Paradigmas da Programação II
LESI + LMCC (1^o ano)

4º Trabalho Prático

Ano Lectivo de 01/02

O trabalho deve ser realizado em grupo, e deve ser entregue acompanhado dum relatório, por submissão electrónica, até 6ªfeira, 24 de Maio.

1 Objectivos e Organização

Este trabalho prático tem como principais **objectivos**:

- continuação da exploração do computador em ambiente linux, e de algumas ferramentas de apoio à programação
- codificação, na linguagem imperativa C, de estruturas de dados e algoritmos mais complexos usando memória dinâmica, execução e teste dos programas escritos;
- documentação dos projectos de programação realizados usando o processador LaTeX.

Para o efeito, esta folha contém cinco enunciados, dos quais deverá resolver um.
O **relatório** a elaborar, deve ser claro e <u>sucinto</u> e, além do respectivo enunciado e da descrição do problema, deverá conter exemplos de utilização e o código do programa. Como foi dito, o relatório terá de ser escrito em LaTeX sugerindo-se que adoptem o esqueleto proposto que pode ser descarregado a partir da página WWW principal da disciplina, no URL:

`http://www.di.uminho.pt/~prh/curpp201.html`

Não se esqueça de indicar na capa do relatório, e como título, o nome do problema que escolheu; no campo *authors* deve indicar o nome completo e o número de cada elemento activo do grupo.

2 Enunciados

2.1 Árvore de Decisão

Muitos dos formulários que preenchemos podem ser vistos como uma sequência de perguntas. Em certos casos, existem perguntas que só devem ser respondidas por certas pessoas (que escolheram determinada opção numa questão anterior). Desta forma, as questões a responder são diferentes de acordo com as respostas dadas.
Uma árvore de decisão funciona de forma idêntica. É dada uma pergunta inicial que o utilizador tem de responder, e um conjunto de opções (por exemplo, o estado civil e as alternativas: solteiro,

casado, divorciado). Dependento da opção escolhida, o utilizador irá ser direccionado para outras perguntas (por exemplo, nome do conjugue no caso de ser casado).

Outro exemplo de árvores de decisão são as tabelas ou chaves dicotómicas usadas para classificar os mais variados tipos de coisas, como sejam as categorias animais.

Pretende-se que se implemente um programa que permita criar árvores de decisão. O programa deverá permitir o uso da árvore (responder a cada questão) e gravar a árvore em disco para ser usada de uma próxima vez.

Estas árvores devem ser implementadas usando memória dinâmica e devem permitir um número arbitrário de respostas.

2.2 Árvore sim/não

Este problema é idêntico ao anterior, mas mais simplificado. Enquanto que no caso anterior podem existir muitas respostas, numa árvore sim/não as perguntas devem ser feitas de forma a que só existam estas duas respostas.

Os objectivos do programa a implementar são os mesmos do problema anterior.

2.3 Árvore Genealógica

Uma árvore genealógica relaciona pessoas por meio de parentescos. O objectivo deste enunciado é a implementação de um editor de árvores genealógicas, que permita responder a questões como:

- A é descendente de B;
- A e B são irmãos;
- A é tio de B;

Para simplificar, considere que todas as pessoas são fiéis, e que não existem filhos bastardos.

Deverá ser possível guardar a árvore em disco para poder ser continuada num outro dia.

2.4 Aeroporto

Quando vamos viajar e passamos por um aeroporto, existe uma tabela de voos. Esta tabela ordena, por ordem crescente de tempo, os voos que estão prestes a partir. No topo da lista encontra-se o próximo avião a levantar voo.

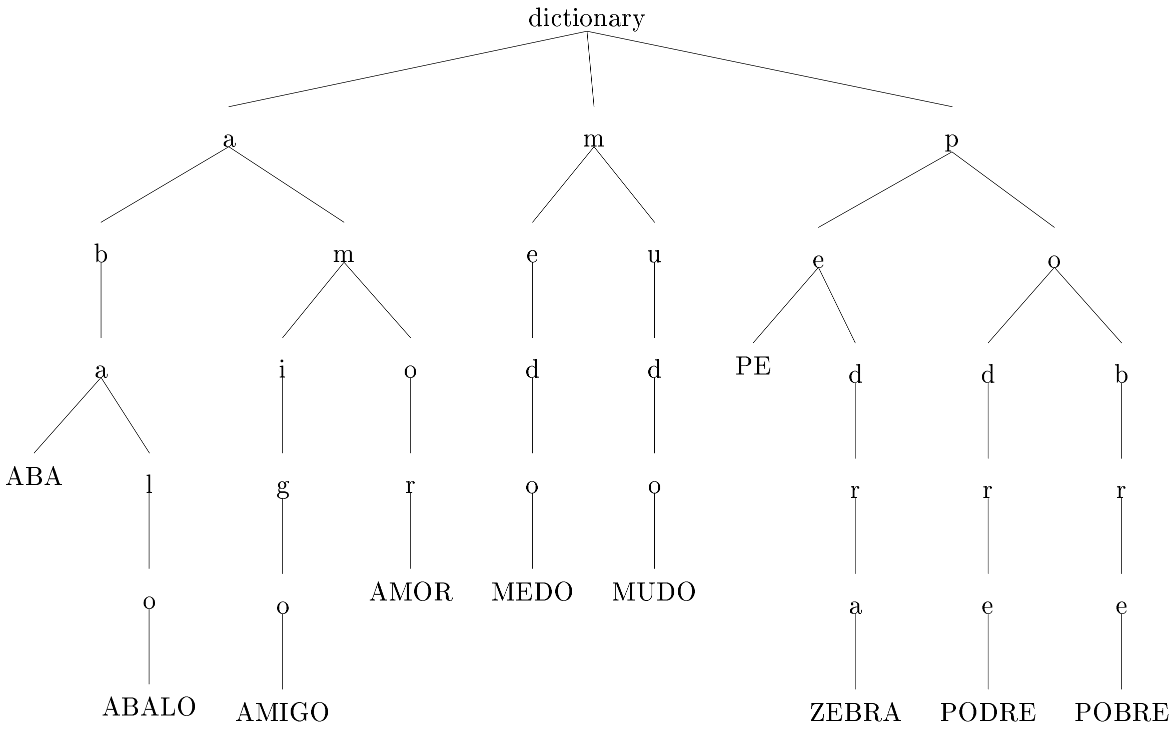
Para controlar esta tabela, pretende-se implementar um programa que por meio de um sistema de menus ou de uma linha de comando permita efectuar as seguintes operações:

- adicionar um voo à lista (no fim da tabela);
- accionar a partida de um avião (retirar da tabela);
- atrasar um avião n voos (mudar a sua posição relativa);
- calcular o número de voos antes de determinado voo levantar;

O programa deverá usar memória dinâmica e permitir gravar a tabela em disco para que se possa voltar a ler num outro dia.

2.5 Dicionário

Pretende-se implementar um sistema de manutenção de um dicionário. Este, terá uma estrutura muito peculiar em memória. Será construida uma árvore em que cada nodo terá apenas uma letra da palavra e cada folha da respectiva árvore terá uma pequena *string* com a definição de cada palavra inserida.



Deve ser possível inserir palavras (palavra e definição) e procurar a definição de determinada palavra. O programa deverá permitir a gravação do dicionário em disco para carregamento posterior.