

# Programação

## Universidade da Beira Interior Departamento de Informática 2007/08

1. Implemente um programa em linguagem C que efectue a gestão do valor de penalização pela assiduidade de cada um dos alunos de Programação.

Para tal, utilize o seguinte tipo de dados:

```
typedef struct{
    int numeroAluno;
    float penalização;
}Penalizacao;
```

Por forma a aumentar a segurança da aplicação, pretende-se que toda a informação seja registada em ficheiros binários, procurando-se minimizar a quantidade de informação em memória. Codifique as seguintes funções:

- a. void adicionaFalta(FILE \*f, int numAluno);  
//”f” é uma stream aberta para leitura e escrita,  
//”numAluno” corresponde ao aluno a que se deseja marcar falta.
- b. float devolvePenalizacao(FILE \*f, int numAluno);  
////”f” é uma stream aberta para leitura e escrita,  
//”numAluno” corresponde ao aluno de que se deseja obter informação  
//É devolvido o valor de penalização do aluno, ou “-1” caso o aluno não exista.
- c. void escreveAlunosAssiduos(FILE \*fb);  
//”fb” é uma stream aberta para leitura  
//A função deverá criar o ficheiro de texto “lista.txt” onde aparecem exclusivamente os números dos alunos que não tiveram penalização por assiduidade (1 número por linha).
- d. int contaMaisFaltosos(FILE \*f, float valor);  
//devolve o número total de alunos com penalização superior ao valor passado por parâmetro.
- e. Codifique um programa em linguagem C que utilize de forma conveniente as funções anteriormente implementadas.

2. Suponha que existe um tipo de dados “Multa” usado para registar as coimas e multas de transito passadas pelos elementos das forças de segurança pública.

```
typedef struct{
    int dia, mes, ano;
    int hora, minuto;
}Hora;

typedef struct{
    int codAgente;      //Código do agente altuante.
    Hora hora;         //Hora da infracção
    char nome[80];     //Nome do condutor altuado;
    int codMulta;      //código do tipo de multa, 1=”estacionamento”, 2=”velocidade”,...
    float valorEuros;  //valor monetário da multa
}Multa;
```

Implemente funções que respondam satisfatoriamente às seguintes alíneas:

- a. int mostraMenu();  
//Mostrar no écran um menu com as seguintes opções e receber uma opção válida do utilizador.

```
*****
MENU
*****
1- Registar multa
2- Visualizar cadastro condutor
3- Imprimir cadastro condutor
4- Mostra Agentes
5- Sair
```

- b. void registaMulta(int codA, Hora h, char nm[80], int codM, float val);  
//Adicionar ao ficheiro “multas.dat” a informação passada em parâmetro
- c. void mostraAgentes();  
//Mostra no écran o código de todos os agentes que já efectuaram alguma altuação. Tome em consideração que o código de cada agente apenas deve ser escrito uma vez.
- d. void mostraCadastro(int codCondutor);  
//Mostra no écran todas as multas passadas ao condutor recebido por parâmetro. A informação deve ser procurada no ficheiro “multas.dat”
- e. void imprimeCadastro(int codCondutor, char nomeficheiro[80]);

//Imprime num ficheiro de texto (nomeFicheiro) todas as multas passadas ao condutor recebido por parâmetro. A informação deve ser procurada no ficheiro “multas.dat”

- f. Não lhe parece que as funções implementadas para as alíneas “c” e “d” se sobrepõem? Que alterações sugeria para resolver este problema?
- g. Codifique um programa em linguagem C que utilize de forma conveniente as funções anteriormente implementadas.