

Dep.	Informática	Unidade Curricular	Estruturas de Dados			
Curso	CTeSP em Desenvolvimento para a Web e Dispositivos Móveis			Ano	1º	Sem. 2º
Ficha	Introdução à Linguagem Programação C				2024/25	

1. Operadores e Expressões

1.1 Para cada um dos seguintes programas em linguagem C, indique, justificando, qual o resultado da sua execução.

a)

```
#include <stdio.h>
void main(void)
{
    int x;
    x = 3 + 4 * 5 - 6;
    printf("%d\n", x);
    x = 3 * 4 % 5 - 6;
    printf("%d\n", x);
    x = (7 + 6) % 5 / 2;
    printf("%d\n", x);
}
```

b)

```
#include <stdio.h>
void main(void)
{
    int x = 2, y, z;
    x *= 3 + 2;
    printf("%d\n", x);
    x *= y = 4;
    printf("%d\t%d\n", x, y);
    z = (x == y);
    printf("%d\t%d\t%d\n", x, y, z);
}
```

c)

```
#include <stdio.h>
void main(void)
{
    int x, y = 1, z;
    x = 0;
    z = 1;
    x = x && y || z;
    printf("%d\n", x);
    printf("%d\n", x || !y && z);
    printf("%d\n", z >= y && y >= x);
}
```

d)

```
#include <stdio.h>
void main(void)
{
    int x, y, z;
    x = y = z = -1;
    ++x || ++y || ++z;
    printf("%d %d %d\n", x, y, z);
    x = y = z = -1;
    ++x && ++y && ++z;
    printf("%d %d %d\n", x, y, z);
}
```

e)

```
#include <stdio.h>
void main(void)
{
    int i, j, k;
    i = j = k = 1;
    i -= j -= k;
    printf("%d\t%d\t%d\n", i, j, k);
    i = j = 1;
    printf("%d\n", i++ - ++j);
    printf("%d\t%d\n", i, j);
}
```

1.2 Indique o resultado da avaliação das seguintes expressões:

- a) $4 > 8 \ || \ 3 \leq 5 \ \&\& \ 7 \neq 9$
- b) $4 / 3 + 2 * 18 / 4$
- c) $(2 == 1) \ || \ (2 == 2)$
- d) $2 == (1 \ || \ 2) == 2$
- e) $2 == 1 \ || \ 2 == 2$

2. Instruções de Entrada/Saída

2.1 Desenvolva um programa que leia dois inteiros e mostre os seus valores e a sua média:

- a) mostrar a média com formato livre;
- b) mostrar a média com duas casas decimais;
- c) mostrar a média com 8 caracteres e três casas decimais;

2.2 Desenvolva um programa que leia uma data no formato dd/mm/aa e a escreva no formato aammdd.

Exemplo:

Introduza a data: 12/04/98

Data introduzida: 980412

2.3 Indique qual a saída produzida pelo seguinte programa.

```
#include <stdio.h>
void main(void)
{
    int i = 10;
    float f = 4.354;
    printf("%f\n", f);
    printf("%.2f\n", f);
    printf("%.4f\n", f);
    printf("%.0f\n", f);
    printf("%.2e\n", f);
    printf("%d\n", i);
    printf("%4d\n", i);
    printf("%-4d\n", i);
    printf("%4.2d\n", i);
    printf("%.4d\n", i);
}
```

2.4 Considere a seguinte instrução de entrada de dados:

```
scanf("%f%d%f%d", &x, &i, &y, &j);
```

Assumindo que estão declaradas as variáveis inteiras **i, j** e reais **x, y**, qual será o seu valor após a chamada desta instrução se o utilizador escrever a seguinte informação:

12 12.4 35.2 3

E se a instrução de leitura for a seguinte?

```
scanf("%d%d%f%f", &i, &j, &x, &y);
```

3. Programas sequenciais simples

3.1 A nota final de um estudante é calculada a partir do desempenho em 3 aspetos:

1ª frequência: 40%

2ª frequência: 40%

trabalho prático: 20%

Desenvolva um programa para calcular a nota final de um aluno (na escala 0-20) a partir das notas parciais. Assuma que as notas das frequências variam entre 0-20 e a nota do trabalho prático varia entre 0-100.

3.2 Desenvolva um programa que calcule a soma dos algarismos de um número inteiro com três dígitos.

Exemplo: o resultado para o número 846 é 18 (8 + 4 + 6).

4. Instruções de decisão

4.1 Desenvolva um programa que leia um número inteiro e mostra uma das mensagens:

O número ##### é negativo

O número ##### é positivo

O número ##### é zero

No caso do número ser positivo acrescente se é PAR ou ÍMPAR.

