Exercícios de Programação e Computadores

Secção de Matemática e Física
Departamento de Engenharia Civil
FEUP

1999–2000

Antes de começar ...

- Os exercícios estão apresentados com dificuldade gradual: os primeiros problemas de cada secção apenas testam a compreensão dos assuntos estudados, progredindo nos problemas seguintes para a aplicação de conhecimentos a novas situações.
- Os exercícios marcados com o símbolo $\triangleright$ foram retirados de exames da disciplina.

1 Algoritmos

1.1 Para cada uma dos seguintes algoritmos, simule a sua execução e indique qual o conteúdo final das variáveis em causa.

(a) $a \leftarrow 3$
$b \leftarrow 2$
$c \leftarrow \sqrt{a^2 + b^2}$

(b) $a \leftarrow 2$
$b \leftarrow 7$
$t \leftarrow a$
$a \leftarrow b$
$b \leftarrow t$

(c) $x \leftarrow 3$
$y \leftarrow (x-1) \times (x+2)$
$y \leftarrow (y-1) \times (y+1)$

(d) $x \leftarrow 2$
$x \leftarrow 1/(1+x)$
$x \leftarrow 1/(2+x)$
$x \leftarrow 1/(3+x)$

(e) $x \leftarrow 0.5$
$y \leftarrow x + 1$
$y \leftarrow y \times x + 2$
$y \leftarrow y \times x + 3$
$y \leftarrow y \times x + 4$

(f) $N \leftarrow 567$
$a \leftarrow \mathrm{MOD}(N, 10)$
$N \leftarrow \mathrm{INT}(N/10)$
$b \leftarrow \mathrm{MOD}(N, 10)$
$c \leftarrow \mathrm{INT}(N/10)$

(g) $a \leftarrow 2$
$b \leftarrow 1$
$c \leftarrow -3$
$d \leftarrow b^2 - 4ac$
se $d < 0$ então
  escreve "`equação impossível`"
senão
  $x_1 \leftarrow (-b + \sqrt{d})/(2a)$
  $x_2 \leftarrow (-b - \sqrt{d})/(2a)$
  escreve x_1, x_2
fim

(h) $N \leftarrow 123$
$M \leftarrow 0$
enquanto $N \neq 0$ fazer
    $M \leftarrow 10 \times M + \mathrm{MOD}(N, 10)$
    $N \leftarrow \mathrm{INT}(N/10)$
fim de ciclo enquanto

(i) $f \leftarrow 1$
para $n = 2$ até 7 fazer
    $f \leftarrow f \times n$
fim de ciclo em n

(j) $n \leftarrow 1$
$f \leftarrow 1$
enquanto $f < 1000$ fazer
    $n \leftarrow n + 1$
    $f \leftarrow f \times n$
fim de ciclo enquanto

1.2 Para trocar os valores de duas variáveis a e b basta usar uma terceira variável "temporária" t e efectuar as atribuições $t \leftarrow a,\ a \leftarrow b,\ b \leftarrow t$.

Escreva uma sequência de acções que troque os valores de *quatro* variáveis (a, b, c, d) entre si de forma a obter (b, c, d, a). Dito de outra forma, o valor final de a deverá ser o valor original de b, etc. Tente usar o menor número de atribuições e variáveis temporárias.

1.3 O algoritmo seguinte compara dois valores dados em a, b e coloca em m o maior deles:

ler x, y
se $x > y$ então $m \leftarrow x$ senão $m \leftarrow y$
fim

Baseado-se neste algoritmo escreva outros algoritmos que:

(a) Dados três valores a, b, c coloque em m o maior deles.

(b) Dados três valores a, b, c colocar em m_1 o maior deles e em m_2 colocar o menor deles.

(c) Dados três valores a, b, c coloque em m o valor do meio (por exemplo: se fosse $a \leq b \leq c$ então deveria ficar $m = b$).

1.4 O *algoritmo de Euclides* permite calcular o máximo divisor comum de dois números naturais a, b: o maior número inteiro que divide simultaneamente a e b. Recorde ainda que "MOD(a, b)" é o resto da divisão inteira de a por b.

Algoritmo de Euclides: máximo divisor comum entre a e b
(o resultado é o valor final de b)
$r \leftarrow \text{MOD}(a, b)$
enquanto $r \neq 0$ fazer:
    $a \leftarrow b$
    $b \leftarrow r$
    $r \leftarrow \text{MOD}(a, b)$
fim de ciclo enquanto

(a) Utilize o algoritmo indicado para calcular o mdc(76,34) e mdc(224,7). Quantas vezes executou o ciclo "enquanto ... " no algoritmo em cada um dos casos?

(b) Modifique o algoritmo para determinar se os dois números a, b são *primos entre sí*, *i.e.*, se o seu máximo divisor comum é 1.

(c) Modifique o algoritmo para calcular o mínimo múltiplo comum entre a e b, usando a relação $\text{mmc}(a, b) = (a \times b)/\text{mdc}(a, b)$.

2 Linguagem FORTRAN 90

2.1 Diga quais dos seguintes nomes são válidos para variáveis de FORTRAN 90, justificando a sua resposta:

`Velocidade`	`João`	`Maria`	`alpha`
`1z2`	`z12`	`alpha_12`	`N/4`
`Este_nome_tem_muitas_letras`	`X-1`	`_1G`	`X&Y`
`Albert Einstein`	`X$`	`PI`	`1.23E14`

2.2 Num programa em FORTRAN 90 foram declaradas as variáveis seguintes:

```
integer ::  J=2, K=2, KK=7, L=-3
real ::  A=2.0, B=3.5, Theta=10.0, XYZ=5.0
```

Qual o valor de cada uma das expressões:

(a) `J*(K-KK)/(9+L)`
(b) `J*((K-KK)/(9+L))`
(c) `-(A+Theta)`
(d) `(A+B)/(J+K)`
(e) `KK/(J+K)`
(f) `KK/(A+K)`
(g) `REAL(KK)/(J+K)`
(h) `INT(B/A) + 1`

2.3 Escreva cada uma das expressões algébricas seguintes como uma atribuição em FORTRAN 90. Caso seja necessário escolha nomes adequados para variáveis e assuma declarações de tipos convenientes.

(a) $z = \frac{1}{x+y}$
(b) $E_c = \frac{1}{2}mv^2$
(c) $T = 2\pi\sqrt{L/g}$
(d) $a = \sin\frac{y}{\sqrt{x^2+y^2}}$
(e) $r = A\sin(\theta - t)$
(f) $x = \sin 2\pi L$
(g) $z = \sqrt[3]{x+y}$
(h) $z = \sqrt[n]{x}, \quad n \in \mathbb{N}$
(i) $r = \log_{10} x$
(j) $\alpha = \log\left(\tan^{-1}\frac{\sqrt{a^2+b^2}}{|c|}\right)$

2.4 Suponha que num programa FORTRAN 90 as variáveis `A`, `B`, `C` foram declaradas como reais e `N`, `M`, `L` como inteiras. Escreva cada uma das seguintes condições como expressões lógicas:

(a) $A > 0 \ \wedge \ B > 0$
(b) $B > A \geq 0$
(c) $\frac{1}{4} < |A| < \frac{1}{2}$
(d) $|A - B| \leq 10^{-3}$
(e) $N = M \ \vee \ N = L$
(f) $N = M = L$
(g) $N = M = L = 2$
(h) $\mathrm{INT}(A + B) = N$
(i) N é múltiplo de 2 ou 3
(j) $M \neq 0$ e N é múltiplo de M

2.5 Qual é o resultado escrito pelos seguintes programas em FORTRAN 90? Simule a sua execução com papel e lápis antes de os tentar executar num computador!

