

Computação 2026/2027

Filipe Rodrigues (frodrigues@iseg.ulisboa.pt)

Exercícios

Guião 1 - Variáveis e Operadores

1. Crie um macro com o nome "First" que quando executado apresente a seguinte mensagem ao utilizador: "Parabéns! O teu primeiro programa está a funcionar corretamente!!". Crie de seguida um botão de execução no excel, associe o macro criado a esse botão e execute-o.
2. Escreva um programa que peça ao utilizador o seu nome, a sua idade e a sua data de nascimento e escreva uma mensagem de parabéns com esses dados. Por exemplo, se o nome do utilizador for "Pedro", se ele tiver "23" anos e a data for 12/1/2000, o programa deve apresentar a seguinte mensagem:
"Olá Pedro, bem-vindo.
O teu aniversário é dia 12/1/2000. Parabéns pelos teus 23 anos."
Nota: Use variáveis do tipo adequado.
3. Escreva um programa que peça ao utilizador um número inteiro e outro decimal. Crie três novas variáveis para guardar a soma, subtração e multiplicação dos valores introduzidos e imprima no ecrã uma mensagem com essa informação.
4. Escreva um programa que permita determinar a área e o perímetro de um retângulo. O programa deve pedir ao utilizador a altura e o comprimento do retângulo e devolver a sua área e o seu perímetro.
5. Escreva um programa que peça ao utilizador 4 valores inteiros da velocidade de um carro e imprima a sua média.
6. Escreva um programa que leia as notas de 4 testes de avaliação e do exame final de um aluno numa disciplina e que calcule a classificação final dessa disciplina. Assuma que cada um dos 4 testes vale 10% e o exame os restantes 60%.
7. Escreva um programa que receba dois números inteiros a e b e mostre o resultado da sua divisão decimal. O programa deve também escrever a divisão dos dois números na forma " $a=\text{quociente}*b+\text{resto}$ ". Por exemplo, se o utilizador introduzir os valores $a = 8$ e $b = 5$ o programa deve escrever:

Divisao inteira: $8=1*5+3$

Divisao decimal: $8/5=1.6$

-
8. Escreva um programa que leia um número inteiro representando uma quantidade de dias e o converta para semanas e dias.
Exemplo: 17 dias = 2 semanas e 3 dias.
9. Escreva um programa que converta segundos para horas/minutos/segundos.
Exemplo: 4000 segundos = 1h6m40s.
10. Escreva um programa que peça ao utilizador um número inteiro de 3 dígitos e escreva esse número pela ordem inversa. Sugestão: Use os operadores da divisão.

Guião 2 - Estruturas de controlo

11. Escreva um programa que receba como input o peso (em kg) e a altura (em metros) de uma pessoa e valide esses dados. Caso algum dos valores seja negativo, o programa deve apresentar uma mensagem de erro e terminar. Caso os valores introduzidos sejam válidos, o programa deve calcular o IMC, classificando a pessoa segundo a escala abaixo. Nota: $IMC = \text{peso} / \text{altura}^2$

IMC	Classificação
<16.9	Muito abaixo do peso
[16.9 , 18.5[	Abaixo do peso
[18.5 , 25[	Peso normal
[25 , 30[	Acima do peso
> 30	Obesidade

12. Escreva um programa (usando o controlador Select Case) que receba como input um número inteiro e apresente um comentário de acordo com a tabela:

Valor	Comentário
<0	Número negativo
1	Nem é primo nem é composto
2,3	Número primo
outros	Não sei se é primo ou não...

13. Numa determinada disciplina existem 5 momentos de avaliação: 4 exercícios e um exame. Para o cálculo da nota final de um aluno são contabilizados apenas os 3 melhores exercícios e o exame final, sendo que o exame vale 50% e os exercícios os restantes 50%. Além disso, existe uma nota mínima de 7 valores no exame para obtenção de aprovação à disciplina. Escreva um programa que permita determinar a classificação final da disciplina e que valide todas as notas introduzidas.
14. Escreva um programa que permite resolver equações de grau menor ou igual a dois.

