

## OPERADORES ARITMÉTICOS

		$A \leftarrow 5$	
		$B \leftarrow 2$	
+	Adição	$A + B$	7
-	Subtração	$A - B$	3
*	Multiplicação	$A * B$	10
/	Divisão	$A / B$	2.5
\	Divisão Inteira	$A \setminus B$	2
^	Exponenciação	$A ^ B$	25
%	Módulo	$A \% B$	1

$$\begin{array}{r} 5 \overline{) 2} \\ 1 \quad 2 \end{array}$$



## ORDEM DE PRECEDÊNCIA

( )	Parênteses
^	Exponenciação
* /	Multiplicação / Divisão
+ -	Adição / Subtração

$$3 + 2 / 2 = 4$$

$$(3 + 2) / 2 = 2.5$$



<pre> algoritmo "Soma" // Função : // Autor : // Data : 20/01/2022 // Seção de Declarações var     N1, N2, S: inteiro inicio     // Seção de Comandos     Escreva ("Digite um número: ")     Leia (N1)     Escreva ("Digite outro número: ")     Leia (N2)     S &lt;- N1 + N2     Escreva ( "A soma entre ", N1, " e ", N2, " é igual a ", S)  finalgoritmo </pre>				^ Início da execução Digite um número: 2 Digite outro número: 5 A soma entre 2 e 5 é igual a 7 Fim da execução.
GLOBAL	N1	I	2	
GLOBAL	N2	I	5	
GLOBAL	S	I	7	

<pre> algoritmo "Média" // Função : // Autor : // Data : 20/01/2022 // Seção de Declarações var     N1, N2: inteiro     M: Real inicio     // Seção de Comandos     Escreva ("Digite um número: ")     Leia (N1)     Escreva ("Digite outro número: ")     Leia (N2)     M &lt;- (N1 + N2) / 2     Escreva ( "A média entre ", N1, " e ", N2, " é igual a ", M)  finalgoritmo </pre>				^ Início da execução Digite um número: 2 Digite outro número: 3 A média entre 2 e 3 é igual a 2.5 Fim da execução.
GLOBAL	N1	I	2	
GLOBAL	N2	I	3	
GLOBAL	M	R	2.5	

# FUNÇÕES ARITMÉTICAS

<b>Abs</b>	Valor Absoluto	<b>Abs(-10)</b>	<b>10</b>
<b>Exp</b>	Exponenciação	<b>Exp(3,2)</b>	<b>9</b>
<b>Int</b>	Valor Inteiro	<b>Int(3.9)</b>	<b>3</b>
<b>RaizQ</b>	Raiz Quadrada	<b>RaizQ(25)</b>	<b>5</b>
<b>Pi</b>	Retorna Pi	<b>Pi</b>	<b>3.14..</b>
<b>Sen</b>	Seno (rad)	<b>Sen(0.523)</b>	<b>0.5</b>
<b>Cos</b>	Cosseno (rad)	<b>Cos(0.523)</b>	<b>0.86</b>
<b>Tan</b>	Tangente (rad)	<b>Tan(0.523)</b>	<b>0.57</b>
<b>GraupRad</b>	Graus para Rad	<b>GraupRad(30)</b>	<b>0.52</b>

Arquivo Editar Exibir Algoritmo Código Ferramentas Ajuda

0.5 s 0 100 Decimais 0

```

algoritmo "conversor"
// Função :
// Autor :
// Data : 20/01/2022
// Seção de Declarações
var
    Angulo, S: Real
inicio
// Seção de Comandos
Escreva ("Informe angulo: ")
Leia (angulo)
S <- Sen(GraupRad (angulo))
Escreva ( "O seno de " , angulo , " é igual a " , S)

finalgoritmo
  
```

Escopo	Nome	Tipo	Valor
GLOBAL	ANGULO	R	90
GLOBAL	S	R	1

^ Início da execução  
 Informe angulo: 90  
 O seno de 90 é igual a 1  
 Fim da execução.

# OPERADORES RELACIONAIS

>	Maior que
<	Menor que
>=	Maior ou igual a
<=	Menor ou igual a
=	Igual a
<>	Diferente de

1920

Arquivo Editar Exibir Algoritmo Código Ferramentas Ajuda

0.5 s

De: 0 Até: 100 Decimais: 0

```

algoritmo "relacionais"
// Função :
// Autor :
// Data : 20/01/2022
// Seção de Declarações
var
A, B, C: Inteiro
inicio
// Seção de Comandos
A <- 2
B <- 3
C <- 5

Escreva (A>B)

finalgoritmo
  
```

Escopo	Nome	Tipo	Valor
GLOBAL	A	I	2
GLOBAL	B	I	3
GLOBAL	C	I	5

^ Início da execução  
FALSO  
Fim da execução.

1920

Arquivo Editar Exibir Algoritmo Código Ferramentas Ajuda

De: 0 Até: 100 Decimais: 0

```

algoritmo "relacionais"
// Função :
// Autor :
// Data : 20/01/2022
// Seção de Declarações
var
  A, B, C: Inteiro
inicio
// Seção de Comandos
A <- 2
B <- 3
C <- 5

Escreva (C = A+B)

finalgoritmo

```

Escopo	Nome	Tipo	Valor
GLOBAL	A	I	2
GLOBAL	B	I	3
GLOBAL	C	I	5

Início da execução  
VERDADEIRO  
Fim da execução.


## OPERADORES LÓGICOS

p	q	p E q
V	V	V
V	F	F
F	V	F
F	F	F

p	q	p OU q
V	V	V
V	F	V
F	V	V
F	F	F

p	NÃO p
V	F
F	V



1920

Arquivo Editar Exibir Algoritmo Código Ferramentas Ajuda

0.5 s De 0 Até 100 Decimais 0

```

algoritmo "logicos"
// Função :
// Autor :
// Data : 20/01/2022
// Seção de Declarações
var
  A, B, C: Inteiro
inicio
// Seção de Comandos
A <- 2
B <- 3
C <- 5

Escreva ((a=b) ou (c>a))

fimalgoritmo

```

Escopo	Nome	Tipo	Valor
GLOBAL	A	I	2
GLOBAL	B	I	3
GLOBAL	C	I	5

Início da execução  
VERDADEIRO  
Fim da execução.

1920

Arquivo Editar Exibir Algoritmo Código Ferramentas Ajuda

0.5 s De 0 Até 100 Decimais 0

```

algoritmo "logicos"
// Função :
// Autor :
// Data : 20/01/2022
// Seção de Declarações
var
  A, B, C: Inteiro
inicio
// Seção de Comandos
A <- 2
B <- 3
C <- 5

Escreva (nao((a=b) ou (c>a)))

fimalgoritmo

```

Escopo	Nome	Tipo	Valor
GLOBAL	A	I	2
GLOBAL	B	I	3
GLOBAL	C	I	5

Início da execução  
FALSO  
Fim da execução.

Obs: Se a proposição  $((a=b) \text{ ou } (c>a))$  é VERDADEIRO, porque **C** é maior que **A**, e quando uma das proposições for verdadeiro, no operador OU o resultado será verdadeiro.  $(a=b)$  é falso e  $(c>a)$  é verdadeiro. Logo: na tabela verdade de OU, fica: F V = V (VERDADEIRO)

Então na expressão:  $(\text{não } ((a=b) \text{ ou } (c>a)))$  eu estou negando toda a proposição, logo será FALSO.

ORDEM DE PRECEDÊNCIA	
Aritméticos	( )
	^
	* /
	+ -
Relacionais	Todos
Lógicos	E
	OU
	NÃO

1920

Arquivo Editar Exibir Algoritmo Código Ferramentas Ajuda

05s 0e 0 Alt 100 Decimal 0

```

algoritmo "logicos_triangulos"
// Função :
// Autor :
// Data : 20/01/2022
// Seção de Declarações
var
  L1, L2, L3: Real
  EQ, ES: logico
inicio
// Seção de Comandos
Escriva ("digite o primeiro lado: ")
Leia (L1)
Escriva ("digite o segundo lado: ")
Leia (L2)
Escriva ("digite o terceiro lado: ")
Leia (L3)
EQ <- (L1 = L2) e (L2 = L3)
ES <- (L1 < L2) e (L2 < L3) e (L3 < L1)
Escrival ( "o triangulo é EQUILATERO? ", EQ)
Escrival ( "o triangulo é ESCALENO? ", ES)

finalgoritmo
  
```

Escopo	Nome	Tipo	Valor
GLOBAL	L1	R	2
GLOBAL	L2	R	3
GLOBAL	L3	R	4
GLOBAL	EQ	L	FALSO
GLOBAL	ES	L	VERDADEIRO

Início da execução  
 digite o primeiro lado: 2  
 digite o segundo lado: 3  
 digite o terceiro lado: 4  
 o triangulo é EQUILATERO? FALSO  
 o triangulo é ESCALENO? VERDADEIRO  
 Fim da execução.



## UTILIZANDO O SCRATCH E O VISUALG

VisuAlg 2.5 - C:\Users\cinth\OneDrive\Documentos\MeusProjetos\Tarefas no VisuAlg\Temperatura.alg

Arquivo Editar Exibir Algoritmo Código Ferramentas Ajuda

0.5 s De: 0 Até: 100 Decimais: 0

```

algoritmo "Creuza03"
// Função :
// Autor :
// Data : 20/01/2022
// Seção de Declarações
var
    F, C: real

inicio
// Seção de Comandos
    EScreval ("Qual a temperatura aqui? ")
    Leia (F)
    C <- (F - 32) / 1.8
    Escreva ("No Brasil estaria: " , C:4:1)

finalgoritmo
  
```

Escopo	Nome	Tipo	Valor
GLOBAL	F	R	100
GLOBAL	C	R	37.7777777777778

Início da execução  
Qual a temperatura aqui?  
100  
No Brasil estaria: 37.8  
Fim da execução.

SCRATCH

Arquivo Editar Compartilhar Ajuda

Movimento Controle Aparência Sensores Som Operadores Caneta Variáveis

objeto1 x: -59 y: 23 direção

Comandos Trajes Sons

quando clicado

pergunte Qual a temperatura aqui? e

mude F para resposta

mude C para  $F - 32 / 1.8$

pense junte No Brasil estaria C

Qual a temperatura aqui?

F 100

C 37.8

sorteie número entre 1 e 10

< = >

e ou não

junte plá mundo

letra 1 de mundo

letras em mundo

resto da divisão por

arredonde

raiz quadrada de 10

Novo sprite: ?

objeto1



## Estruturas Condicionais

(Se, Então)

958

Arquivo Editar Exibir Algoritmo Código Ferramentas Ajuda

0,5 s De: 0 Até: 100 Decimais: 0

```

algoritmo "semnome"
// Função :
// Autor :
// Data : 21/01/2022
// Seção de Declarações
var
    ano, nasc, idade: inteiro
inicio
// Seção de Comandos
escreva ("Em que ano estamos? ")
leia (ano)
escreva ("Em que ano voce nasceu? ")
leia (nasc)
idade <- ano - nasc
escreval ("em ", ano, " voce terá ", idade, " anos")
Se (idade >= 18) Entao
    escreval ("e ja tera atingido a maioridade ")
FimSe

finalgoritmo
  
```

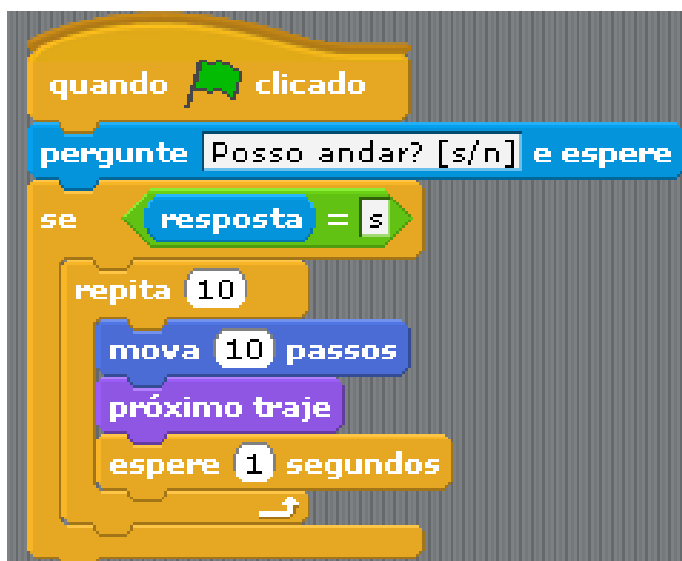
Se (idade >= 18) Entao

Escreval ("e terá atingido a maioridade ")

FimSe

Escopo	Nome	Tipo	Valor
GLOBAL	ANO	I	2022
GLOBAL	NASC	I	1987
GLOBAL	IDADE	I	35

Início da execução  
 Em que ano estamos? 2022  
 Em que ano voce nasceu? 1987  
 em 2022 voce terá 35 anos  
 e ja tera atingido a maioridade  
 Fim da execução.



