

**DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA**

Electrónica e Computadores

# **PROGRAMAÇÃO II**

**AULAS PRÁTICAS**

**A n o   L e c t i v o   0 2 / 0 3**

*Ana Figueiredo, António Barros, Jorge Morais, Jorge Santos,  
Nuno Morgado, Paulo Matos, Sérgio Moreira*

{ana;abarro;jmorais;jorge.santos;nmorgado;paulo.matos;sjm}@dei.isep.ipp.pt



## Índice

1. Strings.....	1
1.1.1. Enunciado.....	1
1.1.2. Enunciado.....	1
1.1.3. Enunciado.....	1
2. Apontadores.....	2
2.1 Introdução.....	2
2.1.1. Enunciado.....	2
2.1.2. Enunciado.....	2
2.2 Aritmética de apontadores.....	3
2.2.1. Enunciado.....	3
2.2.2. Enunciado.....	3
2.2.3. Enunciado.....	3
2.3 Vectores dinâmicos.....	4
2.3.1. Enunciado.....	4
2.3.2. Enunciado.....	4
2.3.3. Enunciado.....	4
2.3.4. Enunciado 2.3.1.....	4
3. Estruturas.....	4
3.1.1. Enunciado.....	4
3.1.2. Enunciado.....	4
3.1.3. Enunciado.....	5
3.1.4. Enunciado.....	5
3.1.5. Enunciado.....	5
3.1.6. Enunciado.....	5
4. Ficheiros.....	6
4.1 Ficheiros de texto.....	6
4.1.1. Enunciado.....	6
4.1.2. Enunciado.....	6
4.1.3. Enunciado.....	6
4.2 Ficheiros binários.....	6
4.2.1. Enunciado.....	6
4.2.2. Enunciado.....	6
4.2.3. Enunciado.....	7
4.2.4. Enunciado.....	7
4.2.5. Enunciado.....	7
5. Estruturas de dados lineares.....	8
5.1 Pilhas.....	8
5.1.1. Enunciado – pilhas c/ vectores.....	8
5.1.2. Enunciado – pilhas c/ memória dinâmica.....	8
5.2 Filas.....	9
5.2.1. Enunciado – filas em vectores.....	9
5.2.2. Enunciado – filas c/ memória dinâmica.....	9
5.2.3. Enunciado - fila circular c/ memória dinâmica.....	9
5.3 Listas simplesmente ligadas.....	9
5.3.1. Enunciado.....	9
5.3.2. Enunciado.....	9



## 1. Strings

### 1.1.1. Enunciado

- a) Construa uma função que receba como argumento uma string contendo a representação hexadecimal de um número e retorne a representação hexadecimal do número consecutivo.
- b) Construa um programa que, lida a representação hexadecimal de um número, imprime os  $n$  números consecutivos a esse número. Considere que o número máximo de dígitos necessários para representar qualquer dos números é 5.

Utilize a função da alínea anterior.

(Baseado no grupo I do exame de recurso de Algoritmia e Programação II realizado em 5/2/2002)

### 1.1.2. Enunciado

- a) Construa uma função que, dada uma *string*, retorne essa *string* invertida.
- b) Altere a função de modo a que, dada uma string, retorne essa string e a string invertida separadas por um espaço (supõe-se que o vector da string tem espaço suficiente para comportar o resultado, isto é, se a string tem tamanho  $n$ , o vector terá tamanho superior ou igual a  $2n + 2$ ).

### 1.1.3. Enunciado

Escrever um programa que receba um conjunto de parâmetros pela função `main()`. O programa deve analisar cada um dos argumentos da seguinte forma:

- criar uma cópia do argumento, utilizando apenas a memória necessária para esse efeito;
- determinar o número de vogais e consoantes existentes nessa palavra.

O programa deve imprimir uma listagem devidamente formatada das palavra analisadas, contendo a palavra e respectiva quantidade de vogais e consoantes.

## 2. Apontadores

### 2.1 Introdução

#### 2.1.1. Enunciado

Escreva uma função que recebe os apontadores para dois valores inteiros e que transforma o menor desses valores no maior.

#### 2.1.2. Enunciado

Escreva uma função que dados os apontadores para três números,  $x$ ,  $y$  e  $z$ , os ordene de forma a que  $x \leq y \leq z$  (defina e use a função `troca`, que dados dois apontadores para inteiros troca os valores apontados).

## **2.2 Aritmética de apontadores**

### **2.2.1. Enunciado**

Imprimir os valores ASCII associados a cada caracter contido numa string.

### **2.2.2. Enunciado**

Defina uma "string" e use strcpy para copiar uma qualquer palavra para ela. Use apontadores para a imprimir caractere a caractere, da frente para trás e de trás para a frente.

