

## Estruturas não homogêneas: Registos (“struct”)

**55** Escreva sub-programas para operar com complexos, declarando *complexo* como uma estrutura do tipo `struct complexo`;

**56** Como se sabe uma matriz quadrada de elementos complexos diz-se hermitica se for igual à sua associada, isto é se:

$$A = A^*$$

em que  $A^*$  é a matriz transposta da matriz conjugada de  $A$ .

1. Considere a seguinte definição:

«Dois números reais dizem-se iguais se a distância entre ambos for inferior a  $10^{-30}$ »

Elabore uma função para verificar a igualdade entre dois números reais de cabeçalho:

```
int iguais(real x, real y);
```

2. Defina o tipo complexo utilizando uma estrutura de `struct complexo`;
3. Considerando as declarações:

```
complexo matriz[20][20];
```

elabore uma função para verificar se uma dada matriz é ou não hermitica, utilizando o conceito de igualdade entre reais definido acima. A função terá como cabeçalho:

```
int hermitica (complexo a[][20], int n);
```

Por razões de eficiência o algoritmo deverá parar logo que encontre um par de elementos correspondentes que não sejam conjugados.

**57** Operações com fracções.

1. Escreva sub-programas para operar com fracções (ler, escrever, simplificar, somar, multiplicar, dividir, subtrair, calcular potências de fracções), declarando as fracções como estruturas do tipo `struct` cujos campos são do tipo `integer`;
2. Elabore um programa para escrever os primeiros  $n$  termos de uma sucessão associada à *série harmónica*:

$$H = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$

sob a forma de fracção. Por exemplo, para  $n = 4$ , a saída do programa deverá ser:

```
1
3/2
11/6
25/12
```

58 Sendo dados os seguintes tipos:

```
typedef struct peca {
    char nome[20];
    int disponivel;
    float precoUnitario;
} Peca;

typedef struct tipo{
    int Codigo;
    peca Peca;
    peca Armazem[ int ];
} Tipo;
```

que descrevem uma armazém de peças. Escreva um sub-programa de facturação que, recebendo os códigos e quantidades das peças encomendadas escreva, em papel, uma factura cujas *linhas de detalhe*, organizadas por coluna, contêm:

1. Para cada peça *encomendada e disponível*, o código, nome, preço unitário, quantidade encomendada e preço total da quantidade encomendada;
2. Para cada peça *encomendada e não disponível*, o código, nome, preço unitário, quantidade encomendada e a mensagem “não disponível” na coluna correspondente ao preço total;

e cuja *linha de total*, posicionada após todas as *linhas de detalhe*, contêm, na coluna correspondente ao preço total, a soma dos preços totais das quantidades encomendadas de todas as peças disponíveis. Declare os tipos e variáveis de que necessitar para elaborar o sub-programa pedido. Indique como invocaria o dito sub-programa.