### Condicional Composta:

VisuAlg 2.5 - Semnome.alg

Arquivo Editar Exibir Algoritmo Código Ferramentas Ajuda

0.5 s De: 0 Até: 100 Decimais: 0

```

algoritmo "Condicional Composta"
// Função :
// Autor :
// Data : 21/01/2022
// Seção de Declarações
var
    dinheiro: real
inicio
// Seção de Comandos
Escreval (" Quanto você economizou? " )
Leia (dinheiro)
Se (dinheiro >= 10000) entao
    Escreval (" Partiu Disney!")
Senao
    Escreva (" #chateado!")
FimSe
fimalgoritmo
  
```

Escopo	Nome	Tipo	Valor
GLOBAL	DINHEIRO	R	5000

^ Início da execução  
 Quanto você economizou?  
 5000  
 #chateado!  
 Fim da execução.

OBS: Para saber se o número é par deve-se dividi-lo por 2 e observar se o resto da divisão. Se o resultado for 0, o número será PAR, contudo se o resto for 1 o número será IMPAR. Para nos auxiliar podemos utilizar o comando (%) que faz esse calculo e observa o resto da divisão.

Se ( $N \% 2 = 0$ ) Entao

Escreva (" O numero " , N , " é par")

Senao

Escreva (" O numero " , N , " é impar ")

FimSe

5	2	
1		2

Logo, 5 é Impar!!!!

VisuAlg 2.5 - C:\Users\cinth\OneDrive\Documentos\MeusProjetos\Curso\_algoritmos - cinthia\Tarefas no VisuAlg - Alg

Arquivo Editar Exibir Algoritmo Código Ferramentas Ajuda

0.5 s De: 0 Até: 100 Decimais: 0

```

algoritmo "ParOuImpar"
// Função :
// Autor :
// Data : 21/01/2022
// Seção de Declarações
var
    N: Inteiro
inicio
// Seção de Comandos
Escreval (" Digite um número: ")
Leia (N)
Se (N % 2 = 0) entao
    Escreval (" O numero " , N , " é par")
Senao
    Escreva (" O numero " , N , " é impar ")
FimSe

fimalgoritmo
  
```

Escopo	Nome	Tipo	Valor
GLOBAL	N	I	10

Início da execução  
 Digite um número:  
 10  
 O numero 10 é par  
 Fim da execução.

## CONDICIONAL ANINHADA

OBS: Quando existem mais de uma alternativa, condições a se observar.

VisuAlg 2.5 - C:\Users\cinth\OneDrive\Documentos\MeusProjetos\Curso\_algoritmos - cinthia\Tarefas no VisuAlg - Algoritmos\condicional\_aninhad...

Arquivo Editar Exibir Algoritmo Código Ferramentas Ajuda

0.5 s De: 0 Até: 100 Decimais: 0

```

algoritmo "Condicional_Aninhada"
// Função :
// Autor :
// Data : 21/01/2022
// Seção de Declarações
var
    dinheiro: real
inicio
// Seção de Comandos
escreval(" Quanto em dinheiro voce juntou? ")
leia (dinheiro)
Se (dinheiro >= 10000) entao
    escreval ( "partiu Disney")
Senao
    Se (dinheiro >= 5000) e (dinheiro < 10000) entao
        EScreval (" visitar familia ")
    Senao
        escreva (" Vai ficar em casa!" )
    FimSe
FimSe

fimalgoritmo
  
```

Escopo	Nome	Tipo	Valor
GLOBAL	DINHEIRO	R	3000

Início da execução  
 Quanto em dinheiro voce juntou?  
 3000  
 Vai ficar em casa!  
 Fim da execução.

## ESTRUTURA ESCOLHA

OBS: Não serve para testar faixa de valores (exe. Não pode-se usar quando for necessário observar entre valores, como entre 5.000 e 7.000, por exemplo, pois neste caso utiliza-se o ANINHADA). E o modo ESTRUTURA ESCOLHA só funciona para valores inteiros.

VisuAlg 2.5 - C:\Users\cinth\OneDrive\Documentos\MeusProjetos\Curso\_algoritmos - cinthia\Tarefas no VisuAlg - Algoritmos\Escolha\_Caso.alg

Arquivo Editar Exibir Algoritmo Código Ferramentas Ajuda

0,5 s De: 0 Até: 100 Decimais: 0

```
// Data : 21/01/2022
// Seção de Declarações
var
  Nome: caractere
  salario, Nsal: real
  dep: inteiro
inicio
  // Seção de Comandos
  EScreval (" Qual o nome do funcionário? ")
  Leia (nome)
  EScreval ("qual o salario do funcionário? R$")
  Leia (salario)
  EScreval ("quantos dependentes possui? ")
  leia (dep)
  EScolha dep
    caso 0
      Nsal <- salario + ((salario*5)/100)
    caso 1, 2, 3
      Nsal <- salario + ((salario*10) / 100)
    caso 4, 5, 6
      Nsal <- salario + ((salario*20) / 100)

  FimEScolha
  EScreva (" O novo salario de " , nome, " sera de R$ " , Nsal)

fimalgoritmo
```

Escopo	Nome	Tipo	Valor
GLOBAL	NOME	C	"cinthia"
GLOBAL	SALARIO	R	5000
GLOBAL	NSAL	R	6000
GLOBAL	DEP	I	5

Início da execução  
 Qual o nome do funcionário?  
 cinthia  
 qual o salario do funcionário? R\$  
 5000  
 quantos dependentes possui?  
 5  
 O novo salario de cinthia sera de R\$ 6000  
 Fim da execução.

## ESTRUTURA DE REPETIÇÃO

(Enquanto..... faça  
FimEnquanto)

Para contar de 0 até 10, deve-se utilizar uma variável que conte (contador), e a estrutura de repetição deve conter as seguintes informações: ENQUANTO , FimEnquanto.

Contador <- 0 (começar a contar do zero)

Enquanto (contador <= 10) faça (enquanto o contador for menor ou igual à zero, faça!)

Escreva (contador)

Contador <- contador + 1 (contador vai receber na sua variável a informação: contador +1)

FimEnquanto

Escreva ("Terminei de contar")

OBS: o comando faz com que o programa entre em um LOOP enquanto as condições estiverem sendo atendidas.

```

algoritmo "ConteAté10"
// Função :
// Autor :
// Data : 21/01/2022
// Seção de Declarações
var
    contador: inteiro

inicio
// Seção de Comandos
contador <- 0
Enquanto (contador <= 10) faça
    Escreva (contador)
    contador <- contador + 1
FimEnquanto
Escreva ("Terminei de contar ")

finalgoritmo
  
```

Escopo	Nome	Tipo	Valor
GLOBAL	CONTADOR	I	11

```

Início da execução
0
1
2
3
4
5
6
7
8
9
10
terminei de contar
Fim da execução.
  
```





**Deixando o usuário escolher até quanto deve-se contar:**

var

contador, valor: inteiro

inicio

Contador &lt;- 0

Escreva ("Quer contar até quanto? ")

Leia (valor)

Enquanto (contador &lt;= valor) faça

Escreval (contador)

contador &lt;- contador + 1

FimEnquanto

Escreval ("terminei de contar")

fimalgoritmo

VisuAlg 2.5 - Semnome.alg

Arquivo Editar Exibir Algoritmo Código Ferramentas Ajuda

0,5 s De: 0 Até: 100 Decimais: 0

```

algoritmo "semnome"
// Função :
// Autor :
// Data : 21/01/2022
// Seção de Declarações
var
contador, valor: inteiro

inicio
// Seção de Comandos
Contador <- 0
EScreva ("Quer contar até quanto? ")
Leia (valor)
Enquanto (contador <= valor) faça
    Escreval (contador)
    contador <- contador + 1
FimEnquanto
Escreval ("terminei de contar")

fimalgoritmo
  
```

Escopo	Nome	Tipo	Valor
GLOBAL	CONTADOR	I	6
GLOBAL	VALOR	I	5

Início da execução  
 Quer contar até quanto? 5  
 0  
 1  
 2  
 3  
 4  
 5  
 terminei de contar  
 Fim da execução.

**Deixando o usuário escolher até quanto deve-se contar e quantos números pular na contagem ( exe.: de 3 em 3):**

VisuAlg 2.5 - C:\Users\cinth\OneDrive\Documentos\MeusProjetos\Curso\_algoritmos - cinthia\Tarefas no VisuAlg - Algoritmos\usuario

Arquivo Editar Exibir Algoritmo Código Ferramentas Ajuda

0,5 s De: 0 Até: 100 Decimais: 0

```

algoritmo "semnome"
// Função :
// Autor :
// Data : 21/01/2022
// Seção de Declarações
var
contador, valor, salto: inteiro

inicio
// Seção de Comandos
Contador <- 0
EScreva ("Quer contar até quanto? ")
Leia (valor)
EScreva (" Qual será o valor do salto? ")
leia (salto)
Enquanto (contador <= valor) faça
    Escreval (contador)
    contador <- contador + salto
FimEnquanto
Escreval ("terminei de contar")

fimalgoritmo
  
```

Escopo	Nome	Tipo	Valor
GLOBAL	CONTADOR	I	12
GLOBAL	VALOR	I	10
GLOBAL	SALTO	I	3

Início da execução  
 Quer contar até quanto? 10  
 Qual será o valor do salto? 3  
 0  
 3  
 6  
 9  
 terminei de contar  
 Fim da execução.

### Estrutura de REPETIÇÃO - (Repita , Ate)

var

Num, Soma: inteiro

resp: caractere

inicio

Soma <- 0

Repita (as regras abaixo)

Escreva ("Digite o valor ==> ")

Leia (Num)

Soma <- Soma + Num (o **Soma** recebe o valor atual de **Soma** + o **Num**, digitado pelo usuário)

Escreva (" Você quer continuar? [S / N] ")

Leia (resp)



Ate (resp = "N" ) (repete ate que a **resp** dada pelo usuário **[S/N]** seja **N**. Agora o programa vai encerrar e somar os **num.** digitados pelo usuário)

EScrevaL ( "A soma dos valores digitados é ", S)

944

Arquivo Editar Exibir Algoritmo Código Ferramentas Ajuda

0,5s De: 0 Até: 100 Decimais: 0

```

algoritmo "repita"
// Função :
// Autor :
// Data : 24/01/2022
// Seção de Declarações
var
Num, Soma: inteiro
resp: caractere

inicio
Soma <- 0
Repita
    Escreva ("Digite o valor ==> ")
    Leia (Num)
    Soma <- Soma + Num
    Escreva (" Voce quer continuar? [S / N] ")
    Leia (resp)

Ate (resp = "N" )
EScrevaL ( "A soma dos valores digitados é ", Soma)

finalgoritmo
  
```

Escopo	Nome	Tipo	Valor
GLOBAL	NUM	I	2
GLOBAL	SOMA	I	7
GLOBAL	RESP	C	"n"

Início da execução  
 Digite o valor ==> 2  
 Voce quer continuar? [S / N] s  
 Digite o valor ==> 3  
 Voce quer continuar? [S / N] s  
 Digite o valor ==> 2  
 Voce quer continuar? [S / N] n  
 A soma dos valores digitados é 7  
 Fim da execução.

