

UNIVERSIDADE FERNANDO PESSOA

FICHA DE EXERCÍCIOS Nº 1: ALGORITMOS E COMPUTAÇÃO

Nota: para todos os exercícios propostos nesta ficha de exercícios, e sempre que aplicável, apresente a resolução em pseudocódigo, o respetivo fluxograma e efetue o teste do algoritmo obtido usando uma tabela de rastreio e arbitrando valores para a entrada de dados.

- 1) Escreva um algoritmo para encontrar o número do telefone correspondente a um determinado nome, numa lista telefónica.
- 2) Escreva um algoritmo para mudar uma lâmpada fundida de um candeeiro.
- 3) Escreva um algoritmo para mudar um pneu vazio de um automóvel.
- 4) Escreva um algoritmo para calcular o maior de três números inteiros.
- 5) Escreva um algoritmo para calcular o maior e o menor de três números inteiros.
- 6) Escreva um algoritmo para calcular, dados três números inteiros, a soma dos dois maiores.
- 7) Escreva um algoritmo para calcular o máximo divisor comum de dois números inteiros.
- 8) Escreva um algoritmo para calcular o mínimo múltiplo comum de dois números inteiros.
- 9) Escreva um algoritmo para calcular a soma de todos os números inteiros positivos, inferiores ou iguais a 1000.
- 10) Escreva um algoritmo para, dado um dado número inteiro N , calcular a soma de todos os números inteiros positivos pares, inferiores ou iguais a esse número N .
- 11) Escreva um algoritmo em que, dado um número inteiro N , com $1 \leq N \leq 10$, apresente a tabuada dos N . Exemplo: Para $N=5$ deverá apresentar: $1 \times 5=5$, $2 \times 5=10$, $3 \times 5=15$, $4 \times 5=20$, $5 \times 5=25$, $6 \times 5=30$, $7 \times 5=35$, $8 \times 5=40$, $9 \times 5=45$, $10 \times 5=50$
 - a) Utilizando unicamente as instruções mínimas do pseudocódigo;
 - b) Utilizando a instrução: *Repita ... Até ...* ;
 - c) Utilizando a instrução: *Enquanto ... Faça ...* ;
 - d) Utilizando a instrução *Para ... = ... Até ... Faça ...*
- 12) Escreva um algoritmo que calcule a capicua de um número, isto é, inverta a ordem dos algarismos que constituem esse número.
- 13) Escreva um algoritmo para calcular o factorial de um número.
 - a) Utilizando unicamente as instruções mínimas do pseudocódigo;
 - b) Utilizando a instrução: *Repita ... Até ...* ;
 - c) Utilizando a instrução: *Enquanto ... Faça ...* ;
 - d) Utilizando a instrução *Para ... = ... Até ... Faça ...*
- 14) Escreva um algoritmo que, dado um número inteiro, determine se ele é ou não um número primo (um número primo é aquele que só é divisível por 1 e por ele próprio).
- 15) Escreva um algoritmo que leia um número inteiro e determine todos os números primos iguais ou inferiores a esse número.
 - a) Utilizando unicamente as instruções mínimas do pseudocódigo;

- b) Utilizando a instrução: *Enquanto ... Faça ...* ;
- c) Utilizando a instrução *Para ... = ... Até ... Faça ...*

- 16) Escreva um algoritmo que dado um determinado número de pontos (NP) e um determinado número de casas (NC), determine todas as combinações possíveis do tipo Ponto - Casa.
- a) Utilizando unicamente as instruções mínimas do pseudocódigo;
 - b) Utilizando a instrução: *Enquanto ... Faça ...* ;
 - c) Utilizando a instrução *Para ... = ... Até ... Faça ...*
- Exemplo: Nº Pontos = 3 Nº Casas = 2 Saída: 1-1 1-2 2-1 2-2 3-1 3-2*
- 17) Escreva um algoritmo que converta um número decimal para a sua representação em sistema binário.
- 18) Escreva um algoritmo que converta um número decimal para a sua representação numa outra base *B* qualquer.
- 19) Escreva um algoritmo que converta um número representado numa qualquer base, para a sua representação na base decimal.
- 20) Escreva um algoritmo que aceite como entradas um determinado valor a pagar e o dinheiro entregue pelo cliente e calcule todas as notas e moedas (usadas em Portugal) a entregar como troco ao cliente.
- 21) Construa um algoritmo com o objectivo de determinar o valor, ao fim de *n* anos, de um depósito bancário da quantia *q*, sabendo que a taxa de juro inicial *j* (superior a 5.0%), decresce todos os anos de um valor 0.5% até atingir o mínimo 5.0%. Suponha que os juros são sempre capitalizados. Indique para cada ano, o capital inicial, taxa de juro, juros e capital final.
- 22) Escreva um algoritmo que receba, como entrada, uma sequência (um vector) desordenada de números (*num[1]*, *num[2]*, ..., *num[n]*) e verifique se um determinado número *m* se encontra nessa sequência.
- 23) Escreva um algoritmo que receba, como entrada, uma sequência (um vector) desordenada de números (*num[1]*, *num[2]*, ..., *num[n]*) e que a ordene por ordem ascendente.

UNIVERSIDADE FERNANDO PESSOA

FICHA DE EXERCÍCIOS Nº 2: INTRODUÇÃO À PROGRAMAÇÃO EM LINGUAGEM C

Nota: para todos os exercícios propostos nesta ficha de exercícios, implemente e teste o programa num compilador de linguagem C e efectue o teste do programa obtido usando o *debugger*.

- 1) Escreva um programa que leia 3 números e:
 - a) Determine qual deles é o maior.
 - b) Determine o menor e o maior.
 - c) Determine a soma dos dois maiores.
 - d) Verifique se são todos iguais, se existem dois valores iguais e um diferente ou se são todos diferentes.
- 2) Escreva um programa que leia dois valores inteiros x e y correspondentes às coordenadas de um ponto num sistema de coordenadas xy no plano cartesiano, e indique se o ponto se encontra no 1º quadrante ($x>0$ e $y>0$), 2º quadrante ($x<0$ e $y>0$), 3º quadrante ($x<0$ e $y<0$) ou 4º quadrante ($x>0$ e $y<0$).
- 3) Escreva um programa que leia uma nota quantitativa, em percentagem, e indique a nota qualitativa correspondente usando a seguinte fórmula de transformação: 0% a 19% - Muito Insuficiente; 20% a 49% - Insuficiente; 50 a 69% - Suficiente; 70% a 89% - Bom; 90% a 99% - Muito Bom; 100% - Excelente.
- 4) Escreva um programa que leia os comprimentos dos três lados de um triângulo e determine de que tipo de triângulo se trata: rectângulo (tem um ângulo de 90°), obtusângulo (um dos ângulos é $>90^\circ$), acutângulo (três ângulos todos $<90^\circ$) ou "impossível". Exemplo:
Entrada: 4 5 3
Saída O triângulo é rectângulo.
- 5) Escreva um programa que determine o bónus de Natal que uma empresa deverá pagar a um empregado, tendo em conta o número de horas extraordinárias que ele trabalhou durante o ano, segundo a seguinte tabela:

Horas extra.....	Bónus de Natal
> 40	500 euros
] 30..40]	400 euros
] 20..30]	300 euros
] 10..20]	200 euros
] 0..10]	100 euros
<= 0	0 euros

- 6) Escreva um programa para calcular o máximo divisor comum (m.d.c.) de dois números inteiros:
 - a) pela procura exaustiva do máximo divisor comum entre os dois inteiros
 - b) usando a decomposição em fatores primos de cada um deles, sendo o mdc o produto dos fatores primos comuns
 - c) usando o método das divisões sucessivas (Algoritmo de Euclides)
- 7) Escreva um programa para calcular o mínimo múltiplo comum de dois números inteiros.