(a)
```
PROGRAM exemplo_1
WRITE(*,*) (-1-sqrt(5.))/2
STOP
END PROGRAM exemplo_1
```

(b)
```
PROGRAM exemplo_2
IMPLICIT NONE
REAL :: pi
pi = ACOS(-1.)
WRITE(*,*) pi, pi/2, pi/3
STOP
END PROGRAM exemplo_2
```

(c)
```
PROGRAM exemplo_3
IMPLICIT NONE
REAL :: a=2.0, b=3.0, r
r = SQRT(a**2 + b**2)
WRITE(*,*) r
STOP
END PROGRAM exemplo_3
```

(d)
```
PROGRAM exemplo_4
IMPLICIT NONE
REAL,PARAMETER ::  Pi=3.14159
REAL ::  Graus, Rads
READ(*,*) Graus
Rads = Graus/180*Pi
WRITE(*,*) Rads
STOP
END PROGRAM exemplo_4
```

(e)
```
PROGRAM exemplo_5
IMPLICIT NONE
REAL ::  x,y
READ(*,*) x,y
IF(x>=0) THEN
  IF(y>=0) THEN
    WRITE(*,*) "1o. Q."
  ELSE
    WRITE(*,*) "2o. Q."
  END IF
ELSE
  IF(y>=0) THEN
    WRITE(*,*) "3o. Q."
  ELSE
    WRITE(*,*) "4o. Q."
  END IF
END IF
STOP
END PROGRAM exemplo_5
```

(f)
```
PROGRAM exemplo_6
IMPLICIT NONE
INTEGER ::  n,f
f = 1
DO n=2, 7
  f = f*n
END DO
WRITE(*,*) f
STOP
END PROGRAM exemplo_6
```

(g)
```
PROGRAM exemplo_7
IMPLICIT NONE
INTEGER ::  i
REAL ::  x
DO i=0, 10
  x = 1.+0.1*i
  WRITE(*,*) x, LOG(x)
END DO
STOP
END PROGRAM exemplo_7
```

(h)
```
PROGRAM exemplo_8
IMPLICIT NONE
INTEGER ::  n=0
REAL ::  f=1.0, y=1.0
DO WHILE(f<1000.AND.y<1000)
  WRITE(*,*) n, f, y
  n = n+1
  f = f*n
  y = EXP(REAL(n))
END DO
STOP
END PROGRAM exemplo_8
```

3 Programas sequenciais

3.1 A área de um qualquer triângulo cujos lados medem a, b, c pode ser calculada pela fórmula $A = \sqrt{s(s-a)(s-b)(s-c)}$, onde $s = (a+b+c)/2$.

Escreva um programa em Fortran 90 que lê os valores de a, b, c e calcula a área do triângulo correspondente.

3.2 Sabendo que o volume V e superfície exterior S de um cilindro de altura h e diâmetro de base d são dados pelas fórmulas

$$V = \frac{h\pi d^2}{4} \qquad S = h\pi d + \frac{\pi d^2}{2}$$

escreva um programa Fortran 90 que leia os valores de h e d e calcule o volume e superfície do cilindro.

3.3 Em engenharia electrotécnica é comum exprimir a relação entre dois valores de potência em *decibeis*, ou dB, dada pela equação

$$\mathrm{dB} = 10 \log_{10} \frac{P_2}{P_1}$$

onde P_2 é o valor de potência a ser medido e P_1 é um valor entendido como referência.

Assumindo como valor de referência 1 miliwatt, escreva um programa em FORTRAN 90 que lê o valor de P_2 e converte-o em dB.

3.4 A força de atracção gravitacional entre dois corpos de massas m_1 e m_2 é dada pela equação

$$F = \frac{G m_1 m_2}{r^2}$$

onde G é a constante de gravitação ($6.672 \times 10^{-11}\ Nm^2/kg^2$), m_1, m_2 são as massas em quilogramas e r é a distância entre os dois corpos em metros.

Escreva um programa em FORTRAN 90 que leia os valores de massas e distância e calcule a força de atracção entre os dois corpos. Teste o seu programa calculando a força exercida pela a Terra sobre um satélite de 800 kg em órbita a 38000 km (a massa da Terra é 5.98×10^{24} kg).

3.5 Numa conta a prazo com capitalização automática o valor do juro é acumulado ao capital inicial no final de cada período. Assim, o capital acumulado ao fim de N períodos é dado pela equação

$$\text{Capital final} = \text{Capital inicial} \times (1+t)^N$$

onde t é a taxa de juro aplicada ($0 < t < 1$).

Por exemplo: com um capital inicial de 1000€ e uma taxa anual de 5% o capital acumulado ao fim de dois anos será $1000€ \times (1 + 0.05)^2 = 1102.5€$.

Escreva um programa em FORTRAN 90 que leia um valor de capital inicial, uma taxa de juro anual e o número de anos e calcule o capital acumulado no final desses anos.

3.6 Para pequenas oscilações, o período dum pêndulo é independente da sua massa e amplitudes máximas, dependendo apenas do comprimento L do fio em que está suspenso (por esta razão os pêndulos foram usados para marcar o tempo nos primeiros relógios mecânicos).

A relação entre a *frequência* (número de oscilações por segundo) e o comprimento do fio é dada pela expressão

$$f = \frac{1}{2\pi}\sqrt{\frac{g}{L}}$$

onde $g \approx 9.80\ m/s^2$ é a constante de aceleração gravitacional ao nível do mar.

O período T de cada oscilação é então o inverso da frequência: $T = 1/f$.

(a) Escreva um programa em FORTRAN 90 que leia o valor do comprimento do fio e que calcule e escreva a frequência e período da oscilação.

(b) Re-escrevendo as expressões dadas, escreva um outro programa que leia o valor do período desejado e calcule qual o comprimento do fio necessário.
Teste esse programa calculando qual o comprimento do fio para obter oscilações de 1/2 s, 1 s e 2 s.

3.7 Escreva um programa em FORTRAN 90 que leia os valores dos três ângulos de um triângulo e o classifique como equilátero (três ângulos iguais), isósceles (dois ângulos iguais) ou escaleno (caso contrário).

Além disso, se algum dos ângulos for 90° o programa deve ainda indicar que o triângulo é rectângulo.

3.8 (a) Escreva um programa FORTRAN 90 que leia as coordenadas (x, y) dum ponto no plano e indique em que quadrante (1º, 2º, 3º ou 4º) este se encontra.

(b) Modifique o programa anterior para indicar também qual o semi-quadrante onde se encontra o ponto (1º, 2º, ... , 8º).

3.9 Uma equação do 2º grau tem a forma genérica $ax^2 + bx + c = 0$ com $a, b, c \in \mathbb{R}$ e $a \neq 0$. Podemos achar as suas raízes usando a conhecida fórmula resolvente

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Escreva um programa em FORTRAN 90 que leia os valores dos parâmetros a, b, c e determine, se possível, as duas raízes reais da equação. Se a equação não tiver raízes, o programa deverá indicar uma mensagem adequada.

3.10 A tabela seguinte apresenta factores de conversão entre algumas unidades de comprimento, volume e massa:

Converter de	**para**	**Factor**
Polegada	Centímetros	×2.54
Pés	Metros	×0.3048
Libras	Quilogramas	×0.45359237
Galões E.U.A	Litros	×3.785411784

(a) Escreva um programa em FORTRAN 90 para converter estas unidades. O programa deve começar por apresentar uma lista com as opções de unidades para converter (por exemplo, numeradas de 1 a 4) e só depois pedir o valor a converter. SUGESTÃO: Torne o seu programa mais estruturado declarando os factores de conversão como `PARAMETER`.

(b) Modifique o programa anterior para permitir também as conversões no sentido inverso. Consegue evitar calcular novos factores de conversão?

3.11 Escreva um programa em FORTRAN 90 que leia o número de um mês (inteiro de 1 a 12) e dum ano (inteiro, por exemplo: 1998) e determine quantos dias tem esse mês seguindo a seguinte informação:

Abril, Junho, Setembro e Novembro:	30 dias
Fevereiro:	28 dias ou 29 dias (ver nota)
restantes:	31 dias

Se o mês introduzido não for válido, o programa deve escrever uma mensagem de erro apropriada (e não escrever o número de dias).