## Repetição com a TABUADA: (Repita – Ate)

VisuAlg 2.5 - C:\Users\cinth\OneDrive\Documentos\MeusProjetos\Curso\_algoritmos - cinthia\Tarefas no VisuAlg - Algoritmos\Repeticao\_tab

Arquivo Editar Exibir Algoritmo Código Ferramentas Ajuda

0,5 s De: 0 Até: 100 Decimais: 0

```

algoritmo "semnome"
// Função :
// Autor :
// Data : 24/01/2022
// Seção de Declarações
var

contador, N, R: inteiro
inicio

Contador <- 1
EscrevaL (" Quer ver a tabuada de qual número? ")
Leia (N)
repita
    R <- N * contador
    EScrevaL (N, " x ", contador, " = ", R)
    contador <- contador + 1
ate (contador > 10)

fimalgoritmo
  
```

Escopo	Nome	Tipo	Valor
GLOBAL	CONTADOR	I	11
GLOBAL	N	I	3
GLOBAL	R	I	30

Início da execução  
 Quer ver a tabuada de qual número?  
 3  
 3 x 1 = 3  
 3 x 2 = 6  
 3 x 3 = 9  
 3 x 4 = 12  
 3 x 5 = 15  
 3 x 6 = 18  
 3 x 7 = 21  
 3 x 8 = 24  
 3 x 9 = 27  
 3 x 10 = 30  
 Fim da execução.

Legenda: (R = resultado) ; (N = número).

## PARA GERAR NÚMEROS/TEXTOS ALEATÓRIOS NO VISUALG

```

1 Algoritmo "Números Aleatórios"
2 // Lógica de Programação e Estruturas de Dados
3 // Fábio dos Reis
4
5 Var
6 NUM, NUM_ALEAT: inteiro
7 RESULTADO: inteiro
8 TEXTO: caractere
9
10 Início
11 escreval ("Digite um número: ")
12 leia (NUM)
13
14 escreva ("Número escolhido pelo computador: ")
15 aleatorio 0, 10
16 leia (NUM_ALEAT)
17 aleatorio off
18 escreval ()
19
20 RESULTADO <- NUM * NUM_ALEAT
21
22 escreval ("O produto dos números é ", RESULTADO)
23 escreval ()
24
25 pausa
26
27 escreval ("Gerando string de texto aleatória:")
28 aleatorio on
29 leia (TEXTO)
30 aleatorio off
31
34 escreval ("Gerando muitos números de forma aleatória:")
35 pausa
36 aleatorio on
37 para NUM de 1 ate 100 faca
38   leia (NUM_ALEAT)
39 fimpara
40 aleatorio off
41
42 Fimalgoritmo

```

## PARA CONTAR QUANTOS NUMEROS NEGATIVOS FORAM DIGITADOS (ENTRE VARIOS)

```

algoritmo "Conta_negativos"
// Função :
// Autor :
// Data : 28/01/2022
// Seção de Declarações
var
N, TotN, C: inteiro
inicio
// Seção de Comandos
TotN <- 0
C <- 1
Repita
    Escreva ("Digite um número: ")
    Leia (N)
    SE (N<0) entao
        TotN <- TotN + 1
    FimSe
    c <- C + 1
Ate (C>5)
escrevaL ("Foram digitado(s) " , TotN, " valor(es) negativo(s).")
fimalgoritmo
  
```

Escopo	Nome	Tipo	Valor
GLOBAL	N	I	5
GLOBAL	TOTN	I	1
GLOBAL	C	I	6

Início da execução  
 Digite um número: 1  
 Digite um número: -5  
 Digite um número: 6  
 Digite um número: 8  
 Digite um número: 5  
 Foram digitados 1 valores negativos.  
 Fim da execução.

### IMPOSTANTE:

**C <- 1** (ou seja, inicia a contagem pelo número 1)

**C <- C + 1** (ou seja, o C que valia 1 vai receber a soma de + 1 ( 1 + 1 ) = 2)

OBS: Como esta no modo de repetição (REPITA / ATE) o programa vai continuar ate que se atenda a regra constante no ATE.

O C passa a valer 2, novamente soma-se + 1, passando a valer 3, novamente soma-se +1, passando a valer 4, novamente soma-se +1, passando a valer 5.

**ATE (C>5)** (ou seja, vai repetir a contagem (somando 1 ao C) ate que alcance o numero 5)

Em outras palavras:

**C <- 1** (significa que a contagem de C começa pelo número 1 e que vai somar +1 no C ate que...)

**ATE (C>5)** (...ate que chegue ao número 5)

OBS: O mesmo ocorre com o TotN que, no exemplo acima, inicia a contagem pelo número 0.

## CALCULANDO O FATORIAL DE UM NÚMERO

Para saber o fatorial de um numero é só calcular o número multiplicado pelo seu antecessor e assim sucessivamente até chegar ao número 1. Exe: Fatorial de 4 é igual a:  $4 \times 3 \times 2 \times 1 = 24$

```

algoritmo "semnome"
// Função :
// Autor :
// Data : 28/01/2022
// Seção de Declarações
var
n, c, F: inteiro
inicio
// Seção de Comandos
EScreva ("digite um numero: ")
Leia (n)
c <- N
F <- 1
Repita
escrevaL (C, " x ")
F <- F * C
C <- c - 1
Ate (c < 1)
escreva (" O valor do fatorial de ", N , " é igual a ", F)
fimalgoritmo
  
```

Escopo	Nome	Tipo	Valor
GLOBAL	N	I	4
GLOBAL	C	I	0
GLOBAL	F	I	24

Início da execução  
 digite um numero: 4  
 4 x  
 3 x  
 2 x  
 1 x  
 O valor do fatorial de 4 é igual a 24  
 Fim da execução.



### FUNÇÃO DESEJA CONTINUAR + LIMPA TELA

---

```

algoritmo "semnome"
// Função :
// Autor :
// Data : 28/01/2022
// Seção de Declarações
var
n, c, F: inteiro
R: caractere
inicio
// Seção de Comandos
Repita
    EScreva ("digite um numero: ")
    Leia (n)
    c <- N
    F <- 1
    Repita
        escrevaL (C, " x ")
        F <- F * C
        C <- c - 1
    Ate (c < 1)
    Escreval (" O valor do fatorial de ", N , " é igual a ", F)
    EScreva (" Deseja continuar? [S/N] ")
    Leia (R)
    LimpaTela
Ate (R = "N")
finalgoritmo

```

---

```

Início da execução
digite um numero: 5
5 x
4 x
3 x
2 x
1 x
O valor do fatorial de 5 é igual a 120
Deseja continuar? [S/N] S
digite um numero: 4
4 x
3 x
2 x
1 x
O valor do fatorial de 4 é igual a 24
Deseja continuar? [S/N] N

Fim da execução.

```

### PARA ACHAR O MAIOR VALOR ENTRE VARIOS NÚMEROS

**SE (N > maior) então** (significa que a cada vez que for lido o N o programa vai verificar Se N é maior que o valor que está guardado, (se for) então o maior valor passa a ser N (numero atual)).

**maior <- N**

```

algoritmo "Somador_Maior_num"
// Função :
// Autor :
// Data : 24/01/2022
// Seção de Declarações
var
Cont, N, S, Maior: inteiro

inicio
Cont <- 1
S <- 0
Enquanto (cont <= 5) faça
    Escreva ("Digite o ", cont, "o. valor: ")
    Leia (N)
    SE (N > maior) então
        maior <- N
    FimSe
    S <- S + N
    Cont <- cont + 1
FimEnquanto
EScrevaL ( "A soma dos valores digitados é ", S)
EScrevaL ( "O maior valor digitados foi ", maior)

fimalgoritmo
  
```

Escopo	Nome	Tipo	Valor	
GLOBAL	CONT	I	6	↑
GLOBAL	N	I	1	
GLOBAL	S	I	23	
GLOBAL	MAIOR	I	8	

Início da execução

Digite o 1o. valor: 5

Digite o 2o. valor: 6

Digite o 3o. valor: 3

Digite o 4o. valor: 8

Digite o 5o. valor: 1

A soma dos valores digitados é 23

O maior valor digitados foi 8

Fim da execução.

### ESTRUTURA DE REPETIÇÃO – PARA

Não pode ser utilizado se não soubermos até qual número será executado, logo para modelos onde o usuário é quem for definir até quanto a contagem irá, escolhendo se deseja continuar [S/N], NÃO PODE-SE USAR O PARA. (ex.: pág. 18). Porém se o usuário definir um número exato para até quando a contagem irá, então pode-se usar o PARA. Ex.: digite um número: 10; será contado até o número 10, ou seja, um número exato/fechado.

**Para** variavel <- inicio **ate** fim [passo Salto] **faca**

Bloco

**FimPara**

**Exemplo: Contando até 10**

**Para** C <- 1 **ate** 10 **passo** 1 **faca**

EscrevaL (C)

**FimPara**

OBS 1: A expressão PASSO significa que vai somar + 1 na variável C (é o SALTO). Então no exemplo acima o C começa recebendo 1 e depois soma-se 1 a cada passagem até alcançar o número 10, estipulado na expressão.

OBS 2: Quando não for especificado o PASSO, entende-se que é **sempre +1**, logo é dispensável colocá-lo na expressão se o PASSO for + 1.

```
algoritmo "semnome"
// Função :
// Autor :
// Data : 31/01/2022
// Seção de Declarações
var
C: inteiro
inicio
// Seção de Comandos
Para C <- 1 ate 10 passo 1 faca
    escrevaL (c)
FimPara

finalgoritmo
```

Escopo	Nome	Tipo	Valor	Início da execução
GLOBAL	C	I	11	1
				2
				3
				4
				5
				6
				7
				8
				9
				10
				Fim da execução.

**Pulando de 2 em 2:** (Passo 2)

```

algoritmo "semnome"
// Função :
// Autor :
// Data : 31/01/2022
// Seção de Declarações
var
C: inteiro
inicio
// Seção de Comandos
Para C <- 1 ate 10 passo 2 faca
    escrevaL (c)
FimPara

fimalgoritmo

```

Escopo	Nome	Tipo	Valor	Início da execução
GLOBAL	C	I	11	1
				3
				5
				7
				9
				Fim da execução.

**CONTAGEM REGRESSIVA:**

```

algoritmo "semnome"
// Função :
// Autor :
// Data : 31/01/2022
// Seção de Declarações
var
C: inteiro
inicio
// Seção de Comandos
Para C <- 10 ate 1 passo -1 faca
    escrevaL (c)
FimPara

fimalgoritmo

```

Início da execução
10
9
8
7
6
5
4
3
2
1
Fim da execução.

### SOMANDO VÁRIOS VALORES DIGITADOS PELO USUÁRIO:

```

algoritmo "semnome"
// Função :
// Autor :
// Data : 31/01/2022
// Seção de Declarações
var
C, N, S: inteiro
inicio
// Seção de Comandos
Para C <- 1 ate 5 passo 1 faca
    escrevaL (" Digite um valor: ")
    Leia (N)
    S <- S + N
FimPara
escrevaL (" A soma de todos os valores é ", S)

finalgoritmo

```

Escopo	Nome	Tipo	Valor	
GLOBAL	C	I	6	
GLOBAL	N	I	1	
GLOBAL	S	I	5	

Início da execução

Digite um valor:

1

Digite um valor:

1

Digite um valor:

1

Digite um valor:

1

Digite um valor:

1

A soma de todos os valores é 5

Fim da execução.

