**CURSO DE ALGORITMOS**

**AULA #01 – PRIMEIRO ALGORITMO.**

**Algoritmos** são conjuntos **finitos** e **organizados** que, quando executados resolvem um determinado **problema**.

**AULA #02 – PRIMEIRO ALGORITMO.**

**Algoritmo** é todo tipo de rotina que você executa.

**Algoritmo computacional** são passos a serem seguidos por um módulo processador e seus respectivos usuários que, quando executados na ordem correta, conseguem realizar determinada tarefa.

Módulo processador – É tudo aquilo que pode efetuar processamento. É tudo aquilo que pode ser programável.

Usuário – É quem utiliza o programa

Realizar uma tarefa – Resolver um problema, atender alguma necessidade do usuário.

Todo algoritmo computacional começa com a **Lógica de Programação** que é escrita através de uma **linguagem de programação** que vai criar um **sistema completo**.

Lógica de programação:

Fluxograma -> Nassi Shneiderman -> **Portugol**

**Portugol** – Maneira de representar a sua lógica de programação antes de passar para uma linguagem de programação.

Ferramenta para aprender e escrever Portugol:

**VISUALG**

Comandos de saída:

1. **Escreva**: Escreve alguma coisa.

ex.:

Escreva (“Qualquer coisa”)

Tudo que estiver entre aspas é uma mensagem. Faz com que o programa escreva as mensagens uma ao lado da outra.

EscrevaL – escreve a mensagem e salta uma linha. Mesma sintaxe do escreva.

1. **Variáveis:** São espaços vazios para serem preenchidos por valores.

var

identificador: tipo

1. **Identificador:** Identificam determinadas variáveis.

6 REGRAS:

1. Deve começar com uma **letra**;

2. Os próximos podem ser **letras** ou **números**;

3. **Não** pode utilizar nenhum **símbolo**, exceto **\_** ;

4. **Não** pode conter **espaços** em branco;

5. **Não** pode conter letras com **acentos**;

6. **Não** pode ser uma **palavra reservada**. (Algoritmo, Var, Inicio, Fimalgorismo)

1. **Tipos Primitivos:**

**A**. **Inteiro**: Vai permitir que sua variável guarde apenas números inteiros.

ex.: 1 3 -5 198 0

**B**. **Real**: Vai permitir que sua variável guarde números com partes fracionárias.

ex.: 0.5 5.0 9.8 -77.3 3.1415

**C**. **Caractere**: É tudo que é colocado entre aspas.

ex.: "Gustavo" "Algoritmo" "123"

**D**. **Logico**: Guarda apenas dois valores.

ex.: Verdadeiro e Falso

1. **Atribuições:**

var

msg: caractere

msg **<-** “Olá, Mundo!”

escreva (msg)

“Olá, Mundo

1. **Comandos de saída:**

Escreva (“msg”)

Escreva (msg)

Escreva (“mensagem”, msg)

Escreval (msg)

**AULA #03 - COMANDO DE ENTRADA E OPERADORES**

1. **Comando de entrada “leia”**

Exemplo 1

Algoritmo “Meu nome”

var

Nome: Caractere

Inicio

Nome <- “Gustavo”

Escreva (“Muito prazer “, Nome)

FimAlgoritmo

Resultado do programa:

“Muito prazer Gustavo”

Exemplo 2:

Algoritmo “Meu nome”

var

Nome: Caractere

Inicio

Escreva (“Digite seu nome: “)

Leia (Nome)

Escreva (“Muito prazer”, Nome)

FimAlgoritmo

Resultado do programa:

“Muito prazer” “nome digitado pelo usuário”

Prática 01:

algoritmo "aula\_02.2\_VALORES"

var

numero1, numero2, soma: Inteiro

inicio

Escreva("Digite o 1º numero: ")

Leia(numero1)

Escreva("Digite o 2º numero: ")

Leia(numero2)

soma <- numero1 + numero2

Escreva("A soma entre ", numero1, " e ", numero2, " e igual a ", soma)

Fimalgoritmo

1. **Operadores Aritméticos:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Operadores Aritméticos** | | | |
|
| **SINAIS** | **OPERADORES** | **A <- 5** | **RESULTADO** |
| **B <- 2** |
| + | Adição | A + B | 7 |
| - | Subtração | A – B | 3 |
| \* | Multiplicação | A \* B | 10 |
| / | Divisão | A / B | 2.5 |
| \ | Divisão Inteira | A \ B | 2 |
| ^ | Exponenciação | A ^ B | 25 |
| % | Módulo | A % B | 1 |

1. **Ordem de Precedência:** Indica a ordem que os operadores serão considerados dentro de uma mesma expressão.

|  |  |
| --- | --- |
| **Ordem de Precedência** | |
|
| ( ) | Parênteses |
| ^ | Exponenciação |
| \* / | Multiplicação e Divisão |
| + - | Adição e Subtração |

Ex.:

3 + 2 / 2 = 4

(3 + 2) / 2 = 2.5

Prática 02:

algoritmo "AULA\_02.3\_MEDIA"

var

numero1, numero2: Inteiro

media: Real

inicio

Escreva("Digite o 1º numero: ")

Leia(numero1)

Escreva("Digite o 2º numero: ")

Leia(numero2)

media <- (numero1 + numero2) / 2

Escreva("A media entre ", numero1, " e ", numero2, " e igual a ", media)

Fimalgoritmo

1. **Funções Aritméticas:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Funções Aritméticas** | | | |
|
| Função | Nome | Exemplo | Resultado |
| Abs | Valor Absoluto | Abs(-10) | 10 |
| Exp | Exponenciação | Exp(3,2) | 9 |
| Int | Valor Inteiro | Int(3.9) | 3 |
| RaizQ | Raiz Quadrada | RaizQ(25) | 5 |
| Pi | Retorna Pi | Pi | 3.1415.. |
| Sen | Seno (rad) | Sen(0.523) | 0.5 |
| Cos | Cosseno (rad) | Cos(0.523) | 0.86 |
| Tan | Tangente (rad) | Tan(0.523) | 0.57 |
| GraupRad | Graus para Rad | GraupRad(30) | 0.52 |

Prática 03:

algoritmo "AULA\_02.4\_FUNCOES\_ARITMETICAS"

var

A: Real

inicio

// Valor Absoluto

Escreva("Valor Absoluto")

A <- Abs(-50)

EscrevaL(A)

// Exponenciacao

Escreva("Exponenciacao")

A <- Exp(9, 5)

EscrevaL(A)

// Raiz quadrada

Escreva("Raiz Quadrada")

A <- RaiZQ(81)

EscrevaL(A)

// Parte Inteira

Escreva("Parte Inteira")

A <- Int(RAizQ(90))

EscrevaL(A)

// Grau para Radiano

Escreva("Grau para Radiano")

A <- GraupRad(90)

EscrevaL(A)

// Seno

Escreva("Seno")

A <- Sen(GraupRad(90))

EscrevaL(A)

Fimalgoritmo

Prática 04:

algoritmo "AULA\_02.5\_CONVERSOR"

var

angulo, seno: Real

inicio

Escreva("Informe um angulo: ")

Leia(angulo)

seno <- Sen(GraupRad(angulo))

Escreva("O Seno de ", angulo, " e igual a ", seno)

Fimalgoritmo

**Aula #04 – OPERADORES LÓGICOS E RELACIONAIS**

1. **Relacionais:** Criam relações entre variáveis ou expressões. Comparar variáveis e relações e obter resultados lógicos, como verdadeiro e falso.