### **2.2.3. Enunciado**

Escreva uma função que dada uma *string*  $s$  e um número  $p$ , retorne o resto da divisão inteira da soma de todos os elementos de  $s$  por  $p$ .

## 2.3 Vectores dinâmicos

### 2.3.1. Enunciado

Ler 20 números para um vector dinâmico e visualizar todos aqueles que são maiores que os seus vizinhos.

(Grupo II da 2ª prova de Programação II realizada em 19/7/99)

### 2.3.2. Enunciado

Ler 20 nomes para um vector dinâmico e listar aqueles que não comecem pela letra A.

(Grupo II da 2ª prova de Programação II realizada em 19/7/99)

### 2.3.3. Enunciado

Elabore um programa em linguagem C que leia do teclado uma sequência de números terminada pela introdução do número 0. Pretende-se saber qual a percentagem de números positivos introduzidos. Utilize um vector dinâmico para armazenar os números.

(Grupo II da 2ª prova de Programação II realizada em 19/7/99)

### 2.3.4. Enunciado 2.3.1

Segundo a conjectura de Collatz, dado qualquer número inteiro  $x$ , a sequência

$$f_{n+1}(x) = \begin{cases} f_n(x)/2 \Leftarrow f_n(x) \text{ par} \\ 3f_n(x)+1 \Leftarrow f_n(x) \text{ impar} \end{cases} \text{ tende para 1.}$$

Escreva uma função que, dado um inteiro  $x$ , construa, apresente e retorne a sequência num vector dinâmico.

## 3. Estruturas

### 3.1.1. Enunciado

Uma empresa de construção civil pretende uma aplicação informática de gestão de recursos humanos. A empresa não prevê ultrapassar os 100 funcionários. Assim, declare uma estrutura capaz de armazenar o número, o nome, a categoria, o vencimento e a data de entrada dos funcionários da empresa

- Escreva uma função para ler os dados dos  $n$  funcionários da empresa.
- Escreva uma função, que para determinada categoria profissional, a perguntar ao utilizador, indique a soma dos vencimentos dos empregados com essa categoria.

(Grupo III da 1ª prova de Programação II realizada em 5/7/99)

### 3.1.2. Enunciado

Pretende-se criar uma aplicação que permita auxiliar na gestão de uma empresa vocacionada para o aluguer de barcos de recreio. Nesse sentido, considerando um vector de estruturas em que cada registo possui os seguintes dados:

- número de barco: inteiro
  - marca: 20 caracteres
  - hora de inicio de aluguer: inteiro
  - disponibilidade (se esta disponível ou não): 1 caracter
- Adicionar informação de um novo barco.
  - Listar todos os barcos disponíveis para alugar.
  - Efectuar o aluguer de um novo barco, actualizando o respectivo registo.

(Grupo III do exame de recurso de Programação II realizado em 11/9/99)

### 3.1.3. Enunciado

Suponha que a Câmara Municipal do Porto decide informatizar os dados respeitantes ao Cadastro Predial (arquivo da documentação referente às propriedades existentes no município), usando para tal um vector. Os dados constantes em cada registo deverão ser:

- Número da matriz
  - Nome do proprietário
  - Número de contribuinte
  - Valor
  - Ano da última alteração
- a) Passando como argumento o ano corrente, escreva uma função que liste todas as propriedades que não tenham sido actualizadas há mais de 10 anos.
- b) Escreva uma função, que recebendo o nome de um dado proprietário, retorne o número de propriedades registadas em seu nome, bem como o valor total.

(Grupo III da 2ª prova de Programação II realizada em 19/7/99)

### 3.1.4. Enunciado

Há várias formas de representar um vector. Duas das formas mais comuns são a representação cartesiana  $(x, y, z)$  e a representação cilíndrica  $(r, \theta, z)$ . Pretende-se escrever um programa que permita converter um vector de uma representação para outra.

- a) Crie dois novos tipos – `vectorcart` e `vectorcil` – baseados em estruturas que representem vectores em coordenadas cartesianas e cilíndricas, respectivamente.
- b) Escreva duas funções para a entrada de um vector em coordenadas cartesianas e em coordenadas cilíndricas.
- c) Escreva duas funções para realizar as conversões cartesiana-cilíndrica e vice-versa. As expressões de conversão são as seguintes:
- $r = \sqrt{x^2 + y^2}$ ;
  - $\theta = \text{atan}(y / x)$ ;
  - $z = z$ ;

### 3.1.5. Enunciado

Pretende-se um programa para a gestão de clientes de uma discoteca. A cada cliente é dado, à entrada, um cartão com um código. De cada vez que o cliente consome algo o código do consumo é associado ao seu cartão. Se o cliente exceder os 10 consumos terá que liquidar a conta e pedir novo cartão. Quando um cliente sai, o programa deverá calcular o preço a pagar e eliminar o seu registo. A lotação do espaço é de 200 clientes e nunca passam por lá mais de 600 numa noite.