### O USUARIO ESCOLHENDO O NÚMERO/REPETIÇÕES (NÚMEROS PARES)

```

algoritmo "semnome"
// Função :
// Autor :
// Data : 31/01/2022
// Seção de Declarações
var
V, C: inteiro
inicio
// Seção de Comandos
EscrevaL ("Digite um valor: ")
Leia (V)
Para C <- 0 ate V passo 2 faca
    escrevaL (C)
FimPara

finalgoritmo

```

```

Início da execução
Digite um valor:
7
0
2
4
6
Fim da execução.

```

## O USUARIO ESCOLHENDO O NÚMERO/REPETIÇÕES (NÚMEROS PARES) - Regressiva

```

algoritmo "semnome"
// Função :
// Autor :
// Data : 31/01/2022
// Seção de Declarações
var
V, C: inteiro
inicio
// Seção de Comandos
EscrevaL ("Digite um valor: ")
Leia (V)
SE (V%2=1) entao
    V <- V - 1
FimSE
Para C <- V ate 0 passo -2 faca
    escrevaL (c)
FimPara

fimalgoritmo

```

Escopo	Nome	Tipo	Valor
GLOBAL	V	I	6
GLOBAL	C	I	-2

^ Início da execução  
 Digite um valor:  
 7  
 6  
 4  
 2  
 0  
 Fim da execução.

OBS: Para mostrar apenas os NUMEROS PARES usando a repetição PARA, deve-se corrigir o código inserindo uma variável com a informação de que se o numero for impar deve-se subtrair (- 1) para que ele fique PAR.

Ex.: **SE** (V%2=1) **então**

V <- V - 1

**FimSE**

### ENTRE 6 VALORES QUANTOS ESTÃO ENTRE 0 E 10

```

algoritmo "semnome"
// Função :
// Autor :
// Data : 31/01/2022
// Seção de Declarações
var
C, V, Tot010: inteiro
inicio
// Seção de Comandos
Tot010 <- 0
Para C <- 1 ate 6 faca
    escreva ("Digite um valor: ")
    Leia (v)
    Se (V >= 0) e (V <= 10)entao
        Tot010 <- Tot010 + 1
    FimSe
FimPara
EScrevaL ("Ao todo foram ", Tot010, " valores entre 0 e 10")

fimalgoritmo

```

Escopo	Nome	Tipo	Valor	
GLOBAL	C	I	7	^
GLOBAL	V	I	6	
GLOBAL	TOT010	I	4	

Início da execução  
 Digite um valor: 12  
 Digite um valor: 15  
 Digite um valor: 10  
 Digite um valor: 2  
 Digite um valor: 5  
 Digite um valor: 6  
 Ao todo foram 4 valores entre 0 e 10  
 Fim da execução.

### SOMA DE VALORES IMPARES QUE ESTEJAM DENTRO DO INTERVALO: 0 E 10

```

algoritmo "semnome"
// Função :
// Autor :
// Data : 31/01/2022
// Seção de Declarações
var
C, V, Tot010, SImp: inteiro
inicio
// Seção de Comandos
Tot010 <- 0
Para C <- 1 ate 6 faca
    escreva ("Digite um valor: ")
    Leia (v)
    Se (V >= 0) e (V <= 10)entao
        Tot010 <- Tot010 + 1
        SE (V%2=1) entao
            SImp <- SImp + V
        FimSE
    FimSe
FimPara
EScrevaL ("Ao todo foram ", Tot010, " valores entre 0 e 10")
escreva (" Nesse intervalo a soma de números impares foi de ", SImp)

fimalgoritmo

```

Escopo	Nome	Tipo	Valor	
GLOBAL	C	I	7	^
GLOBAL	V	I	1	
GLOBAL	TOT010	I	5	
GLOBAL	SIMP	I	13	

Início da execução  
 Digite um valor: 2  
 Digite um valor: 7  
 Digite um valor: 21  
 Digite um valor: 5  
 Digite um valor: 8  
 Digite um valor: 1  
 Ao todo foram 5 valores entre 0 e 10  
 Nesse intervalo a soma de números impares foi de 13  
 Fim da execução.





## SEQUENCIA DE FIBONACCI

```

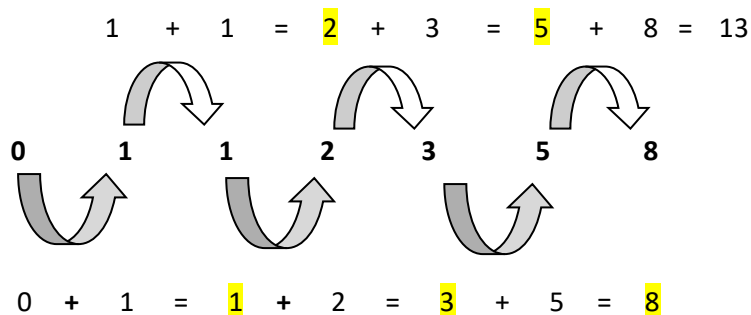
algoritmo "Fibonacci"
// =====
// Algoritmo para mostrar os 15 primeiros termos de uma serie de
// Fibonacci (0,1,1,2,3,5,8,...)
//
// AUTOR: Gustavo Guanabara
// Curso COMPLETO disponivel GRATIS no site www.cursoemvideo.com
// =====
var
C, T1, T2, T3 : Inteiro
inicio
T1 <- 0
Escreva (T1)
T2 <- 1
Escreva (T2)
Para C := 3 ate 15 faca
    T3 := T1 + T2
    Escreva (T3)
    T1 := T2
    T2 := T3
FimPara
fimalgoritmo

```

Escopo	Nome	Tipo	Valor
GLOBAL	C	I	16
GLOBAL	T1	I	233
GLOBAL	T2	I	377
GLOBAL	T3	I	377

Início da execução  
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377  
Fim da execução.

OBS: Na sequencia de Fibonacci deve-se somar o numero com seu sucessor dentro de um intervalo de numero pré definido. Ex.: No intervalo de 7 números começando do zero: Soma-se  $0+1=1$ , então fica: 0,1,1. Depois soma-se o resultado com próximo numero, ou seja:  $1+1=2$ , então fica: 0,1,1,2. Depois soma-se o resultado com o próximo numero, ou seja:  $2+3=5$ , então fica: 0,1,1,2,3,5. Depois soma-se o resultado com o próximo numero, ou seja,  $3+5=8$ .





**PROCEDIMENTO – ROTINA**


---

```

algoritmo "DetectorPesado"
// Função :
// Autor :
// Data : 01/02/2022
// Seção de Declarações
var
I: inteiro
N, pesado: caractere
P, Mai: Real
inicio
// Seção de Comandos
LimpaTela
EScrevaL ("-----")
escrevaL ("DETECTOR DE PESADO ")
EScrevaL ("O maior peso até agora: ", Mai, "Kg")
EScrevaL ("-----")
Para I <- 1 ate 5 faca
    escreva ("Digite o nome: ")
    leia (N)
    escreva ("Digite o peso de ", N, ": ")
    leia (P)
    Se (P>Mai) entao
        Mai <- P
        Pesado <- N
    FimSe
    LimpaTela
    EScrevaL ("-----")
    escrevaL ("DETECTOR DE PESADO ")
    EScrevaL ("O maior peso até agora: ", Mai, "Kg")
    EScrevaL ("-----")
FimPara
EScrevaL ("-----")
escrevaL ("DETECTOR DE PESADO ")
EScrevaL ("O maior peso até agora: ", Mai, "Kg")
EScrevaL ("-----")
EscrevaL ("A pessoa mais pesada foi ", pesado, ", com ", Mai, "quilos. ")
finalgoritmo

```

---

**OBS:** Para evitar a repetição das 5 linhas de (LimpaTela e EscrevaL) podemos usar o procedimento ROTINA. Dentro do campo **VAR**, deve-se colocar todo o conteúdo que vai repetir quando chamado o PROCEDIMENTO., utilizando-se :

procedimento topo()

inicio

//conteúdo a ser repetido quando chamado o comando//

FimProcedimento

<pre> algoritmo "DetectorPesado" // Função : // Autor : // Data : 01/02/2022 // Seção de Declarações var I: inteiro N, pesado: caractere P, Mai: Real procedimento topo() inicio   LimpaTela   EScrevaL ("-----")   EScrevaL ("DETECTOR DE PESADO ")   EScrevaL ("O maior peso até agora: ", Mai, "Kg")   EScrevaL ("-----") FimProcedimento inicio // Seção de Comandos topo() Para I &lt;- 1 ate 3 faca   escreva ("Digite o nome: ")   leia (N)   escreva ("Digite o peso de ", N, ": ")   leia (P)   Se (P&gt;Mai) entao     Mai &lt;- P     Pesado &lt;- N   FimSe topo() FimPara topo() EScrevaL ("A pessoa mais pesada foi ", pesado, ", com ", Mai, " quilos. ") finalgoritmo </pre>				
Escopo	Nome	Tipo	Valor	
GLOBAL	I	I	4	DETECTOR DE PESADO
GLOBAL	N	C	"Ana"	O maior peso até agora: 108Kg
GLOBAL	PESADO	C	"Davi"	-----
GLOBAL	P	R	95	Digite o nome: Ana
GLOBAL	MAI	R	108	Digite o peso de Ana: 95
				-----
				DETECTOR DE PESADO
				O maior peso até agora: 108Kg
				-----
				DETECTOR DE PESADO
				O maior peso até agora: 108Kg
				-----
				A pessoa mais pesada foi Davi, com 108 quilos.
				Fim da execução.

### PROCEDIMENTO SOMA:

<pre> algoritmo "semnome" // Função : // Autor : // Data : 01/02/2022 // Seção de Declarações var x, y: inteiro Procedimento Soma (A, B: inteiro) inicio   escrevaL("Recebi o valor ", A)   escrevaL("Recebi o valor ",B)   escrevaL("A soma dos valores é ", A+B) Fimprocedimento inicio // Seção de Comandos x &lt;- 5 Y &lt;- 3 Soma (x,y) finalgoritmo </pre>				
Escopo	Nome	Tipo	Valor	
GLOBAL	X	I	5	Início da execução
GLOBAL	Y	I	3	Recebi o valor 5
				Recebi o valor 3
				A soma dos valores é 8
				Fim da execução.

### PROCEDIMENTO PAR OU IMPAR:

```

algoritmo "semnome"
// Função :
// Autor :
// Data : 01/02/2022
// Seção de Declarações
var
N: inteiro
Procedimento ParOuImpar (V: inteiro)
inicio
    Se (V%2= 0) entao
        escrevaL ("O número ", V, " é PAR")
    senao
        escrevaL ("O número ", V, " é IMPAR")
    FimSe
FimProcedimento
inicio
// Seção de Comandos
escrevaL ("Digite um número: ")
Leia (N)
ParOuImpar(N)
fimalgoritmo

```

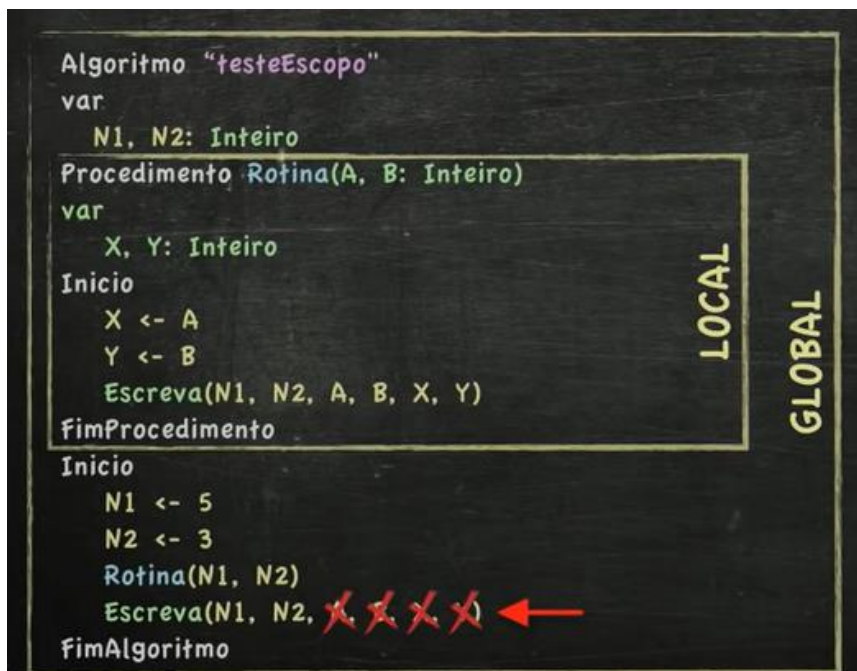
Escopo	Nome	Tipo	Valor	
GLOBAL	N	I	2	

^ Início da execução  
 Digite um número:  
 2  
 O número 2 é PAR  
 Fim da execução.