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

Algoritmo "Operadores Relacionais"

Var

a, b, c: inteiro

Inicio

a <- 2

b <- 3

c <- 5

escreval ("a = 2, ", "b = 3, ", "c = 5")

escreval ("a>b ", a>b)

escreval ("a<b ", a<b)

escreval ("a=b ", a=b)

escreval ("a<>b ", a<>b)

escreval ("a>2 ", a>2)

escreval ("a>=2 ", a>=2)

escreval ("c=a+b ", c=a+b)

escreval ("c>=b^a ", c>=b^a)

escreval ("c<=b^a ", c<=b^a)

escreval ("b<=c%2 ", b<=c%2)

escreval ("1=c%2 ", 1=c%2)

Fimalgoritmo

1. **Operadores Lógicos:** Retornam um valor final lógico e comparam resultados lógicos.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Operadores Lógicos** | | | | | | | | | |
|
|  | | | | | | | | | |
| **p** | **q** | **p E q** |  | **p** | **q** | **p OU q** |  |  | |
| **V** | **V** | **V** | **V** | **V** | **V** | **p** | **NÃO p** |
| **V** | **F** | **F** | **V** | **F** | **V** | **V** | **F** |
| **F** | **V** | **F** | **F** | **V** | **V** | **F** | **V** |
| **F** | **F** | **F** | **F** | **F** | **F** |  | |
|  | | | | | | | | | |

Algoritmo "Operadores Logicos"

Var

a, b, c: inteiro

Inicio

a <- 2

b <- 3

c <- 5

escreval ("a = 2, ", "b = 3, ", "c = 5")

escreval ("verdadeiro e falso - ", verdadeiro e falso)

escreval ("((a=b) ou (c>a)) - ", ((a=b) ou (c>a)))

escreval ("(nao ((a=b) ou (c>a))) - ", (nao ((a=b) ou (c>a))))

Fimalgoritmo

**Ordem de Precedência**

|  |  |
| --- | --- |
| Ordem de Precedência | |
|
| Aritméticos | ( ) |
| ^ |
| \* / |
| + - |
| Relacionais | Todos |
|
| Lógicos | E |
| OU |
| NÃO |

Algoritmo "Teste dos triangulos"

Var

L1, L2, L3: real

EQ, ES, TRI: logico

Inicio

escreva ("Digite o primeiro lado: ")

leia(L1)

escreva ("Digite o segundo lado: ")

leia(L2)

escreva ("Digite o terceiro lado: ")

leia(L3)

TRI <- (L1<L2+L3) e (L2 < L3+L1) e (L3< L1+L2)

EQ <- (L1=L2) e (L2=L3)

ES <- (L1<>L2) e (L2<>L3) e (L3<>L1)

escreval ("Pode formar um triangulo? ", TRI)

escreval ("O triangulo é EQUILATERO? ", EQ)

escreval ("O triangulo é ESCALENTO? ", ES)

Fimalgoritmo

**Aula #05 – INTRODUÇÃO AO SCRATCH**

**Aula #06 – EXERCÍCIOS DE ALGORITMO RESOLVIDOS**

1. **Exercício 01:** Creuza está fazendo aniversário, mas não sabe quantas velas colocar no bolo. Ela sabe apenas o ano do seu nascimento. Ajude Creuza a solucionar este problema.

Algoritmo "creuza\_01"

Var

ano\_atual, ano\_nasc, idade: real

Inicio

escreva ("Em que ano nós estamos? ")

leia(ano\_atual)

escreva ("Em qual ano eu nasci? ")

leia(ano\_nasc)

idade <- (ano\_atual - ano\_nasc)

escreva ("Eu tenho ", idade, " anos")

Fimalgoritmo

1. **Exercício 02:** Creuza vai viajar para os estados unidos e quer saber quanto vai levar em dólares. Ajude ela a descobrir quanto vai levar.

Algoritmo "creuza\_02"

Var

reais, dolares: real

Inicio

escreva ("Quantos reais eu tenho? R$")

leia(reais)

dolares <- (reais / 5.28)

escreva ("Eu tenho ", dolares, " dólares")

Fimalgoritmo

1. **Exercício 03:** Creuza chegou nos EUA olhou para o termômetro e viu 100 graus Fahrenheit, mas ela quer saber quanto é esta temperatura em Graus Celsius. Ajude Creuza.

Algoritmo "creuza\_03"

Var

temp\_c, temp\_f: real

Inicio

escreva ("Digite a temperatura em grau Fahrenheit: ")

leia(temp\_f)

temp\_c <- ((temp\_f - 32) / 1.8)

escreva("Tempetura está em ", temp\_c:4:1, "º")

Fimalgoritmo

1. **Exercício 04:** Creuza Comprou muita muamba e quer saber quanto vai pagar de imposto.

Algoritmo "creuza\_04"

Var

valor, imposto: real

Inicio

escreva("Quanto custou o produto? US$")

leia(valor)

imposto <- ((valor \* 60) / 100)

escreva("Vou pagar de imposto US$", imposto:5:2)

Fimalgoritmo

1. **Exercício 05:** Creuza ficou sem dinheiro e teve que pegar um empréstimo. Ela vai pagar um valor de 20% a mais do valor que ela tomar emprestado e vai poder dividir em quantas parcelas ela quiser. Ajude Creuza a saber o valor de cada parcela.

Algoritmo " creuza\_05"

Var

emprestimo, parcelas, valor\_par: real

Inicio

escreva ("Digite o valor emprestado R$")

leia(emprestimo)

escreva ("Digite em quantas parcelar vai pagar: ")

leia(parcelas)

valor\_par <- ((emprestimo + ((emprestimo \* 20) / 100)) / parcelas)

escreva ("Vou pagar ", parcelas, " parcelas de R$", valor\_par:5:2)

Fimalgoritmo

**Aula #07 – ESTRUTURAS CONDICIONAIS – PARTE 01**

Se **(estrutura condicional)** eu tiver dinheiro **(expressão)** então **(estrutura condicional)** vou fazer uma viagem para a Disney **(ação)**

**VISUALG**

1. **Condicional simples:** onde apenas existe um bloco sendo executado quando uma expressão é verdadeira.

Comando -> Expressão -> Ação

Representação:

Se (expressão) entao

Bloco

FimSe

Exemplo\_01:

Se (dinheiro >= 1000) entao

Escreva(“Partiu Disney”)

FimSe

**Prática\_01:**

algoritmo "Calculo \_Idade"

var

atual, nasc, idade: inteiro

inicio

escreva("Em que ano estamos? ")

leia(atual)

escreva("Em que ano você nasceu? ")

leia(nasc)

idade <- atual - nasc

escreva("Em ", atual, " você terá ", idade, " anos")

se idade >= 21 entao

escreva(" e já terá atingido a maioridade.")

FimSe

fimalgoritmo

1. **Condicional Composta:** Estrutura que comporta o “senao” em sua sintaxe.

Representação:

Se eu tiver dinheiro entao

Vou fazer uma viagem pra Disney

senao

Vou ficar em casa

FimSe

Exemplo\_02:

Se (expressão) entao

Bloco A

Senao

Bloco B

FimSe

**Prática\_02:**

algoritmo "Par\_ou\_Impar"

var

N: real

inicio

escreva("Digite um número: ")

leia(N)

se N % 2 = 1 entao

escreva(N, " é número ímpar.")

senao

escreva(N, " é número par.")