- 8) Escreva um programa para calcular:
- A soma de todos os números inteiros positivos, inferiores ou iguais a 1000.
 - A soma de todos os números inteiros positivos, pares, inferiores ou iguais a 1000.
- 9) Escreva um programa para, dado um número inteiro N , calcular a soma de todos os números positivos pares, inferiores ou iguais a esse número N .
- 10) Escreva um programa para, dado um número inteiro N , com $1 \leq N \leq 10$, apresente a tabuada dos N . Exemplo: para $N=5$ deverá apresentar: $1 \times 5=5$, $2 \times 5=10$, $3 \times 5=15$, $4 \times 5=20$, $5 \times 5=25$, $6 \times 5=30$, $7 \times 5=35$, $8 \times 5=40$, $9 \times 5=45$, $10 \times 5=50$
- Utilizando a estrutura: *do ... while ...*
 - Utilizando a estrutura: *while ...*
 - Utilizando a estrutura: *for ...*
- 11) Um número inteiro positivo é perfeito se for igual à soma dos seus divisores, exceto ele próprio. Por exemplo, $6 = 1 + 2 + 3$, ou, $28 = 1 + 2 + 4 + 7 + 14$, são perfeitos. Escreva um programa que, para dado um número inteiro positivo N , verifique se ele é perfeito.
- 12) A divisibilidade de um inteiro por 3, 4 ou 9 pode ser facilmente verificada usando as regras descritas a seguir. Desenvolva algoritmos para verificar, para um inteiro N , cada um dos casos usando a respectiva regra:
- Um número inteiro N é divisível por 3 se a soma dos seus algarismos for divisível por 3. Exemplo: 1014 é divisível por 3 porque $1+0+1+4=6$ que é múltiplo de 3. O número 142 já não é ($1+4+2=7$).
 - Um número inteiro N é divisível por 4 se o (sub) número constituído apenas pelas unidades e dezenas for múltiplo de 4. Exemplo: $23516 = 23500 + 16$ é múltiplo de 4 porque 16 é múltiplo de 4.
 - Um inteiro N é divisível por 9 se a soma dos seus algarismos for divisível por 9. Exemplo: 1665 é divisível porque $1+6+6+5 = 18$ é múltiplo de 9. Pode aplicar o algoritmo da divisibilidade por 9, e também por 3, de modo recorrente (recursividade): $1+6+6+5=18$, e $1+8=9$ logo 9 divide 1665.
- 13) Escreva um programa que calcule a capicua de um número (inverta a ordem dos algarismos que constituem esse número).
- Exemplo:* *Número inteiro? 1875*
 Número capicua do número 1875 = 5781
- 14) Escreva um programa que leia um número inteiro e calcule a soma dos seus dígitos.
- Exemplo:* *Número inteiro? 1875*
 Soma dos dígitos do número 1875 = 21
- 15) Escreva um programa que dado um ano, A , e verifique se é bissexto. Um ano diz-se bissexto se for múltiplo de 4 mas não de 100; no entanto os anos múltiplos de 400 são bissextos. Por exemplo, o ano 1900 não foi bissexto mas o ano 2000 já foi.
- 16) Escreva um programa que leia uma data no formato " $A M D$ ", onde A , M e D representam valores inteiros correspondentes ao ano, mês e dia, respectivamente, e determine a data do dia seguinte. Tenha em atenção os anos bissextos.
- 17) Escreva um algoritmo que dado um determinado número de pontos (NP) e um determinado número de casas (NC), determine todas as combinações possíveis do tipo Ponto - Casa.
- Utilizando a estrutura: *do ... while ...*
 - Utilizando a estrutura: *while ...*
 - Utilizando a estrutura: *for ...*

- 18) Escreva um programa que leia um número inteiro e determine se ele é ou não um número primo (um número primo é aquele que só é divisível por 1 e por ele próprio).
- 19) Escreva um programa que leia um número inteiro e determine todos os números primos iguais ou inferiores a esse número.
- 20) Escreva um programa que leia uma sequência de valores positivos, terminada por um número negativo ou nulo, e determine o maior elemento dessa sequência e o seu número de ordem:
- a) Usando um ciclo "while ...";
- b) Usando um ciclo "do...while...".

Exemplo:

Número 1 ? 11

Número 2 ? 14

Número 3 ? 3

...

Número 10 ? -1

O maior elemento da sequência lida é 14. O seu número de ordem é 2.

- 21) Escreva um programa que leia uma sequência de valores positivos, terminada por um número negativo ou nulo, e determine a sua soma e média.
- a) Usando um ciclo "while ...";
- b) Usando um ciclo "do ... while...".
- c) Seria possível utilizar as instruções: *for ...*? Porquê?
- 22) Escreva um programa que calcule o factorial de um número. Recorda-se que o factorial de 0 é igual a 1 e que o factorial de n é igual a $1*2*3* \dots *(n-1)*n$.
- a) Usando um ciclo "while ..."
- b) Usando um ciclo "do ... while ..."
- c) Usando o ciclo "For ..."
- 23) Escreva um programa que, dado um inteiro positivo n , calcule o valor dos somatórios,
- a) $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$
- b) $\frac{1}{2} - \frac{1}{2^2} + \frac{1}{2^3} - \frac{1}{2^4} \dots \frac{1}{2^n}$

Note que cada termo do segundo somatório pode ser obtido a partir do termo anterior multiplicando-o por $-1/2$.

- 24) Escreva um programa que simule o funcionamento de um relógio escrevendo a hora no ecrã no formato HH:MM:SS.
- 25) Escreva um programa que leia um carácter '+', '-', '*' ou '/' que indica uma operação aritmética a efectuar com os dois números que são introduzidos a seguir e apresente o resultado da operação.
- 26) Determinar o valor ao fim de n anos de um depósito bancário da quantia q , sabendo que a taxa de juro inicial j (superior a 5.0%), decresce todos os anos de um valor 0.5% até atingir o mínimo 5.0%. Suponha que os juros são sempre capitalizados. Indique para cada ano, o capital inicial, taxa de juro, juros e capital final.

27) Considere a função $f(x, y, z) = x^2 + 3xy + 2xz - 4yz$ em que x, y, z são valores inteiros. Determine o valor máximo da função para valores de x, y, z num intervalo a especificar pelo utilizador. Apresente o valor máximo e os valores de x, y , e z correspondentes.