### PASSAGEM DE PARÂMETRO: POR VALOR

ESCOPO: GLOBAL E LOCAL

Local onde uma determinada variável vai funcionar (poder ser lida).



Quando chamar o procedimento SOMA os valores de X e de Y serão jogados em A e B.

```

algoritmo "semnome"
// Função :
// Autor :
// Data : 01/02/2022
// Seção de Declarações
var
X, Y: inteiro
procedimento Soma(A, B: inteiro)
inicio
    escreval("recebi o valor ", A)
    escreval ("recebi o valor ", B)
    escreval(" a soma entre os dois é de ", A+B)
FimProcedimento
inicio
// Seção de Comandos
X <- 5
Y <- 3
Soma (X, Y)
finalgoritmo

```

Escopo	Nome	Tipo	Valor		
GLOBAL	X	I	5		Início da execução
GLOBAL	Y	I	3		recebi o valor 5
					recebi o valor 3
					a soma entre os dois é de 8
					Fim da execução.

```

algoritmo "semnome"
// Função :
// Autor :
// Data : 01/02/2022
// Seção de Declarações
var
X, Y: inteiro
procedimento Soma(A, B: inteiro)
inicio
    A <- A + 1
    B <- B + 2
    escreval("Valor de A = ", A)
    escreval ("Valor de B = ", B)
    escreval("A soma de A + B = ", A+B)
FimProcedimento
inicio
// Seção de Comandos
X <- 4
Y <- 8
Soma (X, Y)
escreval ("Valor de X = ", X)
escreval ("Valor de Y = ", Y)
finalgoritmo

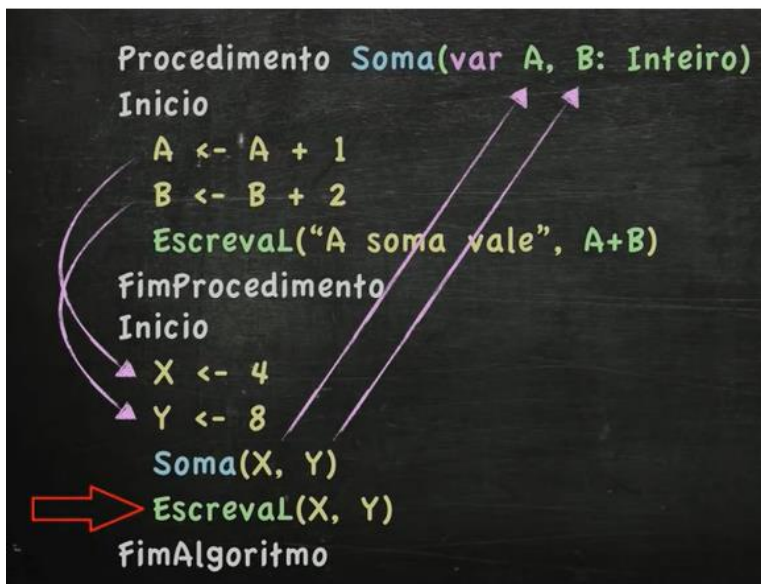
```

Escopo	Nome	Tipo	Valor		
GLOBAL	X	I	4		Início da execução
GLOBAL	Y	I	8		Valor de A = 5
					Valor de B = 10
					A soma de A + B = 15
					Valor de X = 4
					Valor de Y = 8
					Fim da execução.

Note que os valores de X e Y continuam os mesmos no final: 4 e 8. Uma vez o PARAMETRO foi PASSADO POR VALOR, de modo que não altera os valores originais de X e Y.



## PASSAGEM DE PARÂMETRO: POR REFERÊNCIA



Como foi colocado o VAR no início do procedimento SOMA, então quando chamar o procedimento SOMA (X,Y) será passado o endereço de X para dentro de A e o endereço de Y para dentro de B. Então,  $A \leftarrow A + 1$ , vai ficar:  $A \leftarrow 4 + 1$  (pois A recebeu o valor de X que é 4). Logo A passa a valer 5. Assim como,  $B \leftarrow B + 2$ , fica:  $B \leftarrow 8 + 2$  (pois B recebeu o valor de Y que é 8). Logo B passa a valer 10.

```

algoritmo "semnome"
// Função :
// Autor :
// Data : 01/02/2022
// Seção de Declarações
var
X, Y: inteiro
procedimento Soma(var A, B: inteiro)
início
  A ← A + 1
  B ← B + 2
  escreval("Valor de A = ", A)
  escreval("Valor de B = ", B)
  escreval("A soma de A + B = ", A+B)
FimProcedimento
início
  // Seção de Comandos
  X ← 4
  Y ← 8
  Soma (X, Y)
  escreval ("Valor de X = ", X)
  escreval ("Valor de Y = ", Y)
fimalgoritmo
  
```

Escopo	Nome	Tipo	Valor	
GLOBAL	X	I	5	Início da execução
GLOBAL	Y	I	10	Valor de A = 5
				Valor de B = 10
				A soma de A + B = 15
				Valor de X = 5
				Valor de Y = 10
				Fim da execução.

Note que, diferente do que ocorre no PARAMETRO PASSADO POR VALOR, aqui o valor final de X e Y foi alterado para o valor da soma realizada dentro do parâmetro estabelecido no procedimento chamado. No PARAMETRO PASSADO POR REFERÊNCIA qualquer alteração no parâmetro vai afetar no valor original. De modo que X passou a ter valor 5 e Y passou a ter valor 10.

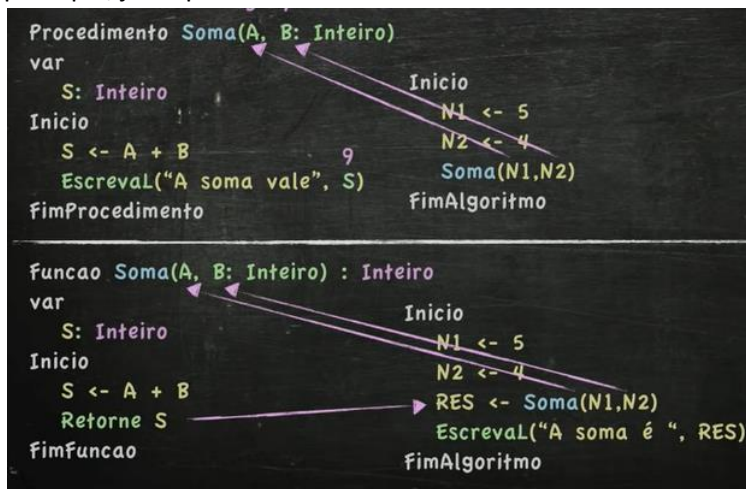
## ROTINAS

- Existem dois TIPOS DE ROTINA:

- 1) PROCEDIMENTOS: Passagem de Parâmetro por VALOR ou por REFERÊNCIA
- 2) FUNÇÕES

## FUNÇÕES

Na função o resultado que é retornado pode ser tratado da maneira que o programador quiser, permitindo que você trate a exibição e a formatação diretamente no seu programa principal, já no procedimento o retorno é mostrado na tela de uma maneira padronizada.



SOMA:

```

algoritmo "Função_SOMA"
// Função :
// Autor :
// Data : 02/02/2022
// Seção de Declarações

var
V1, V2, S: Inteiro
funcao Soma (X, Y: Inteiro) : Inteiro
início
  Retorne X + Y
FimFuncao

início
// Seção de Comandos
escreva("Digite um valor: ")
Leia (V1)
escreva("Digite outro valor: ")
Leia (V2)
S <- Soma(V1, V2)
Escreva ("A soma entre ", V1, " e ", V2, " é igual a ", S)
fimalgoritmo

```

Escopo	Nome	Tipo	Valor
GLOBAL	V1	I	3
GLOBAL	V2	I	5
GLOBAL	S	I	8

Início da execução  
 Digite um valor: 3  
 Digite outro valor: 5  
 A soma entre 3 e 5 é igual a 8  
 Fim da execução.

A FUNÇÃO é criada dentro de VAR e possui a seguinte estrutura:

**FUNCAO** soma (X, Y: Inteiro) : Inteiro

**INICIO**

**RETORNE** X + Y

**FIMFUNCAO**

VERIFICANDO SE O NÚMERO É PAR OU IMPAR:

```
// Data : 02/02/2022
// Seção de Declarações
var
N: inteiro
R: caractere
Funcao ParOuImpar (V: Inteiro): Caractere
inicio
    Se (V%2=0) entao
        Retorne "PAR"
    Senao
        Retorne "IMPAR"
    FimSe
FimFuncao

inicio
// Seção de Comandos
Escreva("Digite um número: ")
Leia (N)
R <- ParOuImpar(N)
Escreva("O número ", N, " é ", R)

finalgoritmo
```

Escopo	Nome	Tipo	Valor
GLOBAL	N	I	8
GLOBAL	R	C	"PAR"

Início da execução  
 Digite um número: 8  
 O número 8 é PAR  
 Fim da execução.

**Funcao** ParOuImpar (V: Inteiro): Caractere

Note que a Função ParOuImpar recebe um valor **inteiro** (V : Inteiro), porém o retorno terá um valor do tipo **caractere**., pois será informando ao usuário se o número digitado é PAR ou IMPAR, sendo exibido essas informações no RETORNO (R).

## CALCULANDO O FATORIAL DE UM NÚMERO:

```

algoritmo "semnome"
// Função :
// Autor :
// Data : 02/02/2022
// Seção de Declarações
var
N, F: Inteiro
Funcao Fatorial (V: Inteiro): Inteiro
var
    Cont, R: inteiro
Inicio
    R <- 1
    Para Cont <- 1 ate V faca
        R <- R * Cont
    FimPara
    Retorne R
FimFuncao
inicio
// Seção de Comandos
Escreva ("Digite um número: ")
Leia (N)
F <- Fatorial(N)
escrevaL ("O valor de " , N, " ! é igual a " , F)
fimalgoritmo

```

Escopo	Nome	Tipo	Valor	
GLOBAL	N	I	3	Início da execução Digite um número: 3 O valor de 3 ! é igual a 6 Fim da execução.
GLOBAL	F	I	6	

## ENTENDENDO O CALCULO DO FATORIAL PELO SISTEMA:

```

R <- 1
Para Cont <- 1 ate V faca
    R <- R * Cont
FimPara
Retorne R

```

O **CONT** irá começar valendo 1 (conforme estipulado pelo programador inicialmente) e vai indo/alterando seu valor até o limite escolhido pelo usuário ao digitar o número, no exemplo, até 3.

Na primeira passagem o **R** vale 1 e o **CONT** vale 1, então  $R <- 1 \times 1 = 1$ . Logo **R** passa a valer 1. (no caso, continua valendo 1, pois ele já tinha esse valor antes, estipulado pelo programador inicialmente).

