vetor corresponde à pista 1, a 2^a à pista 2, e assim sucessivamente. Cada um dos atletas é identificado pelo seu nome e clube a que pertence. Num ficheiro binário estão armazenadas estruturas do tipo *grupo* identificando as várias provas do torneio de natação. Desenvolva uma função em C que apresente no monitor informação detalhada sobre a participação no torneio dos atletas de uma determinada equipa.

A função recebe como argumentos o nome do ficheiro e a identificação do clube. Deve apresentar na consola, para todos os atletas dessa equipa, o seu nome, a hora a que participam, o estilo e a pista em que vão participar. Por exemplo, se o ficheiro binário tiver o seguinte conteúdo:

9:00 CRAWL				
SFGP Iris Rodrigues	GSC Carlota Pinto	CDA Alexandre Matos	AAC Rita Ferreira	AAC Matilde Pinto
9:40 COSTAS				
AACx Marta Lino	SFGP Bianca Lopo	SFGP Iris Rodrigues	SFG Maria Nunes	GSC Concha Jesus

... e for solicitada informação sobre o clube SFGP, deve surgir na consola

```

Atletas do SFGP:
9:00, CRAWL Iris Rodrigues pista 1
9:40, COSTAS Bianca Lopo pista 2
9:40, COSTAS Iris Rodrigues pista 3
  
```

```

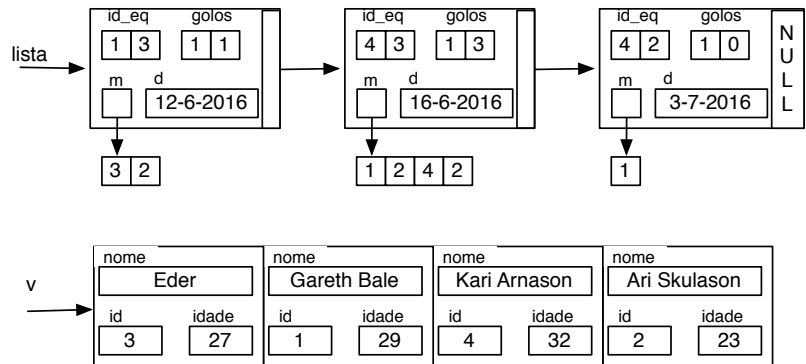
typedef struct res no, *pno;
typedef struct pessoa jog, *pjog;

struct data{int dia, mes, ano;};

struct res{
    int id_eq[2], golos[2], *m;
    struct data d;
    pno prox;};

struct pessoa{
    char nome[50];
    int id;
    int idade;};

```



3. Uma estrutura dinâmica armazena informação sobre jogos realizados no campeonato de futebol Euro-2016 e sobre os marcadores dos golos. A estrutura dinâmica é constituída por uma lista ligada de resultados (nós do tipo *no*) e por um vetor dinâmico de jogadores (elementos do tipo *jog*). A figura ilustra um exemplo em que existem 4 jogadores no vetor dinâmico *v*. Para cada um deles, é armazenado o seu nome, o seu *id* numérico único e a sua idade. O vetor dinâmico não está ordenado por nenhum critério.

Por sua vez, cada nó da lista contém informação sobre um jogo: o campo *d* armazena a data em que realizou, 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 ponteiro *m* de cada nó referencia um vetor dinâmico de inteiros onde estão armazenados os identificadores numéricos dos marcadores dos golos. Num jogo em que a 1ª equipa tenha marcado *X* golos e a 2ª *Y* golos, o vetor tem tamanho *X+Y*: as primeiras *X* posições armazenam os identificadores dos marcadores da 1ª equipa e as restantes guardam a identificação dos marcadores da segunda equipa. **A lista ligada está ordenada por data de realização do jogo.**

No segundo jogo exemplificado na figura, o único golo da equipa 4 foi marcado pelo jogador com *id* 1 e os golos da equipa 3 foram marcados pelos jogadores com *id* 2 (2 golos) e 4.

a) Desenvolva uma função em C que descubra quem é o melhor marcador. A função recebe como argumentos um ponteiro para o início da lista de resultados, um ponteiro para o início do vetor de jogadores e o número de elementos do vetor. Deve escrever na consola o *nome* e o *id* numérico do jogador que marcou mais golos. Em caso de empate, deve identificar o jogador que marcou mais golos e é mais novo. Caso o empate ainda subsista, pode escolher o segundo critério de desempate.

b) Desenvolva uma função em C que adicione informação sobre um novo jogo à estrutura dinâmica. A informação está armazenada em várias linhas de um ficheiro de texto, de acordo com o formato indicado ao lado: *ids* das equipas, data e resultado nas primeiras 3 linhas, seguidos das linhas necessárias para identificar cada um dos marcadores dos golos (*id*, *idade* e *nome*).

```

4 5
2 7 2016
2 1
1 29 Gareth Bale
1 29 Gareth Bale
5 35 Mario Gomez

```

No exemplo pode ver-se que o resultado foi 2-1, sendo necessárias 3 linhas para identificar os marcadores dos golos. Esta função deve criar e inserir ordenadamente na lista um novo nó com os dados do jogo. Caso seja referido algum marcador de golo que ainda não esteja no vetor de jogadores, este deve ser atualizado sendo adicionada uma estrutura com informação sobre o novo jogador. A função tem o seguinte cabeçalho:

```

pjog atualiza(char *txt, pjog v, int *total, pno* lista);

```

Recebe como argumentos o nome do ficheiro de texto, um ponteiro para o início do vetor de jogadores, 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 jogadores atualizado. Caso seja necessário, o número de elementos no vetor e o ponteiro para o início da lista devem ser alterados diretamente através dos argumentos recebidos.