28) O custo do selo de uma carta é de 45 cêntimos para cartas que pesem até 50 gramas. As cartas com peso superior pagam um adicional de 45 cêntimos por cada 20 gramas, ou fracção, em que excedam aquele peso.

a) Escreva um programa que dado um número inteiro representando o peso de uma carta, em gramas, determine o custo do selo respectivo.

b) Altere o programa da alínea anterior de modo a que o peso da carta seja representado por uma variável do tipo real.

29) Escreva um programa que calcule os valores da função $f(x) = e^{-x} \cdot \sin(x)$ para valores de x (real) num intervalo e com um incremento no valor de x a especificar pelo utilizador. A apresentação de resultados deve ser semelhante à ilustrada em seguida.

Exemplo:

Limites do intervalo? -2.5 3.5

Incremento de x ? 0.1

x	$f(x)$
-2.50	-7.29
-2.40	-7.45
-2.30	-7.44
...	...
3.50	-0.01

30) Escreva um programa que determine o número de pontos com coordenadas inteiras que existem no interior de uma circunferência de raio r , indicando também as coordenadas de cada ponto.

Exemplo:

Raio da circunferência ? 1.9

Pontos com coordenadas inteiras, pertencentes ao interior da circunferência:

$(-1,-1), (-1,0), \dots, (1,1)$

Número total de pontos = 9

31) Escreva um programa que receba dois números inteiros representando duas horas do dia no formato $hh\ mm$, em que hh indica as horas $[0..23]$ e mm os minutos $[0..59]$ e calcule a soma dos dois valores. Após efectuar este cálculo deverá apresentar o resultado na forma $d\ hh\ mm$, em que d representa o número de dias $[0..1]$.

Exemplo:

Entrada:	15 20
	20 46
Saída:	1 12 06

32) Escreva um programa que calcule o valor do prémio de um seguro automóvel em função dos seguintes parâmetros de entrada:

Preço base do seguro = 200 euros

Idade da viatura, agravamento:

até 5 anos = 50 euros

entre 6 e 10 anos = 100 euros

mais de 10 anos = 150 euros + 10 euros por ano acima de 11

Idade do condutor:

até 25 anos ou mais de 60 anos, agravamento 80 euros

Número de acidentes:

por cada acidente, agravamento = 60 euros

- 33) Escreva um programa que calcule a raiz de ordem n , com n inteiro, do valor de x , com $x > 1$, através de um processo iterativo que converge para a solução. Em cada iteração do seu algoritmo deve usar como estimativa da raiz o valor $r = (min + max)/2$, em que, inicialmente usa $min=1$ e $max=x$. O algoritmo deve terminar ao fim de k iterações ou quando o valor do erro e for menor do que um valor predefinido pelo utilizador.

- 34) Analise o seguinte fragmento de código escrito em linguagem C.

```
1)  int i,j,k;
2)  for (i=0; i<10; i++) {
3)      j=300;
4)      while (j>100) {
5)          printf("*");
6)          k=500;
7)          do {
8)              printf("+");
9)              k+=100;
10)         } while(k<1000);
11)         j-=50;
12)     }
13)     printf("\n");
14) }
```

- a) Calcule o número total de vezes que a instrução da linha 5 é executada.
b) Calcule o número total de vezes que a instrução da linha 8 é executada.
c) Calcule o número total de vezes que a instrução da linha 13 é executada.
d) Quantas linhas serão impressas no ecrã?
e) Quantos caracteres '*' serão impressos em cada linha?
f) Quantos caracteres '+' serão impressos entre cada par consecutivo de caracteres '*'?
- 35) Escreva um programa que calcule o valor de π (pi) usando a série de Leibniz (http://en.wikipedia.org/wiki/Leibniz_series). A fórmula de Leibniz permite calcular facilmente o valor de π a partir do cálculo de $\pi/4$:

$$\sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots = \frac{\pi}{4}$$

- 36) Analise os seguintes fragmentos de código escritos em linguagem C e, para cada um deles, calcule quantas vezes são executadas as instruções dentro dos respectivos ciclos:

- a) for (i=1; i<=20; i++) ...
b) for (i=1; i<=20; i+=2) ...
c) for (i=-20; i<=20; i++) ...
d) for (i=1; i<=10; i++)
 for (j=1; j<=5; j++) ...

- 37) Escreva um programa para verificar que, para o ciclo apresentado no fragmento de código seguinte, e para $linf \leq lsup$ e $step \geq 1$, o número de vezes que o ciclo é executado é dado pela expressão
- $$\left\lceil \frac{lsup - linf}{step} \right\rceil$$

```
int i, linf, lsup, step;
...
for (i=linf; i<lsup; i=i+step) printf("%d ",i);
```

Nota: $[x]$ representa a operação de *ceiling*, ou arredondamento para cima, de x .

- 38) Escreva um programa para calcular o termo a_n numa progressão aritmética, isto é, o termo que está na posição n da lista de termos. Este termo a_n é obtido pela expressão $a_n = a_1 + (n - 1)r$, sendo a_1 o primeiro termo e r a razão ou diferença entre elementos sucessivos da progressão aritmética.
- 39) Escreva um programa para calcular a soma dos n primeiros termos de uma progressão aritmética. A soma vem dada pela expressão $S_n = \frac{n}{2}(a_1 + a_n)$
- 40) Escreva um programa para calcular o termo a_n numa progressão geométrica, isto é, o termo que está na posição n da lista de termos. Este termo a_n é obtido pela expressão $a_n = a_1 \cdot r^{n-1}$, sendo a_1 o primeiro termo e r a razão entre elementos sucessivos da progressão geométrica.
- 41) Escreva um programa para calcular a soma dos n primeiros termos de uma progressão geométrica. A soma vem dada pela expressão $S_n = \frac{a_1(1-r^n)}{1-r}$
- 42) Escreva um programa para calcular o número de combinações possíveis de n elementos de um conjunto em subconjuntos de k elementos cada e que é dado por $C_k^n = \binom{n}{k} = \frac{n!}{k!(n-k)!}$
- 43) Escreva um programa para calcular o número de arranjos possíveis de n elementos, todos distintos, em diferentes subsequências de k elementos cada e que é dado por $A_k^n = \frac{n!}{(n-k)!}$
- 44) Escreva um programa para, dado um número inteiro lido do utilizador, verificar se ele é par e positivo, e se é menor do que 100 ou maior que 1000.

UNIVERSIDADE FERNANDO PESSOA

FICHA DE EXERCÍCIOS Nº 3: EXPRESSÕES, VARIÁVEIS, CONSTANTES E TIPOS DE DADOS EM C

1) Quais das seguintes declarações de variáveis são correctas em C?

