

Essa página diz
Formulário está OK?

Campo obrigatório 1:

Campo obrigatório 2:

JavaScript

Criação de scripts de automação de páginas WEB

Unidade Curricular: JavaScript

CONTEÚDO FORMATIVO

1. Criação de scripts de automação de páginas WEB

1. Características da linguagem
2. Introdução a lógica de programação
3. Tag Script
4. Sintaxe Básica
5. Operadores
6. Tipos de dados
7. Comparadores lógicos
8. Blocos condicionais
9. Array
10. Blocos de repetição
11. Funções ←
12. Console do navegador
13. Validação de formulário



Capacidades Técnicas

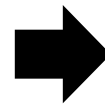
1. Automatizar documentos por meio da linguagem JavaScript
2. Integrar recursos dinâmicos da linguagem JavaScript à páginas HTML

Capacidades sociais, organizativas e metodológicas

1. Planejar tarefas no ambiente de trabalho
2. Demonstrar raciocínio lógico
3. Demonstrar iniciativa na busca por novas informações em diferentes meios

Métodos /Subalgoritmos

Subalgoritmo é um nome dado a um trecho de um algoritmo mais complexo e que, em geral, encerra em si próprio um pedaço da solução de um problema maior – o algoritmo a que ele está subordinado.



- **Subdivisão** de algoritmos complexos, facilitando o seu entendimento;
- **Estruturação** de algoritmos, facilitando principalmente a detecção de erros e a documentação de sistemas;
- **Modularização** de sistemas, que facilita a manutenção de softwares e a reutilização de subalgoritmos já implementados.

Algoritmo <nome_do_algoritmo>

<declaração_de_variáveis>

<subalgoritmos>



Início

<corpo_do_algoritmo>

Fim.

**Algoritmo
Principal**



Biblioteca → Conjunto de algoritmos.

No desenvolvimento de novos sistemas, procura-se ao máximo basear sua concepção em subalgoritmos já existentes na biblioteca, de modo que a quantidade de software realmente novo que deve ser desenvolvido é minimizada.

Algoritmo Principal por onde a execução do algoritmo sempre se inicia. Invocar os demais subalgoritmos.

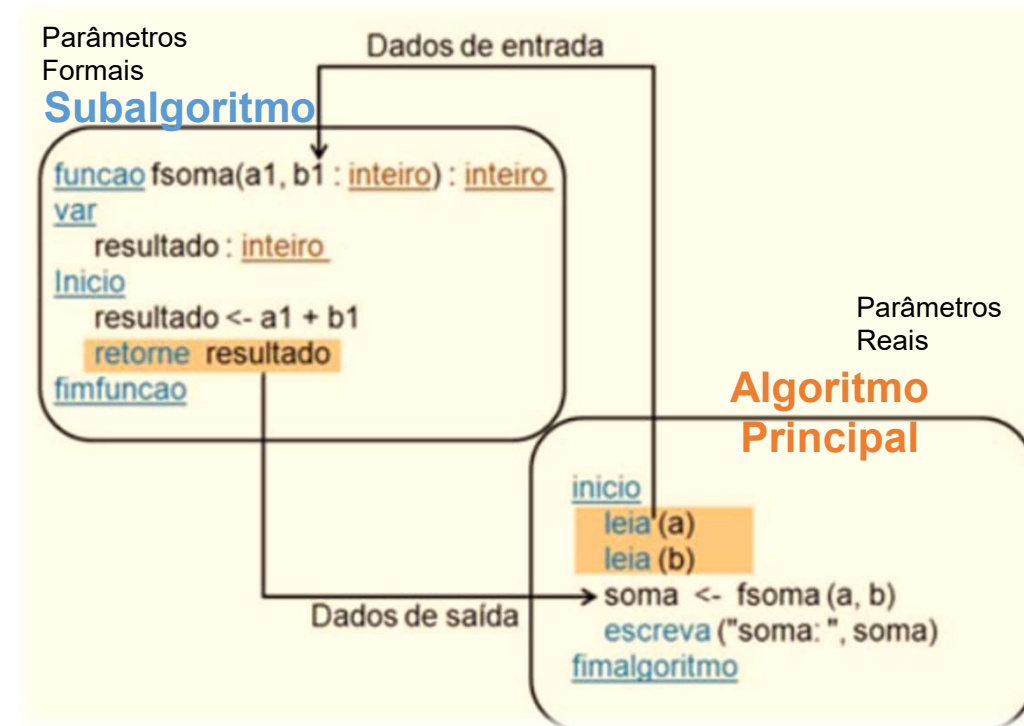
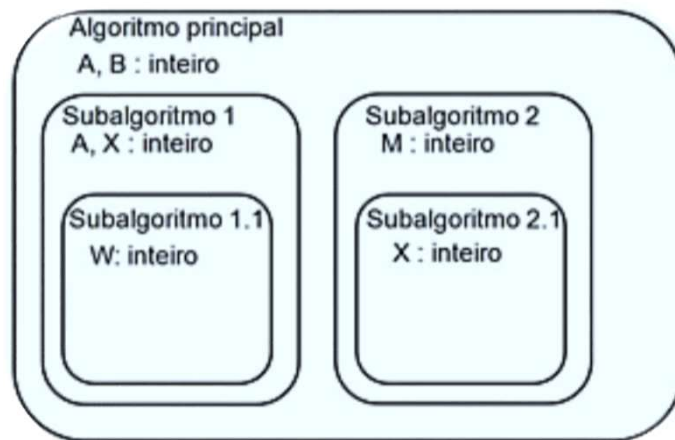
Métodos / **Subalgoritmos**

- Procedimentos → que retornam **zero (nenhum)** ou **mais** valores ao algoritmo chamador.
- Funções → que **retornam** um, e **somente um**, valor ao algoritmo chamador

➤ **Cabeçalho** → onde estão definidos o nome e o tipo do subalgoritmo, bem como os seus parâmetros e variáveis locais;

- **Parâmetros** → são canais por onde os dados são transferidos pelo algoritmo chamador a um subalgoritmo, e vice-versa (dados de entrada e de saída).
- O **nome** deve ser único.

➤ **Corpo** → onde se encontram as instruções (comandos) do subalgoritmo.