1. Defina as estruturas necessárias para a resolução do problema.
2. Crie uma função para inserir clientes.
3. Crie uma função para inserir consumos.
4. Crie uma função para indicar o número de clientes na discoteca.
5. Crie uma função para calcular a despesa de um cliente.

### 3.1.6. Enunciado

Considere que dispõe de um vector de estruturas designado por `tabela classificativa`, que contém informação relativa aos N Clubes que disputam um campeonato de futebol. Este vector está ordenado alfabeticamente em função do nome do clube. Os campos membros da estrutura são: `nome clube`, `total jogos`, `total vitórias`, `total derrotas`, `golos sofridos`, `golos marcados`.

- a) Pretende-se uma função, `f_result`, codificada em C, que tenha como parâmetro de entrada uma string com informação referente a um jogo do campeonato e retorne uma estrutura com quatro campos membro, nome do clube visitado, nº de golos marcados pelo clube visitado, nome do clube visitante, nº de golos marcados pelo clube visitante. A string é da forma “Benfica1 – Porto 2”, podendo ou não haver espaços entre o nome dos clubes e o número de golos, podendo ou não haver o separador -.
- b) Pretende-se uma função, `f_atualiza`, codificada em C, para actualizar a `tabela classificativa`. Esta função tem como parâmetros de entrada a `tabela classificativa` e uma string com informação referente a um jogo do campeonato.

- c) Pretende-se uma função, `ordena_lista`, que mostre a tabela classificativa ordenada decrescentemente em função do total de pontos obtidos. Cada vitória vale 3 pontos e cada empate 1 ponto. Quando há clubes com o mesmo número de pontos, o desempate faz-se através dos critérios aplicados pela ordem seguinte:
- fica primeiro o que tem o menor total jogos;
  - fica primeiro o que tiver maior golos marcados;
  - fica primeiro o que tiver menor golos sofridos;
  - faz-se a ordenação alfabeticamente em função do nome.

## 4. Ficheiros

### 4.1 Ficheiros de texto

#### 4.1.1. Enunciado

Ler 20 nomes e escrevê-los para um ficheiro de nome `nomes.txt`.

#### 4.1.2. Enunciado

Um ficheiro de texto contém informação sobre a ascendência de indivíduos armazenada da seguinte forma:

- cada linha contém um nome;
- cada sequência de 3 linhas representa:
  1. Nome do indivíduo;
  2. Nome do pai;
  3. Nome da mãe

Escreva um programa para ler o ficheiro e armazenar os dados na seguinte estrutura:

- a) Determinar quem são os ascendentes do indivíduo X.
- b) Determinar os nomes dos filhos do indivíduo X.

#### 4.1.3. Enunciado

Escrever um programa que efectue a duplicação de um ficheiro texto. Os nomes dos ficheiros de texto deverão ser enviados por parâmetro para o programa.

### 4.2 Ficheiros binários

#### 4.2.1. Enunciado

Uma das funções da Federação Internacional de Atletismo (F.I.A.) é gerir o *ranking* mundial de prestações alcançadas na especialidade de 100 mts. Dado o elevado nº de provas e atletas, torna-se necessário informatizar esta tarefa. Utilizando um ficheiro de registos estruturados da seguinte forma:

- Nome Atleta (20 caracteres)
- Local Prova (20 caracteres)
- Data Prova (ano,mes,dia)
- Tempo *long*

Elabore um programa em C que permita gerir esse *ranking*, com a seguinte função:

- a) Listagem de todos os registos efectuados num determinado local de prova.

(Grupo IV da 2ª prova de Programação II realizada em 19/7/99)

#### 4.2.2. Enunciado

Considere que um ficheiro binário guarda a seguinte informação sobre cada disciplina do curso Sistemas Eléctricos de Energia:

- Nome da disciplina
- Número de alunos

Elabore um programa que liste as disciplinas cujo número de alunos é superior a 100.

(Grupo IV da 1ª prova de Programação II realizada em 5/7/99)

### 4.2.3. Enunciado

Pretende-se desenvolver um programa que registre num ficheiro binário as leituras realizadas por vários termómetros instalados em vários pontos de um lago.

O nome do ficheiro representa a data em que foram realizadas as medidas, tendo a seguinte forma *aaaammdd.bin*, em que *aaaa* representa o ano, *mm* representa o mês e *dd* o dia.