- a) `char 1a, 2a, 3a;`
- b) `double num1;`
- c) `char else;`
- d) `int numint;`
- e) `char H20;`
- f) `int meuint%;`
- g) `double primeira vez;`
- h) `float taxa_de_juro;`

2) Converta as seguintes expressões algébricas para a Linguagem C :

- a) $y = \frac{a + b^2}{a - b^2}$
- b) $y = (a + b) \frac{c * b}{d^2}$
- c) $y = \sqrt{\frac{2b^2 + 4b - 3}{4 * \sqrt{b} * b + 3b^2}}$
- d) $y = \frac{2[\text{sen}(x) + 2\cos(x)]}{\text{sen}^2(x)}$

3) Justifique porque é que as seguintes expressões estão incorrectas em C:

- a) `w = x (y + z);`
- b) `res = [(a + b) * (a - b)] / 5;`

4) Considere a seguinte declaração:

`int m1, m2 ;`

Supondo que m1 é igual a 2 e m2 é igual a 4, indique o resultado das seguintes expressões:

- a) `res1 = m1 == m2;`
- b) `res2 = m1 != m2;`
- c) `res3 = m1 <= m2;`
- d) `res4 = m1 >= m2;`
- e) `res5 = m1 < m2;`
- f) `res6 = m1 > m2;`
- g) `res7 = m1>1 && m2<6;`
- h) `res8 = m1>1 || m2<6;`
- i) `res9 = m1>5 && m2<6 || m2<=4;`
- j) `res10 = m1>5 || m2<6 && m2<=4;`
- l) `res11 = !(m1>5) && (m2<6) || (m2==m1);`
- m) `res12 = 0 || (m2>m1) && 1;`

5) Simplifique as seguintes expressões:

- a) `!((x>5) || (y<7))`
- b) `!((x<=6) && (y==10))`
- c) `0 && (x<=1) && (1 || (y>6))`

6) Verifique quais os valores das expressões seguintes tendo como base o seguinte código:

`int a, b, c;`
`float x, y;`

`a = 10; b = -15; c = 7;`
`x = 1.85e10;`

y = 0.2;

...

a) `abs(b - 10) + a % (c - 1)`

b) `a + 103 / ((a - c) * (a - c))`

c) `x / y * 0.342 + x`

d) `x * 10.0 / 5.0 * 2.5 / 2.0 * 3.0`

7) Para cada uma das expressões seguintes, escritas em linguagem C, indique:

i) Quantos operadores possui e de que tipo é cada um deles.

ii) Qual a ordem pela qual esses operadores serão avaliados (considere as regras da associatividade e de precedência de operadores da linguagem C).

iii) Qual o resultado, o valor e o seu tipo, produzido pela aplicação de cada operador (pode utilizar uma árvore para representar a avaliação).

iv) Qual o resultado global de avaliação da expressão e, se for usado no contexto de uma expressão condicional, qual o valor lógico obtido, verdadeiro ou falso.

a) `char c;`

`c = 'A' + 1;`

b) `int a=2,b=10,c;`

`c = a + b * b;`

c) `int a=30,b=20,c=10,d;`

`d = a > b > c;`

d) `int a=10,b,c;`

`b = c = a + 10;`

e) `int a=10,b=5,c=10;`

`a > b && a + 10 < c;`

f) `int a=10,b=5,c=10;`

`a > b || a + 10 < c;`

UNIVERSIDADE FERNANDO PESSOA

FICHA DE EXERCÍCIOS Nº 4: UTILIZAÇÃO DAS INSTRUÇÕES PRINTF e SCANF

- 1) Escreva um programa que peça ao utilizador (utilizando as mensagens “Introduza o número x?”) três números reais e escreva a seu produto, formatado com cinco casas decimais (na forma “O produto é x”).
- 2) Escreva um programa que leia dois números reais e escreva uma tabela com os resultados da sua soma, diferença, produto e quociente, formatados com 8 dígitos significativos, sendo reservados 3 para a parte fraccionária.
- 3) Escreva um programa que leia um carácter sem o escrever no ecrã e depois escreva o seu código ASCII no ecrã.
- 4) Escreva um programa que escreva no ecrã todos os caracteres disponíveis, utilizando a forma: código ASCII ----> carácter correspondente.
- 5) Escreva um programa que peça ao utilizador que entre uma letra minúscula e a converta para maiúscula. Tome providências para garantir que o carácter fornecido pelo utilizador corresponde realmente a uma letra minúscula.

Exemplo:

<i>Entrada:</i>	<i>x</i>	<i>Saída:</i>	<i>X</i>
<i>Entrada:</i>	<i>&</i>	<i>Saída:</i>	<i>Não é uma letra minúscula!</i>

- 6) Escreva um programa que dado um número N, escreva no ecrã uma tabuada para esse número, formatada como se segue (utilizando os caracteres ASCII necessários para a tabela):

Exemplo: Qual a Tabuada? 6

1 x 6	= 6
2 x 6	= 12
3 x 6	= 18
4 x 6	= 24
5 x 6	= 30
6 x 6	= 36
7 x 6	= 42
8 x 6	= 48
9 x 6	= 54
10 x 6	= 60

- 7) Considere o seguinte programa:

```
#include <stdio.h>
```

```
void main() {  
    char ch1, ch2;  
    printf("Introduza um caracter: ");  
    scanf("%c", &ch1);  
    fflush(stdin); // limpa o buffer de teclado  
    printf("Introduza outro caracter: ");  
    scanf("%c", &ch2);  
    printf("Os caracteres introduzidos foram %c, %c\n", ch1, ch2);  
}
```

- a) Corra o programa anterior.
- b) Elimine a linha correspondente à instrução `fflush`. Observe o que acontece ao programa. (Nota: quando se lê informação, numa instrução `scanf`, terminada pelo carácter de <newline>, o buffer de entrada (`stdin`) irá ficar com esse carácter armazenado. Para limpar o buffer usa-se a instrução `fflush(stdin)` após cada instrução de `scanf`. Alternativamente, poderá colocar-se um espaço em branco antes do `%c` do segundo `scanf` de modo a retirar do buffer todos os separadores (`'\n'`, `'\t'` e `' '`). Exemplo: `scanf(" %c", &ch2);`)

UNIVERSIDADE FERNANDO PESSOA

FICHA DE EXERCÍCIOS Nº 5: PROGRAMAÇÃO EM C - FUNÇÕES

- 1) Escreva as seguintes funções:
 - a) Uma função *int volume (int comp, int larg, int alt)*; que dado o comprimento, largura e altura de um paralelepípedo, determine e retorne o seu volume.
 - b) Um procedimento cliente com a definição *void clientVolume()*; que leia do utilizador um comprimento, largura e altura de um paralelepípedo, chame a função anterior para determinar o volume e escreva no ecrã o volume calculado.
- 2) Escreva as seguintes funções:
 - a) Escreva um procedimento com a seguinte definição: *void troca (int *x, int *y)*; que dados dois números inteiros, troque os seus valores. Escreva uma função cliente para testar a anterior.
 - b) Construa um programa para efetuar a ordenação de vectores pelo método “Bubble Sort” que utilize a função *troca()* anterior para efetuar as trocas de valores.
- 3) Escreva um programa que leia duas matrizes quadradas, faça o seu produto e escreva o resultado no ecrã. Deverão ser escritos três procedimentos: Leitura, Produto e Escrita. O programa deve poder funcionar com matrizes de qualquer dimensão. Amplie o programa incluindo outras operações de cálculo matricial (Soma, transposição, cálculo do determinante, valores próprios, vectores próprios, etc).
- 4) Escreva um programa que possa ler vectores, que os ordene por ordem crescente e que inclua opções no programa para efectuar a intersecção ou reunião dos vectores (conforme opção do utilizador) e escrever o vector resultante.