Procedimentos

Pode utilizar ou alterar valores de variáveis globais

Algoritmo "troca"

var
n1, n2 : inteiro

procedimento troca ()

var
aux : inteiro

inicio

aux := n1

n1 := n2

n2 := aux

fimprocedimento

Inicio

escreval("Digite o primeiro valor")

leia(n1)

escreval("Digite o segundo valor")

leia(n2)

escreval(n1, " - ", n2)

troca()

escreval(n1, " - ", n2)

Fimalgoritmo

procedimento <nome do procedimento> ()

//Seção de Declarações Internas

inicio

// Seção de comandos

Fimprocedimento

Console simulando o modo texto do MS-DOS

```
(A)-Você chamou o procedimento A?
(B)-Procedimento B presente, chamou?
(C)-Oi, sou o procedimento C, o B que me chamou aqui!

(D)-Sou o procedimento D e chamo todo mundo, veja:
-----
(B)-Procedimento B presente, chamou?
(C)-Oi, sou o procedimento C, o B que me chamou aqui!
-----
(A)-Você chamou o procedimento A?

>>> Fim da execução do programa !
```

```
1 Algoritmo "ExemploVariosProcedimentos"
2
3 Var
4
5 PROCEDIMENTO A ()
6 INICIO
7   escreval("(A)-Você chamou o procedimento A?")
8 FIMPROCEDIMENTO
9
10
11 PROCEDIMENTO B ()
12 INICIO
13   escreval("(B)-Procedimento B presente, chamou?")
14   c()
15 FIMPROCEDIMENTO
16
17 PROCEDIMENTO C ()
18 INICIO
19   escreval("(C)-Oi, sou o procedimento C, o B que me chamou aqui!")
20 FIMPROCEDIMENTO
21
22 PROCEDIMENTO d ()
23 INICIO
24   escreval(" ")
25   escreval(" ")
26   escreval(" ")
27   escreval("(D)-Sou o procedimento D e chamo todo mundo, veja:")
28   escreval("      ----- ")
29   B()
30   escreval("      ----- ")
31   A()
32 FIMPROCEDIMENTO
33
34 Inicio
35   A()
36   B()
37   D()
38 Fimalgoritmo
```


Procedimentos

Pode utilizar ou alterar valores de variáveis globais

Passagem de Parâmetros por Valor:

- O valor do parâmetro real é 'copiado' para o parâmetro formal.
- Não há alteração no valor do parâmetro real

```
1 Algoritmo "parametroPorValor"
2 Var
3 num1, num2 : inteiro
4
5 procedimento multiplica (x, y : inteiro)
6 Inicio
7   x := x * x
8   y := y * y
9   escreval("x: ", x, " e y: ", y)
10 Fimprocedimento
11
12 Inicio
13   escreval("Digite o primeiro número")
14   leia(num1)
15   escreval("Digite o segundo número")
16   leia(num2)
17   multiplica(num1, num2)
18   escreval("Num1: ", num1, " e Num2: ", num2)
19 fimalgoritmo
```

C:\> Console simulando o modo texto do MS-DOS

```
Digite o primeiro número
5
Digite o segundo número
3
x: 25 e y: 9
Num1: 5 e Num2: 3

>>> Fim da execução do programa !
```

Passagem de Parâmetros por Referência:

- Não há reserva de espaço extra na memória para os parâmetros formais.
- Os parâmetros reais e formais correspondentes compartilham o mesmo espaço em memória.

```
1 Algoritmo "ParametroPorreferencia"
2 Var
3 num1, num2 : inteiro
4
5 procedimento multiplica (var x, y : inteiro)
6 Inicio
7   x := x * x
8   y := y * y
9   escreval("x: ", x, " e y: ", y)
10 Fimprocedimento
11
12 Inicio
13   escreval("Digite o primeiro número")
14   leia(num1)
15   escreval("Digite o segundo número")
16   leia(num2)
17   multiplica(num1, num2)
18   escreval("Num1: ", num1, " e Num2: ", num2)
19 fimalgoritmo
```

C:\> Console simulando o modo texto do MS-DOS

```
Digite o primeiro número
20
Digite o segundo número
30
x: 400 e y: 900
Num1: 400 e Num2: 900

>>> Fim da execução do programa !
```

Procedimentos

```

1 Algoritmo "parametro1"
2 Var
3
4 procedimento quadrado (num : inteiro)
5 inicio
6
7     escreval(num * num)
8
9 fimprocedimento
10
11 Inicio
12     quadrado(5)
13
14
15 fimalgoritmo

```

```

1 Algoritmo "tabuada"
2 Var
3 numero : inteiro
4
5 procedimento tabuada (num : inteiro)
6 var
7     i : inteiro
8 inicio
9     para i de 0 ate 10 passo 1 faca
10         escreval(num, " x ", i, " = ", (num*i))
11     fimpara
12 fimprocedimento
13
14 Inicio
15     escreval("Digite um número")
16     leia(numero)
17     tabuada(numero)
18 fimalgoritmo

```

```

procedimento <nome do procedimento> (<parâmetros>)
//Seção de Declarações Internas
inicio
// Seção de comandos
Fimprocedimento

```

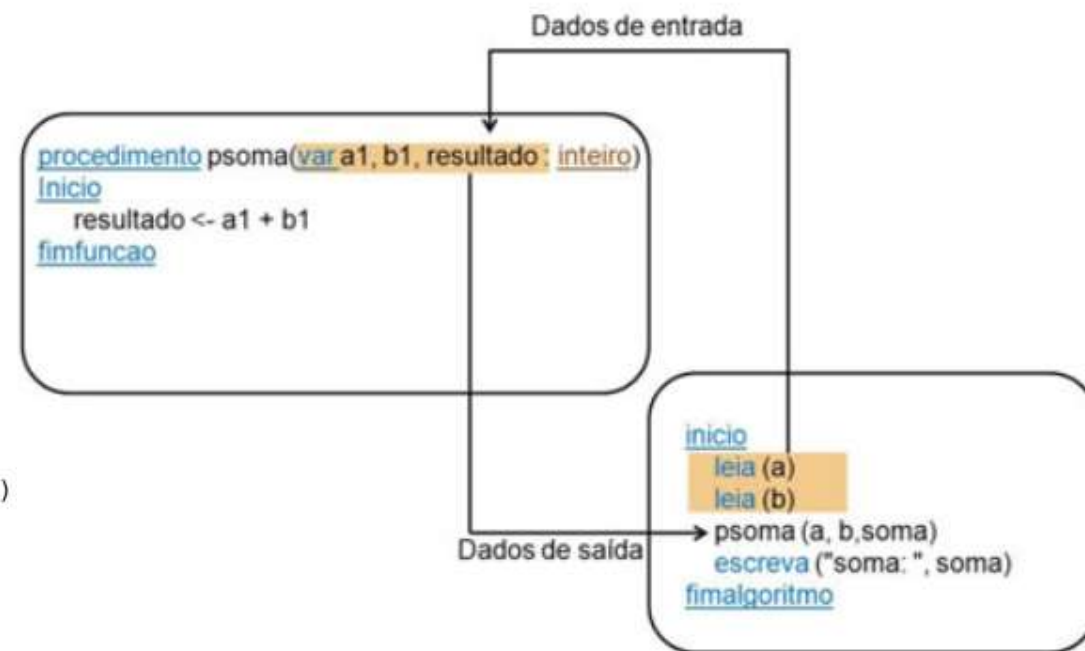
```
procedimento parametrosMesmoTipo (x, y : inteiro)
```

```
procedimento parametrosDiferentes (x, y : inteiro; z : real)
```



FIQUE ALERTA

Você deve prestar muita atenção no tipo de dado definido para o parâmetro na definição do procedimento, pois, ao chamar este procedimento e passar um valor por parâmetro, o tipo do dado deverá ser o mesmo, caso contrário uma exceção ocorrerá.