-
15. Escreva um programa que verifique se um determinado ano é bissexto. Nota: um ano é bissexto quando for múltiplo de 4 e não for múltiplo de 100 ou quando for múltiplo de 400. Teste o programa com os anos 2000 (bissexto), 1900 (não bissexto) e 1996 (bissexto).
16. Escreva um programa que verifique se uma determinada data é válida. A data deve ser pedida ao utilizador através de três inteiros.
17. Escreva um programa que peça ao utilizador a sua data de nascimento e a data atual (sem as validar) e calcule a sua idade.
18. Escreva um programa que receba três números reais e verifique se eles podem ser as medidas dos lados de um triângulo retângulo. Sugestão: use o teorema de pitágoras.
19. Execute as seguintes instruções:
- i. Imprima todos os números desde 1 até 15, conforme apresentado abaixo, usando um ciclo.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 - ii. Imprima todos os números desde 15 até 1 usando um ciclo.
 - iii. Faça um ciclo começado em $i=3$ que execute 12 iterações e que em cada iteração escreva um sinal "+" no ecrã.
 - iv. Peça um número inteiro n_1 ao utilizador. Se esse número for positivo, imprima todos os números desde 0 até n_1 . Caso contrário, imprima o número zero $|n_1|$ vezes.
 - vi. Imprima todos os números pares entre 10 e 50 (inclusivé) de duas formas distintas: a primeira usando um ciclo e uma instrução *if*, e a segunda usando apenas um ciclo.
 - vii. Calcule a soma dos n_3 primeiros números naturais usando um ciclo, sendo n_3 pedido ao utilizador. Exemplo: para $n_3 = 10$, soma = 55.
 - viii. Peça dois valores inteiros a e b ao utilizador tais que $a < b$ e calcule a soma dos números entre a e b (inclusivé).
 - ix. Escreva os quadrados dos 10 primeiros números naturais.
 - x. Peça um número inteiro positivo menor que 10 ao utilizador e calcule o seu fatorial.
 - xi. Calcule a soma dos números entre 1 e 100 que são múltiplos de 2 mas não de 4.
 - xii. Indique quantos números múltiplos de 6 existem entre 100 e 200.
 - xiii. Peça um inteiro m ao utilizador e escreva o output abaixo. O output corresponde ao caso $m = 6$.

```
Inicio...
1 [0, 2]
2 [1, 3]
3 [2, 4]
4 [3, 5]
5 [4, 6]
6 [5, 7]
...Fim!
```

-
20. Escreva um programa que permita determinar o valor da seguinte soma (para um dado valor n pedido):

$$S_n = -\frac{1}{2} + \frac{1}{5} - \frac{1}{10} + \dots + \frac{(-1)^n}{n^2 + 1}$$

21. Escreva um programa que peça ao utilizador um número inteiro positivo e calcule o seu número de divisores.
22. Escreva um programa que peça ao utilizador um número inteiro positivo e imprima todos os seus divisores numa única msgbox.
23. Escreva um programa que peça ao utilizador um número inteiro positivo e verifique se ele é um número primo.
24. Escreva um programa que peça ao utilizador um número inteiro positivo e verifique se ele é um número perfeito. Nota: um número é perfeito se for igual à soma dos seus divisores próprios. Por exemplo, 6 é um número perfeito pois os seus divisores próprios são 1, 2 e 3 e $1+2+3=6$.
25. Escreva um programa que imprima todos os números perfeitos entre 1 e 500.
26. Escreva um programa para determinar o máximo divisor comum entre dois inteiros.
27. Escreva um programa que receba um número inteiro positivo e calcule a soma desde 1 até esse número apresentando a soma por extenso. O programa deve ainda escrever a soma apenas dos números ímpares. Por exemplo, se o valor introduzido for 10, o programa deve escrever:

$$1+2+3+4+5+6+7+8+9+10=55$$

$$1+3+5+7+9=25$$

28. Escreva um programa que indique qual o número inteiro positivo inferior a n que tem mais divisores, sendo n pedido ao utilizador.
29. Escreva um programa que peça sucessivamente números inteiros não negativos ao utilizador. O programa deve terminar quando o utilizador inserir um número inteiro negativo e nessa altura deverá apresentar o mínimo, o máximo e a média dos valores positivos inseridos pelo utilizador.

-
30. Escreva um programa que peça sucessivamente números inteiros ao utilizador. O programa deve terminar quando forem introduzidos 5 números pares e nessa altura deve apresentar a soma de todos os números ímpares introduzidos.
31. Escreva um programa que peça sucessivamente números inteiros ao utilizador. O programa deve terminar quando forem introduzidos consecutivamente dois números iguais e nessa altura deve indicar quantos números foram introduzidos.
32. Escreva um programa que peça um número inteiro positivo n ao utilizador e indique qual o n -ésimo número primo.
33. Peça um inteiro positivo k ao utilizador (fazendo a sua validação) e escreva os outputs abaixo. Os outputs apresentados como exemplo correspondem ao caso $k = 6$.