Cofinanciado por

4.2 Elabore uma aplicação que solicite a nota de dois testes a um utilizador e realize a média desses dois números. As notas devem estar compreendidas entre 0 e 20.

No final deve informar ao utilizador que:

- b) Se a média for menor de 5, chumbou e não pode ir a exame.
- c) Se a média for superior ou igual a 5 e inferior 7, pode ir a exame.
- d) Se tiver entre 7 (inclusive) e 9.5, poderá realizar defesa para passar
- e) Se tiver mais de 9.5 passou

4.3 Desenvolva o algoritmo de um programa que efetue a leitura de três valores inteiros a partir do teclado e escreva o maior no monitor.

5. Instruções de repetição

5.1 Elabore três versões de uma aplicação que apresente no ecrã a tabua decrescente de um numero solicitado ao utilizador. Utilize o `for`, `while` e `do while`.

5.2 Desenvolva um programa que imprima, em várias linhas, os 25 primeiros múltiplos de um dado número inteiro introduzido pelo utilizador. Cada uma das linhas escritas deve conter 5 múltiplos do número especificado

5.3 Desenvolva um programa que leia uma sequência de ordenados do pessoal de uma empresa e calcule o salário médio.

Implemente duas versões: na primeira o utilizador informa, no início da execução, quantos empregados vão ser considerados. Na segunda versão, a introdução termina quando surgir um ordenado nulo ou negativo.

5.4 Desenvolva um programa que escreva no monitor os múltiplos de 5 não múltiplos de 3, compreendidos entre dois valores limite introduzidos pelo utilizador. O programa deve certificar-se que os limites indicados são positivos e que o limite inferior é realmente menor do que o limite superior.

5.5 Desenvolva um programa que determine os divisores de um número inteiro introduzido pelo utilizador.

5.6 Desenvolva um programa que dado um inteiro positivo calcule a soma dos seus dígitos.

5.7 O dono de um hotel concebeu uma forma original de cobrar aos seus clientes. A primeira noite custa €50. A segunda custa €25 (ou seja, €50/2), a terceira €50/3 e a n-ésima noite custa €50/n. Desenvolva um programa que calcule a cobrança a efetuar a um cliente que fique X noites no hotel. O programa deve indicar o preço a pagar por cada noite e também o total.

6. Funções

6.1 Desenvolva uma função que devolva o quadrado de um número real passado como argumento. Escreva um programa que, utilizando a função definida no ponto anterior, leia números e imprima os seus quadrados até que seja introduzido o valor zero.

6.2 Desenvolva uma função que verifique se um número inteiro está entre dois limites (também inteiros). Os três valores são passados como argumento. A função deve devolver 1, se o número estiver dentro dos limites, ou 0, caso contrário.

6.3 Desenvolva uma função que receba, como argumento, um valor inteiro positivo e devolva o numero de dígitos do valor recebido.

6.4 Desenvolva as funções especificadas nas alíneas seguintes:

- a) Função que devolva o cubo de um número inteiro e positivo passado como argumento;
- b) Função que obtenha do utilizador um número inteiro compreendido entre 100 e 999. O valor obtido deve ser devolvido como resultado final;
- c) Função que verifique se um determinado número inteiro N, recebido como argumento, obedece à seguinte propriedade: N é igual à soma do cubo dos seus algarismos. Um exemplo de um número que satisfaz esta propriedade é o $371 = 3^3 + 7^3 + 1^3$. É garantido que o valor recebido como argumento é um número com três dígitos. A função deve devolver 1 se a propriedade se verificar, ou 0, no caso contrário.
- d) Construa um programa que obtenha do utilizador vários números inteiros pertencentes ao intervalo [100, 999] e verifique quais os que satisfazem a propriedade indicada na alínea c). A introdução de números termina quando o utilizador assim o desejar.

6.5 Desenvolva uma função que leia um conjunto de números inteiros e devolva o número de vezes que o valor máximo surgiu. A dimensão da sequência é passada como argumento.

6.6 Desenvolva uma função que devolva o número de divisores de um valor inteiro e positivo, passado como argumento.

Após isso, escreva um programa que leia uma sequência de números inteiros. Sempre que, nessa sequência, surgir um número primo deve ser escrita no monitor a seguinte mensagem:

"O número que introduziu é primo!".

O programa termina quando for introduzido um número negativo ou nulo.

7. Arrays

7.1 Desenvolva uma função que receba um vetor de inteiros e devolva a média dos elementos do mesmo. A inicialização dos elementos do vetor, deverá ser feita pelo utilizador.