Nota: o mês de Fevereiro tem 29 dias nos anos bissextos e 28 nos anos normais. Um ano é bissexto se e só se verifica a condição:

$$(\mathrm{MOD}(ano, 4) = 0 \wedge \mathrm{MOD}(ano, 100) \neq 0)) \;\vee\; \mathrm{MOD}(ano, 400) = 0$$

3.12 Escreva um programa em FORTRAN 90 que leia uma letra numa variável caracter (*i.e.*, tipo `CHARACTER(LEN=1)`) e escreva "`Vogal`" ou "`Consoante`" conforme o tipo da letra lida. Tenha a atenção de considerar as maiúsculas e minúsculas.

Se o caracter lido não for uma letra válida ('a', 'b', ... , 'z' ou 'A', 'B', ... , 'Z') então o programa deve escrever uma mensagem de erro!

3.13 Escreva um programa em FORTRAN 90 que leia um número inteiro entre 1 e 999 e diga se é ou não uma capicua, isto é, se os seus algarimos são os mesmos lidos da direita para a esquerda e vice-versa. O programa deve ainda indicar uma mensagem de erro se o número dado não se encontrar no intervalo especificado.

Por exemplo: 1, 33 e 565 são capicuas; 12 e 234 não são capicuas.

SUGESTÃO: pode obter os algarismos de um inteiro usando o quociente e o resto de divisão inteira por dez (função MOD).

3.14 O custo do aluguer dum automóvel é 0.75€ por Km até aos primeiros 50 Km, 0.65€ por Km para os 100 Km seguintes e 0.50€ por Km acima de 150 Km.

Escreva um programa FORTRAN 90 que lê a distância em quilómetros e calcula o valor total a pagar e a média de custo por quilómetro.

▷ **3.15** As tarifas de um parque de estacionamento, aberto das 8h às 24h, são as seguintes:

1ª hora:	0.50€
2ª e 3ª horas:	0.65€
4ª hora e seguintes:	1.00€

O número de horas a pagar é sempre inteiro (por exemplo, 70 min. corresponde a 2 horas) e não superior a 16h (*i.e.*, os automóveis não podem pernoitar no parque).

Escreva um programa em FORTRAN 90 que leia a hora e minuto (números inteiros) de entrada e saída do parque e calcule o preço a pagar.

3.16 O imposto sobre rendimento de uma pessoa singular é gradativo em diferentes escalões conforme o rendimento colectável. Suponha que os escalões são dados pela tabela seguinte

Rendimento	**Imposto**	
<8,000 €	15% do rendimento	
de 8,000 € até 15,000 €	1,200 € + 28% acima de	8,000 €
>15,000 €	3,160 € + 35% acima de	15,000 €

Por exemplo: uma pessoa com 12,000 € de rendimento estaria no 2º escalão e pagaria 1,200 € + 0.28(12,000 € − 8,000 €) = 2,320 € de imposto.

Escreva um programa em Fortran 90 que leia o valor de rendimento e calcule qual o imposto a pagar. O programa deve ainda indicar qual a taxa de imposto do escalão correspondente.

4 Ciclos e repetição

4.1 Para converter uma temperatura de graus Celsius para graus Fahrenheit podemos usar a relação $T_F = (1.8 \times T_C + 32)$, onde T_C é a temperatura em graus Celsius e T_F a temperatura em graus Fahrenheit.

(a) Escreva um programa que pede o valor de temperatura em graus Celsius e o converta para graus Fahrenheit.

(b) Modifique o programa para que escreva uma tabela de conversão de temperatura desde $0°C$ até $100°C$ com intervalos de $1°C$.

4.2 O *factorial* dum número natural n é $n! = 1 \times 2 \times 3 \times \cdots \times n$. Por exemplo: $5! = 1 \times 2 \times 3 \times 4 \times 5 = 720$. Note ainda que (por convenção) $0! = 1! = 1$.

(a) Escreva um programa em Fortran 90 que calcule o factorial de um número $n \geq 0$ dado.

(b) Escreva um programa em Fortran 90 que tabela a função factorial de 0 até 15. Atenção: o factorial cresce muito rapidamente; em vez de usar variáveis inteiras modifique o programa da alínea anterior de forma a usar variáveis de tipo real de dupla precisão (*i.e.*, declaradas com tipo `REAL(kind=KIND(0d0))`).

▷ **4.3** Um número natural n diz-se *triangular* se $n = 1 + 2 + \cdots + k$ para algum $k \in \mathbb{N}$. Exemplos: 6 é triangular porque $6 = 1 + 2 + 3$, mas 5 não é triangular (porquê ... ?).

(a) Escreva um programa em Fortran 90 que tabele todos os número triangulares menores que 100.

(b) Escreva um outro programa que leia um número n natural e diga se é triangular ou não.

4.4 Na *sequência de Fibonacci* cada termo (excepto os dois primeiros) é a soma dos dois termos anteriores; os dois primeiros termos são 1:

$$1, 1, 2, 3, 5, 8, 11, \ldots$$

(a) Escreva um programa em Fortran 90 que leia um número de termos $n \geq 0$ e escreva n termos desta sequência.

(b) Modifique o programa para que leia um número k e diga se este pertence à sequência de Fibonacci e, em caso afirmativo, indique qual o seu número de ordem na sequência. Em caso negativo, o programa deve indicar que k não é um número de Fibonacci.

4.5 Um número inteiro diz-se um *quadrado perfeito* se é o quadrado de um outro inteiro. Por exemplo: 25 é um quadrado perfeito porque $25 = 5^2$.

(a) Escreva um programa em Fortran 90 que leia um inteiro e diga se é ou não quadrado perfeito.

(b) Modifique o programa anterior para que liste os inteiros entre 1 e 100 que *não são* quadrados perfeitos.

▷ **4.6** De todos os números naturais diferentes de 1 há apenas quatro que podem ser escritos como a soma dos cubos dos seus algarismos. Um destes números é $153 = 1^3 + 5^3 + 3^3$. Sabendo que estes números se encontram entre 100 e 999, escreva um programa em Fortran 90 que os determine.

4.7 Podemos decompor um número natural nos seus algarismos decimais fazendo sucessivas divisões por dez: os restos sucessivos da divisão dão-nos os algarismos do número; quando o quociente for zero, o processo termina.

Vamos, como exemplo, decompor 354 em algarismos: efectuando divisões por 10, os restos sucessivos são os algarismos (das unidades até às centenas); o processo termina quando o quociente é zero.

$$\begin{array}{rrrr} 354 & \underline{|\,10} & & \\ \mathbf{4} & 35 & \underline{|\,10} & \\ & \mathbf{5} & 3 & \underline{|\,10} \\ & & \mathbf{3} & 0 \end{array}$$

(a) Usando esta propriedade, escreva um programa em Fortran 90 que leia um número natural e calcula a soma dos seus algarismos.

(b) Modifique o programa de forma a ler o número natural e calcular o número reverso, *i.e.*, com os algarismos pela ordem contrária.

(c) Agora é fácil fazer um programa que verifique se um número natural qualquer é uma capicua: basta verificar se é ou não igual ao seu reverso.
Modifique o programa da alínea anterior para testar se um número qualquer é ou não capicua usando este método.

4.8 Usando ciclos podemos escrever programas que processam uma sequência de valores, um de cada vez.

O programa seguinte calcula a soma duma sequência de valores, começando por perguntar o seu comprimento (nº de elementos); os elemento são lidos um a um num ciclo.

Modifique o programa dado para calcular:

(a) a **média aritmética** da sequência de valores:

$$\overline{x} = \frac{x_1 + x_2 + \cdots + x_N}{N} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

(b) o **desvio padrão** da sequência de valores:

$$s = \sqrt{\frac{N\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i\right)^2}{N(N-1)}}$$

```
PROGRAM soma
IMPLICIT NONE
INTEGER ::  i, n
REAL ::  x, s
WRITE(*,*) "No. elementos?"
READ(*,*) n
s = 0.
DO i = 1, n
  READ(*,*) x
  s = s + x
END DO
WRITE(*,*) "Soma= ", s
STOP
END PROGRAM soma
```

4.9 Modifique o programa do exercício 4.8 de forma a determinar o maior e menor elementos do sequência e calcular a semi-soma entre eles (recorde o exercício 1.3).