Troca de informações entre algoritmo e procedimento

Desenhista Web

Procedimentos

Vamos Praticar?

1- Faça um procedimento que lê 50 valores inteiros e retorna o maior e o menor deles.



2- Uma locadora de vídeos tem guardada, em um vetor A de 500 posições, a quantidade de filmes retirados por seus clientes durante o ano de 1997. Agora, esta locadora está fazendo uma promoção e, para cada 15 filmes retirados, o cliente tem direito a uma locação grátis. Faça um procedimento que receba o vetor A por parâmetro e retorne, também por parâmetro, um vetor contendo a quantidade de locações gratuitas a que cada cliente tem direito.

3- Construa um algoritmo que receba 3 lados de um triângulo e faça através de procedimentos diferentes:

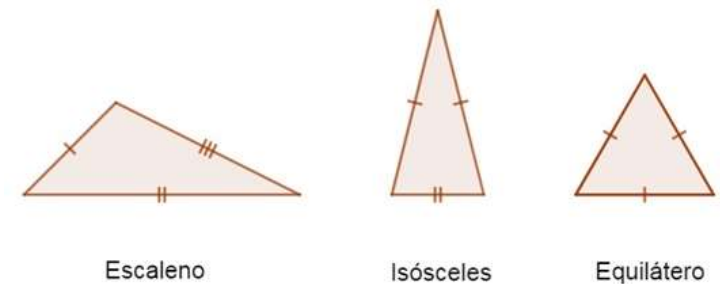
a- Verificar se é um triângulo (a medida de qualquer um dos lados seja menor que a soma das medidas dos outros dois);

b- Verificar os tipos de triângulo (diga se ele é isósceles, escaleno ou equilátero);

c- Apresente o perímetro do triângulo

4- Escreva um procedimento que recebe as 3 notas de um aluno por parâmetro e uma letra.

- Se a letra for A o procedimento calcula a média aritmética das notas do aluno,
- se for P, a sua média ponderada (pesos: 5, 3 e 2)
- e se for H, a sua média harmônica.
- A média calculada também deve retornar por parâmetro.



Procedimentos

Vamos Praticar?

1- Faça um procedimento que lê 50 valores inteiros e retorna o maior e o menor deles.

Feito com 5 para teste

```
1 Algoritmo "procedimentos_maior_menor"
2
3 Var
4 valor, vMenor, vMaior :real
5 i :inteiro
6
7
8 procedimento entradas()
9 Inicio
10   para i de 1 ate 5 passo 1 faca
11     Escreva("Numero",i," = ")
12     leia(valor)
13
14     se i=1 entao //tratando erro
15       vMenor:= valor
16       vMaior:= valor
17     fimse
18
19     compara(valor,vMenor, vMaior)
20   fimpara
21
22   escreval("-----")
23   escreval("Maior valor: ", vMaior)
24   escreval("Menor valor: ", vMenor)
25 fimprocedimento
```

```
28 procedimento compara(var valor,vMenor, vMaior: real)
29 Inicio
30   se (valor> vMaior) entao
31     vMaior := valor
32   fimse
33
34   se (valor< vMenor) entao
35     vMenor := valor
36   fimse
37 fimprocedimento
38
39
40 Inicio
41   entradas()
42 Fimalgoritmo
```

C:\ Console simulando o modo texto do MS-DOS

```
Numero 1 = 4
Numero 2 = 9
Numero 3 = 555
Numero 4 = 1
Numero 5 = 42
```

```
-----
Maior valor:  555
Menor valor:  1
```

```
>>> Fim da execução do programa !
```

Desenhista Web

Procedimentos *Vamos Praticar?*

2- Uma locadora de vídeos tem guardada, em um vetor A de 500 posições, a quantidade de filmes retirados por seus clientes durante o ano de 1997. Agora, esta locadora está fazendo uma promoção e, para cada 15 filmes retirados, o cliente tem direito a uma locação grátis.

Faça um procedimento que receba o vetor A por parâmetro e retorna, também por parâmetro, um vetor contendo a quantidade de locações gratuitas a que cada cliente tem direito.

Console simulando o modo texto do MS-DOS

```
Cliente 1 fez total de locações = 70
Cliente 2 fez total de locações = 410
Cliente 3 fez total de locações = 106
Cliente 4 fez total de locações = 51
Cliente 5 fez total de locações = 773
Cliente 6 fez total de locações = 12
-----
Cliente 1 | Locações gratuitas = 4
Cliente 2 | Locações gratuitas = 27
Cliente 3 | Locações gratuitas = 7
Cliente 4 | Locações gratuitas = 3
Cliente 5 | Locações gratuitas = 51
Cliente 6 | Locações gratuitas = 0

>>> Fim da execução do programa !
```

```
1 Algoritmo "procedimento_locacoes"
2
3
4 Var
5   a: vetor[1..6] de inteiro
6   gratuitas: vetor[1..6] de inteiro
7   i: inteiro
8
9
10 procedimento locacoes(a: real)
11 inicio
12   para i de 1 ate 6 faca
13     gratuitas[i] := ((a[i]) div 15)
14     escrevaL("Cliente ", i, " | Locações gratuitas = ", gratuitas[i])
15   fimPara
16 fimProcedimento
17
18
19 Inicio
20
21   para i de 1 ate 6 faca
22     escreva("Cliente ", i, " fez total de locações = ")
23     //leia(a[i])           usado em teste
24     a[i] := randi(999) // gera valores aleatorios
25     escrevaL(a[i])
26   fimPara
27
28   escrevaL("-----")
29   locacoes(a[i])
30 Fimalgoritmo
```

Feito com poucas posições para teste

Desenhista Web

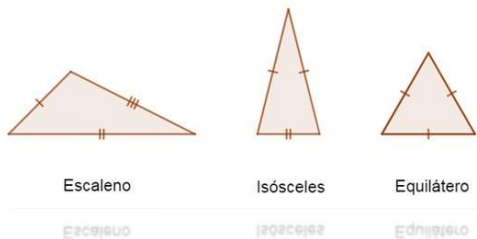
Procedimentos *Vamos Praticar?*

3- Construa um algoritmo que receba 3 lados de um triângulo e faça através de procedimentos diferentes:

a- Verificar se é um triângulo (a medida de qualquer um dos lados seja menor que a soma das medidas dos outros dois);

b- Verificar os tipos de triângulo (diga se ele é isósceles, escaleno ou equilátero);