FimSe

Fimalgoritmo

**Prática\_03:**

algoritmo "calculo\_imc"

var

M, A, IMC: real

inicio

escreva("Digite seu peso: ")

leia(M)

escreva("Digite sua altura: ")

leia(A)

IMC <- (M / (A ^ 2))

Escreval("IMC: ", IMC:5:2)

se (IMC >= 18.5) e (IMC <= 25) entao

escreva("Peso ideal!")

senao

escreva("Fora do peso ideal!")

FimSe

Fimalgoritmo

**Exercício\_01:**

algoritmo "apto\_carteira"

var

N, A, idade: inteiro

inicio

escreval("------------------------------")

escreval("DEPARTAMENTO DE TRÂNSITO")

escreval("------------------------------")

escreva("Ano de nascimento (yyyy): ")

leia(N)

escreva("Ano atual (yyyy): ")

leia(A)

idade <- (A - N)

escreval("------------STATUS------------")

escreval("IDADE: ", idade, " ANOS")

se idade >= 18 entao

escreval("APTO A TIRAR A CARTEIRA")

senao

escreval("INAPTO A TIRAR A CARTEIRA")

FimSe

escreval("------------------------------")

fimalgoritmo

**Exercício\_02:**

algoritmo "ALUNO\_APROVADO\_REPROVADO"

var

N1, N2, M: real

inicio

escreval("--------------------------------")

escreval(" ESCOLA THIAGO CARLOS")

escreval("--------------------------------")

escreva("PRIMEIRA NOTA: ")

leia(N1)

escreva("SEGUNDA NOTA: ")

leia(N2)

escreval("--------------------------------")

M <- ((N1 + N2) / 2)

escreval("MEDIA: ", M:3:1)

se M < 7 entao

escreval("ALUNO REPROVADO")

senao

escreval("ALUNO APROVADO")

FimSe

escreva("--------------------------------")

fimalgoritmo

**Aula #08 – ESTRUTURAS CONDICIONAIS – PARTE 02**

1. **Condicionais aninhada: Encadeamento de estruturas condicionais compostas**

Representação:

Se eu tiver muito dinheiro entao

vou fazer uma viagem para a Disney

Senao Se eu tiver uma graninha entao

vou visitar minha cidade natal

Senao

vou ficar em casa

Exemplo\_01:

Se (situação 1) entao

Bloco A

senao

Se (situação 2) entao

Bloco B

senao

Bloco C

Fimse

FimSe

Exemplo\_02:

Se (dinheiro >= 10000) entao

Escreva (“Partiu Disney”)

senao

Se (dinheiro >= 5000) e (dinheiro < 10000) entao

Escreva (“Visitar família”)

senao

Escreva (“#chateado”)

FimSe

FimSe

**Prática\_04:**

algoritmo "ALUNO\_APROVADO\_REPROVADO\_02"

var

N1, N2, M: real

inicio

escreval("--------------------------------")

escreval(" ESCOLA THIAGO CARLOS")

escreval("--------------------------------")

escreva("PRIMEIRA NOTA: ")

leia(N1)

escreva("SEGUNDA NOTA: ")

leia(N2)

escreval("--------------------------------")

M <- ((N1 + N2) / 2)

escreval("MEDIA: ", M:3:1)

se M >= 7 entao

escreval("ALUNO APROVADO")

senao

se (M >= 5) e (M < 7) entao

escreval ("ALUNO EM RECUPERAÇÃO")

senao

escreval("ALUNO REPROVADO")

FimSe

FimSe

escreva("--------------------------------")

fimalgoritmo

**Prática\_05:**

algoritmo "calculo\_imc\_02"

var

M, A, IMC: real

inicio

escreva("Digite seu peso: ")

leia(M)

escreva("Digite sua altura: ")

leia(A)

IMC <- (M / (A ^ 2))

Escreval("IMC: ", IMC:5:2)

se (IMC < 17) entao

escreval("MUITO ABAIXO DO PESO")

senao

se (IMC >= 17) e (IMC < 18.5) entao

escreval("ABAIXO DO PESO")

senao

se (IMC >=18.5) e (IMC < 25) entao

escreval ("PESO IDEAL")

senao

se (IMC >= 25) e (IMC < 30) entao

escreval ("SOBREPESO")

senao

se (IMC >= 30) e (IMC < 35) entao

escreval ("OBESIDADE")

senao

se (IMC >= 35) e (IMC < 40) entao

Escreval ("OBESIDADE SEVERA")

senao

Escreval ("OBESIDADE MORBIDA")

FimSe

FimSe

FimSe

FimSe

FimSe

FimSe

fimalgoritmo

**Estruturas de Escolha Caso:** Quando se tem muitos testes com valores numéricos simples. Não serve para testar faixa de valores e só funciona para valores inteiros.

**Exemplo\_01:**

Escolha (variável)

Caso valor

Bloco A

Caso valor

Bloco B

Caso valor

Bloco C

Outro caso

Bloco D

FimEscolha

**Prática\_06:**

algoritmo "CRIANCA\_ESPERANCA"

var

D : inteiro

Valor : real

inicio

escreval ("---------------------------")

escreval (" CRIANCA ESPERANCA ")

escreval ("---------------------------")

escreval (" Muito Obrigado por ajudar ")

escreval (" [1] para doar R$10 ")

escreval (" [2] para doar R$25 ")

escreval (" [3] para doas R$50 ")

escreval (" [4] para doar outros valores ")

escreval (" [5] para cancelar ")

leia (D)

escolha D

caso 1

Valor <- 10

caso 2

Valor <- 25

caso 3

Valor <- 50

caso 4

Escreva ("Qual o valor da doacao? R$")

leia (Valor)

caso 5

Valor <- 0

FimEscolha

escreval ("---------------------------")

escreval (" SUA DOACAO FOI DE R$", Valor)

escreval (" MUITO OBRIGADO! ")

escreval ("---------------------------")

fimalgoritmo

**Prática\_07:**

algoritmo "DEPENDENTES\_FUNCIONARIO"

var

nome: caractere

sal, nsal : real

dep : inteiro

inicio

escreva ("Qual o nome do Funcionario? ")

leia (nome)

escreva ("Qual o salario do Funcionario? R$")

leia (sal)

escreva ("Qual e a quantidade de dependentes? ")

leia (dep)

Escolha dep

caso 0

nsal <- sal + (sal \* 5 / 100)

caso 1, 2, 3

nsal <- sal + (sal \* 10 / 100)

caso 4, 5, 6

nsal <- sal + (sal \* 15 / 100)

outrocaso

nsal <- sal + (sal \* 18 / 100)

FimEscolha

escreval (" O novo salario de ", nome, " sera de R$", nsal:5:2)

fimalgoritmo

**Exercício\_01:**

algoritmo "APROVEITAMENTO\_DE\_UM\_ALUNO"

var

N1, N2, M: real

inicio

escreval("--------------------------------")

escreval(" ESCOLA THIAGO CARLOS")

escreval("--------------------------------")

escreva("PRIMEIRA NOTA: ")

leia(N1)

escreva("SEGUNDA NOTA: ")

leia(N2)

escreval("--------------------------------")

M <- ((N1 + N2) / 2)

escreval("MEDIA: ", M:3:1)

se (M >= 9) entao

escreval (" APROVEITAMENTO DO ALUNO: A")

senao

se (M >= 7) entao

escreval (" APROVEITAMENTO DO ALUNO: B")

senao

se (M >= 5) entao

escreval (" APROVEITAMENTO DO ALUNO: C")

senao

se (M >= 3) entao

escreval (" APROVEITAMENTO DO ALUNO: D")

senao

se (M >= 1) entao

escreval (" APROVEITAMENTO DO ALUNO: E")

senao

escreval (" APROVEITAMENTO DO ALUNO: F")

