|  |
| --- |
| Este programa tem por objetivo da r aumento ao funcionário utilizando if e else e variáveis double. |
| Algoritmo "aumento"  Var  salario, porcentagem, aumento, novoSalario, teste : real  Inicio  escreva("Digite o salario da pessoa: ")  leia(salario)  se salario <= 1000.0 entao  porcentagem <- 20  senao  se salario <= 3000.0 entao  porcentagem <- 15  senao  se salario <= 8000.0 entao  porcentagem <- 10  senao  porcentagem <- 5  fimse  fimse  fimse    aumento <- salario \* porcentagem / 100  novoSalario <- salario + aumento    escreval("Novo salario = R$ ", novoSalario:8:2)  escreval("Aumento = R$ ", aumento:8:2)  escreval("Porcentagem = ", porcentagem, " %")  Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este programa calcula a formula de baskara dos 3 coeficientes informados e informa se eles não possuir raízes reais. |
| Algoritmo "baskara"  Var  a, b, c, x1, x2, delta : real  Inicio  escreva("Coeficiente a: ")  leia(a)  escreva("Coeficiente b: ")  leia(b)  escreva("Coeficiente c: ")  leia(c)  delta <- Exp(b, 2) - 4 \* a \* c    se (a = 0) ou (delta < 0) entao  escreval("Esta equacao nao possui raizes reais")  senao  x1 <- (-b + RaizQ(delta)) / (2 \* a)  x2 <- (-b - RaizQ(delta)) / (2 \* a)  escreval("X1 = ", x1:10:4)  escreval("X2 = ", x2:10:4)  fimse  Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este exercício tem por objetivo calcular o raio do círculo usando variáveis double. |
| Algoritmo "circulo"  Var  r, area : real  Inicio  escreva("Digite o valor do raio do circulo: ")  leia(r)    area <- Pi \* r \* r    escreval("AREA = ", area:10:3)  Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este exercício tem por objetivo ir contabilizando quantas pessoas escolheram álcool, gasolina e diesel! |
| Algoritmo "combustivel"  Var  codigo, alcool, gasolina, diesel : inteiro  Inicio  alcool <- 0  gasolina <- 0  diesel <- 0  escreva("Informe um codigo (1, 2, 3) ou 4 para parar: ")  leia(codigo)    enquanto codigo <> 4 faca    escolha codigo  caso 1  alcool <- alcool + 1  caso 2  gasolina <- gasolina + 1  caso 3  diesel <- diesel + 1  fimescolha  escreva("Informe um codigo (1, 2, 3) ou 4 para parar: ")  leia(codigo)    fimenquanto    escreval("MUITO OBRIGADO")  escreval("Alcool: ", alcool)  escreval("Gasolina: ", gasolina)  escreval("Diesel: ", diesel)  Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este exercício tem por objetivo calcular o consumo médio de combustível dividindo a distancia percorrida pelo combustível utilizado. |
| Algoritmo "consumo"  Var  distancia, combustivel, consumo : real  Inicio  escreva("Distancia percorrida: ")  leia(distancia)  escreva("Combustivel gasto: ")  leia(combustivel)    consumo <- distancia / combustivel    escreva("Consumo medio = ", consumo:8:3)  Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este programa pede para o usuário digitar coordenadas de x e y e classifica as coordenas de acordo com os valores digitados |
| Algoritmo "coordenadas"  Var  X, Y : real  Inicio  escreva("Valor de X: ")  leia(X)  escreva("Valor de Y: ")  leia(Y)    se (X > 0) e (Y > 0) entao  escreval("Q1")  senao  se (X < 0) e (Y > 0) entao  escreval("Q2)  senao  se (X < 0) e (Y < 0) entao  escreval("Q3")  senao  se (X > 0) e (Y < 0) entao  escreval("Q4")  senao  se (X = 0) e (Y = 0) entao  escreval("Origem")  senao  se (x = 0) entao  escreval("Eixo Y")  senao  escreval("Eixo X")  fimse  fimse  fimse  fimse  fimse  fimse  Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este exercício pede ao usuário digitar dois números e ele informa se os números estão em ordem crescente. |
| Algoritmo "crescente"  Var  X, Y : real  Inicio  escreval("Digite dois numeros:")  leia(X)  leia(Y)    enquanto X <> Y faca  se X < Y entao  escreval("CRESCENTE!")  senao  escreval("DECRESCENTE!")  fimse    escreval("Digite outros dois numeros:")  leia(X)  leia(Y)  fimenquanto  Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este programa pergunta 3 distancias ao usuário e informa qual dentre essas distancias foi a maior. |