▷ **4.10** Na disciplina de Programação Transcendental I a nota final é dada em função dos trabalhos realizados ao longo do ano segundo este critério: desprezando a pior nota, faz-se a média aritmética das restantes. Por exemplo: com as notas 10, 11, 7 e 9, a nota final será 10 (exactamente).

Escreva um programa em Fortran 90 que leia uma sequência de notas (inteiros entre 0 e 20) e calcule a nota final correspondente.

4.11 Uma outra forma de processar uma sequência de valores, um de cada vez, é usar algun(s) valor(es) como *terminador* da sequência introduzida.

Escreva programas em Fortran 90 para ler uma sequência de números reais positivos, terminada por um qualquer valor ≤ 0 e calcular:

(a) a **média geométrica** $= (x_1 \cdot x_2 \cdots x_N)^{1/N} = \left(\prod_{i=1}^{N} x_i\right)^{1/N}$.

(b) a **média harmónica** $= N/\left(1/x_1 + 1/x_2 + \cdots + 1/x_N\right) = N/\left(\sum_{i=1}^{N} 1/x_i\right)$.

▷ **4.12** Escreva um programa em Fortran 90 que calcula o valor da resistência eléctrica R_E equivalente a um circuito em paralelo de resistências $R_1, R_2, \ldots, R_N$:

$$R_E = \frac{1}{\left(\frac{1}{R_1} + \frac{1}{R_2} + \cdots + \frac{1}{R_N}\right)}$$

O programa deve ler os valores de resistências $R_i > 0$ terminadas por um qualquer valor ≤ 0, e calcular e escreve a resistência equivalente R_E.

▷ **4.13** Sabendo que

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \cdots = \sum_{i=0}^{\infty} (-1)^i \frac{1}{2i+1}$$

escreva um programa em FORTRAN 90 que calcule uma aproximação a π somando os termos da série acima até que o seu valor absoluto seja $< 10^{-5}$.

4.14 As funções trigonométricas como seno, co-seno, tangente, etc. são intrínsecas em FORTRAN 90. No entanto, mesmo que assim não fosse, não seria muito difícil conseguir calculá-las usando as as suas expansões em séries de Taylor:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + \cdots$$
$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots + (-1)^n \frac{x^{2n}}{(2n)!} + \cdots$$

(a) Escreva um programa em FORTRAN 90 que leia um valor do argumento $x \in \mathbb{R}$ (ângulo em radianos) e calcule o seno e co-seno de x usando 10 termos das séries de Taylor acima.
Tente evitar o cálculo desnecessário de potências de x e factoriais escrevendo cada parcela da soma em função da anterior.

(b) Modifique o programa da alínea anterior para somar as parcelas da série até que o seu valor absoluto seja $< 10^{-6}$. O programa deve também indicar quantas parcelas foram somadas até satisfazer este critério (note que o nº de parcelas somadas será diferente para valores diferentes de x).

(c) Note que quando menor for $|x|$ mais rapidamente as séries convergem. Assim, modifique o seu programa de forma a tirar partido da periodicidade do seno e co-seno para reduzir o valor do ângulo x.

▷ **4.15** Alguns integrais não podem ser resolvidos analiticamente mas podem ser convertidos numa soma infinita; por exemplo:

$$F(x) = \int_0^x e^{-t^2}\, dt = x - \frac{x^3}{3 \times 1} + \frac{x^5}{5 \times 2} + \cdots + (-1)^n \frac{x^{2n+1}}{(2n+1)n!} + \cdots$$

Para obtermos uma aproximação numérica basta limitar o número de parcelas da soma; quando mais parcelas tomarmos melhor será a aproximação obtida.

Escreva um programa em FORTRAN 90 que lê um valor de $x \in \mathbb{R}$ e o número de parcelas n e calcula uma aproximação a $F(x)$.

SUGESTÃO: Escreva o termo geral da soma em função do termo da iteração anterior, de forma a evitar re-calcular potências e factoriais.

4.16 Re-escreva o programa do exercício 4.15 de forma a não pedir o número de parcelas n; em vez disso, o programa deve pedir um valor de tolerância $\varepsilon > 0$ e somar parcelas até que o seu valor absoluto seja inferior a ε.

4.17 Embora seja permitido em Fortran 90 usar variáveis reais para controlar um ciclo `DO`, isso é considerado uma má prática de programação: um ciclo em reais como na figura 1 (a) pode executar um nº diferente de vezes em computadores diferentes devido a diferentes erros de arredondamento; por outro lado, o ciclo na figura 1 (b) percorre sempre 5 valores de x, independentemente dos arredondamento efectuados.

```
DO x = 1., 2., 0.25
  WRITE(*,*) x, log(x)
END DO
```

(a)

```
DO k = 0, 4
  x = 1.+k*0.25
  WRITE(*,*) x, log(x)
END DO
```

(b)

Figura 1: Exemplo da conversão dum ciclo de reais para inteiros

Seguindo a ideia do exemplo, diga como converter um ciclo em reais qualquer `DO x =` a`,`b`,`c num ciclo em inteiros "equivalente".

4.18 Em Fortran 90 as funções trigonométricas operam sempre com ângulos em radianos. Para operar com graus é necessário converter o argumento primeiro para radianos.

(a) Escreva um programa que tabele os valores de senos e cosenos para argumentos de $0°$ até $45°$ por intervalos de 1^o.

(b) Modifique o programa anterior para tabelar o valor da tangente mas entre $0°$ e $15°$ por intervalos de $15'$ (recorde que $1' = \frac{1}{60}°$).

▷ **4.19** Se lançarmos verticalmente um projéctil da superfície da Terra com velocidade v este atingirá uma altura máxima dada pela fórmula

$$h = \frac{v^2/(2g)}{1 - v^2/(2gR)}$$

se $v < 2gR$ e, caso contrário, afastar-se-á para sempre com velocidade final

$$v_{final} = \sqrt{v^2 - 2gR}$$

Nestas expressões R é o raio da Terra (aproximadamente 6.366×10^6 m) e g é a aceleração gravitacional (aproximadamente 9.80 m/s^2).

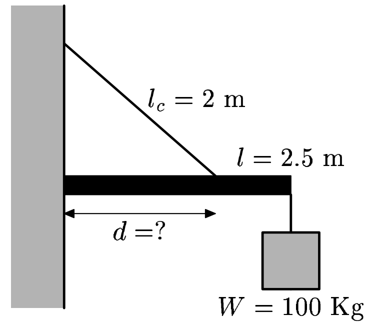
Escreva um programa em Fortran 90 que tabele o resultado obtido (altura máxima ou velocidade final, indicando de qual se trata) para valores de velocidade inicial $v = 10\ \mathrm{m/s}, 100\ \mathrm{m/s}, \ldots 10^6\ \mathrm{m/s}$.

4.20 Um objecto com peso 100 Kg deve ser suspenso de uma haste horizontal de 2.5 m (cujo peso é desprezável). A haste está fixa na parede e é suportada por um cabo de 2 m (ver figura).

A tensão T no cabo é dada pela equação

$$T = \frac{W \times l_c \times l}{d\sqrt{l^2 - d^2}}$$

onde W é o peso do objecto, l_c é o comprimento do cabo, l é o comprimento da haste e d é a distância na haste onde fixamos o cabo.



Escreva um programa em FORTRAN 90 que determine a distância d que minimiza a tensão no cabo. O programa deve calcular a tensão para d de 0.5 m até 2 m por incrementos de 5 cm para localizar a posição de tensão mínima.

5 Sub-programas

5.1 Escreva um sub-programa em FORTRAN 90 para resolver cada um dos seguintes problemas. Em cada caso escolha o tipo de sub-programa mais adequado (`FUNCTION` ou `SUBROUTINE`).

(a) Determinar a média aritmética de quatro valores reais x, y, z, w.

(b) Determinar a mediana de três valores reais x, y, z: se os três valores forem diferentes, a mediana é o valor do meio; se dois ou três forem iguais, a mediana é um desses valores iguais.

(c) Calcular a área e perímetro de uma circunferência dado o raio.

