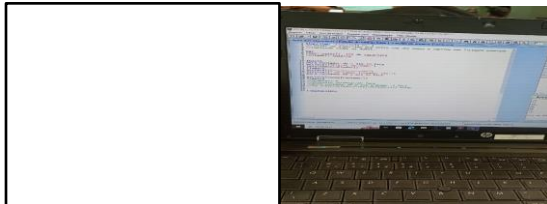


TAREFAS DO AGUSTO

```
=====
=====
algoritmo "lista de nomes – alg 1"
var
nomes: vetor[1..10] de caractere
i: inteiro
inicio
para i <- 1 ate 10 faca
escreval(" digite o ", i, "° nome: ")
leia(nomes[i])
i <- i + 1
fimpara
para i <- 1 ate 10 faca
escreval(" o ", i, "° nome é : ", nomes[i])
i <- i + 1
fimpara
```

Fimalgoritmo

ALGORITMO 1



ALGORITMO 3

```
=====
algoritmo " 3"
var
incremento: inteiro
npm: vetor[1..15] de inteiro
inicio
para incremento <- 1 até 15 passo 1 faca
escreval(" digite o ", incremento, "° valor: ")
leia(npm[incremento])
// escreval(npm[incremento], " x ", npm[1])

// leia(
fimpara
para incremento <- 1 ate 15 passo 1 faca
escreval(npm[incremento], " x ", npm[1])
```

fimpara

Fimalgoritmo

ALGORITIMO 4:

algoritmo "4"

var

va: vetor [1..5] de inteiro

vb: vetor [1..5] de inteiro

somavetor: vetor[1..5] de inteiro

o: inteiro

inicio

va[1] <- 1

va[2] <- 2

va[3] <- 3

va[4] <- 4

va[5] <- 5

vb[1] <- 3

vb[2] <- 6

vb[3] <- 7

vb[4] <- 8

vb[5] <- 9

para o <- 1 ate 5 faca

somavetor[o] <- va[o] + vb[o]

fimpara o <- 1 ate 5 faca

para o<- 1 ate 5 faca

escreva("[",somavetor[o],"]")

fimpara

Fimalgoritmo

Algoritmo "4"

// Disciplina : [Linguagem e Lógica de Programação]

// Professor : Raquel

// Descrição : Aqui você descreve o que o programa faz! (função)

// Autor(a) : Nome do(a) aluno(a)

// Data atual : 13/11/2023

Var

num1: vetor [0..4] de inteiro

num2: vetor [0..4] de inteiro

num3: vetor [0..9] de inteiro

cont: inteiro

Inicio

cont <- 0

para cont de 0 ate 4 faca

escreva("digite o ",cont+1,"do vetor 1:")

leia(num1[cont])

```

    escreva("digite o ",cont+1,"do vetor 2:")
    leia(num2[cont])
fimpara
para cont de 0 ate 4 faca
    num3[cont] <- num1[cont]
fimpara
para cont de 5 ate 9 faca
    num3[cont] <- num2[cont-5]
fimpara
para cont de 0 ate 9 faca
    escreval(num3[cont])
fimpara
Fimalgoritmo

```

```

algoritmo "5"
var
ca: vetor [1..5] de caractere
cb: vetor [1..5] de caractere
vdlr: vetor [1..5] de caractere
i: inteiro
inicio
escreval(" Digite 5 letras para o conjunto A! ")
para i <- 1 ate 5 passo 1 faca
    escreval(" Digite a ",i,"° letra: ")
    leia(ca[i])
fimpara
escreval(" Digite 5 letras para o conjunto B! ")
para i <- 1 ate 5 passo 1 faca
    escreval(" Digite a ",i,"° letra: ")
    leia(cb[i])
fimpara
para i<- 1 ate 5 faca
    se ca[i] = cb[i] entao
        vdlr[i] <- ca[i]
fimse

```

```

fimpara
escreval(" as letras repetidas foram: ")
para i <- 1 ate 5 faca
    escreva(vdlr[i])
fimpara
Fimalgoritmo

```

```

algoritmo "alg 6 - ordem inversa"
var
n: vetor[1..5] de inteiro
i: inteiro

```

```

inicio
para i <- 1 ate 5 passo 1 faca
  escreval(" digite o ", i, "º número: ")
  leia(n[i])
  i <- i + 1
fimpara
escreval(" a ordem inversa dos valores acima é: ")
para i <- 5 ate 1 passo -1 faca
  escreval(n[i])
  i <- i - 1
fimpara

```

Fimalgoritmo

algoritmo "valor de referência-antes do ex7"

var

nid: vetor[1..10] de inteiro

vmqvdr, vvdr, vdr, i: inteiro

inicio

para i <- 1 ate 10 faca

escreva(" Digite o ", i, "º número: ")

leia(nid[i])

fimpara

escreval(" digite um valor de referência: ")

leia(vdr)

vmqvdr <- 0

vvdr <- 0

escreval("os valores que são maiores que ", vdr, ", abaixo: ")

para i <- 1 ate 10 faca

se nid[i] = vdr entao

vvdr <- vvdr + 1

fimse

se nid[i] > vdr entao

escreva(nid[i])

fimse

se nid[i] < vdr entao

vmqvdr <- vmqvdr + 1

fimse

fimpara

escreval(" a quantidade de números menores que o valor de referência:", vdr, " são ", vmqvdr)

escreval(" o valor de referência foi digitado ", vvdr, " vezes.")