FimSe

FimSe

FimSe

FimSe

FimSe

escreva("--------------------------------")

fimalgoritmo

**Exercício\_02:**

algoritmo "BANGU\_X\_MADUREIRA"

var

bangu, madu, dif : inteiro

inicio

escreval ("-------------------------")

escreva ("Quantos gols do BANGU? ")

leia(bangu)

escreva ("Quantos gols do MADUREIRA? ")

leia (madu)

escreval ("-------------------------")

dif <- abs(bangu - madu)

escreval ("DIFERENCA: ", dif)

se (dif < 1) entao

escreval ("STATUS: EMPATE")

senao

se (dif < 4) entao

escreval ("STATUS: NORMAL")

senao

escreval ("STATUS: GOLEADA")

FimSe

FimSe

escreval ("-------------------------")

fimalgoritmo

**Mesmo exercício usando o comando escolha.**

algoritmo "BANGU\_X\_MADUREIRA\_02"

var

bangu, madu, dif : inteiro

inicio

escreval ("-------------------------")

escreva ("Quantos gols do BANGU? ")

leia(bangu)

escreva ("Quantos gols do MADUREIRA? ")

leia (madu)

escreval ("-------------------------")

dif <- abs(bangu - madu)

escreval ("DIFERENCA: ", dif)

escolha dif

caso 0

escreval("STATUS: EMPATE")

caso 1, 2, 3

escreval("STATUS: NORMAL")

outrocaso

escreval ("STATUS: GOLEADA")

FimEscolha

escreval ("-------------------------")

fimalgoritmo

**Aula #09 – ESTRUTURAS DE REPETIÇÃO 01**

**Exemplo\_01:**

Mão <- 0

Enquanto (mão <= 5) faça

troca

mão <- mão + 1

FimEnquanto

**Sintaxe:**

Enquanto expressão faca

bloco

FimEnquanto

**Exemplo\_02:**

Enquanto não arrumar o quarto faca

castigo

Fim enquanto

Liberado

**Prática\_01:**

algoritmo "CONTE\_ATE\_10"

var

contador: inteiro

inicio

contador <- 0

Enquanto (contador <= 10) faca

EscrevaL (contador)

contador <- contador + 1

FimEnquanto

escreval ("Terminei de contar")

fimalgoritmo

**Prática\_02:**

algoritmo "CONTE\_DE\_10\_ATE\_0"

var

contador: inteiro

inicio

contador <- 10

Enquanto (contador >= 0) faca

EscrevaL (contador)

contador <- contador - 2

FimEnquanto

escreval ("Terminei de contar")

fimalgoritmo

**Prática\_03:**

algoritmo "CONTE\_DE\_0\_ATE\_ONDE\_QUISER"

var

contador, v, s: inteiro

inicio

escreva ("Digite até quanto quer contar: ")

leia (v)

escreva ("De quanto será o salto: ")

leia (s)

contador <- 0

Enquanto (contador <= v) faca

EscrevaL (contador)

contador <- contador + s

FimEnquanto

escreval ("Terminei de contar")

fimalgoritmo

**Prática\_04:**

algoritmo "SOMADOR\_NUMERICO"

var

cont, N, S: inteiro

inicio

cont <- 1

S <- 0

Enquanto (cont <= 5) faca

escreva("Digite o ", cont, "o. valor: ")

leia (N)

S <- S + N

cont <- cont + 1

FimEnquanto

escreval("A soma de todos os valores foi ", S)

fimalgoritmo

**Maior e menor valor.**

algoritmo "SOMADOR\_NUMERICO"

var

cont, N, S, maior, menor: inteiro

inicio

cont <- 1

S <- 0

Enquanto (cont <= 5) faca

escreva("Digite o ", cont, "o. valor: ")

leia (N)

se (N > maior) entao

maior <- N

FimSe

se (cont = 1) entao

menor <- N

FimSe

se (N < menor) entao

menor <- N

FimSe

S <- S + N

cont <- cont + 1

FimEnquanto

escreval("A soma de todos os valores foi ", S)

escreval("O maior valor foi: ", maior)

escreval("O menor valor foi: ", menor)

fimalgoritmo

**Prática\_05:**

Algoritmo "CONVERSAO\_MOEDA\_4\_VEZES"

Var

reais, dolares, cont: real

Inicio

cont <- 1

enquanto (cont <= 4) faca

escreva ("Quantos reais eu tenho? R$")

leia(reais)

dolares <- (reais / 5.28)

escreval ("Eu tenho ", dolares:5:2, " dólares")

cont <- cont + 1

FimEnquanto

Fimalgoritmo

**PERGUNTAR AO USUÁRIO QUANTAS CONVERSÕES SERÃO REALIZADAS**

**Algoritmo "CONVERSAO\_MOEDA\_N\_VEZES"**

Var

reais, dolares: real

cont, v: inteiro

Inicio

cont <- 1

escreva ("Quantas conversões você quer converter? ")

leia(v)

enquanto (cont <= v) faca

escreva ("Quantos reais eu tenho? R$")

leia(reais)

dolares <- (reais / 5.28)

escreval(cont, "o. conversão")

escreval ("Eu tenho ", dolares:5:2, " dólares")

cont <- cont + 1

FimEnquanto

Fimalgoritmo

**Exercício\_01:**

algoritmo "CONTADOR\_INTELIGENTE"

var

cont, I, F: inteiro

inicio

escrevaL("CONTAGEM INTELIGENTE")

ESCREVAL("---------------------")

escreva ("INICIO: ")

leia(I)

escreva ("FIM: ")

leia (F)

escreval("---------------------")

escreval(" C O N T A N D O ")

escreval("---------------------")

se (I < F) entao

cont <- I

Enquanto (cont <= F) faca

escreva(cont, ".. ")

cont <- cont + 1

FimEnquanto

senao

Se (F < I) entao

cont <- I

Enquanto (cont >= F) faca

escreva(cont, ".. ")

cont <- cont - 1

FimEnquanto

FimSe

FimSe

escreval("---------------------")

fimalgoritmo

**Exercício\_02:**

algoritmo "MELHOR\_ALUNO"

var

Q\_alunos, contador: inteiro

Nota, M\_N: real

N\_aluno, MELHOR\_ALUNO: caractere

inicio

escreval("---------------------------")

escreval(" ESCOLA SANTA PACIENCIA ")

escreval("---------------------------")

Escreva("QUANTOS ALUNOS A TURMA TEM? ")

leia(Q\_alunos)

contador <- 1

Enquanto (contador <= Q\_alunos) faca

escreval("---------------------------")

Escreval("ALUNO ", contador)

Escreva("NOME DO ALUNO: ")

leia(N\_aluno)