O Programa deverá conter os seguintes procedimentos (n indica o número de elementos):

 - Le_vector (int v[], int n);*
 - Ordena_vector (int v[], int n);*
 - Escreve_vector (int v[], int n);*
 - Reuniao (int v1[], int v2[], int v3[]);*
 - Intersecção (int v1[], int v2[], int v3[]);*
- 5) Escreva procedimentos que lhe permitam ordenar vectores utilizando os métodos estudados (“por selecção”, “inserção”, “bubble-sort”, “partição e troca” e “fusão”). Construa um programa de aplicação desses procedimentos.
- 6) Escreva um subprograma que determine o máximo de três números que lhe são passados como parâmetros e que retorne aquele valor ao programa principal. Resolva este problema recorrendo a:
 - a) Um procedimento: *void maior3(int a, int b, int c, int * maior);*
 - b) Uma função: *int maior3(int a, int b, int c);*
- 7) Escreva um subprograma que determine o máximo e o mínimo de três números que lhe são passados como parâmetros e que retorne aquele valor. Qual o tipo de subprograma que lhe parece mais adequado para resolver este problema ? Justifique a sua resposta.
- 8) Sabendo que $a^b = e^{b \ln(a)}$, escreva uma função que dados *a* e *b*, calcule a^b . Escreva um programa de teste desta função.
- 9) Escreva funções que lhe permitam pesquisar vectores segundo os dois métodos estudados (“pesquisa sequencial” e “pesquisa binária”). As funções devem fornecer como resultado a posição

do elemento pesquisado (caso este exista no vector) ou um valor nulo caso o elemento a pesquisar não se encontre no vector.

- 10) Escreva uma função que verifique se uma determinada data fornecida na forma *DD/MM/AA* é ou não correcta.
- 11) Escreva uma função que dada uma determinada data fornecida na forma *DD/MM/AA*, calcule o dia seguinte ao fornecido.
- 12) Escreva uma função que dada uma determinada data (na forma *DD/MM/AA*) e um determinado número de dias (*DD*), retorne qual a data correspondente à soma entre esses dois valores.
- 13) Escreva uma função que dadas duas datas (na forma *DD/MM/AA*), retorne o número de dias de intervalo entre essas datas.
- 14) Escreva uma função que receba uma string representando um número romano e retorne o valor decimal correspondente.
- 15) Escreva um procedimento que leia três valores booleanos representados pela introdução de 0 para o valor *False*, e 1 para o valor *True*, e que utilize estes valores para calcular o valor da expressão booleana *x AND (y OR z)* para qualquer conjunto de valores das variáveis *x*, *y* e *z*, indicado pelo utilizador, através do teclado
- 16) Escreva as versões recursiva e não recursiva da função que calcula o factorial de um número.
- 17) Escreva as versões recursiva e não recursiva da função que calcula x^n (pelo método tradicional, ou seja, multiplicando *x*, *n* vezes) sendo *x* e *n* (inteiros) parâmetros da função.
- 18) Escreva as versões recursiva e não recursiva de uma função que receba um número inteiro e inverta a ordem dos algarismos desse número.

Exemplo:

Entrada: 14863

Resultado: 36841

- 19)
 - a) Escreva um subprograma que calcule o valor da função $f(x)=x^2-5x+3$ para um valor de *x* (real) dado. Execute também um programa que teste este subprograma.
 - b) Escreva um procedimento que, usando o subprograma da alínea anterior, determine os valores máximo e mínimo de $f(x)$ num intervalo cujos limites *l*he são passados como parâmetros, com uma precisão de 0.1 no valor de *x*.
- 20)
 - a) Escreva um subprograma que calcule a tangente de um ângulo. Admita que o argumento passado para o subprograma nunca conduz a valores infinitos para a tangente. Escreva um programa que teste este subprograma. O ângulo poderá estar em graus ou radianos consoante o valor de uma variável global *Ang*, do tipo *Angulo=(Graus, Radianos)*.
 - b) Altere o programa desenvolvido na alínea anterior de modo a que as unidades em que o ângulo está expresso constituam um dos parâmetros do subprograma.
 - c) Altere o subprograma da alínea anterior de modo a contemplar as situações em que a tangente é infinita. Para isso, o subprograma deve retornar um segundo valor que indique a ocorrência ou não desta situação.
- 21) Escreva procedimentos para efectuar a adição, subtracção, multiplicação e divisão de dois números complexos, integrados num programa de teste. Escreva também procedimentos para fazer a leitura

e a escrita de números complexos. Na leitura deverão ser lidas sempre a parte real e a parte imaginária do número. Os números com parte imaginária não nula deverão ser escritos na forma " $a \pm b j$ ".

Exemplo:

Numero complexo A (xxx.xxx yyy.yyy)? 1 2

Numero complexo B (xxx.xxx yyy.yyy)? -3 2

$$A+B = -2.000 +4.000 j$$

$$A-B = 4.000$$

$$A*B = -7.000 -4.000 j$$

$$A/B = 0.077 -0.615 j$$

- 22)** Pretende-se simular o funcionamento de uma máquina automática de venda de mercadorias. A máquina deve, em função de uma quantia entregue por um comprador para pagar a mercadoria por ela fornecida e de uma quantia a pagar, fornecer o número mínimo de moedas que perfazem o troco. Considere que a máquina aceita e fornece moedas de 2 e 1 euros e de 50, 20, 10 e 5 cêntimos. Escreva um programa que simule o funcionamento da máquina. O programa deve:

- apresentar repetidamente um menu de produtos, com indicação do preço respectivo e pedir ao utilizador que seleccione um produto;
- pedir ao utilizador que introduza moedas que perfaçam um montante igual ou superior ao preço do produto e aceitar as moedas que o utilizador vai introduzindo sucessivamente, até que o total o preço do produto seja atingido ou ultrapassado;
- calcular o troco e determinar o número de moedas de cada tipo a fornecer.

O programa deverá ser constituído à custa de pelo menos quatro subprogramas, implementando as seguintes operações:

- apresentação do menu, retornando o preço do produto seleccionado;
- aceitação das moedas, uma a uma, retornando a quantia total introduzida;
- determinação do número de moedas de cada tipo que constituem o troco, calculado previamente, se a venda for possível;
- "fornecimento" das moedas ao utilizador.