fimalgoritmo

```

algoritmo "10 números, maior valor - 7 "
var
ns: vetor [1..10] de inteiro
i, pdm, maior: inteiro
inicio
pdm<-1
para i <- 1 ate 10 faca
    escreval( "digite o ",i , "" valor: ")
    leia(ns[i])
    se i = 1 entao
        maior <- ns[i]
    fimse
    se ns[i] > maior entao
        maior <- ns[i]
    pdm <- i
    fimse
fimpara
    escreval(" o maior número digitado foi: ",maior, " na posição: ",pdm)
Fimalgoritmo

```

```

algoritmo "8"
var
n: vetor [1..10] de inteiro
i, nd: inteiro

inicio
para i <- 1 ate 10 passo 1 faca
    n[i] <- randi(i)
fimpara
escreval(" digite um número: ")
leia(nd)
para i <- 1 ate 10 faca
    se nd = n[i] entao
        escreval(" o ", nd, " está no vetor na posição ",i)
    senao
        escreval(" o ", nd, " não está no vetor. ")
    fimse
fimpara
Fimalgoritmo

```

```

algoritmo "12"
var
tdp, i: inteiro
vd: vetor[1..10] de inteiro
inicio
escreval(" digite 10 números: ")
para i <- 1 ate 10 faca
    escreval(" digite o ", i, "" número: ")

```

```

leia(vd[i])
fimpara
//tdp <- 0
para i <- 1 ate 10 faca
  se(vd[i] % 2 = 0) entao
    tdp <- tdp + 1
  fimse
fimpara
escreval(" o total de valores pares encontrados foi: ",tdp, " valores. ")
Fimalgoritmo

```

```

algoritmo "13"
var
vd: vetor [1..10] de inteiro
i, maiorv, menorv : inteiro

```

```

inicio
escreval("digite 10 números! ")
para i <- 1 ate 10 passo 1 faca
  escreval("digite o ", i, "º valor: ")
  leia(vd[i])
fimpara
para i <- 1 ate 10 passo 1 faca
  se i=1 entao
    maiorv <- vd[i]
    menorv <- vd[i]
  fimse
  se vd[i] >= maiorv entao
    maiorv <- vd[i]
  fimse
  se vd[i] <= menorv entao
    menorv <- vd[i]
  fimse
fimpara
escreval( " o menor valor inserido foi o ", menorv)
escreval( " o maior valor inserido foi o ",maiorv)
Fimalgoritmo

```

```

algoritmo "14"
var
nda: vetor[1..15] de inteiro
i, tdn,sdn: inteiro
mg: real
inicio
escreval("digite a nota dos 15 alunos, porfavor. ")
para i <- 1 ate 15 passo 1 faca
  escreval(" insira a nota do ", i, "º aluno: ")
  leia(nda[i])

```

```

    sdn <- sdn + nda[i]
    tdn <- tdn + 1
fimpara
mg <- sdn/tdn
escreval(" a média geral entre os ",i , " alunos é ",mg:3:1)

```

Fimalgoritmo

```

algoritmo "15"
var
vrd: vetor[1..5] de inteiro
i, oe:inteiro
inicio
escreval("=====")
escreval("digite 5 valores: ")
escreval("=====")
para i <- 1 ate 5 passo 1 faca
    escreval(" digite o ", i, "º valor: ")
    escreval("=====")
    leia(vrd[i])
fimpara
escreval("=====")
escreval("                Digite o código")
escreval("=====")
escreval("[0] para encerrar o programa. ")
escreval("[1] mostrar os valores inseridos em ordem direta ")
escreval("[2] mostrar os valores inseridos em ordem inversa ")
escreval("=====")
leia(oe)
escreval("=====")
se oe = 0 entao
senao
    se oe = 1 entao
        para i <- 1 ate 5 passo 1 faca
            escreval(vrd[i])
        fimpara
    senao
        se oe = 2 entao
            para i <- 5 ate 1 passo -1 faca
                escreval(vrd[i])
            fimpara
        senao
            escreval(" valor inserido não corresponde as opções em destaque!")
        fimse
    fimse
fimse
Fimalgoritmo

```

```

algoritmo "16"
var
vd: vetor [1..6] de inteiro
i, id: inteiro
sdnp: real
inicio
escreval(" digite 6 números: ")
para i <- 1 ate 6 passo 1 faca
    escreval(" digite o ", i, "º valor: ")
    leia(vd[i])
fimpara
escreval("=====")
escreval(" valores pares digitados: ")
para i <- 1 ate 6 passo 1 faca
    se (vd[i] % 2 = 0) entao
        escreva(vd[i])
        sdnp <- sdnp + vd[i]
    fimse
fimpara
escreval("=====")
escreval(" a soma dos números pares digitados é: ",sdnp)
escreval("=====")
escreval(" os números ímpares digitados foram: ")
para i <- 1 ate 6 passo 1 faca
    se (vd[i] % 2 <> 0) entao
        escreva(vd[i], " ")
        id <- id + 1
    fimse
fimpara
escreval("=====")
escreval(" a quantidade de números ímpares digitados foi de ", id, " números. ")

```

fimalgoritmo

```

algoritmo "mostrar vetores na ordem 17"
var
dvn: vetor[1..10] de inteiro
i, j, aux: inteiro
inicio
escreval(" digite 10 valores numéricos ")
para i <- 1 ate 10 passo 1 faca
    escreval(" digite o ", i, "º valor: ")
    leia(dvn[i])
fimpara
para i <- 1 ate 9 passo 1 faca

```



```
para j <- i + 1 ate 10 passo 1 faca
se(dvn[i] > dvn[j]) entao
  aux <- dvn[i]
  dvn[i] <- dvn[j]
  dvn[j] <- aux
fimse
fimpara
fimpara
para i<- 1 ate 10 passo 1 faca
  escreva("{", dvn[i], "}")
fimpara

finalgoritmo
```