



Algoritmos com Estruturas

Suponha as estruturas data e hora seguintes:

```
struct data {  
    int dia, mes, ano;  
};  
  
struct hora {  
    short int hor, min, seg;  
};
```

- 10.1) Escreva uma função **data pede_data()**; que peça ao utilizador uma data e a retorne. A data deve ser validada (mês entre 1 e 12 e dia entre 1 e o número de dias do ano respetivo).
- 10.2) Escreva uma função **mostra_data(const data da)**; que dada uma determinada data a mostre no ecrã devidamente formatada.
- 10.3) Experimente as funções a cima no programa principal. Experimente sempre também as próximas funções no programa principal com exemplos apropriados.
- 10.4) Escreva uma função **incrementa_data(data &da)**; que incremente uma data para o dia seguinte (com o devido cuidado de verificar se passamos para o mês ou ano seguinte).
- 10.5) Escreva uma função **hora pede_hora()**; que peça ao utilizador uma hora e a retorne. A hora deve ser validada (hora entre 0 e 23, minutos e segundos entre 0 e 59).
- 10.6) Escreva uma função **mostra_hora(const hora ho)**; que dada uma determinada hora a mostre no ecrã devidamente formatada (hora, minutos e segundos).
- 10.7) Escreva uma função **incrementa_hora(hora &ho)**; que incremente uma hora para o segundo seguinte (com o devido cuidado de verificar se passamos para o minuto, hora ou dia seguinte).
- 10.8) Utilizando as funções anteriores crie uma função **relogio(data d, hora hinit)**; que recebe uma data e hora inicial e vai incrementando (a grande velocidade) a hora e data e fazendo o display no ecrã.
- 10.9) Crie uma estrutura **pessoa** que tenha o nome, número do bilhete de identidade, morada e data de nascimento de uma pessoa. Crie funções **pessoa pede_pessoa()**; e **mostra_pessoa(const pessoa pe)**; que permitam introduzir os valores para preencher a estrutura e visualizar no ecrã o seu conteúdo. Utilize a estrutura anterior (data) e respetivas funções para tratar da introdução e visualização da data de nascimento.
- 10.10) Crie no seu trabalho prático (jogo de tabuleiro) uma estrutura designada por movimento ou jogada (ver exemplo no jogo do galo: **struct movimento{ int x, y, jog; };**). Altere o código deo jogo de modo a todo o jogo funcionar com a utilização desta estrutura.