Escreva("NOTA DE ", N\_aluno, ":")

leia(Nota)

se (Nota > M\_N) entao

M\_N <- Nota

MELHOR\_ALUNO <- N\_aluno

FimSe

contador <- contador + 1

FimEnquanto

escreval("---------------------------")

escreva("O MELHOR APROVEITAMENTO FOI DE ", MELHOR\_ALUNO, " COM A NOTA ",M\_N)

fimalgoritmo

**Aula #10 – ESTRUTURAS DE REPETIÇÃO 02**

Repita

Bloco

Ate expressão

**Exemplo\_01:**

Enquanto não arrumar o quarto faca

castigo

FimEnquanto

Liberado

**Exemplo\_02:**

Repita

castigo

Ate arrumar o quarto

Liberado

**Prática\_01:**

algoritmo "MINHA\_TABUADA"

var

cont, N, R: inteiro

inicio

cont <- 1

escrevaL("---------------------")

escrevaL(" MINHA TABUADA ")

escrevaL("---------------------")

escreva("escolha um número: ")

leia(N)

escrevaL("---------------------")

repita

R <- N \* cont

escreval(N, " vezes ", cont, " é igual a ", R)

cont <- cont + 1

ate (cont > 10)

escrevaL("---------------------")

fimalgoritmo

**Prática\_02:**

algoritmo "CONTA\_NEGATIVOS"

var

N, C, TotN: inteiro

inicio

C <- 1

TotN <- 0

Repita

escreva("Digite um numero: ")

leia(N)

Se (N<0) entao

TotN <- TotN + 1

FimSe

C <- C + 1

Ate (C > 5)

Escreval ("Foram digitados ", TotN, " valores negativos")

Fimalgoritmo

**Prática\_03:**

algoritmo "FATORIAL"

var

contador, N, Fa: inteiro

inicio

escreval("-------------------")

escreva("Escolha um numero: ")

leia(N)

contador <- N

Fa <- 1

repita

Fa <- Fa \* contador

contador <- contador - 1

ate (contador = 0)

escreval(" ")

escreval("Fatorial de ", N, " é: ", Fa)

escreval("-------------------")

fimalgoritmo

**Prática\_03 – COM REPETIÇÃO:**

algoritmo "FATORIAL\_REPETICAO"

var

contador, N, Fa: inteiro

R: caractere

inicio

repita

escreval("-------------------")

escreva("Escolha um numero: ")

leia(N)

contador <- N

Fa <- 1

repita

Fa <- Fa \* contador

contador <- contador - 1

ate (contador = 0)

escreval(" ")

escreval("Fatorial de ", N, " é: ", Fa)

escreval("-------------------")

escreva("quer continuar [S/N]? ")

leia(R)

limpatela

ate (R = "N")

fimalgoritmo

**Prática\_04:**

algoritmo "NUMERO\_PRIMO"

var

N, contador, ContDiv: inteiro

inicio

escreva("Digite um numero: ")

leia(N)

contador <- 1

ContDiv <- 0

repita

escreva(contador)

Se(N % contador = 0) entao

ContDiv <- ContDiv + 1

FimSe

contador <- contador + 1

ate (contador > N)

Escreval(" ")

Se (ContDiv = 2) entao

escreva(N, " e PRIMO!")

Senao

escreva(N, " nao e PRIMO")

FimSe

Fimalgoritmo

**Exercício\_01:**

algoritmo "SUPER\_CONTADOR"

var

C, X, N: inteiro

inicio

Repita

escreval("==================")

escreval("| M E N U |")

escreval("==================")

escreval("| [1] De 1 a 10 |")

escreval("| [2] De 10 a 1 |")

escreval("| [3] Sair |")

escreval("==================")

leia(N)

escolha(N)

Caso 1

C <- 1

repita

escreva(C, "..")

C <- C + 1

ate (C > 10)

escreval(" ")

Caso 2

X <- 10

repita

escreva(X, "..")

X <- X - 1

ate (X < 1)

escreval(" ")

Caso 3

escreval("SAINDO...")

FimEscolha

Ate (N = 3)

Fimalgoritmo

**Exercício\_02:**

algoritmo "ESCOLHENDO\_PESSOAS"

var

sexo, resp: caractere

idade, cabelo, homem, mulher: inteiro

inicio

homem <- 0

mulher <- 0

Repita

escreval("==========================")

escreval(" SELETOR DE PESSOAS ")

escreval("==========================")

escreva("Qual o Sexo? [M/F] ")

leia(sexo)

escreva("qual a idade? ")

leia(idade)

escreval("Qual a cor do Cabelo? ")

escreval("--------------------------")

escreval("[1] Preto")

escreval("[2] Castanho")

escreval("[3] Loiro")

escreval("[4] Ruivo")

leia(cabelo)

escreva("Quer continuar? [S/N] ")

leia(resp)

limpatela

Se (sexo = "M") entao

Se (idade > 18) e (cabelo = 2) entao

homem <- homem + 1

Senao

homem <- homem + 0

FimSe

FimSe

Se (sexo = "F") entao

Se (idade>= 25) e (idade <= 30) entao

Se (cabelo = 3) entao

mulher <- mulher + 1

Senao

mulher <- mulher + 0

FimSe

FimSe

FimSe

Ate (resp = "N")

escreval("==========================")

escreval(" RESULTADO FINAL ")

escreval("==========================")

escreval("Total de homens com mais de 18 e cabelos castanhos: ", homem)

escreval("Total de mulheres entre 25 e 30 e cabelos loiros: ", mulher)

fimalgoritmo

**Aula #11 – ESTRUTURAS DE REPETIÇÃO 03**

Revisando estruturas de repetição:

Sintaxe\_01:

Enquanto nao arrumar o quarto faca

Castigo

FimEnquanto

Liberado

Sintaxe\_02:

Repita

Castigo

Ate arrumar o quarto

Liberado

Exemplo\_01:

C <- 1

Enquanto (C<=10) faca

Escreval(C)

C<-C+1

FimEnquanto

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

C <- 1

Repita

Escreval(C)

C <- C + 1

Ate (C>10)

**Estrutura de Repetição “Para”**

Para variável <- inicio ate fim [passo salto] faca

Bloco

FimPara

Sintaxe:

Para C <- 1 ate 10 passo 1 faca

EscrevaL(C)

FimPara

Prática\_01:

algoritmo "CONTAR\_1\_ATE\_10\_PARA"

var

C: inteiro

inicio

EscrevaL("De 1 ate 10")

Para C <- 1 ate 10 passo 1 faca

Escreva(C, "..")

FimPara

Escreval(" ")

Escreval("De 10 ate 0")

Para C <- 10 ate 1 passo -1 faca

Escreva(C, "..")

FimPara

Fimalgoritmo

Prática\_02:

algoritmo "SOMA\_VALORES\_PARA"

var

N, S, C: inteiro

inicio

S <- 0

Para C <- 1 ate 5 faca

EscrevaL("Digite um valor: ")

Leia(N)

S<- S + N

FimPara

EscrevaL("A soma dos valores e ", S)

Fimalgoritmo

Prática\_03:

algoritmo "VALORES\_PARES"

var

n, v: inteiro

inicio

Escreval("PROGRESSIVO")

Escreva("Digite um valor: ")

leia(v)

Para n <- 0 ate v passo 2 faca

Escreva(n, "..")

FimPara

EscrevaL(" ")

EscrevaL("REGRESSIVO")

Se (v % 2 = 1) entao

v <- v - 1

FimSe

Para n <- v ate 0 passo -2 faca

Escreva(n, "..")