| Algoritmo "dardo"  Var  d1, d2, d3, maior : real  Inicio  escreval("Digite as tres distancias:")  leia(d1)  leia(d2)  leia(d3)  se (d1 > d2) e (d1 > d3) entao  maior <- d1  senao  se d2 > d3 entao  maior <- d2  senao  maior <- d3  fimse  fimse    escreval("MAIOR DISTANCIA = ", maior:6:2)  Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este exercício tem por objetivo classificar quantos números estão dentro e fora de 10 e 20! |
| Algoritmo "dentro\_fora"  Var  N, X, i, dentro, fora : inteiro  Inicio  dentro <- 0  fora <- 0  escreva("Quantos numeros voce vai digitar? ")  leia(N)    para i de 1 ate N faca  escreva("Digite um numero: ")  leia(X)    se (X >= 10) e (X <= 20) entao  dentro <- dentro + 1  senao  fora <- fora + 1  fimse  fimpara    escreval(dentro, " DENTRO")  escreval(fora, " FORA")  Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este programa ele pergunta ao usuário quantos casos de divisão ele quer realizar e dá o resultado da operação. |
| Algoritmo "divisao"  Var  N, i : inteiro  x, y, divisao : real  Inicio  escreva("Quantos casos voce vai digitar? ")  leia(N)    para i de 1 ate N faca  escreva("Entre com o numerador: ")  leia(x)  escreva("Entre com o denominador: ")  leia(y)    se y = 0 entao  escreval("DIVISAO IMPOSSIVEL")  senao  divisao <- x / y  escreval("DIVISAO = ", divisao:4:2)  fimse    fimpara  Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este exercício tem por objetivo transformar os segundos em minutos e horas. |
| Algoritmo "duracao"  Var  duracao, horas, minutos, segundos, resto : inteiro  Inicio  escreva("Digite a duracao em segundos: ")  leia(duracao)    horas <- duracao \ 3600  resto <- duracao % 3600    minutos <- resto \ 60  segundos <- resto % 60    escreval(horas, ":", minutos, ":", segundos)  Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este exercício pergunta ao usuário o valor em celsius e converte em fahrenheit e pergunta ao usuário se ele deseja continuar, caso ele responda “n” ele encerra o programa. |
| Algoritmo "exemplo\_repita\_ate"  Var  C, F : real  resp : caractere  Inicio  repita  escreva("Digite a temperatura em Celsius: ")  leia(C)  F <- 9.0 \* C / 5.0 + 32.0  escreval("Equivalente em Fahrenheit: ", F:6:1)  escreva("Deseja repetir (s/n)? ")  leia(resp)  ate resp = "n"  Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este programa ele pergunta ao usuário a quantidade de cobaias e o tipo delas e no final fala o relatório delas. |
| Algoritmo "experiencias"  Var  N, i, qte, totalCobaias, totalSapos, totalCoelhos, totalRatos : inteiro  pratos, psapos, pcoelhos : real  tipoCobaia : caractere  Inicio  totalRatos <- 0  totalCoelhos <- 0  totalSapos <- 0  escreva("Quantos casos de teste serao digitados? ")  leia(N)    para i de 1 ate N faca    escreva("Quantidade de cobaias: ")  leia(qte)  escreva("Tipo de cobaia: ")  leia(tipoCobaia)    se tipoCobaia = "R" entao  totalRatos <- totalRatos + qte  senao  se tipoCobaia = "S" entao  totalSapos <- totalSapos + qte  senao  totalCoelhos <- totalCoelhos + qte  fimse  fimse    fimpara    totalCobaias <- totalRatos + totalSapos + totalCoelhos    pcoelhos <- totalCoelhos / totalCobaias \* 100  pratos <- totalRatos / totalCobaias \* 100  psapos <- totalSapos / totalCobaias \* 100    escreval  escreval("RELATORIO FINAL:")  escreval("Total: ", totalCobaias, " cobaias")  escreval("Total de coelhos: ", totalCoelhos)  escreval("Total de ratos: ", totalRatos)  escreval("Total de sapos: ", totalSapos)  escreval("Percentual de coelhos: ", pcoelhos:4:2)  escreval("Percentual de ratos: ", pratos:4:2)  escreval("Percentual de sapos: ", psapos:4:2)  Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este programa pede ao usuário pra digitar um numero inteiro e o programa exibe valor fatorado desse numero. |
