



## Laboratório 03

### Exercícios Resolvidos

#### Exercícios :

**Exercício 01:** Faça um algoritmo que crie um vetor de 30 posições, onde cada elemento corresponde ao quadrado de sua posição. Imprima depois o vetor resultante.

algoritmo "Ex01"

var

*índice : inteiro*

*vet : vetor[0..29] de real*

inicio

*para índice de 0 ate 29 faça*

*vet[índice]  $\leftarrow$  índice  $^2$*

*escreva ( vet[índice] )*

*fimpara*

fimalgoritmo

**Exercício 02:** Escrever um algoritmo que leia 20 números inteiros e depois imprima em ordem inversa.

algoritmo "Ex02"

var

*índice, iInverso : inteiro*

*vet : vetor[0..19] de inteiro*

*vetInverso : vetor[0..19] de inteiro*

inicio

*para indice de 0 ate 19 faca*

*escreva ( " Digite um valor:" )*

*leia ( vet[indice] )*

*fimpara*

*iInverso  $\leftarrow$  19*

*para indice de 0 ate 19 faca*

*vetInverso[indice]  $\leftarrow$  vet[iInverso]*

*iInverso  $\leftarrow$  iInverso - 1*

*fimpara*

finalgoritmo

**Exercício 03:** Faça um algoritmo que leia 30 números reais e depois exiba os números localizados nas posições ímpares.

algoritmo "Ex03"

var

*índice : inteiro*

*vet : vetor[0..29] de real*

inicio

*para índice de 0 ate 29 faca*

*escreva ( " Informe o “ , ( índice + 1 ) , “ ° valor “ )*

*leia ( vet[índice] )*

*fimpara*

*para índice de 0 ate 29 faca*

*se índice mod 2  $\neq$  0 entao*

*escreva ( vet[índice] )*

*fimse*

*fimpara*

[fimalgoritmo](#)

**Exercício 04:** Escreva um algoritmo que leia 50 valores reais e armazene em um vetor. Modifique o vetor de modo que os valores das posições ímpares sejam aumentados em 5%, e os das posições pares sejam aumentados em 2%. Imprima depois o vetor resultante.

[algoritmo](#) "Ex04"

[var](#)

*índice : inteiro*

*vet : vetor[0..49] de real*

[inicio](#)

*para índice de 0 ate 49 faca*

*escreva ( " Informe o " , ( índice + 1 ) , " ° valor " )*

*leia ( vet[índice] )*

*fimpara*

*para índice de 0 ate 49 faca*

*se índice mod 2 < > 0 entao*

*vet[índice] ← vet[índice] \* 1.05*

*senao*

*vet[índice] ← vet[índice] \* 1.02*

*fimse*

*fimpara*

*para índice de 0 ate 49 faca*

*escreva ( vet[índice] , " " )*

*fimpara*

[fimalgoritmo](#)

**Exercício 05:** Faça um algoritmo para criar um vetor real de 20 posições: as 10 primeiras são informados pelo usuário, e as 10 seguintes são os mesmos números em ordem inversa.

algoritmo "Ex05"

var

*índice, iInverso : inteiro*

*vet : vetor[0..19] de inteiro*

inicio

*para índice de 0 ate 9 faca*

*escreva ( " Digite um valor:" )*

*leia ( vet[índice] )*

*fimpara*

*iInverso  $\leftarrow$  9*

*para índice de 10 ate 19 faca*

*vet[índice]  $\leftarrow$  vet[iInverso]*

*iInverso  $\leftarrow$  iInverso - 1*

*fimpara*

*para índice de 0 ate 19 faca*

*escreva ( vet[índice] )*

*fimpara*

finalgoritmo

**Exercício 06:** Faça um algoritmo para ler uma matriz 2x3 real e depois gerar e imprimir sua transposta (matriz 3x2 equivalente).

algoritmo "Ex06"

var

*matrizA : vetor[ 1..2, 1..3 ] de real*

*matrizB : vetor[ 1..3, 1..2 ] de real*

*linha, coluna : inteiro*

inicio

*para linha de 1 ate 2 faca*

*para coluna de 1 ate 3 faca*

*escreva ( “ Informe um número inteiro para posição “ , linha , “ x” , coluna )*

*leia ( matrizA [ linha, coluna ] )*

*fimpara*

*fimpara*

*escreval ( “ \_\_\_\_\_ Matriz A\_\_\_\_\_ ” )*

*para linha de 1 ate 2 faca*

*para coluna de 1 ate 3 faca*

*escreva ( matrizA[ linha, coluna ] , “ ” )*

*fimpara*

*escreval ( “ ” )*

*fimpara*

*para linha de 1 ate 3 faca*

*para coluna de 1 ate 2 faca*

*matrizB [ linha, coluna ]  $\leftarrow$  matrizA[ coluna, linha ]*

*fimpara*

*fimpara*

*escreval ( “ \_\_\_\_\_ Matriz B\_\_\_\_\_ ” )*

*para linha de 1 ate 3 faca*

*para coluna de 1 ate 2 faca*

*escreva ( matrizB[ linha, coluna ] , “ ” )*

*fimpara*

*escreval ( “ ” )*

*fimpara*

fimalgoritmo

**Exercício 07:** Ler uma matriz 4x3 real e imprimir a soma dos elementos de uma linha L fornecida pelo usuário.

algoritmo "Ex07"var*índice, linha, coluna : inteiro**vet : vetor[ 1..4, 1..3 ] de real**soma : real*inicio*para linha de 1 ate 4 faca**para coluna de 1 ate 3 faca**escreva ( " Informe um número inteiro para posição " , linha , " x " , coluna )**leia ( vet[ linha, coluna ] )**fimpara**fimpara**escreval ( " \_\_\_\_\_ MATRIZ 4 x 3 \_\_\_\_\_ " )**para linha de 1 ate 4 faca**para coluna de 1 ate 3 faca**escreva ( vet[ linha, coluna ] , " " )**fimpara**escreval ( " " )**fimpara**escreva ( " Qual linha deseja somar ? " )**leia ( índice )**para coluna de 1 ate 3 faca**soma ← soma + vet [ índice, coluna ]*

*fimpara*

*escreva* ( “ Linha somada : “ , índice )

*escreva* ( “ Soma da linha : “ , soma )

[finalgoritmo](#)

## DESAFIO:

Faça um algoritmo para ler uma matriz 3x3 real e imprimir a soma dos elementos da Diagonal principal.

[algoritmo](#) "Desafio"

[var](#)

*linha, coluna : inteiro*

*somaDiagonal : real*

*matriz : vetor[1..3, 1..3] de real*

[inicio](#)

*para linha de 1 ate 3 faca*

*para coluna de 1 ate 3 faca*

*escreva* ( “ Informe um número inteiro para posição “ , linha , “ x ” , coluna )

*leia* ( matriz[ linha, coluna ] )

*fimpara*

*fimpara*

*escreval* ( “ \_\_\_\_\_ MATRIZ 3 x 3 \_\_\_\_\_ ” )

*para linha de 1 ate 3 faca*

*para coluna de 1 ate 3 faca*

*escreva* ( matriz[ linha, coluna ] , “ ” )

*fimpara*

*escreval* ( “ ” )

*fimpara*

*para linha de 1 ate 3 faca*

*para coluna de 1 ate 3 faca*

*se linha = coluna entao*

*somaDiagonal  $\leftarrow$  somaDiagonal + matriz[ linha, coluna ]*

*fimse*

*fimpara*

*fimpara*

*escreva ( “ A soma da diagonal principal da matriz informada é : ” , somaDiagonal )*

[finalgoritmo](#)