

Exercício 2.3 São dados os seguintes tipos:

```

type
  Planeta = record Nome          : packed array [1..8] of char;
                Visivel         : boolean;
                RaioOrbital     : real;
            end;
  Sistema = array [1..9] of Planeta;

```

que descrevem o Sistema Solar. Escreva um sub-programa que leia o seguinte ficheiro:

```

Mercúrio s 0.39 Saturno s 9.5
Vénus s 0.72 Úrano n 19.2
Terra s 1.0 Neptuno n 30.1
Marte s 1.5 Plutão n 39.5
Júpiter s 5.2

```

e que escreva o nome e o raio orbital dos planetas visíveis, da Terra, a olho nu.

Exercício 2.4 Suponha as seguintes declarações:

```

type
  data = record dia: 1..31;
            mes: 1..12;
            ano: 1900..2100
        end;
  identidade = record nome          : packed array [1..35] of char;
                data_nas           : data;
                sexo                : 1..2; (* 1=feminino, 2=masculino *)
                estado_civil       : 1..4 (* 1=solt, 2= cas, 3=viu, 4=div *)
            end;

var candidato1, candidato2: identidade;
    dia_casamento: data;

```

1. Faça um sub-programa para saber se um candidato tem 16 anos no dia do previsto casamento.
2. Faça um programa para saber se `candidato1` e `candidato2` podem casar, isto é, se são de sexos opostos, se têm ambos mais de 16 anos no dia do previsto casamento e se não são casados.

Exercício 2.5 Considere os tipos do exercício anterior e:

```

const max = 100;
type lista = array [1..max] of identidade;

```

Escreva um programa que leia um ficheiro no seguinte formato:

```

1 / 2 / 1970 1 2 Fulana
31 / 1 / 1970 2 1 Sicrano
...

```

e escreva (no ecrã) esta lista, ordenada:

1. por ordem alfabética;
2. por idades.

Exercício 2.6 Considere as seguintes declarações:

```
type
  Palavra = packed array [1..30] of char;
  Modalidade = record nome_mod, campeao, nacionalidade: Palavra;
                pontuacao: real;
            end;
var guiness: array [1..1000] of Modalidade;
    atleta, pais, esta_modalidade: Palavra;
    pontos: real;
```

onde o vector `guiness` está ordenado por ordem alfabética do campo `nome_mod`. Em face dos `pontos` obtidos por um atleta do país em `esta_modalidade`, pretende-se actualizar (se necessário) o `guiness` existente.

Elabore um sub-programa para o efeito, tendo em conta que se `esta_modalidade` ainda não faz parte do `guiness`, deve ser introduzida na ordem respectiva.

Exercício 2.7 Sendo dados os seguintes tipos:

```
type
  Codigo = 0..1000;
  Peca = record Nome           : packed array [1..20] of char;
           Disponivel        : boolean;
           PrecoUnitario     : real;
        end;
  Armazem = array [Codigo] of Peca;
```

que descrevem um 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, contém:

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*, contém, 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.

Exercício 2.8 Escreva um programa que resolva uma equação do 2º grau, calcule as suas soluções e as escreva.

SUGESTÃO: Considere os seguintes tipos:

```
type
  numeros = (reais, complexos);
  equacao = record a, b, c: real;
            end;
  complexo = record re, im: real;
            end;
  raizes = record case tipo : numeros of
              reais: (r1, r2: real);
              complexos: (c1, c2: complexo);
            end;
```