| Algoritmo "fatorial"  Var  N, i, fat : inteiro  Inicio  escreva("Digite o valor de N: ")  leia(N)    fat <- 1  para i de 1 ate N faca  fat <- fat \* i  fimpara    escreval("FATORIAL = ", fat)  Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este programa pergunta ao usuário seu nível de glicose e infirma se esta normal, elevado ou diabetes. |
| Algoritmo "glicose"  Var  glicose : real  Inicio  escreva("Digite a medida da glicose: ")  leia(glicose)    se glicose <= 100.0 entao  escreval("Normal")  senao  se glicose <= 140.0 entao  escreval("Elevado")  senao  escreval("Diabetes")  fimse  fimse  Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este programa pergunta ao usuário informações de 2 pessoas como por exemplo nome e a idade dela e no final ele informa a media das idades dessas pessoas. |
| Algoritmo "idades"  Var  nome1, nome2 : caractere  idade1, idade2 : inteiro  media : real  Inicio  escreval("Dados da primeira pessoa:")  escreva("Nome: ")  leia(nome1)  escreva("Idade: ")  leia(idade1)  escreval("Dados da segunda pessoa:")  escreva("Nome: ")  leia(nome2)  escreva("Idade: ")  leia(idade2)  media <- (idade1 + idade2) / 2    escreval("A idade mehdia de ", nome1, " e ", nome2, " eh de ", media:5:1, " anos")  Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este programa o valor que o cliente devera pagar de acordo com o código do item escolhido |
| Algoritmo "lanchonete"  Var  codigo, quantidade : inteiro  valorPago : real  Inicio  escreva("Codigo do produto comprado: ")  leia(codigo)  escreva("Quantidade comprada: ")  leia(quantidade)    escolha codigo  caso 1  valorPago <- 5.0 \* quantidade  caso 2  valorPago <- 3.5 \* quantidade  caso 3  valorPago <- 4.8 \* quantidade  caso 4  valorPago <- 8.9 \* quantidade  caso 5  valorPago <- 7.32 \* quantidade  fimescolha  escreval("Valor a pagar: R$ ", valorPago:8:2)  Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este programa calcula a media das idades digitadas. |
| Algoritmo "media\_idades"  Var  idade, soma, cont : inteiro  media : real  Inicio  soma <- 0  cont <- 0  escreval("Digite as idades:")  leia(idade)    enquanto idade >= 0 faca  soma <- soma + idade  cont <- cont + 1  leia(idade)  fimenquanto    se cont = 0 entao  escreva("IMPOSSIVEL CALCULAR")  senao  media <- soma / cont  escreva("MEDIA = ", media:8:2)  fimse  Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este programa pergunta ao usuário quantos casos ele quer calcular e no final informa as medias dos números digitados. |
| Algoritmo "media\_ponderada"  Var  N, i : inteiro  a, b, c, media : real  Inicio  escreva("Quantos casos voce vai digitar? ")  leia(N)    para i de 1 ate N faca  escreval("Digite tres numeros:")  leia(a)  leia(b)  leia(c)  media <- (a \* 2 + b \* 3 + c \* 5) / 10  escreval("MEDIA = ", media:3:1)  fimpara    Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este programa calcula as medidas do quadrado, triangulo e trapézio de acordo com as medidas informadas. |
| Algoritmo "medidas"  Var  A, B, C, areaQuadrado, areaTriangulo, areaTrapezio : real  Inicio  escreva("Digite a medida A: ")  leia(A)  escreva("Digite a medida B: ")  leia(B)  escreva("Digite a medida C: ")  leia(C)  areaQuadrado <- A \* A  areaTriangulo <- (A \* B) / 2  areaTrapezio <- (A + B) / 2 \* C  escreval("QUADRADO = ", areaQuadrado:10:4)  escreval("TRIANGULO = ", areaTriangulo:10:4)  escreval("TRAPEZIO = ", areaTrapezio:10:4)    Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este código pede que o usuário digite 3 números inteiros e no final ele informa qual dos 3 números digitados tem o menor valor dentre eles. |
| Algoritmo "menor\_de\_tres"  Var  a, b, c, menor : inteiro  Inicio  escreva("Primeiro valor: ")  leia(a)  escreva("Segundo valor: ")  leia(b)  escreva("Terceiro valor: ")  leia(c)    se (a < b) e (a < c) entao  menor <- a  senao  se b < c entao  menor <- b  senao  menor <- c  fimse  fimse    escreval("MENOR = ", menor)  Fimalgoritmo |
|  |
|  |