(d) Calcular o volume de um cone dado o raio da base R e a altura h (Volume do cone = $\pi R^2 h/3$).

(e) Escrever a lista dos divisores de um número inteiro n (d é um divisor de n se $\mathrm{MOD}(n, d) = 0$).

(f) Determinar se um número inteiro n é perfeito, *i.e.*, se é igual à soma dos seus divisores menores que n.

(g) Calcular a distância entre dois pontos (x_1, y_1) e (x_2, y_2) no plano.

(h) Converter coordenadas polares (r, θ) de um ponto no plano em coordenadas rectangulares (x, y).

$\triangleright$ **5.2** Dois números naturais n e m dizem-se *amigáveis* se a soma dos divisores próprios de n é igual à de m (os divisores próprios de um número são divisores d tais que $1 \leq d < n$).

(a) Escreva um sub-programa `FUNCTION` em FORTRAN 90 (chamado, por exemplo, "`divisores`") que calcula a soma dos divisores próprios dum numero.

(b) Usando o sub-programa da alínea anterior, escreva um programa principal que leia dois números e diga se são ou não amigáveis.

5.3 Recorde a definição de duas funções importantes em combinatória: as *permutações* e as *combinações* de n elementos p-a-p:

$$P(n,p) = \frac{n!}{(n-p)!} \qquad C(n,p) = \frac{n!}{p!(n-p)!}$$

(a) Escreva dois sub-programas em FORTRAN 90, "`perm`" e "`comb`", que calculem respectivamente estas duas funções.
Tenha o cuidado de minimizar o número de operações aritméticas efectuadas, quer por uma questão de eficiência, quer para evitar "*overflow*" desnecessário. Por exemplo: $P(100, 98) = 100!/98! = 100 \times 99 = 9900$.

(b) Verifique os seus sub-programas escrevendo um programa principal que resolve os seguintes problemas:

- Quatro livros são colocados à sorte numa prateleira. Quantas ordenações existem? R: $P(4, 4)$
- Quantas mãos distintas de 4 cartas é possível tirar de um baralho com 52 cartas? R: $C(52, 4)$
- Qual a probabilidade de acertar no prémio máximo do Totoloto com uma só aposta? R: $1/C(49, 6)$

▷ **5.4** Para determinar o dia da semana a que corresponde uma data dada por três variáveis A, M, D, correspondendo respectivamente ao valor do ano, mês e dia (todas inteiras, com $1 \leq D \leq 31,\ 1 \leq M \leq 12,\ A \geq 1582$) pode usar-se o seguinte algoritmo:

(*Primeiro calculamos um valor FACTOR:*)
se o mês for Janeiro ou Fevereiro:

$$\text{FACTOR} = 365 \times A + 31 \times (M-1) + D + \text{INT}((A-1)/100+1)$$

para os restantes meses:

$$\begin{aligned}\text{FACTOR} = {} & 365 \times A + 31 \times (M-1) + D - \text{INT}(0.4 \times M + 2.3) + \\ & + \text{INT}(A/4) - \text{INT}(0.75 \times (\text{INT}(A/100)+1))\end{aligned}$$

O dia da semana é então dado por SEMANA = MOD(FACTOR,7)
com a correspondência 0≡sábado, 1≡domingo, 2≡segunda, ... 6≡sexta.

(a) Escreva um sub-programa `FUNCTION SEMANA(A,M,D)` em FORTRAN 90 que determina o dia da semana duma data caracterizada por `A`,`M`,`D`.

(b) Escreva um programa principal que use a função da alínea (a) para tabelar o dia de semana do Natal de 1980 até ao ano 2050.

NOTA: Repare que este algoritmo não sofre do famoso "bug do ano 2000"!

5.5 O **método de bissecções sucessivas** permite obter uma aproximação à raiz duma equação $f(x) = 0$: partindo dum intervalo $[a, b]$ onde se encontrar a raiz, vamos dividimo-lo a metade e escolhemos aquela em que a função troca de sinal; o processo termina quando a amplitude do intervalo for inferior à tolerância ε pretendida na aproximação da raiz.

Método de bissecções sucessivas:
$f_1 \leftarrow f(a)$
enquanto $b - a > \varepsilon$
  $m \leftarrow (a+b)/2$
  $f_2 \leftarrow f(m)$
  se $f_1 \times f_2 < 0$ então
    $b \leftarrow m$
  senão
    $a \leftarrow m,\ f_1 \leftarrow f_2$
fim de ciclo enquanto
solução $\leftarrow (a+b)/2$

(a) Escreva um sub-programa em que implemente o método descrito para determinar a raiz de uma função real em FORTRAN 90 `FUNCTION F(X)`.

(b) Escreva um programa principal que use o sub-programa anterior para determinar a raiz de $x^3 - x - 3 = 0$ partindo do intervalo $[1, 2]$ com uma tolerância 0.001.

5.6 A **regra dos trapézios** é um método numérico que permite obter um valor aproximado dum integral $\int_a^b f(x)\,dx$ (que corresponde à área limitada pela gráfico da função $y = f(x)$ entre $[a, b]$).

O método consiste em dividir o intervalo $[a, b]$ em n sub-intervalos e aproximar a função por um segmento de recta em cada sub-intervalo; a fórmula desta aproximação é

$$\int_a^b f(x)\,dx \;\approx\; \frac{h}{2}\left(f(a) + f(b) + 2\sum_{i=1}^{n-1} f(a + i \times h)\right)$$

onde $h = (b-a)/n$ é a amplitude de cada sub-intervalo.

(a) Escreva um sub-programa que aproxime um integral duma função real em FORTRAN 90 `FUNCTION F(X)` usando a regra dos trapézios.

(b) Escreva um programa principal que use o sub-programa anterior para aproximar $\ln 2 = \int_1^2 1/x\,dx$ com 2, 4, 8 e 16 sub-intervalos. Compare os diferentes resultados com o valor dado pela função intrínseca `LOG`.

5.7 A sub-rotina `random_number` é intrínseca em FORTRAN 90. Esta sub-rotina pode ser usada para obter um número "pseudo-aleatório" no intervalo $[0, 1)$. A sequência obtida por sucessivas chamadas desta sub-rotina é uniformemente distribuída e satisfaz outros critérios de "aleatoridade"; no entanto, não é verdadeiramente aleatória e duas execuções do programa produzirão a mesma sequência[1].

Podemos usar esta sub-rotina para fazer programas que simulam acontecimentos aleatórios como jogos de azar. O programa seguinte simula o resultado de fazer dez lançamentos de um dado com seis faces (numeradas de 1 a 6):

[1]Excepto se modificar a "semente" de geração usando `random_seed` — ver um manual de FORTRAN 90 para mais explicações.

```
PROGRAM dado
    IMPLICIT NONE
    INTEGER ::  i      !  contador de lançamentos
    REAL ::  x
    DO i = 1, 10
      CALL random_number(x) !  x é pseudo-aleatório em [0,1)
      WRITE(*,*) 1+INT(6*x) !  converter x para 1,2,3,4,5 ou 6
    END DO
    STOP
END PROGRAM dado
```

(a) Altere o programa de forma a contabilizar o número de vezes que saíram valores pares (2, 4 ou 6) ao longo de 100 lançamentos.
(Note que havendo tantas faces pares como ímpares devemos esperar que aproximadamente 50% dos lançamentos saiam pares ...)

(b) Altere o programa de forma a efectuar 100 lançamentos e contabilizar o número de vezes que dois lançamentos sucessivos somaram 7 ou 11.

▷ **5.8** A Alice e o Bob vão jogar um jogo de azar: lançam alternadamente uma moeda ao ar, cujo resultado pode ser "cara" ou "coroa" (representado por 0 ou 1). Assumimos que a moeda é equilibrada (*i.e.*, "cara" e "coroa" têm a mesma probabilidade de sair) e que a moeda não pode cair em pé ... A Alice ganha o jogo se a sequência "001" sair primeiro e o Bob ganha se a sequência "011" sair primeiro.

Pretende-se simular vários jogos e contabilizar quem ganha mais vezes. Para tal, pode usar a sub-routina intrínseca `random_number` de Fortran 90 (ver o exercício 5.7).