c- Apresente o perímetro do triângulo



Console simulando o modo texto do MS-DOS

```
Digite os lados do triângulo:
Lado A: 5
Lado B: 4
Lado C: 6
-----
É um triângulo ESCALENO de Perímetro 15
>>> Fim da execução do programa !
```

```
48 Início
49 escreval("Digite os lados do triângulo:")
50 escreva("Lado A: ")
51 leia(ladoA)
52 escreva("Lado B: ")
53 leia(ladoB)
54 escreva("Lado C: ")
55 leia(ladoC)
56
57 verificar(ladoA,ladoB,ladoC)
58 Fim algoritmo
```

```
1 Algoritmo "procedimentos_triângulo"
2
3 Var
4 ladoA, ladoB, ladoC :real
5
6
7 procedimento verificar(var ladoA,ladoB,ladoC: real)
8 Início
9
10 escreval("-----")
11
12 se (ladoA< ladoB+ladoC)E (ladoB< ladoA+ladoC)E (ladoC< ladoB+ladoA) ENTAO
13 escreva("É um triângulo ")
14
15 classifica(ladoA,ladoB,ladoC) //b)
16 perimetro(ladoA,ladoB,ladoC) //c)
17 senao
18 escreval("Não é um triângulo")
19 fimSE
20 fimprocedimento
21
22 //b)
23 procedimento classifica(var ladoA,ladoB,ladoC: real)
24 Início
25
26 se (ladoA= ladoB)E (ladoA<>ladoC) ou (ladoA= ladoC)E (ladoA<>ladoB) ou (ladoC= ladoB)
27 escreva("ISÓSCELES")
28 senao
29 se (ladoA= ladoB)E (ladoB=ladoC) ENTAO
30 escreva("EQUILÁTERO")
31 senao
32 escreva("ESCALENO")
33 fimSE
34 fimSE
35
36 fimprocedimento
37
38 //c)
39 procedimento perimetro(var ladoA,ladoB,ladoC: real)
40 Início
41 escreval(" de Perímetro ", ladoA+ladoB+ladoC)
42 fimprocedimento
```

A lógica continua comparando todos os lados

Desenhista Web

Procedimentos *Vamos Praticar?*

4- Escreva um procedimento que recebe as 3 notas de um aluno por parâmetro e uma letra.

- Se a letra for A o procedimento calcula a média aritmética das notas do aluno,
- se for P, a sua média ponderada (pesos: 5, 3 e 2)
- e se for H, a sua média harmônica.
- A média calculada também deve retornar por parâmetro.

Calculadora de média harmônica online <https://miniwebtool.com/br/harmonic-mean-calculator/>

```
46 procedimento aritimerica(var nota1,nota2,nota3 :real)
47 Inicio
48   escreva("Média aritmética = ", (nota1+nota2+nota3)/3)
49 fimprocedimento
50
51
52 procedimento ponderada(var nota1,nota2,nota3 :real)
53 Inicio
54   escreva("Média ponderada = ", (nota1*5+nota2*3+nota3*2)/10)
55
56 fimprocedimento
57
58
59 procedimento harmonica(var nota1,nota2,nota3 :real)
60 Inicio
61   escreva("Média harmônica = ", 3/( (1/nota1)+(1/nota2)+(1/nota3) ) )
62 fimprocedimento
63
64
65 Inicio
66   escreva("Digite as notas e escolha em seguida a opção de cálculo")
67   entradas()
68 Fimalgoritmo
```

$$\bar{X} = \frac{(X_1 + X_2 + X_3 + \dots + X_n)}{n}$$

$$M_p = \frac{p_1 \cdot X_1 + p_2 \cdot X_2 + \dots + p_n \cdot X_n}{p_1 + p_2 + \dots + p_n}$$

$$\bar{h} = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \dots + \frac{1}{x_n}}$$

```
1 Algoritmo "procedimentos_notas"
2
3 Var
4 nota1,nota2,nota3 :real
5 resp: caracter
6
7
8 procedimento entradas()
9 Inicio
10  escreva("nota1 = ")
11  leia(nota1)
12  escreva("nota2 = ")
13  leia(nota2)
14  escreva("nota3 = ")
15  leia(nota3)
16
17  escreva("----- ")
18  escreva("A- média aritmética")
19  escreva("P- média ponderada (pesos: 5, 3 e 2)")
20  escreva("H- média harmônica")
21  leia(resp)
22
23  opcoes (resp)
24 fimprocedimento
25
26
27 procedimento opcoes(var resp :caracter)
28 Inicio
29
30  se (resp="a") ou (resp="A") entao
31    aritimerica(nota1,nota2,nota3)
32  senao
33    se (resp="p") ou (resp="P") entao
34      ponderada(nota1,nota2,nota3)
35    senao
36      se (resp="h") ou (resp="H") entao
37        harmonica(nota1,nota2,nota3)
38      senao
39        escreva("Opção inválida")
40      fimse
41    fimse
42  fimse
43 fimprocedimento
```

Funções

Funções Predefinidas

CTRL+J o Visualg mostra uma Lista de funções predefinidas (palavras-chaves)

Raizq(valor : real) : real

Nome
da função

Parâmetro :
Tipo do parâmetro

Tipo
de retorno

```
1 Algoritmo "raizQuadrada"
2 Var
3 numero, raiz : real
4
5 Inicio
6 escreval("Digite um número")
7 leia(numero)
8 raiz := Raizq(numero)
9 escreval("A raiz quadrada de ", numero, " é: ", raiz)
10 Fimalgoritmo
```

Raizq(valor : real) : real

FUNÇÃO	DESCRIÇÃO
Abs (valor : real) : real	Valor absoluto
Arccos (valor : real) : real	Arco cosseno
Arcsen (valor : real) : real	Arco seno
Arctan (valor : real) : real	Arco tangente
Asc (s : caracter) : inteiro	Retorna o código ASCII
Compr (c : caracter) : inteiro	Retorna a dimensão do caractere
Copia (c : caracter , posini, posfin : inteiro) : caracter	Copia um determinado trecho do caractere
Cos (valor : real) : real	Cosseno
Cotan (valor : real) : real	Co-tangente
Exp (<base>, <expoente>)	Potenciação
Grauprad (valor : real) : real	Converte grau para radiano
Int (valor : real) : inteiro	Converte o valor em inteiro
Log (valor : real) : real	Logaritmo de base 10
Logn (valor : real) : real	Logaritmo natural (ln)

Maiusc (c : caracter) : caracter	Converte em Maiúscula
Minusc (c : caracter) : caracter	Converte em Minúscula
Numpcarac (n : inteiro ou real) : caracter	Converte um numero inteiro ou real para caractere
Pi : real	Valor Pi
Pos (subc, c : caracter) : inteiro	Retorna a posição do caractere.
Quad (valor : real) : real	Elevado quadrado
Radpgrau (valor : real) : real	Converte Radiano para grau.
Raizq (valor : real) : real	Raiz quadrada
Rand : real	Gerador de números aleatórios entre 0 e 1
Randi (limite : inteiro) : inteiro	Gerador de números inteiros aleatórios com um limite determinado
Sen (valor : real) : real	Seno
Tan (valor : real) : real	Tangente

Funções

Funções Predefinidas

O Visualg possui uma função predefinida que recebe como parâmetro uma variável (tipo caractere) e retorna um valor (inteiro) correspondente à quantidade de caracteres existentes

```
algoritmo "Uso_de_Funcao_Predefinida"  
var  
    palavra: caractere  
    qtd: inteiro  
inicio  
    escreva("Digite uma palavra qualquer: ")  
    leia(palavra)  
    qtd <- Compr(palavra)  
    escreva("A palavra que você digitou possui ", qtd, " caracteres")  
fimalgoritmo
```

Exemplo: Criar um algoritmo que, dada uma palavra qualquer, informe ao usuário quantas letras essa palavra possui.

