

Estruturas (struct)

1. Construa o tipo **struct PONTO** para armazenar um ponto de \mathbf{R}^3 e resolva as questões que a seguir se apresentam.

a) Escreva um subprograma para ler um ponto de \mathbf{R}^3 (inicializar uma variável do tipo PONTO). Duas alternativas:

```
PONTO lerPONTO ();  
void lerPONTO (PONTO*);
```

b) Escreva um subprograma para escrever um ponto de \mathbf{R}^3 (variável do tipo PONTO).

```
void escreverPONTO (PONTO, PONTO);
```

c) Escreva um subprograma para determinar a distância entre dois pontos de \mathbf{R}^3 .

```
float distanciaEntreDoisPontos (PONTO, PONTO);
```

2. Uma revista musical organiza, semanalmente, uma pesquisa de opinião sobre a popularidade de discos. Os resultados da pesquisa estão guardados no ficheiro "discos.txt" (encontra-se na página web da disciplina), com a seguinte informação por linha: nome do autor, nome do disco, discos vendidos. Escrever um programa que leia esta informação para um array de registos e escreva os discos que têm vendas superiores a um certo valor.
3. Usando o tipo **COMPLEXO** abaixo transcrito, escreva um programa que retorne a soma de dois números complexos. Lembre-se que um número complexo é um número que pode ser escrito na forma $\mathbf{a+bi}$, em que a e b são números reais, sendo a designado por parte real e b por parte imaginária dum número complexo.

```
typedef struct {  
    float a; /* parte real */  
    float b; /* parte imaginaria */  
} COMPLEXO;
```

Utilize subprogramas para:

- a) ler um número complexo.
- b) escrever um número complexo.
- c) calcular a soma de dois números complexos.

4. Considere o ficheiro de texto "dados1.txt" (encontra-se na página web da disciplina) que deverá conter uma pauta com a seguinte informação (em 4 linhas seguidas): número do aluno, nome do aluno, classificação da frequência, classificação do exame. Escreva um programa que crie outro ficheiro, contendo o número de aluno e a classificação final (melhor das duas notas). Deve ainda filtrar os alunos com classificação final igual ou superior a 10. Deve usar arrays de estruturas para resolver o problema.

5. Considere o ficheiro de texto "dados2.txt" (encontra-se na página Web da disciplina), que deverá conter informação associadas a várias pessoas com a seguinte informação (em 4 linhas seguidas): número de CC, nome, peso e altura.

Definir um tipo de dados com o nome PESSOA da seguinte forma:

```
typedef struct{  
    int    CC;  
    char nome[100];  
    float peso;  
    int    altura;  
} PESSOA;
```

Utilizando subprogramas implemente um programa que:

- a) passe o conteúdo do ficheiro para um array A;
 - b) solicite ao utilizador uma letra e copie todos os registos cujo campo nome comece por essa letra para um outro array B;
 - c) apresente o campo nome para o registo com o valor de peso mais elevado do array B;
 - d) acrescente um novo registo (campos introduzidos pelo utilizador) ao array A.
6. As notas dos alunos da disciplina de Programação estão guardadas no ficheiro de texto "dados3.txt". Cada linha deste ficheiro contém a seguinte informação: número de aluno e nota final obtida (por esta ordem). Implementar um programa em C para determinar a nota média obtida na disciplina. Para tal, construir e usar os seguintes subprogramas:
- a) um subprograma para ler um array a partir dos dados do ficheiro.
 - b) um subprograma para calcular a soma das notas obtidas pelos alunos (usar o array).
 - c) um subprograma para calcular a média das notas dos alunos aprovados (usar o array).
7. As notas dos alunos da disciplina de Programação estão guardadas no ficheiro de texto "dados4.txt". Cada linha deste ficheiro contém a seguinte informação: número de aluno, nota dos trabalhos práticos (TP) e nota do teste escrito (T) (por esta ordem).

Definir um tipo de dados com o nome ALUNO da seguinte forma:

```
typedef struct{  
    int    numero;  
    int    notaTP;  
    int    notaT;  
    int    notaFinal;  
} ALUNO;
```

Implementar um programa para determinar as notas final de todos os alunos e o número de alunos aprovados na disciplina. Para tal, construir e usar os seguintes subprogramas:

- a) um subprograma para ler um array com os dados do ficheiro;
- b) um subprograma para atualizar o array anterior determinado a nota final;
- c) um subprograma para calcular o número de alunos aprovados (usar o array).