Output 1

```
Inicio...
1 -> 0
2 -> 1 0
3 -> 2 1 0
4 -> 3 2 1 0
5 -> 4 3 2 1 0
6 -> 5 4 3 2 1 0
...Fim!
```

Output 2

```
Inicio...
1 -> Soma desde 1 ate 1 = 1
2 -> Soma desde 1 ate 2 = 3
3 -> Soma desde 1 ate 3 = 6
4 -> Soma desde 1 ate 4 = 10
5 -> Soma desde 1 ate 5 = 15
6 -> Soma desde 1 ate 6 = 21
...Fim!
```

Output 3

```
6 5 4 3 2 1
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```

Output 4

```
1*****
12*****
123*****
1234***
12345**
123456*
1234567
```

34. Implemente o jogo do *hi-lo*. Primeiro, o programa deve gerar um número aleatório entre 1 e 100. Ao utilizador são dadas 7 tentativas para adivinhar o número gerado pelo computador.

Se o utilizador não adivinhar o número correto, o programa deve dizer ao utilizador se o seu palpite está abaixo ou acima do número correto. Se o utilizador acertou no número correto, o programa deve dizer ao utilizar que ganhou. Se o utilizar não consegue adivinhar o número correto ao fim das 7 tentativas o programa deve dizer-lhe que perdeu e indicar qual o número correto.

Nota: Para gerar um número inteiro aleatório entre 1 e 100 deve usar os seguintes comandos:

```
Dim numeroAleatorio As Integer
```

```
numeroAleatorio = Int(99 * Rnd) + 1.
```

Exemplo:

Tens 7 tentativas para o adivinhar o numero em que estou a pensar.

```
Tentativa #1: 50
O numero introduzido e muito baixo.

Tentativa #2: 75
O numero introduzido e muito baixo.

Tentativa #3: 87
O numero introduzido e muito alto.

Tentativa #4: 80
O numero introduzido e muito alto.

Tentativa #5: 77
O numero introduzido e muito baixo.

Tentativa #6: 79
Correto! Ganhou!
```

35. Escreva um programa interativo que apresente o seguinte menu ao utilizador:

```
-----  
|Brincar com números|  
| 1. Inserir número novo|  
| 2. Mostrar mínimo|  
| 3. Mostrar média|  
| 4. Mostrar máximo|  
| 5. Maior diferença absoluta entre números consecutivos|  
| 6. Terminar programa|  
| Opção:|  
-----
```

Cada vez que uma determinada opção é escolhida, a respectiva ação deve ser executada.

36. (Para corajosos...) Escreva um programa que peça ao utilizador um mês, um ano, e o dia da semana em que esse mês começa e escreva o calendário desse mês. Exemplo: Ao introduzir o ano 2023, o mês 3 e o dia da semana quarta, o programa deverá imprimir o seguinte output:

```
|S |T |Q |Q |S |S |D |  
| | |1 |2 |3 |4 |5 |  
|6 |7 |8 |9 |10|11|12|  
|13|14|15|16|17|18|19|  
|20|21|22|23|24|25|26|  
|27|28|29|30|31| | |
```

Guião 3 - Variáveis Indexadas

37. Escreva um programa que permita criar os seguintes vetores de forma automática. Depois de criar cada um dos vetores, imprima-o no ecrã.
- i) $v = (1, 2, 3, \dots, 25)$;
 - ii) $u = (2, 4, 6, \dots, 50)$;
 - iii) $z = (0, 1, 0, 1, \dots, 1)$ (vetor com dimensão 30);
 - iv) $y = (30, 40, 50, \dots, 200)$
 - v) $w = (1, 0.5, 0.25, 0.125, \dots)$ (vetor com dimensão 15);
38. Escreva um programa que peça um vetor de inteiros ao utilizador. O programa deve primeiro pedir a dimensão do vetor e só depois os seus elementos. Altere depois o vetor obtido de modo que cada coordenada seja o triplo da coordenada inicial e imprima o vetor modificado.
39. Escreva um programa para determinar a soma dos elementos de um vetor. O programa deve primeiro pedir ao utilizador que indique a dimensão do vetor e só depois ler os seus elementos.
40. Escreva um programa para determinar a norma de um vetor. O programa deve primeiro pedir ao utilizador que indique a dimensão do vetor e só depois ler os seus elementos.
41. Escreva um programa que peça um vetor de 10 inteiros ao utilizador. Após receber o vetor, e sem nunca o ordenar, o programa deve fazer as ações abaixo. Use um ciclo for para cada alínea e não um ciclo for para todas.
- i) Imprimir o vetor;
 - ii) indicar quantos números pares existem no vetor;
 - iii) calcular o valor mínimo do vetor;
 - iv) determinar o índice (posição) do primeiro número par existente no vetor. Caso não existam números pares no vetor, deverá ser indicada a posição -1;
 - v) determinar o índice do último número par existente no vetor. Caso não existam números pares no vetor, deverá ser indicada a posição -1;
 - vi) calcular o último índice do valor máximo do vetor;
 - vii) verificar se existem valores negativos no vetor;
 - viii) verificar se existem elementos repetidos no vetor;
 - ix) criar um novo vetor que contenha apenas os números ímpares presentes no vetor inicial;
 - x) Verificar quantas vezes é que em posições consecutivas do vetor existem números pares.
42. Escreva um programa que peça sucessivamente valores inteiros positivos ao utilizador e os guarde num vetor. O programa deve parar de pedir valores quando for introduzido o valor zero. Após terminar a leitura dos valores, o programa deve calcular e apresentar a soma dos valores introduzidos. Nota: Uma vez que não sabe quantos elementos serão inseridos à priori deve fazer o redimensionamento do vetor cada vez que for inserido um novo valor.

43. Escreva um programa para determinar o produto interno entre dois vetores. O programa deve primeiro pedir a dimensão dos dois vetores e só depois ler os seus elementos.

44. Pretende-se escrever um programa que use como base um vetor de números reais. Comece por pedir ao utilizador quantos elementos quer colocar no vetor. Crie um vetor com esse número de elementos e preencha-o pedindo os valores ao utilizador. De seguida, o programa deve apresentar um menu interativo que permita ao utilizador:

- i) Imprimir o vetor;
- ii) Mostrar o vetor escrito por ordem inversa;
- iii) Calcular a norma do vetor;
- iv) Calcular a média dos valores contidos no vetor;
- v) Verificar se existem elementos repetidos no vetor;
- vi) Verificar se um elemento (pedido ao utilizador) está no vetor;
- vii) Mostrar o vetor ordenado por ordem crescente;
- ix) Adicionar um novo elemento (pedido ao utilizador) ao vetor;
- x) Pedir um valor i ao utilizador e remover o i -ésimo elemento do vetor;
- xi) Pedir um valor x ao utilizador e remover a primeira ocorrência do elemento x do vetor;
- xii) Pedir um valor x ao utilizador e remover todas as ocorrências do elemento x do vetor;
- xiii) Pedir um valor i ao utilizador e remover o i -ésimo número par do vetor, se ele existir;
- xiv) Abandonar o programa.

45. Escreva um programa que peça 7 strings ao utilizador e as guarde num vetor de strings. O programa deverá depois alterar o vetor de strings criado de modo a que todas as strings com mais do que 5 caracteres sejam substituídas pela letra G e as strings com 5 ou menos caracteres sejam substituídas pela letra P. Sugestão: a função $Len(s)$ indica o número de caracteres da string s .