Compr(c : caractere) : inteiro

Nome da função	Parâmetro : Tipo do parâmetro	Tipo de retorno
-------------------	----------------------------------	--------------------

Funções *Vamos Praticar?*

Funções Predefinidas

1- Construa um algoritmo que receba o nome completo de uma pessoa e escreva esse nome todo com letras maiúsculas

2- Construa um algoritmo que receba dois valores do usuário (a e b) e realize a subtração entre eles. Apresente sempre o resultado positivo do cálculo. Ex: Para $a = 2$ e $b = 9$, o resultado de $a - b$ deverá ser 7 (e não -7).

3- Construa um algoritmo que resolva uma equação de segundo grau. Receba os três coeficientes a, b e c e informe as raízes da equação.



4 - Criar um algoritmo que, dado o nome de uma pessoa (Nome + Sobrenome), retorne apenas o sobrenome. Ex: “Alba Lopes” , retorna somente “Lopes”

5- Construa um algoritmo que receba o e-mail de uma pessoa e verifique se o e-mail é válido. Considere que o e-mail é válido se possuir um @ e, no mínimo, 5 caracteres.

Funções *Vamos Praticar?*

Funções Predefinidas

1- Construa um algoritmo que receba o nome completo de uma pessoa e escreva esse nome todo com letras maiúsculas

```
1 Algoritmo "Nome_Maiusculo"  
2  
3 Var  
4   nome : caracter  
5  
6 Inicio  
7   Escreval("Digite seu nome:")  
8   Leia(nome)  
9   nome <- Maiusc(nome)  
10  Escreval(" ")  
11  Escreval(nome)  
12 Fimalgoritmo
```

C:\ Console simulando o modo texto do MS-DOS

Digite seu nome:

chuck norris

CHUCK NORRIS

>>> Fim da execução do programa !

Funções *Vamos Praticar?*

Funções Predefinidas

2- Construa um algoritmo que receba dois valores do usuário (a e b) e realize a subtração entre eles. Apresente sempre o resultado positivo do cálculo. Ex: Para a = 2 e b = 9, o resultado de $a - b$ deverá ser 7 (e não -7).

```
1 Algoritmo "Sempre_Positivo"  
2  
3 Var  
4   a , b, resultado : real  
5  
6 Inicio  
7   Escreval("Digite o valor do numero a:")  
8   Leia(a)  
9   Escreval("Digite o valor do numero b:")  
10  Leia(b)  
11  resultado <- abs(a-b)  
12  Escreval(" ")  
13  Escreval(resultado)  
14 Fimalgoritmo
```

C:\ Console simulando o modo texto do MS-DOS

Digite o valor do numero a:

20

Digite o valor do numero b:

250

230

>>> Fim da execução do programa !

Funções *Vamos Praticar?*

Funções Predefinidas

3- Construa um algoritmo que resolva uma equação de segundo grau. Receba os três coeficientes a, b e c e informe as raízes da equação.

$$ax^2+bx+c=0$$
$$x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$$

```
1 Algoritmo "Equacao_Segundo_Grau"
2
3 Var
4   a , b, c, xUm, xDois, delta, raiz : real
5
6 Inicio
7   Escreval("Equação 2º Grau = ax² + bx + c:")
8   Escreval("Digite o valor de a:")
9   Leia(a)
10  Escreval("Digite o valor de b:")
11  Leia(b)
12  Escreval("Digite o valor de c:")
13  Leia(c)
14
15  Escreval(" ")
16  Escreval("A equação criada foi:")
17  Escreval(a, "x² +", b, "x +", c, ":")
18
19  Escreval(" ")
20  Escreval("As raízes da equação criada serão:")
21
22  delta:= (b) ^ 2 - 4 * a * c
23  raiz := RAIZQ(delta)
24  xUm := ((-b + raiz)/(2 * a))
25  xDois := ((-b - raiz)/(2 * a))
26
27  Escreval("x1 = ", xUm)
28  Escreval("x2 = ", xDois)
29
30 Fimalgoritmo
```

C:\ Console simulando o modo texto do MS-DOS

```
Equação 2º Grau = ax² + bx + c:
Digite o valor de a:
1
Digite o valor de b:
2
Digite o valor de c:
-15

A equação criada foi:
1x² + 2x + -15:

As raízes da equação criada serão:
x1 = 3
x2 = -5

>>> Fim da execução do programa !
```

Calculadora Online de Equação do 2º grau
<http://www.calculadoraonline.com.br/equacao-2-grau>

Funções *Vamos Praticar?*

Funções Predefinidas

algoritmo "Retorna_Sobrenome"

var

nomeCompleto, sobrenome: caractere
qtdCaracteres, localEspaco: inteiro

inicio

nomeCompleto <- "Alba Lopes"
qtdCaracteres <- Compr(nomeCompleto)
localEspaco <- Pos(" ", nomeCompleto)
sobrenome <- Copia(nomeCompleto, localEspaco + 1, qtdCaracteres)
escreva("O sobrenome é: ", sobrenome)

fimalgoritmo

Compr (c : caractere) : inteiro	Retorna a dimensão do caractere
--	---------------------------------

Pos (subc, c : caractere) : inteiro	Retorna a posição do caractere.
--	---------------------------------

sobrenome <- Copia(nomeCompleto, localEspaco + 1, qtdCaracteres)

Copia (c : caractere , posini, posfin : inteiro) : caractere	Copia um determinado trecho do caractere
---	--

4 - Criar um algoritmo que, dado o nome de uma pessoa (Nome + Sobrenome), retorna apenas o sobrenome. Ex: "Alba Lopes" , retorna somente "Lopes"

Funções *Vamos Praticar?*

Funções Predefinidas

5- Construa um algoritmo que receba o e-mail de uma pessoa e verifique se o e-mail é válido. Considere que o e-mail é válido se possuir um @ e, no mínimo, 5 caracteres.

```
1 Algoritmo "ValidaEmail"
2
3 Var
4   email : caracter
5   qntCaracteres, localArroba: inteiro
6
7 Inicio
8   ESCREVAL("Digite o seu email")
9   LEIA(email)
10
11  qntCaracteres := COMPR(email)
12  localArroba := POS("@", email)
13
14  SE localArroba = 0 ENTÃO
15    ESCREVAL("Não é um email. Falta o '@'")
16
17  SENÃO
18    SE qntCaracteres <= 5 ENTÃO
19      ESCREVAL("Não é um email. Precisa ter no mínimo 5 caracteres")
20
21  SENÃO
22    ESCREVAL("É um email válido")
23  FIMSE
24 FIMSE
25 Fimalgoritmo
```