FimPara

Fimalgoritmo

algoritmo "QUANTOS\_ENTRE\_0\_E\_10"

var

s, v, C, impar: inteiro

inicio

s <- 0

impar <- 0

Para C <- 1 ate 6 faca

Escreva("Digite um valor: ")

leia(v)

Se (v >= 0) e (v <= 10) entao

s <- s + 1

FimSe

Se (v % 2 = 1) entao

impar <- impar + v

FimSe

FimPara

escreval("Existem ", s, " valores entre 0 e 10")

escreval("A soma dos valores impares foi: ", impar)

fimalgoritmo

Prática\_04:

algoritmo "COMBINACOES"

var

C1, C2 : inteiro

inicio

Para C1 <- 1 ate 3 faca

Para C2 <- 1 ate 3 faca

EscrevaL (C1, C2)

FimPara

FimPara

Fimalgoritmo

algoritmo "COMBINACOES"

var

C1, C2, V1, V2 : inteiro

inicio

escreva("Digite o primeiro valor: ")

leia(V1)

escreva("Digite o segundo valor: ")

leia (V2)

Escreval(" ")

Para C1 <- V1 ate V2 faca

Para C2 <- V1 ate V2 faca

EscrevaL (C1, ",", C2)

FimPara

EscrevaL(" ")

FimPara

Fimalgoritmo

Exercício\_01:

algoritmo "SEQUENCIA\_DE\_FIBONACCI"

var

X1, X2, X3, S: inteiro

inicio

X1 <- 0

X2 <- 1

Escreva(X1)

Escreva(X2)

Para S <- 1 ate 13 faca

X3 <- X1 + X2

Escreva(X3)

X1 <- X2

X2 <- X3

FimPara

Fimalgoritmo

Exercício\_02:

algoritmo "ANALISADOR\_DE\_VALORES"

var

valor, C, soma, divi, SomaPar: inteiro

nulo: real

inicio

soma <- 0

divi <- 0

nulo <- 0

SomaPar <- 0

Para C <- 1 ate 5 faca

escreva("Digite o ", C, "o. valor: ")

leia(valor)

soma <- soma + valor

Se (valor % 5 = 0) entao

divi <- divi + 1

FimSe

Se (valor = 0) entao

nulo <- nulo + 1

FimSe

Se (valor % 2 = 0) entao

SomaPar <- SomaPar + valor

FimSe

FimPara

EscrevaL("A soma dos valores e: ", soma)

EscrevaL("A media entre os valores e: ", soma / C)

EscrevaL("Os valores divisiveis por cinco: ", divi)

EscrevaL("Valores nulos: ", nulo)

EscrevaL("A soma dos valores pares: ", SomaPar)

Fimalgoritmo

**Aula #12 – PROCEDIMENTOS**

**ROTINA**

algoritmo "DETECTOR\_PESADO"

var

I: Inteiro

N, Pesado: Caractere

P, Mai: Real

Procedimento Topo()

Inicio

LimpaTela

EscrevaL("------------------------------------")

EscrevaL(" D E T E C T O R DE P E S A D O ")

EscrevaL(" Maior Peso ate agora: ", Mai, "kg")

EscrevaL("------------------------------------")

FimProcedimento

inicio

Topo()

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

Topo()

FimPara

Topo()

EscrevaL("A pessoa mais pesada foi ", Pesado, ", com ", Mai, " quilos.")

Fimalgoritmo

**PASSAGEM DE PARÂMETRO**

**TIPO 1: POR VALOR**

Procedimento Soma (A, B: Inteiro)

Inicio

EscrevaL(“Recebi o valor”, A)

EscrevaL(“Recebi o valor”, B)

EscrevaL(“A soma vale”, A+B)

FimProcedimento

algoritmo "FAZ\_SOMA\_PROCEDIMENTO"

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 e ", a + b)

FimProcedimento

inicio

x <- 5

y <- 3

Soma(x, y)

Fimalgoritmo

algoritmo "PAR\_OU\_IMPAR\_PROCEDIMENTO"

var

N: Inteiro

Procedimento ParOuImpar(V: Inteiro)

inicio

Se (V % 2 = 0) entao

Escreva("o numero ", V, " e PAR")

Senao

Escreva("O numero ", V, " e IMPAR")

FimSe

FimProcedimento

inicio

Escreva("Digite um numero: ")

Leia(N)

ParOuImpar(N)

Fimalgoritmo

**ESCOPO**: Local onde determinada variável vai funcionar.

**TIPO 2: POR REFERÊNCIA**

Procedimento Soma(var A, B: Inteiro)

Inicio

A <- A + 1

B <- B + 2

EscrevaL(“A soma vale”, A+B)

FimProcedimento

Inicio

X <- 4

Y <- 8

Soma(X, Y)

EscrevaL(X, Y)

FimAlgoritmo

algoritmo "SEQUENCIA\_DE\_FIBONACCI\_REFERENCIA"

var

C, X1, X2, X3: inteiro

procedimento ProximoFibonacci (var A, B: Inteiro)

var C: Inteiro

inicio

C <- A + B

Escreva (C)

A <- B

B <- C

FimProcedimento

inicio

X1 <- 0

Escreva(X1)

X2 <- 1

Escreva(X2)

Para C <- 3 ate 10 faca

ProximoFibonacci (X1, X2)

FimPara

Fimalgoritmo

**Aula #13 – FUNÇÕES**

algoritmo "SOMA\_FUNCAO"

var

V1, V2, S: Inteiro

Funcao soma(X, Y: Inteiro): Inteiro

Inicio

Retorne X + y

FimFuncao

inicio

Escreva("Digite o primeiro valor: ")

Leia(V1)

Escreva("Digite o segundo valor: ")

Leia(V2)

S <- Soma(V1, V2)

EscrevaL("A soma entre ", V1, " e ", V2, " e igual a ", S)

fimalgoritmo

algoritmo "ParImpar\_Funcao"

var

N: Inteiro

R: Caractere

Funcao ParOuImpar(V: Inteiro): Caractere

Inicio

Se (V % 2 = 0) entao

Retorne "PAR"

Senao

Retorne "IMPAR"

FimSe

FimFuncao

inicio

Escreva("Digite um numero: ")

Leia(N)

R <- ParOuImpar(N)

Escreva (" O numero ", N, " e um valor ", R)

Fimalgoritmo

algoritmo "FatorialFuncao"

var

N, F: Inteiro

Funcao Fatorial(V: Inteiro): Inteiro

Var

C, R: Inteiro

Inicio

R <- 1

Para C <- 1 ate V faca

R <- R \* C

FimPara

Retorne R

FimFuncao

inicio

Escreva("Digite um numero: ")

Leia(N)

F <- Fatorial(N)

EscrevaL("O valor de ", N, "! e igual a ", F)

Fimalgoritmo

PASSAGEM DE PARAMETRO:

algoritmo "SEQUENCIA\_DE\_FIBONACCI\_FUNCAO"

var

I, T1, T2, T3: inteiro

Funcao ProximoFibonacci (var N1, N2: Inteiro): Inteiro

var

N3: Inteiro

inicio

N3 <- N1 + N2

N1 <- N2

N2 <- N3

Retorne N3

FimFuncao

inicio

T1 <- 0

Escreva(T1)

T2 <- 1

Escreva(T2)

Para T3 <- 3 ate 10 faca

T3 <- ProximoFibonacci (T1, t2)

Escreva(T3)

FimPara

Fimalgoritmo

* **FUNÇÕES DO VISUALG:**

**VALORES DE CARACTERE:**

algoritmo "ANALIZADOR\_DE\_STRINGS"

var

n: caractere

c: inteiro

inicio

Escreva("Digite seu nome: ")

Leia(n)

EscrevaL("Total de letras do seu nome: ", Compr(n))

EscrevaL("Seu nome em maiusculas e ", Maiusc(n))

EscrevaL("Seu nome em minusculas e ", Minusc(n))

EscrevaL("A primeira letra do seu nome e ", Copia(n, 1, 1))

EscrevaL("A ultima letra do seu nome e ", Copia(Maiusc(n), Compr(n), 1))

EscrevaL("Seu nome tem a letra A na posicao ", Pos("A", Maiusc(n)))