UNIVERSIDADE FERNANDO PESSOA

FICHA DE EXERCÍCIOS Nº 6: VARIÁVEIS INDEXADAS – VECTORES (ARRAYS) E MATRIZES

- 1) Escreva um programa para ler uma sequência de valores reais, positivos, inferiores ou iguais a 100 (percentagens obtidas por um conjunto de alunos num exame), terminados por um valor negativo, e guardar a sequência num vector. Em seguida deve determinar a média e o desvio padrão dos valores lidos e quantos valores estão compreendidos nos intervalos $[0..10[$, $[10..20[$, ..., $[90..100]$. Admita que o número de alunos não é superior a 500. Construa em seguida um histograma para os intervalos considerados.
- 2) Escreva um programa que leia 10 caracteres para um vector e os escreva depois, no ecrã, em ordem inversa.
- 3) Escreva um programa para gerar uma aposta no totoloto, constituída por 6 números inteiros, não repetidos, pertencentes ao intervalo $[1..49]$. (Sugestões: utilize a função de geração de números inteiros aleatórios (*rand()*, precedida de *srand(int seed)*).
- 4) Altere o programa anterior de forma a evitar que surjam números repetidos na chave fornecida. Para tal utilize um vector de Booleanos para manter informação sobre os números que vão sendo gerados.
- 5) Considere um conjunto de pontos de coordenadas (x_i, y_i) . Escreva um programa que determine os dois pontos mais distantes, utilizando:
 - a) Dois vectores lineares, $x[]$ e $y[]$, para guardar separadamente as coordenadas em cada um dos eixos.
 - b) Um único vector bidimensional, *Coordenadas[]*.
(a distância euclidiana entre dois pontos de coordenadas (x_i, y_i) e (x_j, y_j) é dada por $d = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$)
- 6) Escreva um programa que preencha um vector com uma sequência de números positivos terminada por zero e que seguidamente elimine todos os valores repetidos.
 - a) Suponha que a sequência de valores não repetidos é guardada num outro vector.
 - b) Suponha que a sequência de valores não repetidos é guardada no mesmo vector.
- 7) Escreva um programa que leia dois vectores e efectue a sua fusão eliminando todos os elementos que estejam repetidos.
- 8) Escreva um programa que leia uma matriz de 5x5, representando as altitudes de uma determinada região e determine as coordenadas do(s) ponto(s) mais alto(s). A leitura dos elementos da matriz deve ser feita de tal modo que a posição do écran onde são lidos os elementos corresponda à posição desses elementos na matriz.
- 9) Construa uma função que calcule a soma dos elementos da diagonal principal de uma matriz quadrada $n \times n$
- 10) Escreva um programa que permita efectuar a leitura de um vector e em seguida permita efectuar a sua ordenação por ordem decrescente ou crescente (selecção efectuada pelo utilizador).
- 11) Considere um conjunto de n pontos, de coordenadas (x_i, y_i) . Escreva um programa que determine os parâmetros da recta de regressão (recta que melhor se ajusta ao conjunto de pontos), $y(x)$, de acordo com as fórmulas seguintes correspondentes a um modelo de regressão simples:

$$y = a_0 + a_1 x$$

$$a_1 = \frac{n \sum_{i=0}^{n-1} (x_i y_i) - \sum_{i=0}^{n-1} x_i \sum_{i=0}^{n-1} y_i}{n \sum_{i=0}^{n-1} (x_i^2) - (\sum_{i=0}^{n-1} x_i)^2} = \frac{\overline{xy} - \bar{x}\bar{y}}{\overline{x^2} - \bar{x}^2} = \frac{Cov[x, y]}{Var[x]}$$

$$a_0 = \frac{\sum_{i=0}^{n-1} y_i - a_1 \sum_{i=0}^{n-1} x_i}{n} = \bar{y} - a_1 \bar{x}$$

Calcule também o coeficiente de correlação amostral, que indica a qualidade do ajuste da recta de regressão ao conjunto de pontos, o qual é dado por:

$$r_{xy} = \frac{n \sum_{i=0}^{n-1} (x_i y_i) - \sum_{i=0}^{n-1} x_i \sum_{i=0}^{n-1} y_i}{\sqrt{n \sum_{i=0}^{n-1} (x_i^2) - (\sum_{i=0}^{n-1} x_i)^2} \sqrt{n \sum_{i=0}^{n-1} (y_i^2) - (\sum_{i=0}^{n-1} y_i)^2}} = \frac{\sum_{i=0}^{n-1} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=0}^{n-1} (x_i - \bar{x})^2 \sum_{i=0}^{n-1} (y_i - \bar{y})^2}}$$

- 12) Considere a seguinte declaração de um programa em C:

```
typedef int tabnotas [25][3];
```

a) Escreva um programa que leia as notas (de 0 a 20) dos alunos de uma turma (no máximo 25), em três disciplinas, para um vector do tipo *tabnotas*, e que determine a nota média de cada aluno e a nota média de cada disciplina. Durante a leitura deve ser apresentado no ecrã o número de ordem do aluno e o número da disciplina cuja nota o utilizador deve introduzir.

b) Altere o programa da alínea anterior, considerando a existência de dois vectores "paralelos", onde são guardados os nomes dos alunos e das disciplinas. Neste caso, deve-se começar por pedir ao utilizador que indique o nome de todos os alunos e de todas as disciplinas e só depois as notas.

c) Altere o programa da alínea anterior, de modo a que sejam apresentados no ecrã os nomes dos alunos aprovados. Um aluno considera-se aprovado se tiver nota superior ou igual a 10, em todas as disciplinas.

- 13) Escreva um programa que permita efectuar a introdução de uma matriz (de dimensão $n \times m$ com uma dimensão máxima de 10 linhas por 10 colunas) e ordenar inicialmente todas as linhas dessa matriz por ordem crescente e seguidamente todas as colunas por ordem decrescente, escrevendo o resultado final, sobre a forma matricial no ecrã.

Exemplo:

Número de Linhas: 4 Número de Colunas: 4

$M[1,1]= 2$	$M[1,2]= 3$	$M[1,3]= 5$	$M[1,4]= 4$
$M[2,1]= 1$	$M[2,2]= 4$	$M[2,3]= 1$	$M[2,4]= 1$
$M[3,1]= 8$	$M[3,2]= 5$	$M[3,3]= 6$	$M[3,4]= 7$
$M[4,1]= 3$	$M[4,2]= 1$	$M[4,3]= 3$	$M[4,4]= 2$

$\begin{bmatrix} 2 & 3 & 5 & 4 \\ 1 & 4 & 1 & 1 \\ 8 & 5 & 6 & 7 \\ 3 & 1 & 3 & 2 \end{bmatrix}$	$\begin{bmatrix} 2 & 3 & 4 & 5 \\ 1 & 1 & 1 & 4 \\ 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 3 \end{bmatrix}$	$\begin{bmatrix} 5 & 6 & 7 & 8 \\ 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 3 \end{bmatrix}$
--	--	--

- 14) Escreva uma função que indique o número de vezes que o maior número de uma tabela de N elementos ocorre. (Nota: Só deve ser efectuada uma passagem pela tabela).
- 15) Escreva um programa que leia uma matriz de 6×6 , representando as altitudes de uma determinada região e determine quantos pontos possuem coordenadas inferiores a 10 e são simultaneamente mínimos locais (ou seja, têm um valor menor que todos os seus vizinhos). A leitura dos elementos da

matriz deve ser feita de tal modo que a posição do ecrã onde são lidos os elementos corresponda à posição desses elementos na matriz.