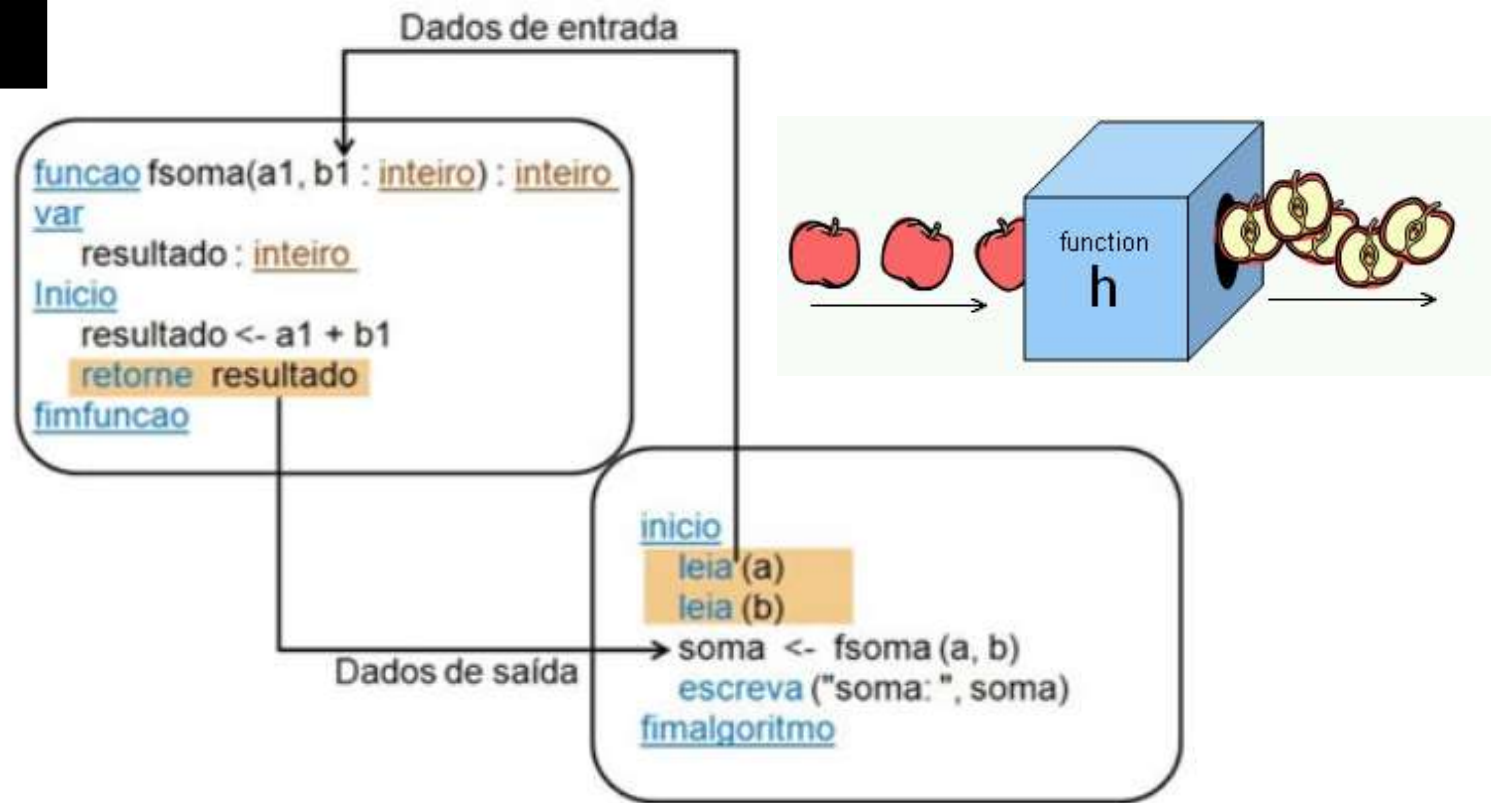
Funções

Algoritmo "funcaoSoma"

Var
num1, num2, res : inteiro

```
funcao soma(x, y : inteiro) : inteiro
var
    resultado : inteiro
inicio
    resultado := x + y
    retorne resultado
fimfuncao
```

```
Inicio
escreval("Digite o primeiro número")
leia(num1)
escreval("Digite o segundo número")
leia(num2)
res := soma(num1, num2)
escreval("Resultado: ", res)
Fimalgoritmo
```



Troca de informações entre algoritmo e função

```
funcao <nome da função> (<lista de parâmetros>) : <tipo de retorno>
//Seção de declaração de variáveis locais
inicio
//Seção de comandos
fimfuncao
```

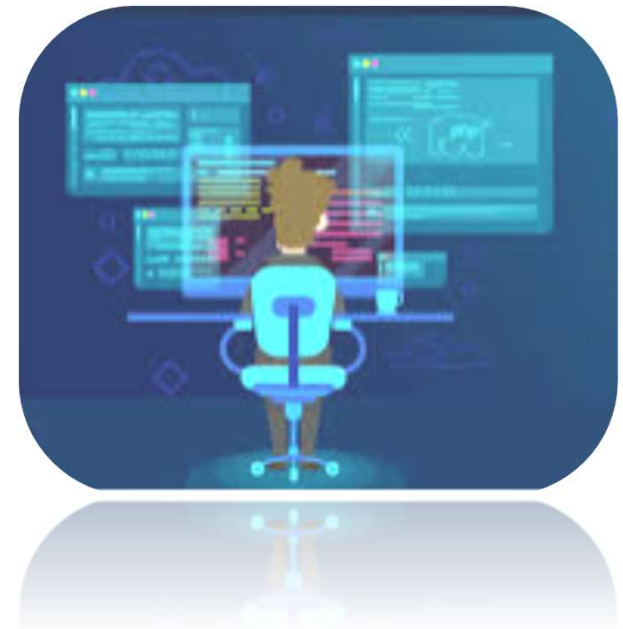
Funções

Vamos Praticar?

1- Faça duas funções que recebam por parâmetro o raio de uma esfera e calcule o seu volume ($v = \frac{4}{3} \cdot P \cdot R^3$) e a área da superfície da esfera ($A = 4 \cdot P \cdot R^2$). Use arredondamento com três casas decimais quando apresentar os resultados.

2- Escreva uma função que recebe, por parâmetro, dois valores X e Z e calcula e retorna X^Z . (sem utilizar funções ou operadores de potência prontos)

3- Faça uma função que recebe, por parâmetro, a altura (alt) e o sexo de uma pessoa e retorna o seu peso ideal. Para homens, calcular o peso ideal usando a fórmula **peso ideal = $72.7 \times \text{alt} - 58$** e, para mulheres, **peso ideal = $62.1 \times \text{alt} - 44.7$** .




