

# Programação Imperativa 2021/2022 (CC1003), DCC/FCUP

## Folha 10

Folha baseada em exercícios elaborados pela Professora Ana Paula Tomas  
(CC1003, 2020/2021)

**10.1** Dada uma base de dados com informação sobre alunos e disciplinas de cursos de uma faculdade, pretendemos consultá-la para obter informações. As estruturas de dados que vão ser usadas pelo programa, obedecem à especificação seguinte.

```
#define MAX 11
#define MAXNOME 100

typedef struct aluno {
    char nome[MAXNOME], codigo[MAX];
    int disc[MAX], nd;
} ALUNO;

typedef struct disciplina {
    char nome[MAXNOME];
    int ano;
} DISCIPLINA;

int NDisc = 9;
DISCIPLINA Disc[MAXDISC] = {
    {"Programacao Imperativa",1},
    {"Calculo Infinitesimal I",1},
    {"Programacao Estruturada",1},
    {"Topologia",4},
    {"Geometria Computacional",4},
    {"Probabilidades e Estatistica",2},
    {"Teoria de Numeros",3},
    {"Analise Infinitesimal",3},
    {"Logica",3}
};

int NAlunos = 6;
ALUNO Alunos[MAXALUNOS] = {
    {"Mariana Matias","2001018003", {2,1,9,5,3,7}, 6},
    {"Anabela Moreira", "2000020004",{6,2,3,7,8,9}, 6},
    {"Sonia Silva", "2000020005",{5,6,1,8,9},5},
    {"Joao Maria Peixoto", "2000018001",{3,4,1,5}, 4},
    {"Joao Mario Pereira","2004010005",{8,1,2,3,4,5,6},7},
```

```
{ "Maria Reis", "2000020050", {5, 6, 1, 7, 3, 9}, 6}
};
```

No exemplo, *Programação Imperativa* é a disciplina 1 e *Programação Estruturada* a disciplina 3. As disciplinas a que um aluno está inscrito são identificadas pelos seus números de ordem no array `Disc`. O código de um aluno é uma sequência de 10 dígitos: os quatro primeiros identificam o ano de inscrição, os três seguintes o código do curso, e os três últimos identificam o aluno.

1. Escreva uma função `void procura(char palavranome[])` para imprimir o código e nome dos alunos cujo nome contém uma palavra dada por `palavranome`. Por exemplo, se a palavra fosse "Maria", deveria imprimir:

```
Alunos cujo nome inclui Maria

2001018003 Mariana Matias
2000018001 Joao Maria Peixoto
2000020050 Maria Reis
```

Não deverá ser feita qualquer ordenação dos dados. A menos que a ordenação seja explicitamente solicitada, no output deverá ser mantida a ordem de ocorrência dos dados nas estruturas de dados. Para testar o programa, pode usar os dados que constam do ficheiro `dados.h`. Para isso, coloque a instrução `#include "dados.h"`.

2. Escreva uma função `int codigo2Aluno(char codigo[], char nome[])` que escreve na cadeia de caracteres `nome` o nome de um aluno dando o seu código (a função retorna 0 se não existir, e 1 caso contrário).
3. Escreva uma função `int disciplinas_de_Aluno(char nome[])` que imprime na saída padrão a lista de disciplinas a que um aluno está inscrito, sendo dado o nome do aluno (e retorna o número de disciplinas a que está inscrito).
4. Escreva uma função `int alunos_de_disciplina(char disciplina[], int indices[])` que guarda no vetor `indices` os índices (das posições em `Alunos`) dos alunos que estão inscritos a uma disciplina, dando o nome da disciplina (retorna o número de alunos nessas condições).
5. Escreva uma função `int numero_medio(char curso[])` que retorna o número médio de disciplinas em que os alunos dum dado curso estão inscritos. Esse valor deve ser retornado arredondado por defeito, se necessário.
6. Escreva uma função `int muitas_disciplinas(char curso[], int indices[])` que guarda no vetor `indices` os índices das posições (em `Alunos`) dos alunos dum dado curso que estão inscritos a mais disciplinas do que o número médio a que se encontram inscritos os alunos desse curso e que retorne o número de alunos nessas condições.
7. Escreva uma função `int numero_maximo(char cursoano[])` que retorna o número máximo de disciplinas a que os alunos que entraram num dado curso num dado ano estão inscritos. A sequência de caracteres dada por `cursoano` foi obtida por concatenação do ano de entrada com o código do curso (`dados`), ou seja, se fosse "2006" e "018", `cursoano` apontaria "2006018".
8. Escreva uma função `int numero_disc(int posaluno, int ano)` que retorna o número de disciplinas dum dado ano a que um dado aluno está inscrito. Assume-se que `posaluno` contém o índice da posição do aluno em `Alunos`, sendo uma posição válida.

9. Escreva uma função `int varios_anos(int posaluno)` que verifica se um dado aluno está inscrito a disciplinas de vários anos. A função retorna um valor qualquer diferente de zero em caso afirmativo, e 0 (zero) caso contrário. Assume-se que `posaluno` contém o índice da posição do aluno em `Alunos`, sendo uma posição válida.

**10.2** Considere a seguinte descrição dum programa para implementar um dicionário usando programação modular. Primeiro, o programa lê palavras e as definições correspondentes da entrada padrão e as grava num dicionário. Um utilizador pode depois inserir uma palavra na entrada padrão e o programa procura tal palavra no dicionário. Caso esta palavra seja contida no dicionário, o programa retorna a definição da palavra na saída padrão. O programa continua a funcionar até que o utilizador decida sair.

O programa é constituído pelos seguintes ficheiros: `main.c`, `dict.c`, `dict.h`. O conteúdo destes ficheiros é descrito brevemente a seguir:

`main.c`

```
#include <stdio.h>
#include <stdlib.h>
#include "dict.h"

int main() {
    ...
}
```

`dict.c`

```
#include "dict.h"

/* estrutura de dados para o dicionário */

char * dicionario[1000];

void carrega_dicionario() {
    ...
}

char * procura(char []) {
    ...
}
```

`dict.h`

```
/* estrutura de dados para o dicionário */
char * dicionario[1000];

/* protótipos das funções */
void carrega_dicionario();

char * procura(char []);
```

Responda às seguintes perguntas baseadas na estrutura de programa descrita.

1. Na implementação do programa quer aceder à variável `dicionario` a partir dos ficheiros `main.c` e `dict.c`. Doutro lado, sendo que o ficheiro *header* `dict.h` é incluído em ambos os ficheiros *source*, a variável é declarada em ambos os ficheiros, criando assim ambiguidade. Como pode resolver tal ambiguidade?
2. Assuma que já completou a escrita do programa. Qual é o comando necessário para compilar o programa e gerar o ficheiro executável `dicionario.o`?