Na segunda passagem o contador **CONT** vai valer 2. Então **R** (que ainda vale 1) x **CONT** ( que agora vale 2), Fica:  $1 \times 2 = 2$ . Agora **R** passa a valer 2.

Na terceira passagem, **R** vale 2 e **CONT** vale 3. Então fica:  $2 \times 3 = 6$ .

→ No final  $R <- 6$  (R recebe 6).

## STRINGS - FUNÇÕES DO SISTEMA

### - TOTAL DE LETRAS, TODAS LETRAS EM MAÍUSCULAS / MINUSCULAS:

```

algoritmo "semnome"
// Função :
// Autor :
// Data : 02/02/2022
// Seção de Declarações
var
N: caractere
inicio
// Seção de Comandos
escreval("Digite um nome: ")
Leia (N)
Escreval ("Total de letras do seu nome: ", Compr (N))
Escreval ("Seu nome em maiusculas é: ", Maiusc (N))
Escreval ("Seu nome em maiusculas é: ", Minusc (N))
finalgoritmo

```

Escopo	Nome	Tipo	Valor	
GLOBAL	N	C	"Spoc"	

Início da execução  
 Digite um nome:  
 Spoc  
 Total de letras do seu nome: 4  
 Seu nome em maiusculas é: SPOC  
 Seu nome em maiusculas é: spoc  
 Fim da execução.

### - CÓPIA

```

algoritmo "semnome"
// Função :
// Autor :
// Data : 02/02/2022
// Seção de Declarações
var
N: caractere
inicio
// Seção de Comandos
escreval("Digite um nome: ")
Leia (N)
Escreval ("Qual a primeira letras do seu nome: ", Copia (N, 1, 1))
finalgoritmo

```

Escopo	Nome	Tipo	Valor	
GLOBAL	N	C	"Cinthia"	

Início da execução  
 Digite um nome:  
 Cinthia  
 Qual a primeira letras do seu nome: C  
 Fim da execução.

**Copia (N, 1, 1))**

No caso da função COPIA, os números indicados após a variável ao qual vai ser feito a copia dos dados (no exemplo, **N**), são os indicativos de qual posição será iniciado a cópia e quantas letras após o inicio serão copiadas. Neste caso ela iniciará na primeira letra copiando apenas uma letra, ou seja, quer-se apenas a primeira letra do nome.

Caso fosse: **Copia (N, 2, 5))** então teríamos como resultado: **inthe**. Pois iniciou a cópia a partir da segunda letra do nome (i) e copiando-se 5 letras ao todo, ficando: **INTHI**.

## COMBINANDO FUNÇÕES: Maiúsculas + Última Letra

<pre> <b>algoritmo</b> "Ultima_letra_nome" // Função : // Autor : // Data : 02/02/2022 // Seção de Declarações <b>var</b> N: <u>caractere</u> <b>inicio</b> // Seção de Comandos escreval("Digite um nome: ") Leia (N) Escreval ("Qual a ultima letras do seu nome: ", Copia(Maiusc(N), Compr(N),1)) <b>fimalgoritmo</b> </pre>				
Escopo	Nome	Tipo	Valor	
GLOBAL	N	C	"Lory"	Início da execução Digite um nome: Lory Qual a ultima letras do seu nome: Y Fim da execução.

Maiúsculas: `Maiusc(N)`Última Letra: `Compr(N), 1`Obs: Primeira Letra: `(N, 1, 1)`

## - POSIÇÃO DAS LETRAS

<pre> <b>algoritmo</b> "Ultima_letra_nome" // Função : // Autor : // Data : 02/02/2022 // Seção de Declarações <b>var</b> N: <u>caractere</u> <b>inicio</b> // Seção de Comandos escreval("Digite um nome: ") Leia (N) Escreval ("Seu nome tem a letra I na posição ", Pos("I" , Maiusc(N))) <b>fimalgoritmo</b> </pre>				
Escopo	Nome	Tipo	Valor	
GLOBAL	N	C	"Thiago"	Início da execução Digite um nome: Thiago Seu nome tem a letra I na posição 3 Fim da execução.

## - NOME DE TRAS PARA FRENTE

```

algoritmo "Ultima_letra_nome"
// Função :
// Autor :
// Data : 02/02/2022
// Seção de Declarações
var
N: caractere
C: inteiro
inicio
// Seção de Comandos
escreval("Digite um nome: ")
Leia (N)
Para C<- Compr(N) ate 1 passo -1 faca
    escreva (copia(maiusc(N), C, 1))
FimPara
finalgoritmo

```

Escopo	Nome	Tipo	Valor
GLOBAL	N	C	"Cinthia"
GLOBAL	C	I	0

Início da execução  
 Digite um nome:  
 Cinthia  
 AIHTNIC  
 Fim da execução.

## VARIAVEIS COMPOSTAS

LEMBRANDO:

Variável Simples: É composta por apenas um elemento. Ex. N1: Inteiro. (composta de apenas 1 valor em uma única posição).

**VARIÁVEL COMPOSTA HOMOGÊNEA** → É aquela que possui vários elementos e todos com o mesmo tipo.

Exemplo: N: vetor[1..4] de inteiro (significa que o N vai receber valores da posição 1 até a 4)

N (nome da variável)

Vetor (indicativo de que será uma variável composta)

[1..4] (posta em colchetes, tem-se o ÍNDICE que indica quantas posições a variável terá, os números são separados por dois pontos **..** que indicam as posições, no caso, posição 1 até posição 4). OBS: ÍNDICE: Indica a posição/endereço dentro da variável.

De Inteiro (especifica o tipo da variável)

\*Aqui podemos observar que todos os valores inseridos no vetor são do mesmo tipo, ou seja, INTEIRO. Logo, trata-se de uma variável composta homogeneia.

**VARIÁVEIS COMPOSTAS HOMOGÊNEAS**

```

var
  n1: inteiro
  n2, n3, n4: inteiro
  n: vetor[1..4] de inteiro
inicio
  n[1] <- 3
  n[2] <- 5
  n[3] <- 1
  n[4] <- 0

```

Na VARIÁVEL SIMPLES, tem-se 4 variáveis para guardar 4 valores.

Já na VARIVEL COMPOSTA, tem-se uma única variável para guardar 4 valores.

N [1] <- 3 (leia-se: N na posição 1 recebe o valor 3)

```

var
  n: vetor[1..4] de inteiro
  i: inteiro

```

### 1) VARIÁVEL COMPOSTA HOMOGÊNEA UNIDIMENSIONAL:

```

algoritmo "semnome"
// Função :
// Autor :
// Data : 03/02/2022
// Seção de Declarações
var
V: vetor[1..6] de inteiro
c: inteiro
inicio
// Seção de Comandos
Para c <- 1 ate 6 faça
    escreva("Digite o ", C, "o. valor: ")
    leia (v[c]) //leia-se: leia o vetor V na posição C
FimPara

para c <- 1 ate 6 faça
    escreva ("{" , v[c], "}")
FimPara
fimalgoritmo

```

Escopo	Nome	Tipo	Valor
GLOBAL	V[1]	I	5
GLOBAL	V[2]	I	4
GLOBAL	V[3]	I	2
GLOBAL	V[4]	I	1
GLOBAL	V[5]	I	0
GLOBAL	V[6]	I	6
GLOBAL	C	I	7

Início da execução  
 Digite o 1o. valor: 5  
 Digite o 2o. valor: 4  
 Digite o 3o. valor: 2  
 Digite o 4o. valor: 1  
 Digite o 5o. valor: 0  
 Digite o 6o. valor: 6  
 { 5 } { 4 } { 2 } { 1 } { 0 } { 6 }  
 Fim da execução.

A variável **V** é composta, pois possui vários espaços/posições.

É homogeneia, pois todos os espaços/posições são do mesmo tipo. (inteiro)

E é unidimensional, pois para identificar cada posição utiliza-se o mesmo endereço (V[1] ; V[2] ; V[3], etc.). (leia-se: **V** na posição 1; **V** na posição 2; **V** na posição 3, etc.)



## PRATICA 01: LER 7 VALORES E MOSTRAR QUANTOS SÃO PARES.

```

algoritmo "semnome"
// Função :
// Autor :
// Data : 03/02/2022
// Seção de Declarações
var
val: vetor[1..7] de inteiro
i, TotPar: inteiro
inicio
// Seção de Comandos
TotPar <- 0

Para i <- 1 ate 7 faca
    escreva ("Digite o ", i, "o. valor: ")
    leia (val[i]) //leia o vetor VAL na posição i
    Se (val[i] % 2 = 0) entao
        TotPar <- TotPar + 1
    FimSe
FimPara
Escreva ("O total de números pares digitados foi de", TotPar, " números")
finalgoritmo

```

Escopo	Nome	Tipo	Valor
GLOBAL	VAL[1]	I	5
GLOBAL	VAL[2]	I	2
GLOBAL	VAL[3]	I	6
GLOBAL	VAL[4]	I	8
GLOBAL	VAL[5]	I	7
GLOBAL	VAL[6]	I	2
GLOBAL	VAL[7]	I	3
GLOBAL	I	I	8
GLOBAL	TOTPAR	I	4

Início da execução  
 Digite o 1o. valor: 5  
 Digite o 2o. valor: 2  
 Digite o 3o. valor: 6  
 Digite o 4o. valor: 8  
 Digite o 5o. valor: 7  
 Digite o 6o. valor: 2  
 Digite o 7o. valor: 3  
 O total de números pares digitados foi de 4 números  
 Fim da execução.

## MOSTRANDO AS POSIÇÕES DOS VALORES PARES:

```

// Data : 03/02/2022
// Seção de Declarações
var
val: vetor[1..7] de inteiro
i, TotPar: inteiro
inicio
// Seção de Comandos
TotPar <- 0

Para i <- 1 ate 7 faca
    escreva ("Digite o ", i, "o. valor: ")
    leia (val[i]) //leia o vetor VAL na posição i
    Se (val[i] % 2 = 0) entao
        TotPar <- TotPar + 1
    FimSe
FimPara
Escreval ("O total de números pares digitados foi de", TotPar, " números.")
para i <- 1 ate 7 faca
    Se (val[i] % 2 = 0) entao
        escreval ("Valor PAR na posição", i)
    FimSe
FimPara
finalgoritmo

```

Escopo	Nome	Tipo	Valor
GLOBAL	VAL[1]	I	7
GLOBAL	VAL[2]	I	6
GLOBAL	VAL[3]	I	5
GLOBAL	VAL[4]	I	4
GLOBAL	VAL[5]	I	3
GLOBAL	VAL[6]	I	2
GLOBAL	VAL[7]	I	1
GLOBAL	I	I	8
GLOBAL	TOTPAR	I	3

Início da execução  
 Digite o 1o. valor: 7  
 Digite o 2o. valor: 6  
 Digite o 3o. valor: 5  
 Digite o 4o. valor: 4  
 Digite o 5o. valor: 3  
 Digite o 6o. valor: 2  
 Digite o 7o. valor: 1  
 O total de números pares digitados foi de 3 números  
 Valor PAR na posição 2  
 Valor PAR na posição 4  
 Valor PAR na posição 6  
 Fim da execução.