Cada termómetro envia a seguinte informação ao sistema:

- identificação do sensor (número inteiro positivo);
  - tempo (hora/minutos) da medida;
  - temperatura lida.
- a) Crie um novo tipo baseado em estruturas que permita conter uma mensagem de um termómetro.
  - b) Escreva uma função que escreva uma mensagem no ficheiro. A função deverá receber a mensagem e o apontador para o ficheiro a escrever.
  - c) Escreva uma função que determine a que horas um dado termómetro registou a leitura máxima, para uma determinada data.

### 4.2.4. Enunciado

O ficheiro "artigos.dat" contém os seguintes elementos:

- int código;
- char descrição;
- float preço unitário.

Faça um programa que permita:

- Inserir dados no ficheiro
- Consultar o ficheiro
- Alterar os registos que tenham preço unitário superior à média

Ordenar o ficheiro, copiando o ficheiro inicial para um "backup" (ex, "artigos.bak")

### 4.2.5. Enunciado

Implemente um programa para armazenar e gerir os produtos de um pequeno estabelecimento comercial. Os dados a guardar são:

- Código
- Nome do produto
- Preço de venda sem iva.
- Quantidade em stock.

Deverá guardar a informação num ficheiro binário.

1. Defina as estruturas de dados necessárias ao programa.
2. Crie uma função para inserir um novo produto.
3. Crie uma função para pesquisar um produto e que permita a actualização da sua quantidade em stock.
4. Crie uma função que permita eliminar um registo de produto.
5. Crie uma função para listar todos os produtos cujo stock se encontra abaixo de uma determinada quantidade.

## **5. Estruturas de dados lineares**

### **5.1 Pilhas**

#### **5.1.1. Enunciado – pilhas c/ vectores**

Escreva um programa que inverta uma “string” obtida a partir do teclado, utilizando uma pilha.

#### **5.1.2. Enunciado – pilhas c/ memória dinâmica**

Escreva um programa que permita ler uma sequência de números inteiros (sem permitir repetições) terminada pelo valor sentinela **-1** e os armazene numa stack. O programa deve também permitir eliminar o elemento do topo da stack.

Utilize as rotinas já desenvolvidas para manipulação de uma stack, indicando apenas as alterações necessárias, caso as haja.

## 5.2 Filas

### 5.2.1. Enunciado – filas em vectores

Escreva um programa que permita inserir e eliminar números inteiros numa fila. Deve criar as funções específicas que realizam a inicialização da lista e que verificam se a lista está cheia ou vazia. Utiliza a versão de implementação com vectores.

### 5.2.2. Enunciado – filas c/ memória dinâmica

Escreva um programa que permita inserir e eliminar números inteiros numa fila. Deve criar as funções específicas que o permitam fazer utilizando a implementação de memória dinâmica.

### 5.2.3. Enunciado - fila circular c/ memória dinâmica

Uma lavandaria tem um armazém circular com 20 posições. Cada posição pode estar **vazia** ou ter um **cabide grande** ou um **cabide pequeno**. Cada posição tem também associada o estado da peça respectiva (**pronta/por tratar**) e uma referência (**alfanumérica** com máximo de 20 caracteres). O armazém só roda num sentido e todas as operações são feitas sequencialmente a partir da última posição da operação anterior.

Escreva um programa em C que execute as seguintes tarefas (por ordem):

- reserve memória para guardar todos os dados necessários (inicialmente o armazém está vazio, todas as posições marcadas como “prontas” e com referências numéricas de “1” a “20”).
- liste o estado do armazém (indicando para todas as posições, a referência, o tipo e o estado). Se preferir pode utilizar valores numéricos para indicar o tipo e estado.
- preencha os seguintes dados (a partir da posição actual)
  - vazio
  - vazio
  - “ABC”, Cabide Pequeno, Por Tratar
  - vazio
  - vazio
  - “XPT”, Cabide Pequeno, Por Tratar
  - “KX25”, Cabide Grande, Pronta
- liste o estado do armazém
- procure a primeira posição “Por Tratar” e passe-a a “vazia” e “Pronta”
- liste o estado do armazém

## 5.3 Listas simplesmente ligadas

### 5.3.1. Enunciado

Crie um programa para gerir as entradas e saídas de pessoas de um determinado edifício. De cada vez que uma pessoa entra no edifício, o seu nome é adicionado à lista, bem como a hora e minutos de entrada. Quando uma pessoa sai o seu registo é apagado. A lista deverá estar ordenada por ordem alfabética.

- a) Escreva as estruturas necessárias para este programa.
- b) Crie funções para inserir, listar, eliminar e pesquisar pessoas no edifício.

### 5.3.2. Enunciado

Escreva um programa que implemente um conjunto de rotinas que permitam inserir (sem permitir repetições, no início e no fim), pesquisar, listar, alterar e remover elementos de uma lista simplesmente ligada. Cada elemento da lista deve possuir a seguinte informação:

- produto (30 caracteres)
- preço (inteiro)