7.2 Desenvolva uma função, que receba dois vetores de inteiros (e respetiva dimensão) e verifique se estes são iguais. A função deve devolver 1 se os dois vetores forem iguais ou devolver 0 se forem diferentes. Considere que dois vetores são iguais se, na mesma posição, tiverem elementos com o mesmo valor.

7.3 Defina uma função que receba como argumentos um vetor de inteiros e a sua dimensão e devolva a soma de todas as ocorrências do maior número do vetor.

Exemplo:

```
Vetor: 1 2 8 8 1 8 devolve : 24
Vetor: 2 2 2 2 2 2 devolve : 12
Vetor: 1 2 3 2 -1 0 devolve : 3
```

Usando a função faça um programa que:

- a) Declare um vetor de inteiros de tamanho **TAM**.
- b) Peça ao utilizador valores inteiros para preencher o vetor.
- c) Chame a função anterior enviando como argumentos o vetor e a sua dimensão.
- d) Imprima o valor devolvido pela função.

7.4 Considere um vetor com N componentes inteiros.

- a) Desenvolva uma função que devolva a posição do maior dos elementos do vetor. No caso de haver repetições do valor no vetor, a função pode, por exemplo, fornecer como resultado a posição com maior índice.
- b) Desenvolva uma função que desloque todos os seus elementos, uma posição para a direita. O último elemento deve deslocar-se para a primeira posição.
- c) Elabore um programa que, após a leitura das N componentes inteiras, faça as necessárias *rotações* para a direita até que o elemento de maior valor do vetor se encontre na última posição desse vetor.

- 7.5 Desenvolva um programa que inicialize um *array* 10X10 com valores aleatórios entre 0 e 10. Após a inicialização, o programa deve contar o número de zeros existentes no *array*.
- 7.6 Desenvolva um programa que inicialize um *array* 10X3 da seguinte forma: em cada uma das linhas, a primeira coluna deve ficar com um inteiro entre 1 e 100 introduzido pelo utilizador, a segunda coluna com o quadrado deste valor e a terceira com o cubo. Após a inicialização, o programa deve contar quantas posições do *array* têm valores superiores a 1000.
- 7.7 Um treinador de atletismo treina 3 atletas e faz 5 sessões de treino por semana. Em cada sessão, cada atleta percorre uma distância que é cronometrada. Os valores dos tempos, em segundos, são registados sob a forma de uma matriz $T(3, 5)$, onde cada linha diz respeito a um atleta e cada coluna a uma sessão de treino.

Escreva um programa para:

- a) Calcular e escrever a média dos tempos realizados em cada sessão de treinos.
 - b) Determinar o melhor tempo realizado por cada um dos atletas nas 5 sessões.
- 7.8 Na matriz $IP(24, 6)$ de elementos inteiros, encontram-se registadas as notas dos 24 alunos de cada uma das 6 turmas de Introdução à Programação. Sabendo que todos os elementos de IP são valores entre 0 e 20, elabore um programa que determine e escreva:
- a) o número de alunos aprovados (nota ≥ 10).
 - b) a melhor nota em cada uma das turmas.
 - c) a turma com maior número de alunos aprovados.

8. Strings

- 8.1 Desenvolva um programa que leia uma frase introduzida pelo utilizador e a escreva invertida.

Exemplo: Hoje e Domingo!

O programa deverá escrever: !ognimoD e ejoH

- 8.2 Desenvolva uma função, cujo protótipo seja `int contaPrimeiro(char[])`, que receba por argumento uma *string* e devolva o número de vezes que o caracter inicial (o primeiro carácter da frase que não seja um espaço em branco) surge ao longo da frase. O programa não deve distinguir entre letras maiúsculas e minúsculas.

Por exemplo: `contaPrimeiro(" Amanha nao e Domingo!");`

Deve devolver o valor: 4

- 8.3 Desenvolva um programa que leia uma frase introduzida pelo utilizador e escreva cada uma das palavras que constituem a frase numa linha separada. Considere que as palavras estão separadas por um ou mais espaços em branco, podendo também existir espaços no início e fim da frase.

Exemplo: Hoje e Domingo!

O programa deverá escrever:

Hoje

e

Domingo!

Dicas: `isspace()` -> devolve se o carater está em branco

- 8.4 Desenvolva um programa que leia uma frase introduzida pelo utilizador e verifique quantas vezes a primeira palavra se repete. O programa não deve distinguir entre letras maiúsculas e minúsculas.

Exemplo: ter ou nao ter

O programa deverá escrever: A palavra *ter* repete-se duas vezes.