<pre>// Data : 03/02/2022 // Seção de Declarações var val: vetor[1..7] de inteiro i, TotPar: inteiro inicio // Seção de Comandos TotPar &lt;- 0  Para i &lt;- 1 ate 7 faca     escreva ("Digite o ", i, "o. valor: ")     leia (val[i]) //leia o vetor VAL na posição i FimPara  para i &lt;- 1 ate 7 faca     Se (val[i] % 2 = 0) entao         TotPar &lt;- TotPar + 1         Escreval ("Valor PAR na posição", i)     FimSe FimPara Escreval ("O total de números pares digitados foi de", TotPar, " números.") finalgoritmo</pre>					
Escopo	Nome	Tipo	Valor	Início da execução	
GLOBAL	VAL[1]	I	4	Digite o 1o. valor: 4	
GLOBAL	VAL[2]	I	5	Digite o 2o. valor: 5	
GLOBAL	VAL[3]	I	6	Digite o 3o. valor: 6	
GLOBAL	VAL[4]	I	7	Digite o 4o. valor: 7	
GLOBAL	VAL[5]	I	8	Digite o 5o. valor: 8	
GLOBAL	VAL[6]	I	9	Digite o 6o. valor: 9	
GLOBAL	VAL[7]	I	10	Digite o 7o. valor: 10	
GLOBAL	I	I	8	Valor PAR na posição 1	
GLOBAL	TOTPAR	I	4	Valor PAR na posição 3	
				Valor PAR na posição 5	
				Valor PAR na posição 7	
				O total de números pares digitados foi de 4 números.	
				Fim da execução.	

Outra forma de fazer, com as condições de SE todas juntas, para não precisar repetir as condições já impostas, neste caso, a condição que estabelece se o número é PAR ( $N \% 2 = 0$ ), seria conforme o exemplo ao lado.

<pre>nome: vetor[1..4] de caractere n1: vetor[1..4] de real n2: vetor [ 1..4] de real m: vetor [1..4] de real i: inteiro  inicio // Seção de Comandos Para i &lt;- 1 ate 4 faca     Escreval ("ALUNO ", i)     Escreva ("Nome: ")     Leia (nome[i])     Escreva ("Primeira nota: ")     Leia (n1[i])     Escreva ("Segunda nota: ")     Leia (n2[i])     m[i] &lt;- (n1[i] + n2[i]) / 2 FimPara LimpaTela Escreval ("LISTAGEM DE ALUNOS") Escreval ("-----") Para i &lt;- 1 ate 4 faca     Escreval (nome[i]:15, m[i]) FimPara</pre>					
Escopo	Nome	Tipo	Valor	ALUNO 2	
GLOBAL	NOME[1]	C	"Gabi"	Nome: Igor	
GLOBAL	NOME[2]	C	"Igor"	Primeira nota: 8	
GLOBAL	NOME[3]	C	"Thais"	Segunda nota: 9.5	
GLOBAL	NOME[4]	C	"Geonava"	ALUNO 3	
GLOBAL	N1[1]	R	10	Nome: Thais	
GLOBAL	N1[2]	R	8	Primeira nota: 10	
GLOBAL	N1[3]	R	10	Segunda nota: 9.5	
GLOBAL	N1[4]	R	10	ALUNO 4	
GLOBAL	N2[1]	R	9.5	Nome: Geonava	
GLOBAL	N2[2]	R	9.5	Primeira nota: 10	
GLOBAL	N2[3]	R	9.5	Segunda nota: 10	
GLOBAL	N2[4]	R	10	LISTAGEM DE ALUNOS	
GLOBAL	M[1]	R	9.75	-----	
GLOBAL	M[2]	R	8.75	Gabi 9.75	
GLOBAL	M[3]	R	9.75	Igor 8.75	
GLOBAL	M[4]	R	10	Thais 9.75	
GLOBAL	I	I	5	Geonava 10	
				Fim da execução.	

### PRÁTICA 01: LISTAGEM DE TURMA (Lista com a média de cada aluno)

Note-se que foi escolhido a letra **i** na estrutura PARA a fim de remeter ao nome **ÍNDICE**, mas poderia ser qualquer outra letra.

## PRATICA 02: RELACIONAR QUAIS ALUNOS ESTÁ ACIMA DA MÉDIA

```

algoritmo "semnome"
// Função :
// Autor :
// Data : 03/02/2022
// Seção de Declarações
var
nome: vetor[1..4] de caractere
n1: vetor[1..4] de real
n2: vetor [ 1..4] de real
m: vetor [1..4] de real
SM, MT: real
i, tot: inteiro

inicio
// Seção de Comandos
Para i <- 1 ate 4 faca
    Escreva ("ALUNO ", i)
    Escreva ("Nome: ")
    Leia (nome[i])
    Escreva ("Primeira nota: ")
    Leia (n1[i])
    Escreva ("Segunda nota: ")
    Leia (n2[i])
    m[i] <- (n1[i] + n2[i]) / 2
    SM <- SM + m[i] // SM + o vetor m(média) na posição i
FimPara
MT <- SM / 4 // SM (soma das medias) dividido por 4 que é o número de alunos
LimpaTela
Escreva ("LISTAGEM DE ALUNOS")
Escreva ("-----")
Para i <- 1 ate 4 faca
    Escreva (nome[i]:15, m[i])
    Se (m[i] > MT) entao
        Tot <- Tot + 1
FimSe
FimPara
escreva ("Ao Todo temos ", Tot, " alunos acima de média da turma que é ", MT)

fimalgoritmo

```

				Início da execução
				ALUNO 1
				Nome: Pedrinho
				Primeira nota: 10
				Segunda nota: 6.9
GLOBAL	NOME[1]	C	"Pedrinho"	ALUNO 2
GLOBAL	NOME[2]	C	"Ana"	Nome: Ana
GLOBAL	NOME[3]	C	"Luana"	Primeira nota: 10
GLOBAL	NOME[4]	C	"Agatha"	Segunda nota: 8.5
GLOBAL	N1[1]	R	10	ALUNO 3
GLOBAL	N1[2]	R	10	Nome: Luana
GLOBAL	N1[3]	R	8.7	Primeira nota: 8.7
GLOBAL	N1[4]	R	5	Segunda nota: 9
GLOBAL	N2[1]	R	6.9	ALUNO 4
GLOBAL	N2[2]	R	8.5	Nome: Agatha
GLOBAL	N2[3]	R	9	Primeira nota: 5
GLOBAL	N2[4]	R	10	Segunda nota: 10
GLOBAL	M[1]	R	8.45	LISTAGEM DE ALUNOS
GLOBAL	M[2]	R	9.25	-----
GLOBAL	M[3]	R	8.85	Pedrinho 8.45
GLOBAL	M[4]	R	7.5	Ana 9.25
GLOBAL	SM	R	34.05	Luana 8.85
GLOBAL	MT	R	8.5125	Agatha 7.5
GLOBAL	I	I	5	Ao Todo temos 2 alunos acima de média da turma que é 8.5125
GLOBAL	TOT	I	2	Fim da execução.

## PRATICA 03: GUARDANDO SOMENTE NOMES QUE COMECEM COM A LETRA C

```

algoritmo "semnome"
// Função :
// Autor :
// Data : 03/02/2022
// Seção de Declarações
var
Nome: caractere
soC: vetor[1..4] de caractere
c, tot: inteiro
inicio
// Seção de Comandos
tot <- 0
Para c <- 1 ate 5 faca
    escreva ("Digite seu nome: ")
    Leia (Nome)
    Se (copia(maiusc(nome), 1, 1) = "C") entao
        tot <- tot + 1
        soC[tot] <- nome // o vetor soC na posição tot recebe o nome da pessoa
    FimSe
FimPara

LimpaTela
EScreval("LISTAGEM FINAL")
Escreval ("-----")
Para c <- 1 ate tot faca
    escreval (soC[c])
FimPara
fimalgoritmo

```

Escopo	Nome	Tipo	Valor	
GLOBAL	NOME	C	"diego"	^
GLOBAL	SOC[1]	C	"camila"	
GLOBAL	SOC[2]	C	"carina"	
GLOBAL	SOC[3]	C	"carol"	
GLOBAL	SOC[4]	C	" "	
GLOBAL	C	I	4	
GLOBAL	TOT	I	3	

Início da execução

Digite seu nome: camila

Digite seu nome: carina

Digite seu nome: carol

Digite seu nome: andre

Digite seu nome: diego

LISTAGEM FINAL

-----

camila

carina

carol

Fim da execução.

## Prática 04: ORDENAÇÃO EM VETOR

```

algoritmo "semnome"
// Função :
// Autor :
// Data : 03/02/2022
// Seção de Declarações
var
vet: vetor[1..4] de inteiro
i, j, aux: inteiro
inicio
// Seção de Comandos
Para i <- 1 ate 4 faca
    Escreva ("Digite Um valor: ")
    Leia (vet[i])
FimPara
Para i <- 1 ate 3 faca
    Para j<- i+1 ate 4 faca
        Se (vet[i] > vet[j]) entao
            aux <- vet[i]
            vet[i] <- vet[j]
            vet[j] <- aux
        FimSe
    FimPara
FimPara
para i <- 1 ate 4 faca
    escreva ("{" , vet[i], "}")
fimPara
fimalgoritmo

```

Escopo	Nome	Tipo	Valor
GLOBAL	VET[1]	I	0
GLOBAL	VET[2]	I	5
GLOBAL	VET[3]	I	6
GLOBAL	VET[4]	I	9
GLOBAL	I	I	5
GLOBAL	J	I	5
GLOBAL	AUX	I	9

Início da execução  
 Digite Um valor: 5  
 Digite Um valor: 9  
 Digite Um valor: 0  
 Digite Um valor: 6  
 { 0}{ 5}{ 6}{ 9}  
 Fim da execução.

➔ Aqui a intenção é deixar os números digitados pelo usuário em ordem crescente.

## 2) VARIAVEIS COMPOSTAS HOMOGÊNEAS MULTIDIMENSIONAIS (matrizes)

Var

M: vetor[1..3, 1..2] de inteiro

M: nome da variável

**Vetor[1..3]** : Aqui fica estabelecido quantas linhas terão, no caso, 3 linhas

**1..2]** : aqui fica estabelecido quantas colunas terão, no caso 2 colunas.

**OBS:** Separa-se o numero das linhas e o numero das colunas com a **vírgula**.

1	4		
2			
3			
	1	2	

```
var
  m: vetor[1..3, 1..2] de inteiro
inicio
  m[1,2] <- 4
```

No INÍCIO estabelecemos os valores que irão preencher as linhas e colunas, da seguinte forma: **M[1,2] <- 4**. Então, estamos dizendo ao programa que a variável **M** vai receber o valor 4 na primeira linha (1) da segunda coluna (2).

### MAIS DIMENSÕES

```
var
  m: vetor[1..3, 1..2] de inteiro
inicio
  m[1,2] <- 4
  m[2,2] <- 5
  m[3,1] <- 8
```

1	4		
2	5		
3	8		
	1	2	

#### Var

M: vetor[1..3, 1..2] de inteiro

L,C: inteiro

#### Inicio

Para l <- 1 ate 3 faça

    Para c <- 1 ate 2 faça

        Leia(m[l,c])

    FimPara

FimPara

## PRATICA 01: PREENCHER UMA MATRIZ 3 X 2 COM VALORES VIA TECLADO

```
// Data : 04/02/2022
// Seção de Declarações
var
mat: vetor[1..3, 1..2] de inteiro
l, c: inteiro
inicio
// Seção de Comandos
Para l <- 1 ate 3 faca
    Para c <- 1 ate 2 faca
        escreva ("Digite o valor da posição [", l, ", ", c, "] : ")
        leia (mat[l,c])
    FimPara
FimPara


Para l <- 1 ate 3 faca
    Para c <- 1 ate 2 faca
        Escreva (mat[l,c]:5)
    FimPara
    EscrevaL ()
FimPara
Fim
```

Escopo	Nome	Tipo	Valor
GLOBAL	MAT[1,1]	I	1
GLOBAL	MAT[1,2]	I	5
GLOBAL	MAT[2,1]	I	2
GLOBAL	MAT[2,2]	I	3
GLOBAL	MAT[3,1]	I	6
GLOBAL	MAT[3,2]	I	4
GLOBAL	L	I	4
GLOBAL	C	I	3

```
Início da execução
Digite o valor da posição [ 1, 1] : 1
Digite o valor da posição [ 1, 2] : 5
Digite o valor da posição [ 2, 1] : 2
Digite o valor da posição [ 2, 2] : 3
Digite o valor da posição [ 3, 1] : 6
Digite o valor da posição [ 3, 2] : 4
  1  5
  2  3
  6  4

Fim da execução.
```

**DICA:** para preencher automaticamente os números e não precisar digitar um por um, pode-se clicar no botão da imagem a seguir, estabelecendo de qual numero ate qual número o sistema ira usar para preencher, além de informar quantas casas decimais poderão ser usadas, no caso de números do tipo INTEIRO, coloca-se zero na casa decimal.