Exemplo: Ao introduzir o vetor (pao, arroz, batata, farinha, massa, queijo, chocolate) o programa deve criar e escrever o vetor (P, P, G, G, P, G, G).

46. Escreva um programa que peça dois vetores de inteiros de dimensão 7 ao utilizador, fazendo a validação dos seus valores. O primeiro vetor representa as notas de 7 alunos no primeiro teste de uma cadeira e o segundo representa as notas desses mesmos alunos no segundo teste. De seguida:

- Indique em qual dos testes a média de notas foi superior;
- Calcule qual a maior subida e a maior descida do primeiro para o segundo teste;
- Indique se algum aluno manteve a sua nota em ambos os testes;
- Indique quantos alunos chumbaram no primeiro teste;
- Verifique se algum aluno chumbou no primeiro teste e passou no segundo;
- Indique se houve mais alunos a melhorar ou a piorar do primeiro para o segundo teste;
- Verifique se algum aluno teve negativa em ambos os testes;

-
- Verifique se uma determinada nota (pedida ao utilizador) existe quer no primeiro quer no segundo teste;
 - Verifique se algum aluno teve no primeiro teste uma nota igual à nota que outro aluno teve no segundo teste;
 - Crie um novo vetor que contenha os elementos dos dois vetores ordenados.
47. Escreva um programa que peça um vetor de inteiros positivos ao utilizador com dimensão 10. Após receber o vetor, o programa deve indicar quantos e quais os números primos que existem no vetor (sem os repetir).
48. Escreva um programa que peça um vetor de inteiros ao utilizador e que crie um novo vector com os valores normalizados, isto é, com cada entrada do vetor a ser dividida pelo máximo dos elementos do vetor. Escreva o vetor inicial e o vetor normalizado para o ecrã.
49. Escreva um programa que peça um vetor de inteiros ou utilizador. De seguida:
- i) Imprima o vetor;
 - ii) Remova do vetor todos os elementos múltiplos de 3;
 - iii) Incremente uma unidade em todos os elementos primos existentes no vetor;
 - iv) Crie um novo vetor que contenha apenas os números pares existentes no vetor original;
 - v) Troque todos os elementos maiores que 5 pelo número primo seguinte mais próximo. Por exemplo, se existir no vetor o número 8, ele será substituído pelo número 11.
50. Para um dado n impar pedido ao utilizador crie e imprima as seguintes matrizes:
- $$\begin{array}{llll}
 i. & \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 2 & 2 & \cdots & 2 \\ \vdots & \vdots & \ddots & \vdots \\ n & n & \cdots & n \end{bmatrix} & ii. & \begin{bmatrix} 1 & 0 & \cdots & 1 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 1 \end{bmatrix} & iii. & \begin{bmatrix} a & * & \cdots & * \\ 0 & a & \cdots & * \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a \end{bmatrix} & iv. & \begin{bmatrix} 2 & 3 & \cdots & n+1 \\ 3 & 4 & \cdots & n+2 \\ \vdots & \vdots & \ddots & \vdots \\ n+1 & n+2 & \cdots & 2n \end{bmatrix}
 \end{array}$$
51. Escreva um programa para calcular a soma de duas matrizes. Comece por pedir ao utilizador a dimensão das matrizes e em seguida os seus elementos.
52. Escreva um programa que verifique se todos os elementos de uma qualquer matriz pedida ao utilizador são positivos.

53. Escreva um programa que verifique se uma matriz é simétrica. Comece por pedir ao utilizador a dimensão da matriz e em seguida os seus elementos. Nota: Uma matriz A é simétrica se para todas as suas entradas se verificar $a_{ij} = a_{ji}$.

54. Escreva um programa que leia uma matriz quadrada A de dimensão $n \times n$, e um vetor $\vec{v}$ de dimensão n . O programa deve depois imprimir o resultado da operação matricial $A\vec{v}$.

Exemplo:

$$A \times \vec{v} = \begin{bmatrix} 1 & 3 & 5 \\ 7 & 8 & 9 \\ 13 & 4 & 2 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 9 \\ 24 \\ 19 \end{bmatrix}$$

55. Escreva um programa onde comece por criar um vetor de dimensão 100 com o nome *vetorAleatorio* e preencha-o com números inteiros aleatórios entre 0 e 20 (ver nota abaixo). Tendo por base o vetor criado, crie depois um novo vetor *histograma* que na i -ésima posição contém o número de valores iguais a i do vetor *vetorAleatorio*. Imprima o histograma no ecrã.

Nota: Para gerar um vetor de tamanho 100 com números inteiros aleatórios entre 0 e 20 deve usar os seguintes comandos:

```
Dim vetorAleatorio(1 To 100) As Integer
For i = 1 To 100
    vetorAleatorio(i) = (Round(Rnd() * 20))
Next i
```

Exemplo: O programa deve imprimir o seguinte *output*:

```
Nota | F |
0    | 4 | ****
1    | 5 | *****
2    | 8 | *********
3    | 5 | *****
4    | 5 | *****
5    | 1 | *
6    | 6 | *****
7    | 8 | *********
8    | 5 | *****
9    | 0 |
10   | 4 | ****
11   | 3 | ***
12   | 3 | ***
13   | 5 | *****
14   | 5 | *****
15   | 6 | *****
16   | 2 | **
17   | 5 | *****
18   | 2 | **
19   | 2 | **
20   | 6 | *****
```

-
56. Escreva um programa com menu interativo que permita simular o sistema de atendimento do MacDonalds. Cada novo pedido deve ser identificado pelo nome do cliente (não é necessário saber qual o produto pedido)

```

-----
|Pedidos MacDonalds|
|  1. Inserir novo pedido  |
|  2. Remover pedido       |
|  3. Número de pedidos por servir  |
|  4. Número de pessoas servidas  |
|  5. Ordenar pedidos por ordem alfabética  |
|  6. Terminar programa  |
|  Opção:                  |
-----

```

57. Escreva um programa que peça ao utilizador uma matriz 4x4 e que verifique se os elementos existentes nas primeiras duas linhas são os mesmos que existem nas duas últimas linhas (eventualmente numa ordem diferente). Isto acontece por exemplo na matriz

$$A = \begin{bmatrix} 1 & 3 & 5 & 0 \\ 7 & 8 & 9 & 0 \\ 3 & 0 & 0 & 7 \\ 9 & 8 & 1 & 5 \end{bmatrix}$$

58. Escreva um programa para calcular o produto de duas matrizes. Comece por pedir ao utilizador a dimensão das matrizes, validá-las, e em seguida pedir os seus elementos. Nota: dadas duas matrizes $A_{n \times p}$ e $B_{p \times m}$, temos $C_{ij} = \sum_{k=1}^p A_{ik} \times B_{kj}$.

Guião 4 - Funções

59. Implemente as funções/procedimentos abaixo (que deverá chamar no programa principal). Cada função recebe três números inteiros.

- Uma função que devolva o máximo dos valores introduzidos;
- Uma função que devolva o mínimo dos valores introduzidos;
- Uma função que devolva a média dos valores introduzidos;
- Um procedimento que escreva os valores por ordem decrescente;
- Uma função booleana que verifique se algum dos três números é repetido;
- Uma função que devolva a diferença entre o valor máximo e mínimo introduzidos. Não é permitida a utilização de instruções condicionais na implementação desta função.

60. Implemente as seguintes funções/procedimentos:

- Uma função que receba um vetor e o devolva escrito na forma de string.
- Uma função que receba um vetor de inteiros e calcule o máximo desse vetor;
- Uma função que receba um vetor de inteiros e calcule a média dos seus elementos;
- Uma função booleana que receba um vetor de inteiros e verifique se nele existem números primos;
- Uma função que receba um vetor de inteiros e devolva um novo vetor contendo apenas os seus elementos pares;
- Um procedimento que receba um vetor de inteiros e substitua todos os seus elementos ímpares por zero;

61. Escreva uma função que receba um número inteiro positivo e retorne o seu número de divisores. Crie depois um programa que chame a função criada.

62. Escreva um procedimento que receba um número inteiro positivo e escreva todos os divisores desse número por ordem decrescente. Crie depois um programa que chame a função criada.

63. Escreva um programa que permita fazer operações com matrizes. O programa deve permitir:

- i) somar matrizes;
- ii) subtrair matrizes;
- iii) multiplicar matrizes;
- iv) multiplicar matrizes por escalares;

Não se esqueça de validar as dimensões das matrizes introduzidas pelo utilizador.

64. Escreva duas funções que calculem o fatorial de um inteiro positivo n , sendo uma delas iterativa e a outra recursiva. Crie depois um programa que chame ambas as funções e escreva os respectivos resultados.

65. A sucessão de Fibonacci é dada por $F_n = F_{n-1} + F_{n-2}$ para $n > 2$, com $F_1 = F_2 = 1$. Escreva uma função iterativa e outra recursiva que permitam determinar o n -ésimo número da sequência de Fibonacci.

66. A divisão entre dois números de Fibonacci consecutivos tende para a razão de ouro $\frac{1+\sqrt{5}}{2}$, isto é,

$$\lim_{n \rightarrow \infty} \frac{F_{n+1}}{F_n} = \frac{1 + \sqrt{5}}{2}$$

Usando esta informação, escreva um programa que determine qual o menor valor de n que garante $|\frac{F_{n+1}}{F_n} - \frac{1+\sqrt{5}}{2}| < \epsilon$, sendo ϵ um valor pedido ao utilizador. Exemplo: Para $\epsilon = 0,001$ teríamos $n = 9$.

67. Considere a sucessão abaixo onde x um número pedido ao utilizador. Escreva uma função que calcule o n -ésimo termo desta sucessão.

$$x_0 = x \text{ e } x_{n+1} = \begin{cases} x_n/2, & \text{se } x_n \text{ é par,} \\ 3x_n + 1, & \text{se } x_n \text{ é ímpar,} \end{cases} \quad n \geq 0,$$

68. Escreva uma função booleana recursiva que verifique se um vetor tem algum elemento nulo.

69. Escreva uma função recursiva que calcule a soma de um vetor de inteiros.

70. (Para corajosos...) Escreva uma função recursiva que escreva os divisores de um número inteiro positivo e os divisores de cada um dos seus divisores. Por exemplo, para o valor 12, deve ser escrito o seguinte output:

```

12
  6
    3
      1
        2
          1
            4
              2
                1
                  3
                    1
                      2
                        1
                          1

```

71. (Para corajosos...) Pretende-se implementar o jogo do galo. Para tal, execute as instruções abaixo.

- Crie um módulo independente e nele defina uma variável global para representar uma matriz 3x3.
- Implemente uma função que verifique se uma linha, coluna ou diagonal da matriz está preenchida com o mesmo símbolo ($\times$ ou o). Se tal acontecer, a função deve devolver esse símbolo, caso contrário deve devolver o símbolo $-$;
- Uma função que receba o número da linha, o número da coluna e um símbolo e coloque esse símbolo na respetiva posição da matriz, caso tal seja possível;
- Uma função que verifique se a matriz de jogo está completamente preenchida;
- Um procedimento que imprima a matriz de jogo;

De seguida, crie um procedimento principal que servirá de base ao jogo do Galo. O primeiro jogador ficará com o símbolo $\times$ e o segundo com o símbolo o . Nesse procedimento deve jogar alternadamente cada um dos jogadores, começando pelo primeiro jogador. Na vez de cada jogador, deve ser pedido ao utilizador o número da linha e coluna onde ele pretende jogar. Após cada jogada, deve ser impressa a matriz de jogo. O jogo termina quando um dos jogadores vencer o jogo ou quando todo o tabuleiro de jogo estiver preenchido.

Guião 5 - Classes e objetos

72. Crie uma classe **Quadrado** para representar um quadrado. A classe deve ter um atributo público que seja o lado do quadrado e dois métodos para calcular a sua área e o seu perímetro. No programa principal, crie dois **Quadrados**, um com lado 3 e outro com lado 5, e calcule a diferença absoluta entre as áreas e perímetros desses **Quadrados**.

73. Crie uma classe **Circunferencia** com atributos privados para representar uma circunferência. A classe deve calcular o perímetro da circunferência e a sua área. Deve também permitir definir as coordenadas (x_0, y_0) do centro da circunferência e escrever a sua equação. Implemente também um método que permita verificar se um determinado ponto (x, y) se encontra dentro, fora, ou na fronteira da circunferência. Nota: A área da circunferência é $A = \pi r^2$, o seu perímetro é $P = 2\pi r$ e a sua equação é $(x - x_0)^2 + (y - y_0)^2 = r^2$. Pode aceder ao valor de π usando a instrução *WorksheetFunction.Pi()*. A classe deve funcionar com o procedimento principal abaixo:

```
Sub teste()  
    Dim c As Circunferencia  
    Set c = New Circunferencia  
  
    Call c.Raio(5)  
    Call c.Centro(2, -5)  
  
    Dim x As Double  
    Dim y As Double  
    x = InputBox("x: ")  
    y = InputBox("y: ")  
  
    MsgBox ("Area: " & c.Area & vbCrLf & _  
        "Perimetro: " & c.Perimetro & vbCrLf & _  
        "Equacao: " & c.Equacao & vbCrLf & _  
        "O ponto (" & x & ", " & y & ") esta " & c.Posicao(x, y) & _  
        " da circunferencia ")  
End Sub
```

74. Crie uma classe para representar uma pessoa com os atributos privados: nome, idade, altura e peso. Crie um método público que permita preencher os dados de uma pessoa, outro que permita imprimir os dados da pessoa e outro para calcular o seu índice de massa corporal $IMC = \text{peso}/(\text{altura}^2)$. Teste a classe no programa principal de forma adequada.

75. Crie uma classe **Vetor** para representar um vetor (x, y, z) . A classe deve usar três valores reais como atributos públicos, deve permitir definir as coordenadas do vetor, imprimir as suas coordenadas e calcular a sua norma. Após definir a classe, crie no programa principal dois objetos do tipo **Vetor** e calcule o seu produto interno. Calcule ainda a norma de cada **Vetor**.

76. Crie uma classe para representar polinômios de segundo grau, ou seja, $ax^2 + bx + c$. Deve usar apenas um atributo na classe. Escreva depois um método que permita alterar um coeficiente do polinômio, outro que devolva um coeficiente do polinômio e outro que permita imprimir o polinômio. A classe deve também ter um método para determinar as raízes do polinômio. Após definir a classe, crie dois objetos do tipo polinômio e calcule a sua soma.

```
Sub teste()  
    Dim s As String  
    Dim p1 As Polinomio  
    Set p1 = New Polinomio  
    Call p1.AlterarCoef(1,5)  
    Call p1.AlterarCoef(2,3)  
    Call p1.AlterarCoef(3,7)  
    s = "p1: " & p1.Imprime & VbCrLf  
  
    Dim p2 As Polinomio  
    Set p2 = New Polinomio  
    Call p2.AlterarCoef(1,1)  
    Call p2.AlterarCoef(2,1)  
    Call p2.AlterarCoef(3,1)  
    s = s & "p2: " & p2.Imprime & VbCrLf  
    s = s & "Raizes de p2: " & p2.Raizes & VbCrLf  
  
    Dim psoma As Polinomio  
    Set psoma = New Polinomio  
  
    'Escreva código para definir psoma como a soma de p1 e p2  
    '...  
  
    s = s & "psoma: " & psoma.imprime & VbCrLf  
    MsgBox (s)  
End Sub
```

-
77. Crie uma classe que permita representar frações. A classe deve ter um método privado para calcular o máximo divisor comum entre dois números. Além disso, deve ter um método público para escrever uma fração e outro método também público para simplificar uma fração (usando o método privado implementado). O programa deve funcionar com o seguinte procedimento de teste.

