

Programação

Universidade da Beira Interior Departamento de Informática 2007/08

1. Crie um programa que, com base no tipo de dados “Hora” abaixo definido, receba do teclado a informação relativa a uma hora e emita uma das seguintes mensagens: “Manhã” (06:00-12:00), “Tarde” (12:01-20:00) ou “Noite” (20:01-5:59). Deverá ainda informar o utilizador se a hora recebida for inválida.

```
typedef struct{
    int hora;
    int minuto;
}Hora;
```

2. Defina um tipo de dados (Data) adequado para registar uma data (composta por um dia, mês e ano). De seguida, implemente funções para:
 - a. Mostrar uma data no écran (formato: dd/mm/aaaa).
Protótipo: void mostraData(Data d);
 - b. Verificar se uma determinada data é ou não válida.
Protótipo: int dataValida(Data d);
 - c. Receber os valores de uma data válida.
Protótipo: Data recebeData();
 - d. Comparar duas datas.
Protótipo: int comparaDatas (Data d1, Data d2);
 - e. Somar dias a uma data.
Protótipo: Data somaDias(Data d, int dias);
 - f. Contar os dias entre duas datas válidas.
Protótipo: int contaDias(Data d1, Data d2);
 - g. Crie um programa onde estejam incluídas as funções anteriormente codificadas: Deverá ser repetidamente mostrado ao utilizador o menu abaixo indicado, bem como implementadas as respectivas funcionalidades

MENU

- 1- Inserir data 1
- 2- Inserir data 2
- 3- Contar dias entre datas
- 4- Comparar datas
- 5- Sair

3. Para guardar informação sobre as notas dos alunos de programação, foi decidido definir as seguintes estruturas de dados.

```
typedef struct{
    int numAluno;
    float notas_freq[3]; //notas das 3 frequências
    float assiduidade; //penalização pela assiduidade
    float notas_exames[2];
} NotaProg;
```

Suponha que toda a informação é registada num vector de 168 elementos (o número de pessoas inscritas à disciplina).

- a. Codifique uma função que, com base nos critérios de avaliação da disciplina (<http://www.di.ubi.pt/~hugomcp/programacao1>), devolva a nota final de um aluno

Protótipo: float notaAluno(NotaProg v[168], int numA);

- b. Crie uma função que devolva “1” ou “0” consoante um aluno tenha ou não obtido aprovação.

Protótipo: int estadoAluno(NotaProg v[168], int numA);

- c. Implemente a função que lista o número dos alunos aprovados:

Protótipo: void mostraAprovados(NotaProg v[168]);

- d. Implemente a função que conta o número de alunos que não foram avaliados, isto é, os que tiveram classificação “-1” em todas as provas:

Protótipo: int contaNaoAvaliados(NotaProg v[168]);

- e. Implemente uma função que mostra os alunos que, apesar de terem sido aprovados por frequência, ainda melhoraram a sua nota num dos exames.

Protótipo: void mostraMelhorias(NotaProg v[168]);

4. Suponha que foram definidos os seguintes tipos de dados para registar informação sobre a assiduidade dos empregados de uma empresa.

```
typedef struct{
    int dia, mes, ano;
}Data;
```

```
typedef struct{
    int hora, minuto;
}Hora;
```

```
typedef struct{
    int codEmpregado; //código do empregado
    int codPorta; //Identificação da porta por onde foi registada a entrada
    Data data;
    Hora hora; //Hora a que o empregado entrou
} Registo;
```

Suponha que toda a informação está registada num vector com 10000 posições completas.

- a. Implemente uma função que conte o total de empregados diferentes registados:

Protótipo: int totalEmpregados(registo v[10000]);

- b. Implemente uma função que conte o número de vezes que um empregado entrou por uma determinada porta.

Protótipo: int totalVezePassaPorta(registo v[10000], int codE, int codP);

- c. Implemente a função que conta o número de portas diferentes por onde um empregado passou:

Protótipo: int totalPortasEmpregado(registo v[10000], int codE);

5. Crie uma função que mostre no écran o código dos empregados que alguma vez entraram depois das 08:00:

Protótipo: void mostraAtrasados(registo v[10000]);

6. Batalha naval. Para registar informação acerca de um tabuleiro da batalha naval, foi pensada a seguinte estrutura de dados:

```
typedef struct{
    int codPeca; //0=água, 1=submarino, ...5=porta aviões
    int estado; //0=não pedido, 1=pedido
} Casa;
```

Casa tabuleiro[10][10]; //1 tabuleiro é uma matriz de casas

- a. Crie uma função que, com base num tabuleiro recebido e numa posição pedida, emita uma das seguintes mensagens: “Água”, “Casa já pedida”, “Tiro no submarino, ...

Protótipo: void recebePedido(Casa tab[10][10], int linha, int coluna);

- b. Implemente uma função que verifica se um tabuleiro já está esgotado, isto é, se já não existem barcos por pedir (1=esgotado, 0=não esgotado):

Protótipo: int tabuleiroEsgotado(Casa tab[10][10]);

- c. Crie a função que mostra em formato matricial o tabuleiro. As casas não pedidas devem conter “?”, as já pedidas e com água devem conter “A” e as com barcos devem conter “X”.

Protótipo: void mostraTabuleiro(Casa tab[10][10]);

- d. Crie um programa que, através de 2 tabuleiros, simule uma partida de batalha naval.