De:  Até:  Decimais: 

## PRÁTICA 02: LER UMA MATRIZ 3 X 3 (tbm chamada de Matriz de terceira ordem) E MOSTRAR OS NÚMEROS PARES.

A expressão: **Escreval ()** faz com que os números fiquem alinhados em forma de tabela, respeitando as linhas e colunas pré-estipulados.

```

        escreva ("Digite o valor da posição [", l, ",", c, "] : ")
        leia (valores[l,c])
    FimPara
FimPara

EscrevaL ()
TotPar <- 0
Para l <- 1 ate 3 faca
    Para c <- 1 ate 2 faca
        Se (valores[l,c] % 2 = 0) entao
            Escreva ("(", valores[l,c]:2, ")")
            TotPar <- TotPar + 1
        Senao
            Escreva (valores[l,c]:5)
        FimSe
    FimPara
FimPara
Escreva ("Ao todo foram digitados ", TotPar, " valores PARES")
fimalgoritmo

```

Escopo	Nome	Tipo	Valor	^	Início da execução
GLOBAL	VALORES[1,1]	I	10		Digite o valor da posição [ 1, 1] : 10
GLOBAL	VALORES[1,2]	I	18		Digite o valor da posição [ 1, 2] : 18
GLOBAL	VALORES[1,3]	I	16		Digite o valor da posição [ 1, 3] : 16
GLOBAL	VALORES[2,1]	I	10		Digite o valor da posição [ 2, 1] : 10
GLOBAL	VALORES[2,2]	I	18		Digite o valor da posição [ 2, 2] : 18
GLOBAL	VALORES[2,3]	I	12		Digite o valor da posição [ 2, 3] : 12
GLOBAL	VALORES[3,1]	I	20		Digite o valor da posição [ 3, 1] : 20
GLOBAL	VALORES[3,2]	I	18		Digite o valor da posição [ 3, 2] : 18
GLOBAL	VALORES[3,3]	I	2		Digite o valor da posição [ 3, 3] : 2
GLOBAL	L	I	4		{10}{18}
GLOBAL	C	I	3		{10}{18}
GLOBAL	TOTPAR	I	6		{20}{18}
					Ao todo foram digitados 6 valores PARES
					Fim da execução.

### MATRIZ IDENTIDADE DE TERCEIRA ORDEM

É uma matriz de terceira ordem (3,3, ou seja, 3 linhas e 3 colunas)

**PRÁTICA 03:**

**CRIAR UMA MATRIZ IDENTIDADE DE 3ª ORDEM**

matID

1	1		
2		1	
3			1
	1	2	3

**Diagonal principal:** Tem linha e coluna iguais.

Por exemplo, o número 1 (um) está na linha 1 da coluna 1(1,1); depois temos o número 1 na linha 2 da coluna 2(2,2) e por fim, o número 1 na linha 3 da coluna 3(3,3).



```

i, j: Inteiro
inicio
// Seção de Comandos
Para i <- 1 ate 3 faca
    Para j <- 1 ate 3 faca
        Se (i=j) entao
            mID [i,j] <- 1
        Senao
            mID [i,j] <- 0
        FimSe
    FimPara
FimPara

Para i <- 1 ate 3 faca
    Para j <- 1 ate 3 faca
        EScreva (mID[i,j]:3)
    FimPara
    EscrevaL ()
FimPara
fimalgoritmo

```

Escopo	Nome	Tipo	Valor	^	Início da execução
GLOBAL	MID[1,1]	I	1		1 0 0
GLOBAL	MID[1,2]	I	0		0 1 0
GLOBAL	MID[1,3]	I	0		0 0 1
GLOBAL	MID[2,1]	I	0		
GLOBAL	MID[2,2]	I	1		
GLOBAL	MID[2,3]	I	0		
GLOBAL	MID[3,1]	I	0		
GLOBAL	MID[3,2]	I	0		
GLOBAL	MID[3,3]	I	1		
GLOBAL	I	I	4		
GLOBAL	J	I	4		

Fim da execução.

# PRÁTICA 04: PREENCHER UMA MATRIZ DE QUARTA ORDEM:

## - Soma dos Valores da Diagonal Principal.

```

algoritmo "semnome"
// Função :
// Autor :
// Data : 04/02/2022
// Seção de Declarações
var
m: vetor[1..4, 1..4] de Inteiro
l, c, sDP: Inteiro
inicio
// Seção de Comandos
sDP <- 0 // Soma da Diagonal Principal começa com zero
Para l <- 1 ate 4 faca
    Para c <- 1 ate 4 faca
        Escreva ("Digite o valor da posição [", l, ", ", c, "]")
        Leia (m[l,c])
        Se (l=c) entao
            sDP <- sDP + m[l,c] // sDP recebe a soma da sDP + o valor q acabou de ser digitado.
        FimSe
    FimPara
FimPara

Para l <- 1 ate 4 faca // CODIGO PARA EXIBIR A MATRIZ NA TELA
    Para c <- 1 ate 4 faca
        Escreva (m[l,c]:5)
    FimPara
    EscrevaL ()
FimPara
Escreval ("A soma dos valores da Diagonal Principal é ", sDP)
fimalgoritmo

```

```

Se (l=c) entao
    sDP <- sDP + m[l,c]
FimSe

```

(l=c)

Escopo	Nome	Tipo	Valor	
GLOBAL	M[1,1]	I	17	Início da execução
GLOBAL	M[1,2]	I	17	Digite o valor da posição [ 1, 1]17
GLOBAL	M[1,3]	I	5	Digite o valor da posição [ 1, 2]17
GLOBAL	M[1,4]	I	2	Digite o valor da posição [ 1, 3]5
GLOBAL	M[2,1]	I	20	Digite o valor da posição [ 1, 4]2
GLOBAL	M[2,2]	I	18	Digite o valor da posição [ 2, 1]20
GLOBAL	M[2,3]	I	19	Digite o valor da posição [ 2, 2]18
GLOBAL	M[2,4]	I	0	Digite o valor da posição [ 2, 3]19
GLOBAL	M[3,1]	I	8	Digite o valor da posição [ 2, 4]0
GLOBAL	M[3,2]	I	14	Digite o valor da posição [ 3, 1]8
GLOBAL	M[3,3]	I	3	Digite o valor da posição [ 3, 2]14
GLOBAL	M[3,4]	I	7	Digite o valor da posição [ 3, 3]3
GLOBAL	M[4,1]	I	5	Digite o valor da posição [ 3, 4]7
GLOBAL	M[4,2]	I	10	Digite o valor da posição [ 4, 1]5
GLOBAL	M[4,3]	I	16	Digite o valor da posição [ 4, 2]10
GLOBAL	M[4,4]	I	14	Digite o valor da posição [ 4, 3]16
GLOBAL	L	I	5	Digite o valor da posição [ 4, 4]14
GLOBAL	C	I	5	17 17 5 2
GLOBAL	SDP	I	52	20 18 19 0

```

17 17 5 2
20 18 19 0
8 14 3 7
5 10 16 14

```

DESTA FORMA: 17 + 18 + 3 + 14 = 52

**PRODUTO DOS NUMEROS DA SEGUNDA LINHA:**

-Deve-se fixar a segunda linha, variando apenas as colunas.

**m[2, c]**

```

algoritmo "semnome"
// Função :
// Autor :
// Data : 04/02/2022
// Seção de Declarações
var
m: vetor[1..4, 1..4] de Inteiro
l, c, sDP, p2L: Inteiro
inicio
// Seção de Comandos
sDP <- 0 // Soma da Diagonal Principal começa com zero
p2L <- 1 // Produto da segunda linha, como é multiplicação começa com 1
Para l <- 1 ate 4 faca
    Para c <- 1 ate 4 faca
        Escreva ("Digite o valor da posição [", l, ", ", c, "]")
        Leia (m[l,c])
        Se (l=c) entao
            sDP <- sDP + m[l,c] // sDP recebe a soma da sDP + o valor q acabou de ser digitado.
        FimSe
    FimPara
FimPara

Para l <- 1 ate 4 faca // CODIGO PARA EXIBIR A MATRIZ NA TELA
    Para c <- 1 ate 4 faca
        EScreva (m[l,c]:5)
    FimPara
    EscrevaL ()
FimPara
Para c <- 1 ate 4 faca
    p2L <- p2L * m[2,c] // A linha será sempre a mesma, linha 2.
FimPara
Escreval ("A soma dos valores da Diagonal Principal é ", sDP)
Escreval ("Os valores do produto da segunda linha é ", p2L)
fimalgoritmo

Para c <- 1 ate 4 faca
    p2L <- p2L * m[2,c]
FimPara

```

```

11      6      10      2
17      3      20      13
19      2      17      10
18     19      16       6
A soma dos valores da Diagonal Principal é  37
Os valores do produto da segunda linha é  13260

```

DESTA FORMA:  $17 \times 3 \times 20 \times 13 = 13260$ .

## MOSTRANDO O MAIOR VALOR DA TERCEIRA COLUNA

-Deve-se fixar a terceira coluna, variando apenas as linhas.

```
Se (m[1,3] > mai3C) entao
    mai3C <- m[1,3]
```

FimSE

Se o “m” na posição linha (que vai variar a cada passagem) e na coluna fixa 3 (a linha varia, mas a coluna será sempre 3), for maior que o maior numero da terceira coluna ate o momento, então ele passa a ser o maior numero guardado.

```
algoritmo "semnome"
// Função :
// Autor :
// Data : 04/02/2022
// Seção de Declarações
var
m: vetor[1..4, 1..4] de Inteiro
l,c, sDP, p2L, mai3C: Inteiro
inicio
// Seção de Comandos
sDP <- 0 // Soma da Diagonal Principal começa com zero
p2L <- 1 // Produto da segunda linha, como é multiplicação começa com 1
Para l <- 1 ate 4 faca
    Para c <- 1 ate 4 faca
        Escreva ("Digite o valor da posição [", l, ", ", c, "]")
        Leia (m[l,c])
        Se (l=c) entao
            sDP <- sDP + m[l,c] // sDP recebe a soma da sDP + o valor q acabou de ser digitado.
        FimSe
    FimPara
FimPara

Para l <- 1 ate 4 faca // CODIGO PARA EXIBIR A MATRIZ NA TELA
    Para c <- 1 ate 4 faca
        EScreva (m[l,c]:5)
    FimPara
    EscrevaL ()
FimPara

Para c <- 1 ate 4 faca
    p2L <- p2L * m[2,c] // A linha será sempre a mesma, linha 2.
FimPara

Para l <- 1 ate 4 faca
    Se (m[l,3] > mai3C) entao
        mai3C <- m[l,3]
    FimSE
FimPara

Escreval ("A soma dos valores da Diagonal Pirincipal é ", sDP)
Escreval ("Os valores do produto da segunda linha é ", p2L)
EScreva (" O maior valor da terceira coluna é ", mai3C)
finalgoritmo
```

```
Para l <- 1 ate 4 faca
    Se (m[l,3] > mai3C) entao
        mai3C <- m[l,3]
    FimSE
FimPara
```

(m[1,3] > mai3C)

15	4	14	5
4	4	2	4
9	19	16	15
9	18	0	5

A soma dos valores da Diagonal Pirincipal é 40  
 Os valores do produto da segunda linha é 128  
 O maior valor da terceira coluna é 16