Desenhista Web

Funções

Vamos Praticar?

1- Faça duas funções que recebam por parâmetro o raio de uma esfera e calcule o seu volume ($v = 4/3 \cdot P \cdot R^3$) e a área da superfície da esfera ($A = 4 \cdot P \cdot R^2$). Use arredondamento com três casas decimais quando apresentar os resultados.

```
1 Algoritmo "funcao_esfera"
2
3 Var
4   raio, volume, areaSuperficie: real
5
6
7 funcao calc_volume(r: real):real
8 var resul: real
9 Inicio
10   resul:= 4/3*PI*r^3
11   retorne resul
12 fimFuncao
13
14
15 funcao calc_area(rai: real):real
16 var resposta: real
17 Inicio
18   resposta:= 4*PI*rai^2
19   retorne resposta
20 fimFuncao
```

 Console simulando o modo texto do MS-DOS

```
Calculadora de Raio e Área da Superfície da Esfera
Raio (cm) = 5

Área da Superfície da Esfera = 314.159 cm²
Volume = 523.599 cm³

>>> Fim da execução do programa !
```

```
23 Inicio
24   Escreval("Calculadora de Raio e Área da Superfície da Esfera")
25   Escreva("Raio (cm) = ")
26   leia(raio)
27
28   volume:= calc_volume(raio)
29   areaSuperficie:= calc_area(raio)
30
31   Escreval(" ")
32   Escreval("Área da Superfície da Esfera = ", areaSuperficie:1:3, " cm²")
33   Escreval("Volume = ", volume:1:3, " cm³")
34 Fimalgoritmo
```


Desenhista Web

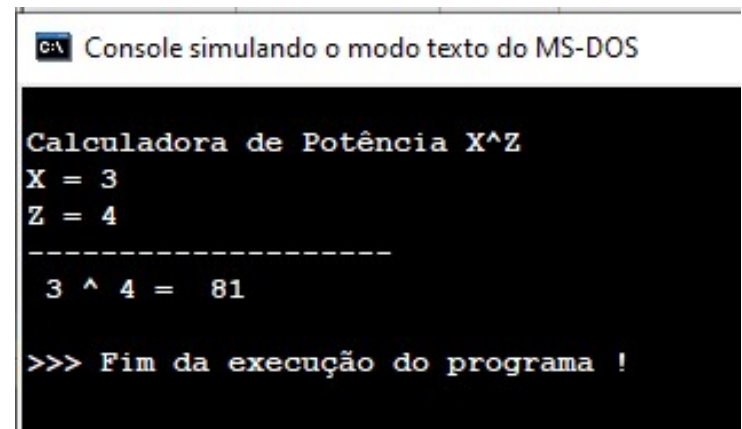
Funções

Vamos Praticar?

2- Escreva uma função que recebe, por parâmetro, dois valores X e Z e calcula e retorna X^Z . (sem utilizar funções ou operadores de potência prontos)

```
1 Algoritmo "funcao_potencia"
2
3 Var
4   x,z, potencia: inteiro
5
6
7 funcao cal_potencia(base, expoente: inteiro): inteiro
8 var
9   resultante, auxiliar: inteiro
10 Inicio
11   resultante:= base
12
13   para auxiliar de 2 ate expoente passo 1 faça
14     resultante:= resultante*base
15   fimPara
16
17   retorne resultante
18 fimFuncao
```

```
21 Inicio
22   Escreval("Calculadora de Potência X^Z")
23   Escreva("X = ")
24   leia(x)
25   Escreva("Z = ")
26   leia(z)
27
28   potencia:= cal_potencia(x,z)
29
30   Escreval("-----")
31   Escreval(x, " ^", z, " = ", potencia)
32 Fimalgoritmo
```



C:\ Console simulando o modo texto do MS-DOS

```
Calculadora de Potência X^Z
X = 3
Z = 4
-----
3 ^ 4 = 81

>>> Fim da execução do programa !
```

Desenhista Web

Funções *Vamos Praticar?*

Versão 2

2- Escreva uma função que recebe, por parâmetro, dois valores X e Z e calcula e retorna X^Z . (sem utilizar funções ou operadores de potência prontos)

```
1 Algoritmo "funcao_potencia_v2"
2
3 Var
4   x,potencia: real
5   z :inteiro
6
7
8
9
10
11
12 Inicio
13   Escreval("Calculadora de Potência X^Z")
14   Escreva("X = ")
15   leia(x)
16   Escreva("Z = ")
17   leia(z)
18
19   potencia:= cal_potencia(x,z)
20
21   Escreval("-----")
22   Escreval(x," ^",z," = ",potencia)
23 Fimalgoritmo
```

```
8 funcao cal_potencia(base :real; expoente:inteiro):real
9 var
10   resultante:real
11   auxiliar: inteiro
12 Inicio
13
14   se expoente>0 entao
15     resultante:= base
16
17     para auxiliar de 2 ate expoente passo 1 faca
18       resultante:= resultante*base
19     fimPara
20
21   senao
22     se expoente<0 entao
23
24       expoente:=expoente*(-1)
25       resultante:= (1/base)
26
27       para auxiliar de 2 ate expoente passo 1 faca
28         resultante:= resultante*(1/base)
29         escreval("auxiliar= ",auxiliar)
30         escreval("expoente= ",expoente)
31       fimPara
32
33     senao //para expoente=0
34       resultante:= 1
35     fimSe
36   fimSe
37
38   retorne resultante
39 fimFuncao
```

Desenhista Web

Funções

Vamos Praticar?

3- Faça uma função que recebe, por parâmetro, a altura (alt) e o sexo de uma pessoa e retorna o seu peso ideal. Para homens, calcular o peso ideal usando a fórmula peso ideal = $72.7 \times \text{alt} - 58$ e, para mulheres, peso ideal = $62.1 \times \text{alt} - 44.7$.

```
1 Algoritmo "funcao_peso_ideal"
2
3 Var
4   alt,ideal :real
5   sexo :caracter
6
7
8 funcao cal_peso_ideal(altura :real; sex:caracter):real
9 var
10  pesoIdeal:real
11 Inicio
12
13   se (sex="m") ou (sex="M") entao
14     pesoIdeal := 72.7 * altura - 58
15   senao
16     se (sex="f") ou (sex="F") entao
17       pesoIdeal := 62.1 * altura - 44.7
18     senao
19       Escreval(" ")
20       Escreval("Opção de sexo invalida para a operação")
21     fimSe
22   fimSe
23   retorne pesoIdeal
24
25 fimFuncao
```