Escreva um programa em Fortran 90 que simula N jogos (onde N é dado pelo utilizador) e contabiliza os jogos ganhos por cada jogador. No final o programa deve apresentar também a frequência com que cada jogador ganhou (=nº jogos ganhos/nº jogos total).

6 Variáveis indexadas

6.1 Num programa em Fortran 90 foram declaradas as seguintes variáveis indexadas:

```
REAL ::  A(10,10), X(10)
INTEGER ::  B(5,5)
```

Escreva uma expressão ou sequência de instruções em Fortran 90 que calcule cada um dos seguintes valores:

(a) a soma dos elementos na 2ª linha de `A`;

(b) a média aritmética dos elementos na 3ª coluna de `A`;

(c) a contagem do nº de elementos positivos de `A`;

(d) a soma dos inversos dos elementos positivos nas duas primeiras linhas de `A`;

(e) a média dos elementos positivos de `A`;

(f) a média do maior e menor valor de `A`;

(g) o produto dos módulos dos elementos de `X`;

(h) o nº de elementos pares em `B`;

(i) a soma das raízes quadradas dos elementos de `B`;

(j) o número de elementos pares em `B`;

(k) o número de elementos de `A` maiores que a média aritmética do vector `X`;

(l) valor lógico: existe algum elemento par em `B`?

(m) valor lógico: serão todos os elementos de `B` maiores que 0?

6.2 (a) Escreva um sub-programa em FORTRAN 90 que calcule a *norma euclidiana* de um vector $x \in \mathbb{R}^n$: $||x|| = \sqrt{x_1^2 + \cdots + x_n^2} = \left(\sum_{i=1}^n x_i^2\right)^{1/2}$.

(b) Baseado no sub-programa da alínea anterior, escreva um outro sub-programa que calcule a *norma de Hölder*: $||x||_p = \left(\sum_{i=1}^n |x_i|^p\right)^{1/p}$, com $p \geq 1$.
(Note que a norma euclidiana corresponde ao caso particular $p = 2$.)

(c) Escreva um programa principal que calcule a norma de Hölder do vector $x = (1, 1, 1) \in \mathbb{R}^3$ para $p = 1$, $p = 3/2$, $p = 2$ e $p = 3$.

6.3 Em álgebra linear é possível definir várias *normas de matrizes* que que traduzem o "tamanho" duma matriz de forma análoga à norma dum vector.

Escreva sub-programas em FORTRAN 90 que calculem as seguintes normas duma matriz $A \in \mathbb{R}^2$ com dimensões $N \times M$:

(a) a norma de Frobenius: $||A||_F = \left(\sum_{i=1}^{N}\sum_{j=1}^{M} a_{ij}^2\right)^{1/2}$

(b) a norma-1: $||A||_1 = \max_{1\leq j\leq M} \sum_{i=1}^{N} |a_{ij}|$

(c) a norma-∞: $||A||_\infty = \max_{1\leq i\leq N} \sum_{j=1}^{M} |a_{ij}|$

SUGESTÃO: Use secções da matriz e a função intrínseca `SUM`.

6.4 Escreva um sub-programa `angulo3d` que determine o ângulo entre dois vectores $x, y \in \mathbb{R}^3$. Para tal, pode usar a relação

$$\cos\theta = \frac{x \cdot y}{|x||y|}$$

onde θ é o ângulo entre os dois vectores, $x \cdot y$ é o seu produto interno e $|x|$ é a norma de x. Para determinar θ sabendo o seu co-seno, pode usar a função intrínseca `ACOS` em FORTRAN 90.

6.5 As taxas de conversão entre o Euro e algumas moedas europeias em Outubro de 1999 eram as seguintes:

1 € corresponde a:	
200.482	Escudos
40.3399	Francos belgas
6.55957	Francos franceses
1.95583	Marcos
166.386	Pesetas
2.20371	Florins
1936.27	Liras

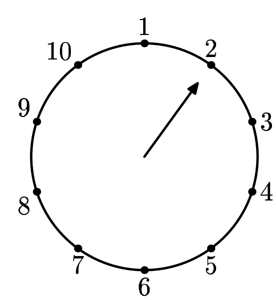
(a) Usando esta informação escreva um programa em FORTRAN 90 que permita converter um valor em Euros para uma destas moedas. O programa deve comecar por apresentar uma lista para o utilizador escolher a conversão desejada.
SUGESTÃO: Represente a tabela de conversão com os nomes e as taxas de conversão usando variáveis indexadas declaradas com `PARAMETER`.

(b) Modifique o programa anterior de forma a permiter converter entre qualquer uma destas moedas (*i.e.*, de Francos franceses para Escudos, por exemplo) usando o câmbio com o Euro.
Procure evitar ter de guardar mais taxas de conversão do que as declaradas no programa anterior ...

▷ **6.6** Considere o seguinte jogo: n pessoas, numeradas de 1 a n são dispostas num círculo. Começando na 2ª pessoa, vamos removendo do círculo "pessoa-sim-pessoa-não" e o círculo aperta-se. O jogo termina quando resta apenas uma pessoa.

Por exemplo, para $n = 10$ a ordem das eliminações é: 2, 4, 6, 8, 10, 3, 7, 1, 9 e 5 é o sobrevivente (ver figura).

Escreva um programa em FORTRAN 90 que leia n e escreva a ordem das eliminações e o sobrevivente de acordo com este algoritmo.



SUGESTÃO: Represente o círculo de pessoas com uma variável indexada.

▷ **6.7** Um **quadrado mágico** é uma matriz quadrada de inteiros em que a soma de qualquer linha, coluna, diagonal principal ou secundária é sempre a mesma (ver exemplo na figura 2).

Escreva um programa em FORTRAN 90 que leia N e uma matriz quadrada A de dimensão $N \times N$ e determine se A é ou não um quadrado mágico.

SUGESTÃO: Use secções da matriz e a função intrínseca `SUM`.

6	1	8
7	5	3
2	9	4

28	19	10	1	48	39	30
29	27	18	9	7	47	38
37	35	26	17	8	6	46
45	36	34	25	16	14	5
4	44	42	33	24	15	13
12	3	43	41	32	23	21
20	11	2	49	40	31	22

Figura 2: Dois quadrados mágicos de 3×3 e 7×7.

6.8 Podemos usar o seguinte algoritmo para gerar um quadrado mágico de $n \times n$ inteiros para n ímpar (ver os exemplos da figura 2):

1. Começe por colocar "1" no meio do 1ª linha;
2. Mova-se na diagonal para cima e para a esquerda (se sair fora do quadrado, reaparece pelo lado oposto como se todo o plano fosse preenchido por quadrados iguais). Coloque na nova casa o "2";
3. Repita o processo para o "3", "4", ... Se atingir uma casa já preenchida, então desça uma casa e continue.

Escreva formalmente o algoritmo acima e codifique-o em FORTRAN 90 num programa que, dado o n ímpar (até, por exemplo, $n \leq 20$), gera e escreva o quadrado mágico.

6.9 **O problema dos aniversários**: Num grupo de n pessoas qual é a probabilidade de que duas ou mais pessoas façam anos no mesmo dia? Podemos usar simulação para responder a esta questão.

(a) Escreva um programa em FORTRAN 90 que leia n e calcule a probabilidade de duas ou mais pessoas em n fazerem anos num mesmo dia.
O programa deve gerar aleatoriamente um vector de dimensão n com n datas de aniversário (inteiros de 1 a 365) usando `random_number` (ver exercício 5.7); depois deve verificar se pelo menos duas datas coincidem.

Para estimar a probabilidade pretendida, o programa deve repetir este processo 1000 vezes e calcular a frequência com que encontrou datas coincidentes, ou seja: "nº de datas coincidentes/1000".

(b) Re-escreva o programa abordando o problema de outra forma: construir um vector com dimensão 365 (uma entrada para cada dia do ano) e seguidamente gerar n datas de aniversário aleatoriamente. A verificação de datas coincidentes torna-se então mais expedita ...