|  |
| --- |
|  |
| Algoritmo "multiplos"  Var  x, y : inteiro  Inicio  escreval("Digite dois numeros inteiros:")  leia(x)  leia(y)    se (x % y = 0) ou (y % x = 0) entao  escreval("Sao multiplos")  senao  escreval("Nao sao multiplos")  fimse  Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este programa pergunta 2 notas aos usuário e infirma a ele no final se foi reprovado ou não. |
| Algoritmo "notas"  Var  nota1, nota2, notaFinal : real  Inicio  escreva("Digite a primeira nota: ")  leia(nota1)  escreva("Digite a segunda nota: ")  leia(nota2)    notaFinal <- nota1 + nota2    escreval("NOTA FINAL = ", notaFinal:5:1)    se notaFinal < 60.0 entao  escreval("REPROVADO")  fimse  Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este programa informa o valor a ser pago de acordo com os minutos digitados, antes de 100 minutos ele informa 50R$. |
| Algoritmo "operadora"  Var  minutos : inteiro  valorPago : real  Inicio  escreva("Digite a quantidade de minutos: ")  leia(minutos)    valorPago <- 50.0  se minutos > 100 entao  valorPago <- valorPago + 2 \* (minutos - 100)  fimse    escreval("Valor a pagar: R$ ", valorPago:8:2)  Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este programa calcula o valor a ser pago a cada funcionário de acordo com suas horas trabalhadas e o valor da hora. |
| Algoritmo "pagamento"  Var  nome : caractere  valorHora, pagamento : real  horasTrabalhadas : inteiro  Inicio  escreva("Nome: ")  leia(nome)  escreva("Valor por hora: ")  leia(valorHora)  escreva("Horas trabalhadas: ")  leia(horasTrabalhadas)    pagamento <- valorHora \* horasTrabalhadas    escreval("O pagamento para ", nome, " deve ser ", pagamento:8:2)  Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este programa pergunta quantos números o usuário irá digitar e informa se eles são pares ou impares positivos ou negativos. |
| Algoritmo "par\_impar"  Var  N, i, x : inteiro  Inicio  escreva("Quantos numeros voce vai digitar? ")  leia(N)  para i de 1 ate N faca  escreva("Digite um numero: ")  leia(x)  se x = 0 entao  escreval("NULO")  senao  se x % 2 = 0 entao  se x < 0 entao  escreval("PAR NEGATIVO")  senao  escreval("PAR POSITIVO")  fimse  senao  se x < 0 entao  escreval("IMPAR NEGATIVO")  senao  escreval("IMPAR POSITIVO")  fimse  fimse  fimse  fimpara  Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este programa s mostra os pares consecutivos do numero digitado no programa usando variáveis inteiras. |
| Algoritmo "pares\_consecutivos"  Var  X, soma : inteiro  Inicio  escreva("Digite um numero inteiro: ")  leia(X)  enquanto X <> 0 faca  se X % 2 <> 0 entao  X <- X + 1  fimse    soma <- 5 \* X + 20  escreval("SOMA = ", soma)    escreva("Digite um numero inteiro: ")  leia(X)    fimenquanto  Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este programa pede ao usuário digitar as coordenadas de x e y e informa no final a qual quadrante pertence usando if e else. |
| Algoritmo "quadrante"  Var  X, Y : real  Inicio  escreval("Digite os valores das coordenadas X e Y:")  leia(X)  leia(Y)    enquanto (X <> 0) e (Y <> 0) faca    se (X > 0) e (Y > 0) entao  escreval("QUADRANTE Q1")  senao  se (X < 0) e (Y > 0) entao  escreval("QUADRANTE Q2")  senao  se (X < 0) e (Y < 0) entao  escreval("QUADRANTE Q3")  senao  escreval("QUADRANTE Q4")  fimse  fimse  fimse  escreval("Digite os valores das coordenadas X e Y:")  leia(X)  leia(Y)    fimenquanto  Fimalgoritmo |
|  |
|  |