9. Estruturas e arrays com estruturas

9.1 Defina o tipo de dados Fração (numerador/denominador) e implemente as operações de soma, subtração, divisão e multiplicação de frações. Os numeradores, denominadores e operação são pedidos ao utilizador.

9.2 Suponha a seguinte informação de **clientes** de uma loja online:

Nome (100 caracteres);
Número de cliente (8 caracteres: constituído por 1 caractere alfa e 7 dígitos);
Email (50 caracteres);
Total de compras realizadas (número real);
Número de itens adquiridos (número inteiro);

Defina um tipo de dados chamado CLIENTE que permita guardar a informação mencionada.

- Declare o array `CLIENTES_LOJA` que permite guardar a informação de um máximo de 100 clientes de uma determinada loja.
- Escreva as instruções para atribuir 5 itens adquiridos e um total de compras realizadas de 1250€ ao 3º cliente da loja.
- Elabore uma função `MenorCliente` que devolva o número de itens adquiridos pelo cliente com menor valor em compras realizadas.
- Elabore uma função de nome `ClienteInf800` que liste no monitor o nome de todos os clientes que já realizaram um total de compras inferior a 800€.
- Preencha o array com dados aleatórios e teste as funções elaboradas nas alíneas anteriores.

9.3 Crie uma variável que seja um vetor em que cada elemento é uma estrutura composta por número, nome e nota do aluno. Elabore um programa que solicite ao utilizador N nomes e respetivos número e nota e depois, dado o número de um aluno, indique o respetivo nome e nota.

9.4 Elabore um programa que permita armazenar em memória a informação relativa a um novo livro que chega a uma biblioteca. Considere que, para cada livro, deve ser guardada a seguinte informação:

- título (máximo 30 caracteres);
- autor (máximo 30 caracteres);
- área (máximo 20 caracteres);
- ano.

O programa deve permitir a inserção de novos livros, bem como listar e retirar livros da lista.

10. Ponteiros

10.1 Implemente uma função que permita trocar o valor de duas variáveis. Teste a função num pequeno programa para o efeito.

10.2 Elabore um programa que leia dois vetores de n elementos cada (inteiros) e determine um terceiro com $2n$ elementos que contenha os elementos do primeiro vetor intercalado com os do segundo vetor, mas estes pela ordem inversa da original. Mostre o vetor final no monitor. Considere $n = 4$ elementos por ex.:

```
vetor x = [1 3 5 7]
=> vetor final = [1 8 3 6 5 4 7 2]
vetor y = [2 4 6 8]
```

10.3 Indique o resultado escrito pelo seguinte bloco de código:

```
int vetor[5] = {44, 55, 66};
printf("%d\n", *vetor+2);
```

10.4 Considere a seção de código à esquerda. Complete à direita o output respetivo, atendendo ao resultado do primeiro `printf` já fornecido. **Nota:** considere que o espaço ocupado por uma variável do tipo `int` é de **4 bytes**.

<pre>int a[7]={-4, 2, -3, 4, -7, 3, 2}; int *p; p = &a[2]; printf("&a = %p\n", a); printf("p-a = %d\n", p-a); printf("p-1 = %p\n", p-1); printf("* (p-2) = %d\n", *(p-2)); printf("2+*(p+1) = %d\n", 2+*(p+1));</pre>	<pre>//Output: &a = 520 p-a = _____ p-1 = _____ * (p-2) = _____ 2+*(p+1) = _____</pre>
---	---

10.5 Elabore um programa que:

- Leia um vetor de n elementos inteiros (n é especificado pelo utilizador)
- Apresente no monitor os elementos do vetor
- Apresente no monitor os elementos do vetor pela ordem inversa.
- Determine o máximo do vetor;
- Determine o mínimo do vetor.
- Determine a soma das componentes
- Determine a média dos elementos do vetor.
- Determine quantos elementos do vetor são superiores à média e quantos são inferiores.
- Duplique para outra variável os valores lidos.

11. Ficheiros

11.1 Elabore um programa que pergunte repetidamente ao utilizador *strings* e as grave num ficheiro. O programa deve terminar a escrita no ficheiro quando a *string* introduzida for o caracter '`.`'.

11.2 Elabore um programa que leia e mostre no monitor as *strings* contidas no ficheiro gravado pelo programa anterior.

11.3 Faça uma função que permita saber o tamanho (número de *bytes*) de um ficheiro. O nome do ficheiro de que se quer saber o tamanho deve ser pedido ao utilizador.

11.4 Complete o programa do exercício 9.4 de forma a permitir:

- armazenar os dados num ficheiro;
- ler os dados do ficheiro;

A função que grava os dados em ficheiro deve armazenar apenas as fichas efetivamente preenchidas.