Teste os seus programas para 5, 10, 20 e 40 pessoas.

▷ **6.10** Dado um conjunto de n pontos de coordenadas (X_i, Y_i), o **método de regressão linear** permite determinar os coeficientes da recta $Y = A \cdot X + B$ que se aproxima melhor do conjunto de pontos dado[2]; os coeficientes são determinados por

$$A = \frac{nS_{XY} - S_X S_Y}{nS_{XX} - (S_X)^2} \qquad B = \frac{S_Y - AS_X}{n}$$

onde $S_X = \sum_{i=1}^{n} X_i$ e $S_{XY} = \sum_{i=1}^{n} X_i Y_i$.

Escreva um programa em FORTRAN 90 que leia a dimensão N e dois vector X e Y com as coordenadas dos N pontos e calcule os coeficientes A e B.

▷ **6.11** Considere a seguinte tabela de valores nutricionais por 100g de alguns alimentos comuns:

Alimento	**Calorias**	**Glúcidos**	**Lípidos**	**Proteínas**
pão	239.	49.	1.2	8.
arroz	354.	77.	1.7	7.6
banana	90.	20.	0.5	0.4
maçã	52.	12.	0.3	0.3
couve-flor	30.	4.9	0.2	2.4
tomate	22.	4.	0.3	1.

Pretende-se calcular uma medida estatística da *correlação* entre duas sequências de dados. Sejam $X = [x_1, x_2, \ldots, x_n]$ e $Y = [y_1, y_2, \ldots, y_n]$ as duas sequências; supomos ainda que já calculamos as suas médias aritméticas $\overline{X}$ e $\overline{Y}$:

$$\overline{X} = \tfrac{1}{n} \sum_{i=1}^{n} x_i \qquad \overline{Y} = \tfrac{1}{n} \sum_{i=1}^{n} y_i$$

O *factor de correlação* $\rho(X, Y)$ é dado por

$$\rho(X, Y) = \frac{S_{XY}}{\sqrt{S_{XX} S_{YY}}}$$

onde

$$\begin{aligned} S_{XY} &= \textstyle\sum_{i=1}^{n} (x_i - \overline{X})(y_i - \overline{Y}) \\ S_{XX} &= \textstyle\sum_{i=1}^{n} (x_i - \overline{X})^2 \\ S_{YY} &= \textstyle\sum_{i=1}^{n} (y_i - \overline{Y})^2 \end{aligned}$$

O factor de correlação ρ dá um valor entre -1 e 1: um valor de 1 indica total correlação, 0 indica nenhum correlação e -1 indica total anti-correlação.

(a) Escreva um sub-programa em FORTRAN 90 `CORRELATE` que calcula o coeficiente de correlação ρ de dois vectores reais X, Y.

(b) Escreva um programa principal que lê uma tabela de valores nutricionais como no exemplo dado e que calcula $\rho(cal, glu)$, $\rho(cal, lip)$ e $\rho(cal, prot)$.
Indique ainda como deveria introduzir os dados do exemplo no seu programa.

[2] No sentido estatístico dos "mínimos quadrados".

▷ **6.12** Pretende-se fazer a classificação das alturas de uma população de 1000 pessoas distribuindo-as por 10 intervalos:

$1^{\underline{o}}$ intervalo:	<1.55m
$2^{\underline{o}}$ intervalo:	1.55m – 1.60m
$3^{\underline{o}}$ intervalo:	1.60m – 1.65m
⋮	
$9^{\underline{o}}$ intervalo:	1.90m – 1.95m
$10^{\underline{o}}$ intervalo:	$\geq$ 1.95m

Escreva um programa em FORTRAN 90 que leia as alturas das 1000 pessoas e faça a contagem do número de pessoas em cada intervalo.

SUGESTÃO: Represente as contagens por uma variável indexada pelo número do intervalo. O objectivo é não ter de escrever 10 instruções "`IF ... ENDIF`", uma por cada intervalo!

▷ **6.13** Em Portugal os deputados à Assembleia da República são eleitos segundo o Método de Hondt. Este método consiste em dividir os votos expressos em cada partido pelo número de deputados já eleitos por esse partido mais um, e escolher o maior quociente resultante. O método é aplicado sucessivamente até todos os lugares a eleger estarem preenchidos.

Exemplo: se quisermos eleger 7 deputados de entre os partidos A, B e C cujos votos expressos são dados na $2^{\underline{a}}$ coluna da tabela seguinte

Partido	**Votos**			
A	2460 ($1^{\underline{o}}$)	1230 ($3^{\underline{o}}$)	820 ($6^{\underline{o}}$)	615 ($7^{\underline{o}}$)
B	1830 ($2^{\underline{o}}$)	915 ($5^{\underline{o}}$)	610	
C	960 ($4^{\underline{o}}$)	480		

então o partido A elege 4 deputados, B elege 2 e C elege apenas 1; os deputados são eleitos pela ordem indicada dentro de parêntesis.

Escreva um programa em FORTRAN 90 que leia um vector V de votos expressos nos NP partidos (*i.e.*, NP = dimensão de V) bem como o número total de deputados a eleger (ND), e determine os deputados eleitos por cada partido.

6.14 Em muitas situações é conveniente ordenar uma sequência de valores de forma crescente ou decrescente.

Um dos algoritmos clássicos para ordenar uma sequência $x_1, x_2, \ldots, x_n$ de n valores numéricos por ordem crescente consiste em seleccionar sucessivamente o menor elemento e colocá-lo no início da sequência; este algoritmo é designado *ordenação por selecção*:

Ordenação dum vector $(x_1, x_2, \ldots, x_n)$ *por selecção:*
para i de 1 até $n-1$
    $k \leftarrow i$
    para j de $i+1$ até n
        se $x_j < x_k$ então $k \leftarrow j$
    fim de ciclo em j
    $t \leftarrow x_i$
    $x_i \leftarrow x_k$
    $x_k \leftarrow t$
fim de ciclo em i

Escreva um sub-programa em Fortran 90 que ordene um vector númerico usando este método.

6.15 Se o vector $(x_1, \ldots, x_n)$ estiver já quase ordenado então o algoritmo de ordenação "*bubblesort*" será mais eficiente do que o do exercício 6.14: a ideia é efectuar sucessivas "passagens" em que vamos trocando pares de elementos que não estão por ordem.

Ordenação dum vector $(x_1, x_2, \ldots, x_n)$ *por* "bubblesort":
$m \leftarrow n-1$
enquanto $m > 0$ fazer:
    $l \leftarrow 0$
    para $j = 1$ até m
        se $x_j > x_{j+1}$ então:
            $t \leftarrow x_j$
            $x_j \leftarrow x_{j+1}$
            $x_{j+1} \leftarrow x_j$
            $l \leftarrow j$
    fim de ciclo em j
    $m \leftarrow l$
fim de ciclo enquanto

Escreva um sub-programa Fortran 90 que ordene um vector numérico usando este método.

6.16 Como poderia modificar os algoritmos e programas dos exercício 6.14 e 6.15 de forma a ordenarem por ordem decrescente?

6.17 A procura de um elemento numa sequência ordenada é muito mais fácil do que numa sequência desordenada. Por exemplo: compare o trabalho de procurar na lista telefónica um nº de telefone dado o nome ou procurar o nome dado o nº de telefone!

Podemos formalizar um processo de pesquisa dictómica parecido com o que usamos ao procurar na lista telefónica: dividimos a lista a meio; se acertamos no elemento procurado não há mais nada a fazer; se não, verificamos se o elemento procurado é menor ou maior e continuamos o processo na metade da lista da esquerda ou da direita, respectivamente.

Procura dictómica de y num vector $(x_1, x_2, \ldots, x_n)$ ordenado
Resultado: o índice $l \in \{1, 2, \ldots, n\}$ se o elemento foi encontrado;
caso contrário, $l = 0$
$i \leftarrow 1,\ j \leftarrow n,\ l \leftarrow 0$
enquanto $i < j$
    $k \leftarrow \mathrm{INT}((i + j)/2)$
    se $x_k = y$ então $l \leftarrow k$ e termina o algoritmo;
    senão se $x_k < y$ então $i \leftarrow k$
    senão $j \leftarrow k - 1$
fim de ciclo enquanto

Escreva um sub-programa em FORTRAN 90 para pesquisa dictómica num vector de inteiros usando o método descrito acima.

