



**FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA**

Departamento de Engenharia Eletrotécnica

PROGRAMAÇÃO DE MICROPROCESSADORES

2021 / 2022

Licenciatura em Engenharia Eletrotécnica
e Computadores

1º ano

1º semestre

Trabalho nº 4 Testes, condições e ciclos (Parte 2)

1 Introdução

Nos capítulos 3 e 4 do livro “Linguagem C” de Luís Damas, recomendado para a disciplina de Programação de Microprocessadores, são apresentados respectivamente os testes e condições, e os ciclos. Esta aula, e a anterior, visam consolidar estas matérias através de um conjunto de exercícios. Faça todos os exercícios pedidos em ficheiros separados e **GUARDE O CÓDIGO desenvolvido na memória USB**. Durante a aula o docente pode pedir-lhe para mostrar o código desenvolvido.

2 Sequências e sequências, Lda

Observe o programa seguinte. Ele deveria escrever os números sequencialmente até ao número máximo que o utilizador colocou (que tem de ser menor do que um máximo absoluto), e sempre que esse número é múltiplo de 4, repetir a escrita. Crie um ficheiro chamado *seq.c*, confirme se o programa se comporta desta maneira, e se não se comportar altere-o.

```
/*
 * sequencias e sequencias, Lda
 * Ficheiro: seq.c
 */
#include <stdio.h>
#include <stdlib.h> /* define exit() para sair do programa */

main () {
    int i, n=0;

    printf ("Introduza o limite de elementos pretendido: ");
    if (scanf (" %d", &n) != 1) { /* Se não leu um elemento */
        printf ("Leitura do limite inválida\n");
        exit (1); /* Sai do programa */
    }
    if (n > 200) {
        printf (" Número muito elevado. Tente outra vez\n");
        exit (1);
    }
    for (i = 1; i < n; i++) { /* i tem valores entre 1 e n-1 */
        if (!(i%4))
            printf ("%d ", n);
        printf ("%d\n", n);
    }
}
```

EXERCÍCIO 1: Crie um ficheiro novo com o nome *par.c* e escreva o código de um programa que escreve no ecrã apenas os números pares desde 0 até ao valor que o utilizador definiu, e quando chegar ao fim apresente a soma de todos os números ímpares intermédios. Por exemplo, se o número for 5, apresenta 0, 2, 4, e soma = 9.

EXERCÍCIO 2: Copie o ficheiro anterior para outro com o nome *pot.c* que escreve todos os números desde 1 até ao valor que o utilizador definiu, mas a seguir ao número escreve também o quadrado desse número. Por exemplo:

```
1 1
2 4
3 9
4 16
Etc.
```

EXERCÍCIO 3: O matemático do séc XIII (imaginem) chamado Leonardo Fibonacci, nasceu em Pisa e definiu a seguinte sequência de Fibonacci:

- Os dois primeiros termos são 1
- Cada um dos termos seguintes é a soma dos dois termos anteriores.

Para os alunos universitários do século XXI que ainda não perceberam, aqui vão os primeiros termos:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55,...

Crie um ficheiro chamado ***fibo.c*** com o código para escrever esta sequência até um certo número definido pelo utilizador.

EXERCÍCIO 4: O matemático do séc XIX chamado Edouard Lucas voltou a estudar as sequências (e os trabalhos do Fibonacci) e veio com a sequência de Lucas.

Aqui estão os primeiros elementos da sequência de Lucas

1, 3, 4, 7, 11, 18, 29, 47, 76, 123,...

Crie um ficheiro chamado ***luc.c*** com o código para escrever esta sequência até um certo número definido pelo utilizador

EXERCÍCIO 5: (Esta é mesmo para os bons, para fazer depois do exercício final). Estas sequências parecem brincadeiras, mas a divisão de um elemento pelo anterior em cada uma delas (a partir do quarto elemento) vai dar uma razão que se aproxima (à medida que vamos evoluindo na sequência) do **número de ouro**, ou da **razão de ouro**. Mas isso já você sabe pois é matéria do 8º ano do 3º ciclo. Nada de mais!...

...Quer calcular o **número de ouro** com aproximações sucessivas?

3 Dígitos e algarismos

Imagine que só existem inteiros de três algarismos.

EXERCÍCIO 6: Crie um ficheiro novo com o nome ***dig.c*** que leia um valor introduzido pelo utilizador como inteiro (com o ***scanf*** e ***%d***). Tem de ter três algarismos (se o número que o utilizador introduziu não tem três algarismos, o programa deve ir pedindo um novo até ter três algarismos). O programa deve depois escrever as seguintes três frases em três linhas (para o número 236):

- O algarismo das centenas é o dois;
- O algarismo das dezenas é o três
- O algarismo das unidades é o seis.

O programa tem de ter obrigatoriamente pelo menos um ***switch***. Se executou o programa com três ***switch***, então modifique-o agora para ter apenas um ***switch***.

Um exemplo é o seguinte:

```
Introduza um número com três algarismos: 23
Erro. Não tem três algarismos.
Introduza um número com três algarismos: 418
O algarismo das centenas é o quatro
O algarismo das dezenas é o um
O algarismo das unidades é o oito
Introduza um número com três algarismos:
```

EXERCÍCIO 7: (Para fazer depois do exercício final) Crie um ficheiro novo com o nome *cheque.c* que aproveite o código anterior tanto quanto possível e que a partir de um inteiro inferior a um milhão, e considerando duas casas decimais, escreva por extenso o valor desse inteiro, como se tem de escrever nos cheques. Por exemplo, 1250,34 (mil duzentos e cinquenta euros e trinta e quatro cêntimos).

4 Exercício final

Pretende-se desenvolver um programa para ler valores de tempo no formato “*h(hora) m(minuto) s(segundo)*”, devolvendo o número de segundos associado ao tempo introduzido.

No entanto, o utilizador deve poder omitir campos (e.g. introduzir só “h23”). Para terminar a leitura o utilizador pode introduzir um carácter diferente de ‘h’, ‘m’ ou ‘s’. Pretende-se também que o programa valide os valores introduzidos, não deixando introduzir valores de minutos ou segundos superiores a 59, nem introduzir campos repetidos (só se pode introduzir 0 ou 1 campo de hora; 0 ou 1 campo de minuto; 0 ou 1 campo de segundo). Repare que:

- São necessárias variáveis para guardar cada componente do tempo;
- Só há leitura da parte numérica de uma componente do tempo se o carácter for válido;
- Dependendo do carácter lido, o valor numérico contribui para o valor final de tempo com um peso diferente;
- O ciclo de leitura de componentes do tempo não tem um número de componentes fixo, nem uma ordem fixa.

Quando terminar o exercício, chame o docente e mostre a aplicação a funcionar durante a aula.

Apresenta-se na figura seguinte um exemplo de utilização com a aplicação pretendida:

```
./final2
Introduza o tempo no formato h(hora) m(minuto) s(segundo):
m56 h37 s23 x
Leu 37h 56m 23s = 136583 segundos

./final2
Introduza o tempo no formato h(hora) m(minuto) s(segundo):
h1 x
Leu 1h 0m 0s = 3600 segundos

./final2
Introduza o tempo no formato h(hora) m(minuto) s(segundo):
h25 m70 x
Erro: número de minutos inválidos

./final2
Introduza o tempo no formato h(hora) m(minuto) s(segundo):
h25 h70 x
Erro: campo hora duplicado
```