- 16) Escreva um programa que simule o jogo do MasterMind. Neste jogo, inicialmente é gerado aleatoriamente um código de cinco dígitos, sendo possível ao utilizador, em seguida, efectuar tentativas para acertar nesse código. Em cada tentativa, é indicado ao utilizador o número de dígitos do código em que acertou na posição correcta e o número de dígitos em que acertou mas numa posição incorrecta.

Exemplo:

Supondo códigos com 5 dígitos e que o Código gerado foi 13349

Nº Tentativa	Código	Certas	Fora do Sítio	
Tentativa 1:	45367	1	1	
Tentativa 2:	12547	2	0	
Tentativa 3:	13837	2	1	
Tentativa 4:	13943	2	3	
Tentativa 5:	13349	5	0	Acertou

- 17) Escreva uma função *int kmaior(vector x, int k, int n)*, que determine o valor do k-ésimo maior elemento de um vector com n elementos (*k* e *n* devem ser passados como parâmetros à função).

Exemplo:

Para o vector x[] =: 10 20 13 14 18 3 5 8 15 2

Kmaior(x, 3, 10) = 15 (pois o 3º maior elemento do vector é 15)

Kmaior(x, 8, 10) = 3 (pois o 8º maior elemento do vector é 3)

- 18) Escreva uma função que, dado um array de inteiros já preenchido que admite elementos repetidos, conte o número de elementos repetidos. Exemplo: no array *a[]*=2,4,1,4,2,3,4,3 existem 4 elementos repetidos, que são 4,2,4,3. Teste a função com recurso a uma função cliente de teste.
- 19) Escreva uma função que, dado um array de inteiros, conte o número de elementos compreendidos entre *vmin* e *vmax* fornecidos pelo utilizador. Exemplo: se *vmin*=10 e *vmax*=20, para o array *a[]*=10,14,9,7,16,13,20,7 existem 5 elementos entre *vmin* (10) e *vmax* (20), que são 10,14,16,13,20. Teste a função com recurso a uma função cliente de teste.

UNIVERSIDADE FERNANDO PESSOA

FICHA DE EXERCÍCIOS Nº 7: CADEIAS DE CARACTERES (“STRINGS”)

- 1)
 - a) Escreva um programa que leia uma frase e que converta todas as letras para maiúsculas.
 - b) Altere o programa de forma a que todas as palavras comecem por uma letra maiúscula.
- 2) Escreva um programa que leia uma frase e elimine todos os caracteres "espaço" que apareçam repetidos entre duas palavras.
- 3)
 - a) Escreva um programa que leia uma frase e a decompõe nas palavras constituintes.
 - b) Altere o programa de forma a que as palavras sejam armazenadas num vector de palavras e seja indicado qual o comprimento médio das palavras encontradas.
- 4) Escreva um programa que leia uma frase e substitua todas as ocorrências de uma dada palavra por uma outra palavra. Estas duas palavras deverão ser indicadas pelo utilizador. Só são consideradas palavras as sequências de caracteres delimitadas por um carácter “espaço”, de cada lado; no entanto, as palavras a indicar pelo utilizador não deverão incluir estes caracteres.
- 5) Escreva um programa que leia um número em base 8, representado sob a forma de uma string e o converta para a base 10.
- 6) Escreva um programa que determine quantas vezes uma sub-cadeia de caracteres (fornecida pelo utilizador) aparece no interior de uma outra cadeia (também fornecida inicialmente pelo utilizador). Nesta contagem, não considere as sobreposições entre sub-cadeias, ou seja, considere que a cadeia ‘BABABABA’ contém duas sub-cadeias ‘BAB’ (‘BABABABA’) e não três (‘BABBABA’). Altere o programa anterior de forma a considerar como válidas na contagem as sobreposições.
- 7) Escreva um programa que, dado um texto cifrado usando uma cifra de César com parâmetro de deslocamento k (tipo de encriptação simples em que cada letra é substituída por outra que está deslocada k posições, sendo o parâmetro de deslocamento k o mesmo para todas as letras), encontre o texto original.
- 8) Escreva um programa que converta um nome de uma pessoa de um “formato normal” para um “formato compacto”, como se exemplifica a seguir:

Exemplo:

<i>Formato normal:</i>	<i>Zeferino Pedro Duarte Aguiar</i>
<i>Formato compacto:</i>	<i>Aguiar, Z. P. D.</i>

- 9) Escreva um programa que leia uma string representando uma frase (com um máximo de 60 caracteres) e leia uma palavra (com um máximo de 20 caracteres) e permita determinar o número de ocorrências da palavra na frase e qual a sua respectiva localização. As localizações devem ser armazenadas num vector de inteiros e devem ser escritas na forma apresentada no exemplo.

Exemplo de funcionamento:

Entre a Frase: *A casa onde foi o casamento era ao lado da minha casa*

Entre a Palavra: casa

Número de ocorrências da palavra casa: 2

Localizadas nas posições: 3-6 , 50-53

Nota: Cuidado com as palavras que contêm a palavra a encontrar (ex. “casamento”) e com as ocorrências da palavra que se encontram no início ou no fim da frase.

- 10) Escreva um programa que leia um texto (introduzido pelo utilizador) e o apresente no ecrã "justificado" a 20 linhas. Cada linha do texto resultado, deve conter 20 caracteres e deve estar encostada simultaneamente à margem esquerda e direita
- 11) Altere o programa anterior incluindo as opções de centrar o texto, alinhamento à esquerda e alinhamento à direita.
- 12) Escreva um programa que leia um vector de strings *ST* e em seguida uma string *S* que indica qual (quais) os valores a procurar no vector. Esta string *S* pode englobar o carácter *, o qual pode substituir qualquer string incluindo a string vazia (funcionando como um carácter *wildcard* de correspondência de padrões de texto). A saída do programa deve consistir em todos os elementos do vector *ST* que possam ser identificados pela string *S*. Considere que a string *S* só pode conter (no máximo) um símbolo *.

Exemplo:

ST = ('Antonio', 'Antonio Sequeira', 'Paulo Oliveira', 'Alberto Mouta', 'Antonio Reis')

<u>Entrada (S)</u>	<u>Saída</u>
<i>Paulo Oliveira</i>	<i>Paulo Oliveira</i>
<i>Luis Reis</i>	<i>O Nome não consta da lista</i>
<i>Antonio*</i>	<i>Antonio</i>
	<i>Antonio Sequeira</i>
	<i>Antonio Reis</i>
<i>*eira</i>	<i>Antonio Sequeira</i>
	<i>Paulo Oliveira</i>
<i>A*a</i>	<i>Antonio Sequeira</i>
	<i>Alberto Mouta</i>

UNIVERSIDADE FERNANDO PESSOA

FICHA DE EXERCÍCIOS Nº 8: PROGRAMAÇÃO EM C - REGISTOS E FICHEIROS

- 1) Escreva um programa que permita a criação de um ficheiro binário com uma sequência de valores reais, positivos, inferiores ou iguais a 100 (percentagens obtidas por um conjunto de alunos num exame). O programa deve também permitir efectuar a leitura do ficheiro, guardando a sequência de valores lidos, num vector. Em seguida deve determinar a média e o desvio padrão dos valores lidos e quantos valores estão compreendidos nos intervalos [0..10], [10..20], ..., [90..100]. Admita que o número de alunos não é superior a 500. Construa em seguida um histograma para os intervalos considerados.