6.18 Numa situação de estabilidade a temperatura num qualquer ponto de uma placa metálica será a média das temperaturas dos pontos à sua volta. Este facto pode ser usado para aproximar iterativamente a distribuição de temperaturas na placa.

A figura 3 representa a placa metálica quadrada dividida numa malha de 10×10 nós. A temperatura em cada nó T_{ij} é uma variável bi-indexada. Supomos que na fronteira a temperatura é mantida constante igual 20° por um sistema de arrefecimento, e que no nó $(3, 8)$ é de 100° pela exposição a água em ebulição.

20°	20°	20°	20°	20°	20°	20°	20°	20°	20°
20°									20°
20°		100°							20°
20°									20°
20°					T_{ij}				20°
20°									20°
20°									20°
20°									20°
20°									20°
20°	20°	20°	20°	20°	20°	20°	20°	20°	20°

Figura 3: Uma placa metálica dividida numa malha 10×10.

Uma nova estimativa da temperatura no nó (i, j) pode ser obtida pela média dos nós vizinhos:

$$T_{ij} = \frac{1}{4}\left(T_{i+1,j} + T_{i-1,j} + T_{i,j-1} + T_{i,j+1}\right)$$

Para determinar a temperatura em cada nó, inicializamos a 50° todas as temperaturas não constantes na figura 3. Em seguida aplicamos a estimativa acima aos nós não constantes até que a diferença entre o valor de temperatura anterior e actual seja pequena. Nessa altura encontramos uma configuração estável.

Escreva um programa em FORTRAN 90 que determine a configuração estável da malha na figura 3, aplicando a estimativa até que a máxima diferença de temperatura em qualquer nó entre duas iterações seja inferior a $0.01°$. Qual é a temperatura estável no nó $(5, 5)$?

6.19 O *Jogo da Vida* inventado pelo matemático J. H. Conway desenrola-se num tabuleiro quadrado com n casas de lado; o jogo é mais interessante para tabuleiros grandes (*i.e.*, $n \geq 10$).

Cada casa do tabuleiro pode conter uma célula ou estar vazia. Consideram-se ainda *vizinhas* de cada casa as 8 casas mais próximas, segundo as direcções cardeais e diagonais (ver figura 4). Note-se ainda que as casas da fronteira do tabuleiro têm menos vizinhos.

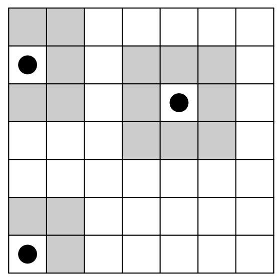


Figura 4: Casas vizinhas de algumas células num tabuleiro de 7×7.

Partindo duma configuração inicial do tabuleiro, o jogo desenrola-se por gerações: na passagem duma geração para outra algumas células morrem (*i.e.*, passam a uma casa vazia), outras mantêm-se e outras nascem (*i.e.*, numa casa vazia surge uma célula). As regras que regulam essa evolução são:

- uma célula morre se tiver menos de 2 ou mais do que 3 células vizinhas;
- se uma casa vazia tiver exactamente 3 células vizinhas então nasce uma nova célula no seu lugar.

Dependendo da configuração de células inicial, a colónia pode extinguir-se ao fim de poucas gerações, pode estabilizar ou mesmo aumentar indefinidamente.

Pretende-se escrever um programa em FORTRAN 90 que simule a evolução duma colónia: deve começar por distribuir aleatoriamente (usando a sub-rotina `random_number`) um certo número de células pelo tabuleiro; em seguida deve simular a passagem de gerações, desenhando o tabuleiro após cada nova geração.

O número de células colocadas inicialmente no tabuleiro bem como o número de gerações devem ser dados pelo utilizador.

NOTA: Tenha em atenção que a mudança de gerações deve ser efectuada simultaneamente em todas as células, o que se torna mais fácil se dispuser de dois tabuleiros: um com a situação actual e outro com a situação futura.

7 Saída Formatada

7.1 Assuma que as variáveis `a`, `b`, `c`, `i`, `j`, `k` são declaradas e inicializadas com:

```
REAL ::  a=-0.0001, b=6.02E23, c=3.141593
INTEGER ::  i=32767, j=24, k=-1010101
```

Indique qual o resultado das seguintes instruções:

(a) `WRITE(*,"(X,3F10.4)") a,b,c`

(b) `WRITE(*,"(X, F10.3, 2X, E10.3, 2X, F10.5)") a,b,c`

(c) `WRITE(*,"(X,3I8)") i,j,k`

(d) `WRITE(*,"(3(X,I8))") i,j,k`

(e) `WRITE(*,"(3(X,I2))") i,j,k`

(f) `WRITE(*,"(X,I5,X,I2,X,I8)") i,j,k`

(g) `WRITE(*,"(X,I8,2X,I8.8,2X,I8)") i,j,k`

7.2 Escreva um programa em FORTRAN 90 que escreve a tabuada dum algarismo dado (de 1 a 9). O programa deverá produzir um resultado com o formato seguinte:

```
Tabuada de?  7
1 * 7 = 7
2 * 7 = 14
3 * 7 = 21
4 * 7 = 28
5 * 7 = 35
6 * 7 = 42
7 * 7 = 49
8 * 7 = 56
9 * 7 = 63
10 * 7 = 70
```

7.3 Escreva um programa em FORTRAN 90 que escreve os seguintes "versos":

```
Numero de elefantes (>=1, <=99)?  10
Se 1 elefantes incomodam muita gente,
2 elefantes incomodam muito mais.
Se 2 elefantes incomodam muita gente,
3 elefantes incomodam muito mais.
⋮
Se 9 elefantes incomodam muita gente,
10 elefantes incomodam muito mais.
```

O "número de elefantes" é dado pelo utilizador . . . .

7.4 A aceleração da gravidade a uma altitude h sobre a superfície da Terra é dada pela expressão

$$g = -G\frac{M}{(R+h)^2}$$

onde $G \approx 6.672 \times 10^{11}$ Nm^2/Kg^2 é a constante de gravitação, $M \approx 5.98 \times 10^{24}$ Kg é a massa da Terra, $R \approx 6371$ Km é o raio médio da Terra e h é a altitude. Se M é medido em Kg e R e h em metros, a aceleração g vem em m/s^2.

Escreva um programa em FORTRAN 90 que tabela a aceleração da gravidade de uma altitude de 0 Km até 10,000 Km por incrementos de 500 Km. A tabela deve aparecer com o seguinte formato:

```
Altitude         Aceleração
    0 Km       -9.830 m/s^2
  500 Km       -9.828 m/s^2
 1000 Km       -9.827 m/s^2
       ⋮                  ⋮
```

7.5 Escreva um programa em FORTRAN 90 que escreva uma tabela de logaritmos na base 10 entre 1 e 10 por incrementos de 0.1. A tabela deverá ter o seguinte formato:

log10:	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9
1.	0.000	0.041	0.079	0.114	⋯					
2.	0.301	0.322	0.342	0.362	⋯					
3.	⋯									
4.	⋯									
⋮										
10.										

7.6 Modifique o programa do exercício 6.8 para gerar um quadrado mágico e escrevê-lo formatado. Escreva cada número no quadrado com 2 algarismos e um espaço de intervalo.

7.7 Escreva um programa em FORTRAN 90 que leia um valor total de segundos desde o início do dia (este valor estará entre 0 e $24 \times 3600 = 86400$ s) e o escreva na forma `HH:MM:SS`, onde `HH` é a hora (0–23), `MM` o minuto (0–59) e `SS` o segundo (0–59).

SUGESTÕES: Para decompor os segundos em horas e minutos pode usar a divisão inteira por 60. Ao escrever, use o especificador de formato `I`$w.m$ para introduzir zeros à esquerda nos campos `MM` e `SS`.