```
Sub teste()
    Dim s As String
    Dim f1 As Fracao
    Dim f2 As Fracao
    Set f1 = New Fracao
    Set f2 = New Fracao

    s = "f1: " & f1.Imprimir & vbCrLf      '-> escreve 0
    Call f1.Definir(8, 24)                 '-> cria fração 8/24
    Call f1.Simplificar                    '-> converte f1 para 1/3
    s = s & "f1: " & f1.Imprimir & vbCrLf '-> escreve 1/3

    Call f2.Definir(10, 12)                 '-> cria fração 10/12

    'Soma:
    Dim soma As fracao
    Set soma = New fracao
    soma = '...
    '...
    s = s & "soma: " & soma.Imprimir & vbCrLf '-> escreve 7/6

    'Subtração:
    '...

    'Multiplicação:
    '...

    'Divisão:
    '...

    MsgBox (s)
End Sub
```

Guião 6 - Vetores de Classes e Objetos

78. Execute as seguintes instruções:

- (a) Copie a table seguinte para uma folha do excel com o nome "Inicio"

Nome	Idade	Filhos	Filhas
Gervásio	30	1	1
Clutilde	37	2	0
Gilmario	31	0	0
Flausina	45	1	3
Aurora	27	0	1
Idalécio	22	1	0

- (b) Guarde o ficheiro excel como livro com permissão para macros.
- (c) Crie uma classe **Pessoa** que tenha como atributos o nome da pessoa, a sua idade, o número de filhos e o número de filhas.
- (d) Defina na classe uma função **Imprime** que devolva uma string com a informação sobre a pessoa na forma:

Nome: ...
Idade: ...
Filhos: ...

- (e) Crie um modulo com o nome **Principal** e dentro dele um procedimento com o nome **Teste**. Nele comece por declarar um vetor de objetos do tipo **Pessoa**.
- (f) Crie uma segunda folha do excel com o nome "Resultados" usando a instrução: *Worksheets.Add Before:=Worksheets(Worksheets.Count) e ActiveSheet.Name = "Resultados"*
- (g) No modulo **Principal** preencha o vetor que criou com a informação que está na folha "Inicio". Para tal, use uma das instruções: *Application.Cells(... , ...)* ou *ActiveSheet.Range("...").Value*. Para garantir que está a ler os dados da folha correta, comece por tornar a folha "Inicio" a folha ativa do excel. Para isso use a instrução: *Worksheets("...").Activate*
- (h) Imprima cada elemento do vetor que criou usando a função **Imprime** que definiu na classe.
- (i) Mude o nome da folha "Inicial" para "Dados" usando a instrução *ActiveSheet.Name = "Dados"*
- (j) Na folha "Resultados" escreva (de forma automática) uma nova tabela com 3 colunas. A primeira coluna deve ter o nome de cada pessoa. A segunda deve ter o número total de filhos. A terceira deve ter uma mensagem que depende do número total de filhos. Se o número de filhos for 0, a mensagem deve ser "Já está na altura!". Caso o número de filhos seja 1, a mensagem deve ser "Não mexas mais!". Caso o número de filhos seja maior que 1, a mensagem deve ser "Nunca terá descanso!".

-
- (k) Escreva uma message box na qual indique qual é a pessoa mais velha da lista, qual a que tem mais filhos e quantas pessoas têm mais do que 2 filhos.
 - (l) Use a instrução *Worksheets("...").Delete* para apagar a folha "Resultados".
 - (m) Use a instrução *Application.ActiveWorkbook.Save* para guardar as alterações que efetuou.
 - (n) Use a instrução *Application.Quit* para fechar o Excel. TENHA A CERTEZA que o ficheiro foi previamente guardado como livro com permissão para macros pois caso contrário perderá tudo!
 - (o) Faça uma cópia do ficheiro excel em que estava a trabalhar. Na cópia que criou, altera os atributos da classe de modo que o número de filhos e filhas seja guardado num vetor com duas posições em vez de serem guardados um em cada inteiro. Altera o que for necessário no programa para que tudo funcione após esta modificação.