Console simulando o modo texto do MS-DOS

Calculadora de Peso Ideal

Altura (m)= 1,75

Sexo = (M/F) m

Peso Ideal = 69.23 kg

>>> Fim da execução do programa !

```
27 Inicio
28   Escreval("Calculadora de Peso Ideal")
29   Escreva("Altura (m)= ")
30   leia(alt)
31   Escreva("Sexo = (M/F) ")
32   leia(sexo)
33
34
35   ideal:= cal_peso_ideal(alt,sexo)
36
37   Escreval("-----")
38   Escreval("Peso Ideal = ", ideal:1:2, " kg")
39
40 Fimalgoritmo
```

Desenhista Web

Algoritmo "notas"

Var

nota1, nota2, nota3, media : real

i : inteiro

funcao calcularMedia(n1, n2, n3, media : real) : real

var

media : real

inicio

media := (n1+n2+n3)/3

retorne media

fimfuncao

procedimento verificarSituacao(media : real)

inicio

se media >= 7 entao

escreval("Aprovado")

senao

se media >= 5 entao

escreval("Em exame")

senao

escreval("Reprovado")

fimse

fimse

fimprocedimento

Função + Procedimento

Inicio

para i de 0 ate 9 passo 1 faca

escreval("Digite a primeira nota")

leia(nota1)

escreval("Digite a segunda nota")

leia(nota2)

escreval("Digite a terceira nota")

leia(nota3)

media := calcularMedia(nota1, nota2, nota3)

verificarSituacao(media)

fimpara

fimalgoritmo

Recursividade

Quando uma função internamente chama a si própria

$$6! = 6 \times 5 \times 4 \times 3 \times 2 \times 1$$

Algoritmo "fatorial"

Var

numero, i : inteiro

Inicio

escreval("Digite um número")

leia(numero)

para i de (numero - 1) ate 1 passo -1 faca

 numero := numero * i

fimpara

escreval("O fatorial é:", numero)

fimalgoritmo

REPETIÇÃO	numero	i	numero := numero * i
1	6	5	(6 := 6 * 5) => 30
2	30	4	(30 := 30 * 4) => 120
3	120	3	(120 := 120 * 3) => 360
4	360	2	(360 := 360 * 2) => 720
5	720	1	(720 := 720 * 1) => 720

NÃO RECURSIVO

Lógica de Programação - Recursividade – 30

<https://www.youtube.com/watch?v=M7c-m2xN9FQ>

Lógica de Programação

Recursividade

Quando uma função internamente chama a si própria

$$6! = 6 \times 5 \times 4 \times 3 \times 2 \times 1$$

Algoritmo "fatorialRecursivo"

Var

numero : inteiro

funcao fatorial(num : inteiro) : inteiro

inicio

se (num = 1) ou (num = 0) entao

retorne 1

senao

retorne num * fatorial(num - 1)

fimse

fimfuncao

Inicio

escreval("Digite um número")

leia(numero)

escreval(numero, "!", fatorial(numero))

fimalgoritmo

EMPILHAMENTO DA EXECUÇÃO	DESEMPILHAMENTO DA EXECUÇÃO
Fatorial(6)	Retorne numero * fatorial(5)
Fatorial(5)	Retorne numero * fatorial(4)
Fatorial(4)	Retorne numero * fatorial(3)
Fatorial(3)	Retorne numero * fatorial(2)
Fatorial(2)	Retorne numero * fatorial(1)
Fatorial(1)	Retorne 1

DESEMPILHA FATORIAL(1)	retorne 1
Desempilha Fatorial(2)	Retorne (2 * 1) = 2
Desempilha Fatorial(3)	Retorne (3 * 2) = 6
Desempilha Fatorial(4)	Retorne (4 * 6) = 24
Desempilha Fatorial(5)	Retorne (5 * 24) = 120
Desempilha Fatorial(6)	Retorne (6 * 120) = 720

RECURSIVO

Lógica de Programação

Recursividade *Vamos Praticar?*

A sequência [0, 1, 1, 2, 3, 5, 8, 13, 21, ...] é conhecida como sequência ou série de Fibonacci e tem aplicações teóricas e práticas, na medida em que alguns padrões na natureza parecem segui-la. Pode ser obtida através da definição recursiva:

$$Fib(n) = \begin{cases} 0 & \text{se } n = 0 \\ 1 & \text{se } n = 1 \\ Fib(n-1) + Fib(n-2) & \text{se } n > 1 \end{cases}$$

Lógica de Programação

Recursividade *Vamos Praticar?*

```
1 Algoritmo "recursividade_fibonacci"
2
3 Var
4   termoFinal, i: inteiro
5
6 funcao fibonacci(n : inteiro): inteiro
7 inicio
8   se (n= 0) ou (n= 1) entao
9     retorne n
10  senão
11    retorne (fibonacci(n-2)+ fibonacci(n-1))
12  fimSe
13 fimFuncao
14
16 Inicio
17   Escreva("Digite um o termo desejado: ")
18   leia(termoFinal)
19   EscrevaL(" ")
20
21   escrevaL("Fibonacci do termo ", termoFinal, " = ", fibonacci(termoFinal))
22   EscrevaL(" ")
23
24   escrevaL("Sequencia de Fibonacci do termo ", termoFinal, " : ")
25
26   para i de 1 ate termoFinal passo 1 faca
27     escreva(" ", fibonacci(i)) //Não aproveita da recursividade
28   fimPara
29   EscrevaL(" ")
30 Fimalgoritmo
```

$$Fib(n) = \begin{cases} 0 & \text{se } n = 0 \\ 1 & \text{se } n = 1 \\ Fib(n-1) + Fib(n-2) & \text{se } n > 1 \end{cases}$$

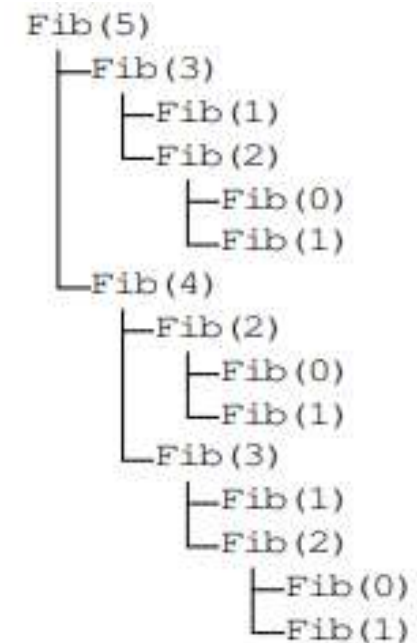
C:\> Console simulando o modo texto do MS-DOS

```
Digite um o termo desejado: 8

Fibonacci do termo 8 = 21

Sequencia de Fibonacci do termo 8:
 1 1 2 3 5 8 13 21

>>> Fim da execução do programa !
```



I - Diagrama de execução de Fib(5).

A sequência [0, 1, 1, 2, 3, 5, 8, 13, 21, ...] é conhecida como sequência ou série de Fibonacci e tem aplicações teóricas e práticas, na medida em que alguns padrões na natureza parecem segui-la. Pode ser obtida através da definição recursiva: