AULAS 5 E 6 - TIPOS DE DADOS ABSTRATOS

Para compreender a implementação de tipos de dados abstratos na linguagem C utilizando o paradigma modular, deve começar por ler o capítulo 2 do livro *Estruturas de Dados e Algoritmos em C*, analisando a implementação do tipo de dados abstrato COMPLEX (ponto 2.7 − Exemplo de um tipo de dados elemento matemático, páginas 50-57).

O tipo de dados abstrato TIME é constituído pelo ficheiro de interface **time.h** e pelo ficheiro de implementação **time.c** e implementa a criação de informações horárias na forma "*horas***:***minutos***:***segundos*" e de operações sobre elas. Este tipo de dados tem capacidade de múltipla instanciação e usa um controlo centralizado de erros.

Para armazenar as componentes de um tempo é usada uma *struct* (estrutura ou registo), com três campos inteiros para armazenar as horas, minutos e segundos. Comece por ler com atenção os ficheiros indicados e compare esta implementação com a da linguagem Java (ficheiro **time.java**) com a qual deve estar familiarizado.

De seguida compile o módulo, usando para o efeito o comando **gcc -c -Wall time.c**. A compilação deve gerar o ficheiro objeto **time.o**. De seguida compile e teste a aplicação **simtime.c**, que testa as principais operações do módulo. Não se esqueça que para compilar as aplicações, tem que mencionar o ficheiro objeto do tipo de dados (time.o) no comando de compilação. Também é fornecida a *makefile* **mktime**, para compilar o módulo e a aplicação. Teste convenientemente toda a funcionalidade deste tipo de dados. Por exemplo, implementando um programa simples para validar as restantes operações.

- Pretende-se desenvolver o tipo de dados abstrato DATE para processar datas. O tipo de dados deve ter capacidade de múltipla instanciação e controlo centralizado de erros.

A funcionalidade pretendida é especificada pelo ficheiro de interface **date.h**, sendo também fornecido o esqueleto do ficheiro de implementação **date.c**. É ainda fornecida a aplicação **simdate.c** e a *makefile* **mkdate**, para compilar o módulo e a aplicação.

Comece por ler com atenção os ficheiros e complete o módulo. Teste convenientemente toda a funcionalidade do tipo de dados e determine os resultados das seguintes operações:

Considere a data de 31/1/2000. Qual é o dia seguinte?
Considere a data de 22/3/2000. Qual é o dia seguinte?
Considere a data de 28/2/2000. Qual é o dia seguinte?
Considere a data de 29/2/2000. Qual é o dia seguinte?
Considere a data de 31/12/1999. Qual é o dia seguinte?
Considere a data de 29/2/2010. Qual é o dia seguinte?

Considere a data de 1/1/2000. Qual é o dia anterior?
Considere a data de 1/3/2000. Qual é o dia anterior?
Considere a data de 29/2/2000. Qual é o dia anterior?
Considere a data de 31/12/1999. Qual é o dia anterior?
Considere a data de 1/11/2000. Qual é o dia anterior?
Considere a data de 29/2/2010. Qual é o dia anterior?

NOME: **Nº MEC:**

Pretende-se desenvolver um tipo de dados abstrato para armazenar números inteiros, fazendo duas implementações baseadas em duas estruturas de dados diferentes:

- Na implementação estática (SEQ_ARRAY) os números inteiros são armazenados numa sequência (*array*). Como estamos perante uma estrutura de dados estática temos que indicar a sua capacidade de armazenamento, o que é feito na invocação do construtor. E como a sequência pode ficar cheia, também pode ser necessário implementar uma operação para duplicar a sua capacidade de armazenamento. A funcionalidade pretendida é especificada pelo ficheiro de interface **seqarray.h**, sendo também fornecido o esqueleto do ficheiro de implementação **seqarray.c** e a aplicação de simulação **simsequence.c**.

- Na implementação dinâmica (SEQ_LIST) os números inteiros são armazenados numa lista biligada (*double link list*). Como estamos perante uma estrutura de dados dinâmica ela vai crescendo em função das necessidades. A funcionalidade pretendida é especificada pelo ficheiro de interface **seqlist.h**, sendo também fornecido o esqueleto do ficheiro de implementação **seqlist.c** e a aplicação de simulação **simlistseq.c**.