- 2) Considere as seguintes declarações:

```
#define NUM_DISCIPLINAS 9
typedef char Nome[30];
typedef enum {mat, fis, qui, por, fra, ing, fil, cie, efi} DISCIPLINAS ;
typedef struct {
    Nome nome_al;
    int notas[NUM_DISCIPLINAS];
} ALUNO;
```

- a) Escreva um programa que crie um ficheiro contendo registos *ALUNO*, contendo os nomes e as notas dos alunos de uma escola.

- b) Escreva um procedimento que determine para cada uma das disciplinas, a média das notas, o número de aprovações e o número de reprovações.

- c) Altere o programa anterior incluindo um procedimento que determine também, para cada aluno, o número de disciplinas a que obteve aprovação.

- d) Crie ainda um novo procedimento que dado um nome de um aluno, permita calcular a média de todas as disciplinas a que obteve aprovação.

- 3) Os elementos de dois ficheiros são registos contendo um nome e um salário:

```
typedef struct { char nome_al[30]; float salario; } ELEMENTO;
```

Ambos os ficheiros estão ordenadas por ordem alfabética crescente de nomes. Construa um programa que funda os dois ficheiros, criando um ficheiro de texto com todos os nomes (e respectivos salários) contidos nos dois ficheiros, ordenados por ordem alfabética. Deve ser colocado um nome e um salário em cada uma das linhas do novo ficheiro criado.

- 4) Escreva um programa que crie um ficheiro binário contendo todas as datas dos anos entre 1900 e 1999. Para tal utilize o seguinte tipo:

```
typedef struct { int dia, mes, ano; } DATA;
```

O cálculo de todas as datas deve ser realizado de forma automática.

- 5) Considere as seguintes declarações de um programa:

```
enum genero {masc, fem};
enum areas_interesse {música, literatura, pintura, cinema, desporto};
typedef struct { char nome[50]; int idade; enum genero sexo; int
    interesse[5]; } pessoa;
FILE * f_pessoa;
```

Uma agência de casamentos pretende fazer uma lista de casamentos possíveis entre as pessoas constantes de um ficheiro *f_pessoa*. O ficheiro é de acesso aleatório (binário). Considera-se que um casamento é possível entre duas pessoas se elas forem de sexo diferente, se a sua diferença de idades não for superior a 20 anos e se tiverem pelo menos duas áreas de interesse em comum (no ficheiro não

constam pessoas casadas!).

- a) Escreva um programa que construa um ficheiro do tipo apresentado.
- b) Escreva um programa que, usando o ficheiro construído na alínea anterior, gere a referida lista de possíveis casamentos, guardando-a num ficheiro de texto, com a formatação que se indica a seguir:

Ficheiro de Texto Exemplo:

1 - MANUEL 30 M música cinema desporto

Pode Casar-se com:

 JOANA 45 F cinema desporto

 LUÍSA 21 F música pintura cinema

2 - JOAO 25 M cinema pintura

Fica solteiro !

3 - PAULO

...

24 - JOANA 45 F cinema desporto

Pode Casar-se com:

 MANUEL 30 M música cinema desporto

 RAUL 33 M pintura cinema desporto

- 6) Escreva um programa que crie um ficheiro de acesso directo ('FUTEBOL.DAT') contendo a seguinte informação acerca dos clubes que disputam um campeonato de futebol:

- ◇ Nome do clube
- ◇ Número de golos marcados
- ◇ Número de golos sofridos
- ◇ Pontuação

O programa deverá ler o nome de cada um dos clubes e escrevê-lo no ficheiro, inicializando com o valor 0 cada um dos restantes campos de cada registo. Os nomes devem ser introduzidos por ordem alfabética. Quando houver nomes de clubes com mais do que uma palavra, as palavras serão escritas sem espaço entre si (ex: VITORIAGUIMARAES).

- 7) Escreva um programa que actualize o ficheiro criado pelo programa anterior e que, no final escreva o conteúdo do ficheiro actualizado. Após cada jornada é preparado um ficheiro de texto ('RESULT.DAT'), contendo os resultados da jornada, com o aspecto que se exemplifica a seguir:

Ficheiro Exemplo:

PORTO 4 BENFICA 0

VITORIAGUIMARAES 1 SPORTING 1

BELENENSES 0 BOAVISTA 1

...

O programa deverá ler o ficheiro 'RESULT.DAT' e actualizar o ficheiro 'PONTOS.DAT'. O programa deverá conter subprogramas que executem as seguintes tarefas:

- ◇ Ler uma linha do ficheiro *RESULT.DAT* e decompô-la nos seus elementos (admita que entre estes existe apenas um espaço);
- ◇ Procurar um clube no ficheiro *PONTOS.DAT*;
- ◇ Actualizar um registo do ficheiro *PONTOS.DAT*;
- ◇ Escrever o conteúdo do ficheiro *PONTOS.DAT*.

- 8) Reformule o problema da ordenação pelo método “Bubble-sort” de modo a que os elementos do vector a ordenar sejam lidos a partir de um ficheiro texto, com o nome ‘Entrada.txt’ (que pode ser criado através de um editor de texto). Escreva 2 versões do programa:
- a) uma em que os resultados sejam escritos no écran;
 - b) outra em que os resultados sejam escritos noutra ficheiro de texto ‘Saida.txt’.
- 9) Os elementos de dois ficheiros de texto são palavras com o máximo de 20 caracteres. As sequências de palavras dos dois ficheiros estão ordenadas por ordem alfabética crescente. Faça um programa que funda as duas sequências e escreva o resultado num ficheiro do mesmo tipo.
- 10) Escreva um programa que leia de um ficheiro de texto um conjunto de nomes de pessoas, converta todas as letras minúsculas para maiúsculas e elimine as partículas ‘E’, ‘DE’, ‘DA’, ‘DO’, ‘DOS’, ‘DAS’ e ‘D’. Os nomes devem ser lidos um por um. Escreva os resultados noutra ficheiro de texto.
- 11) Escreva um programa que leia linhas de um ficheiro de texto e que as apresente no terminal (de 80 colunas) devidamente justificadas, ou seja, de forma a que a primeira palavra de cada linha comece na primeira coluna da linha e a última palavra termine na última coluna dessa linha. Os espaços entre as palavras devem ser distribuídos de forma uniforme. Supõe-se que nenhuma linha do ficheiro excede 80 caracteres e que cada linha tem pelo menos duas palavras e que estas estão separadas por um único espaço.
- 12) Escreva um programa que converta todos os nomes de pessoas contidos num ficheiro de texto (um nome por linha), de um “formato normal” para um “formato compacto”, como se exemplifica a seguir. O resultado (nomes num formato compacto) devem ser colocados num outro ficheiro de texto.

Exemplo:

<i>Formato Normal:</i>	<i>Zeferino Pedro Dinis Aguiar</i>
<i>Formato Compacto:</i>	<i>Aguiar, Z. P. D.</i>