



Frequência 1

1. Considere as seguintes estruturas de dados relativas a uma lista duplamente ligada, utilizada para guardar informação sobre os trabalhadores de uma determinada empresa.

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

typedef struct NODO{
    int BI;           %critério de ordenação (ascendente)
    char nome[80];
    float salario;
    Data dtNascimento;
    struct NODO *nseg;
    struct NODO *nant;
}Nodo;
```

- a. (4 valores) Implemente uma função que verifique se a lista está bem ordenada.

Protótipo: int bemOrdenada(Nodo* L);
//0=Não, 1=Sim

- b. Crie uma função que calcule o valor total de salários dos trabalhadores com determinada idade (especificada em anos).

Protótipo: float totalSalarios(Nodo* L, int anos);

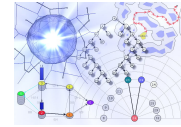
Considere que tem à sua disposição as seguintes funções:

```
int comparaDatas(Data d1, Data d2);
//Devolve "0" caso as datas sejam iguais. "1" caso D1 seja anterior a D2. "-1" caso D2 seja anterior a "D1".
```

```
Data adicionaData(Data D, int dias, int meses, int anos);
//Adiciona "dias", "meses" e "anos" a uma determinada data. Devolve a data actualizada. Aceita valores negativos.
```

```
Data now();
Coloca em "dia", "mês", "ano" os valores respeitantes à data actual.
```

- c. (4 valores) Implemente uma função que mostre no écran o nome completo de todas as pessoas cujo BI acabe num determinado algarismo.



Protótipo: void mostraEcran(Nodo* L, int digitoBI);
//digitoBI= último dígito de BI desejado

- d. (4 valores) Crie uma função que divida a lista de trabalhadores segundo o seu salário. Na primeira lista deverão ficar os 50% com maiores salários. Na segunda lista, deverão ficar os restantes elementos.

Protótipo: Nodo* divideLista(Nodo* L, Nodo **primeira, Nodo **segunda);
//A função deve retornar um apontador para a lista original (vazia)

Considere que tem à sua disposição a seguinte função:

`Nodo* insertLast(Nodo* L, Nodo *nv);`

2. Considere a seguinte estrutura de dados relativa a uma "Fila".

```
typedef struct STACK{  
    int chave;  
    struct NODO *nseg;  
}Stack;
```

- a. (4 valores) Codifique uma função que verifique a igualdade de duas filas. Considere que tem à sua disposição as funções de manuseamento de pilhas/filas. Após a execução da função, as filas não deverão estar alteradas.

```
Stack* push(Stack *F, Stack *nv);    //Push para filas  
Stack* pop(Stack ** PF);            //Pop para filas
```

Protótipo: int filasIguais(Stack* L1, Stack *L2);
//1=Sim, 0=Não