EscrevaL("O codigo da letra A e ", Asc("A"))

EscrevaL("A letra de codigo 65 e ", Carac(65))

Para c <- Compr(n) ate 1 passo -1 faca

Escreva(Copia(Maiusc(n), C, 1))

FimPara

Fimalgoritmo

**AULA 14 – VETORES – VARIÁVEIS COMPOSTAS, PARTE 1**

algoritmo "teste\_vetor"

var

v: vetor[1..6] de inteiro

c: inteiro

inicio

Para c <- 1 ate 6 faca

Escreva("Digite o ", c, "o. valor: ")

Leia (v[c])

FimPara

Para c <- 1 ate 6 faca

Escreva ( "{", v[c], "}")

FimPara

Fimalgoritmo

Prática 01:

algoritmo "valores\_pares\_vetores"

var

v: vetor[1..7] de inteiro

c, p: inteiro

inicio

p <- 0

Para c <- 1 ate 7 faca

Escreva("Digite o ", c, "o. valor: ")

Leia(v[c])

Se (v[c] % 2 = 0) entao

p <- p + 1

FimSe

FimPara

Escreval(p, " sao numeros PARES")

Fimalgoritmo

algoritmo "valores\_pares\_vetores\_02"

var

v: vetor[1..7] de inteiro

c, p: inteiro

inicio

p <- 0

Para c <- 1 ate 7 faca

Escreva("Digite o ", c, "o. valor: ")

Leia(v[c])

FimPara

Para c <- 1 ate 7 faca

Se (v[c] % 2 = 0) entao

p <- p + 1

EscrevaL("Valor PAR na ", c, "o. posicao")

FimSe

FimPara

Escreval(p, " sao numeros PARES")

Fimalgoritmo

Prática 02:

algoritmo "listagem\_da\_turma"

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

i: inteiro

inicio

Para i <- 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] <- (n1[i] + n2[i]) / 2

FimPara

Limpatela

EscrevaL(" LISTAGEM DE ALUNOS ")

EscrevaL("--------------------")

Para i <- 1 ate 4 faca

EscrevaL(nome[i]:15, m[i]:4:1)

FimPara

Fimalgoritmo

algoritmo "listagem\_da\_turma\_02"

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

i: inteiro

inicio

Para i <- 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] <- (n1[i] + n2[i]) / 2

FimPara

Limpatela

EscrevaL(" LISTAGEM DE ALUNOS ")

EscrevaL("--------------------")

Para i <- 1 ate 4 faca

EscrevaL(nome[i]:15, m[i]:4:1)

FimPara

EscrevaL(" STATUS DOS ALUNOS ")

EscrevaL("--------------------")

Para i <- 1 ate 4 faca

Se (m[i] >= 7) entao

EscrevaL(nome[i]:15, " APROVADO")

Senao

EscrevaL(nome[i]:15, " REPROVADO")

FimSe

FimPara

Fimalgoritmo

algoritmo "listagem\_da\_turma\_03©"

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

Para i <- 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] <- (n1[i] + n2[i]) / 2

SM <- SM + m[i]

FimPara

MT <- SM/4

Limpatela

EscrevaL(" LISTAGEM DE ALUNOS ")

EscrevaL("--------------------")

Para i <- 1 ate 4 faca

EscrevaL(nome[i]:15, m[i]:4:1)

Se (m[i] > MT) entao

Tot <- Tot + 1

FimSe

FimPara

EscrevaL("Ao todo temos ", Tot, " alunos acima da media da turma que e ", MT:4:1)

Fimalgoritmo

Prática 03:

algoritmo "SO\_COM\_C"

var

nome: Caractere

soC: vetor[1..10] de Caractere

c, tot: Inteiro

inicio

tot <- 0

Para c <- 1 ate 10 faca

Escreva("Digite seu nome: ")

Leia(nome)

Se (copia(maiusc(nome), 1, 1) = "C") entao

tot <- tot + 1

soC[tot] <- nome

FimSe

FimPara

Limpatela

EscrevaL("LISTAGEM FINAL")

Para c <- 1 ate tot faca

EscrevaL(soC[c])

FimPara

Fimalgoritmo

Prática 04:

algoritmo "ORDENA\_VETOR"

var

vet: vetor[1..10] de Inteiro

i, j, aux: Inteiro

inicio

Para i <- 1 ate 10 faca

Escreva("Digite um valor: ")

Leia(vet[i])

FimPara

Para i <- 1 ate 9 faca

Para j <- i + 1 ate 10 faca

Se (vet[i] > vet[j]) entao

aux <- vet[i]

vet[i] <- vet[j]

vet[j] <- aux

FimSe

FimPara

FimPara

Para i <- 1 ate 10 faca

Escreva("{", vet[i], "}")

FimPara

Fimalgoritmo

Exercício 01:

algoritmo "CAMPEONATO\_FUTEBOL"

var

Time: vetor[1..3] de Caractere

C, B: inteiro

Procedimento Topo()

Inicio

EscrevaL("----------------------")

EscrevaL(" CAMPEONATO FUTEBOL ")

EscrevaL("----------------------")

FimProcedimento

inicio

Topo()

Para C <- 1 ate 3 faca

Escreva("Digite o ", C, "o. time: ")

Leia(Time[c])

FimPara

Limpatela

TOpo()

Para C <- 1 ate 3 faca

Para B <- 1 ate 3 faca

Se (Time[C] <> Time[B]) entao

EscrevaL(Time[C]:10, " [ ] x [ ] ", Time[B])

FimSe

FimPara

FimPara

Fimalgoritmo

Exercício 02:

algoritmo "CORRIGINDO\_PROVAS"

var

Gabarito: vetor[1..5] de caractere

Nome: vetor[1..3] de Caractere

RespAluno: vetor[1..5] de caractere

NotaAluno: vetor[1..5] de Inteiro

C, R, Q, SomaNotas: Inteiro

MediaTurma: Real

inicio

EscrevaL("PASSO 1 - Cadastro de gabarito")

EscrevaL("------------------------------")

Para C <- 1 ate 5 faca

Escreva("Questao ", C, ": ")

Leia(Gabarito[C])

FimPara

LimpaTela

SomaNotas <- 0

Para C <- 1 ate 3 faca

EscrevaL("------------------------------")

EscrevaL("ALUNO ", C)

EscrevaL("------------------------------")

Escreva("Nome: ")

Leia(Nome[C])

EscrevaL("RESPOSTAS DADAS")

R <- 0

Para Q <- 1 ate 5 faca

Escreva("Questao", Q,": ")

Leia(RespAluno[Q])

Se RespAluno[Q] = Gabarito[Q] entao

R <- R + 2

FimSe

NotaAluno[C] <- R

FimPara

LimpaTela

SomaNotas <- SomaNotas + NotaAluno[C]

MediaTurma <- SomaNotas/C

FimPara

LimpaTela

EscrevaL("------------------------------")

EscrevaL(" N O T A S F I N A I S ")

EscrevaL("------------------------------")

Para C <- 1 ate 3 faca

EscrevaL(Nome[C]:10, NotaAluno[C]:4:1)

FimPara

EscrevaL("------------------------------")

EscrevaL("Media da Turma: ", MediaTurma:4:1)

Fimalgoritmo

Exercício 03:

algoritmo "RESERVAR\_CINEMA"

Var

cad: vetor[1..10] de inteiro

cont, n: inteiro

r: Caractere

inicio

Repita

Para cont <- 1 ate 10 faca

Se(cad[cont] = 1) entao

Escreva("[--]")

Senao

Escreva("[B", cont, "]")

FimSe

FimPara

EscrevaL(" ")

EscrevaL("---------------------------------------------------")

Escreva("Reservar a cadeira: B")

Leia(n)

Se (cad[n] = 1) entao

EscrevaL("ERRO, cadeira ja reservada!")