|  |
| --- |
|  |
| Algoritmo "retangulo"  Var  base, altura, area, perimetro, diagonal: real  Inicio  escreva("Base do retangulo: ")  leia(base)  escreva("Altura do retangulo: ")  leia(altura)  area <- base \* altura  perimetro <- 2 \* (base + altura)  diagonal <- RaizQ(base \* base + altura \* altura)  escreval("AREA = ", area:8:4)  escreval("PERIMETRO = ", perimetro:8:4)  escreval("DIAGONAL = ", diagonal:8:4)  Fimalgoritmo |
|  |
|  |

|  |
| --- |
|  |
| Algoritmo "senha\_fixa"  Var  senha : caractere  Inicio  escreva("Digite a senha: ")  leia(senha)    enquanto senha <> "2002" faca  escreva("Senha Invalida! Tente novamente: ")  leia(senha)  fimenquanto    escreval("Acesso permitido!")  Fimalgoritmo |
|  |
|  |

|  |
| --- |
|  |
| Algoritmo "sequencia\_impares"  Var  X, i : inteiro  Inicio  escreva("Digite o valor de X: ")  leia(X)    para i de 1 ate X faca  se i % 2 <> 0 entao  escreval(i)  fimse  fimpara  Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este programa e bem simples e tem apenas o proposito de somar 2 valores digitados e final ele informa o resultado da soma. |
| Algoritmo "soma"  Var  X, Y, soma : inteiro  Inicio  escreva("Digite o valor de X: ")  leia(X)  escreva("Digite o valor de Y: ")  leia(Y)    soma <- X + Y    escreval("SOMA = ", soma)  Fimalgoritmo |
|  |
|  |

|  |
| --- |
|  |
| Algoritmo "soma\_impares"  Var  X, Y, troca, i, soma : inteiro  Inicio  escreval("Digite dois numeros:")  leia(X)  leia(Y)    se X > Y entao  troca <- x  x <- y  y <- troca  fimse    soma <- 0  para i de x+1 ate y-1 faca  se i % 2 <> 0 entao  soma <- soma + i  fimse  fimpara    escreval("SOMA DOS IMPARES = ", soma)  Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este programa calcula e imprime na tela a tabuada do valor digitado. |
| Algoritmo "tabuada"  Var  N, i, resultado : inteiro  Inicio  escreva("Deseja a tabuada para qual valor? ")  leia(N)    para i de 1 ate 10 faca  resultado <- i \* N  escreval(N, " x ", i, " = ", resultado)  fimpara  Fimalgoritmo |
|  |
|  |

|  |
| --- |
| Este programa tem o objetivo de perguntar em qual escala ele quer a temperatura e informa respectivamente. |
| Algoritmo "temperatura"  Var  C, F : real  unidade : caractere  Inicio  escreva("Voce vai digitar a temperatura em qual escala (C/F)? ")  leia(unidade)    se unidade = "F" entao  escreva("Digite a temperatura em Fahrenheit: ")  leia(F)  C <- 5 / 9 \* (F - 32)  escreva("Temperatura equivalente em Celsius: ", C:6:2)  senao  escreva("Digite a temperatura em Celsius: ")  leia(C)  F <- 9 \* C / 5 + 32;  escreva("Temperatura equivalente em Fahrenheit: ", F:6:2)  fimse  Fimalgoritmo |
|  |
|  |

|  |
| --- |
|  |
| Algoritmo "tempo\_de\_jogo"  Var  horaInicial, horaFinal, duracao : inteiro  Inicio  escreva("Hora inicial: ")  leia(horaInicial)  escreva("Hora final: ")  leia(horaFinal)    se horaInicial < horaFinal entao  duracao <- horaFinal - horaInicial  senao  duracao <- 24 - horaInicial + horaFinal  fimse    escreval("O JOGO DUROU ", duracao, " HORA(S)")  Fimalgoritmo |
|  |
|  |

|  |
| --- |
|  |
| Algoritmo "terreno"  Var  largura, comprimento, metroQuadrado : real  area, preco : real  Inicio  escreva("Digite a largura do terreno: ")  leia(largura)  escreva("Digite o comprimento do terreno: ")  leia(comprimento)  escreva("Digite o valor do metro quadrado: ")  leia(metroQuadrado)    area <- largura \* comprimento  preco <- area \* metroQuadrado    escreval("Area do terreno = ", area:10:2)  escreval("Preco do terreno = ", preco:10:2)  Fimalgoritmo |
|  |
|  |

|  |
| --- |
|  |
| Algoritmo "troco"  Var  preco, dinheiro, troco : real  quantidade : inteiro  Inicio  escreva("Preco unitario do produto: ")  leia(preco)  escreva("Quantidade comprada: ")  leia(quantidade)  escreva("Dinheiro recebido: ")  leia(dinheiro)    troco <- dinheiro - (preco \* quantidade)    escreval("TROCO = ", troco:8:2)  Fimalgoritmo |
|  |
|  |

|  |
| --- |
|  |
| Algoritmo "troco\_verificado"  Var  preco, dinheiro, troco, resto : real  quantidade : inteiro  Inicio  escreva("Preco unitario do produto: ")  leia(preco)  escreva("Quantidade comprada: ")  leia(quantidade)  escreva("Dinheiro recebido: ")  leia(dinheiro)  se dinheiro >= (preco \* quantidade) entao  troco <- dinheiro - preco \* quantidade  escreval("TROCO = ", troco:8:2)  senao  resto <- preco \* quantidade - dinheiro  escreval("DINHEIRO INSUFICIENTE. FALTAM ", resto:8:2, " REAIS")  fimse  Fimalgoritmo |
|  |
|  |

|  |
| --- |
|  |
| Algoritmo "validacao\_de\_nota"  Var  nota1, nota2, media : real  Inicio  escreva("Digite a primeira nota: ")  leia(nota1)    enquanto (nota1 < 0) ou (nota1 > 10) faca  escreva("Valor invalido! Tente novamente: ")  leia(nota1)  fimenquanto  escreva("Digite a segunda nota: ")  leia(nota2)  enquanto (nota2 < 0) ou (nota2 > 10) faca  escreva("Valor invalido! Tente novamente: ")  leia(nota2)  fimenquanto  media <- (nota1 + nota2) / 2  escreval("MEDIA = ", media:6:2)  Fimalgoritmo |
|  |
|  |