FimSe

cad[n] <- 1

Escreva("Quer reservar outra cadeira? [S/N] ")

Leia(R)

LimpaTela

Ate R = "N"

Fimalgoritmo

**AULA 15 – VETORES, PARTE 2 - VARIÁVEIS COMPOSTAS HOMOGÊNEAS MULTIDIMENSIONAIS (MATRIZES)**

**var**

**m: vetor[1..3, 1..2] de inteiro**

**l, c: inteiro**

**inicio**

**para l <- 1 ate 3 faca**

**para c <- 1 ate 2 faca**

**leia(m[l,c])**

**fimpara**

**fimpara**

Prática 01:

algoritmo "LER\_MATRIZ\_3x2"

var

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

linha, coluna: inteiro

inicio

Para linha <- 1 ate 3 faca

Para coluna <- 1 ate 2 faca

Escreva("Digite um valor: ")

Leia(mat[linha,coluna])

Limpatela

FimPara

FimPara

Fimalgoritmo

Aprimoramento:

algoritmo "LER\_MATRIZ\_3x2"

var

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

linha, coluna: inteiro

inicio

Para linha <- 1 ate 3 faca

Para coluna <- 1 ate 2 faca

Escreva("Digite o valor da posicao [", linha, ",", coluna, "]: ")

Leia(mat[linha,coluna])

Limpatela

FimPara

FimPara

Para linha <- 1 ate 3 faca

Para coluna <- 1 ate 2 faca

Escreva(mat[linha,coluna]:5)

FimPara

EscrevaL()

FimPara

Fimalgoritmo

Prática 02:

algoritmo "MATRIZ\_3x3\_PAR"

var

mat: vetor[1..3, 1..3] de inteiro

linha, coluna, TotPar: inteiro

inicio

Para linha <- 1 ate 3 faca

Para coluna <- 1 ate 3 faca

Escreva("Digite o valor da posicao [", linha, ",", coluna, "]: ")

Leia(mat[linha,coluna])

FimPara

FimPara

EscrevaL("---------------")

EscrevaL(" MINHA MATRIZ")

EscrevaL("---------------")

TotPar <- 0

Para linha <- 1 ate 3 faca

Para coluna <- 1 ate 3 faca

Se (mat[linha, coluna] % 2 = 0) entao

Escreva("{", mat[linha,coluna]:2, "}")

TotPar <- TotPar + 1

Senao

Escreva(mat[linha, coluna]:4)

FimSe

FimPara

EscrevaL()

FimPara

EscrevaL("---------------")

EscrevaL("Foram digitados ", TotPar, " numeros PAR!")

Fimalgoritmo

Prática 03:

algoritmo "MATRIZ\_IDENTIDADE\_3x3"

var

Matrizz: vetor[1..3, 1..3] de Inteiro

linha, coluna: Inteiro

inicio

Para linha <- 1 ate 3 faca

Para coluna <- 1 ate 3 faca

Se (linha = coluna) entao

Matrizz[linha, coluna] <- 1

Senao

Matrizz[linha, coluna] <- 0

FimSe

FimPara

FimPara

EscrevaL()

Para linha <- 1 ate 3 faca

Para coluna <- 1 ate 3 faca

Escreva(Matrizz[linha, coluna])

FimPara

EscrevaL()

FimPara

Fimalgoritmo

Prática 04:

algoritmo "MATRIZ\_4x4"

var

Matrizz: vetor[1..4, 1..4] de Inteiro

linha, coluna, SomaPri, ProLin2, Maior3Col: Inteiro

inicio

Para linha <- 1 ate 4 faca

Para coluna <- 1 ate 4 faca

Escreva("Digite o valor da posicao [", linha, ", ", coluna, "]: ")

Leia(Matrizz[linha, coluna])

FimPara

FimPara

EscrevaL()

EscrevaL("----------------------")

SomaPri <- 0

ProLin2 <- 1

Maior3Col <- 0

Para linha <- 1 ate 4 faca

Para coluna <- 1 ate 4 faca

Se (linha = coluna) entao

SomaPri <- SomaPri + Matrizz[linha, coluna]

FimSe

Se (linha = 2) entao

ProLin2 <- ProLin2 \* Matrizz[linha, coluna]

FimSe

Se (coluna = 3) entao

Se (Matrizz[linha, coluna] > Maior3Col) entao

Maior3Col <- Matrizz[linha, coluna]

FimSe

FimSe

Escreva(Matrizz[linha, coluna]:5)

FimPara

EscrevaL()

FimPara

EscrevaL("----------------------")

EscrevaL("A soma da Diagonal Principal e ", SomaPri)

EscrevaL("O Produto da segunda linha e ", ProLin2)

EscrevaL("O maior valor da terceira coluna e ", Maior3Col)

Fimalgoritmo

Exercício 01:

algoritmo "DISSECANDO\_MATRIZ"

var

v: vetor[1..4, 1..4] de inteiro

linha, coluna, OPCAO: Inteiro

Procedimento Topo()

Inicio

EscrevaL("-------------------")

EscrevaL(" MINHA MATRIZ ")

EscrevaL("-------------------")

FimProcedimento

inicio

Para linha <- 1 ate 4 faca

Para coluna <- 1 ate 4 faca

Escreva("Digite o valor para a posicao [", linha, ",", coluna, "]: ")

Leia(v[linha, coluna])

FimPara

FimPara

LimpaTela

Repita

// MATRIZ COMPLETA

Se OPCAO = 1 entao

Topo()

Para linha <- 1 ate 4 faca

Para coluna <- 1 ate 4 faca

Escreva(v[linha, coluna]:3)

FimPara

EscrevaL()

FimPara

FimSe

// DIAGONAL PRINCIPAL

Se OPCAO = 2 entao

Topo()

Para linha <- 1 ate 4 faca

Para coluna <- 1 ate 4 faca

Se (linha = coluna) entao

Escreva(v[linha, coluna]:3)

Senao

Escreva(" ":3)

FimSe

FimPara

EscrevaL()

FimPara

FimSe

// TRIANGULO SUPERIOR

Se OPCAO = 3 entao

Topo()

Para linha <- 1 ate 4 faca

Para coluna <- 1 ate 4 faca

Se (linha < coluna) entao

Escreva(v[linha, coluna]:3)

Senao

Escreva(" ":3)

FimSe

FimPara

EscrevaL()

FimPara

FimSe

// TRIANGULO INFERIOR

Se OPCAO = 4 entao

Topo()

Para linha <- 1 ate 4 faca

Para coluna <- 1 ate 4 faca

Se (linha > coluna) entao

Escreva(v[linha, coluna]:3)

Senao

Escreva(" ":3)

FimSe

FimPara

EscrevaL()

FimPara

FimSe

EscrevaL()

EscrevaL(" MENU DE OPCOES ")

EscrevaL("==================")

EscrevaL("[1] Mostrar a Matriz")

EscrevaL("[2] Diagonal Principal")

EscrevaL("[3] Triangulo Superior")

EscrevaL("[4] Triangulo Infeiror")

EscrevaL("[5] Sair")

Escreva("======== OPCAO: ")

Leia(OPCAO)

LimpaTela

até OPCAO = 5

Escreva("Saindo...")

Fimalgoritmo

Exercício 02: