

Unidade 3 - Algoritmos Resolvidos

algoritmo "Vetor"

var

alunos : vetor[1..50] de caractere

pesos : vetor[1..35] de real

idades : vetor[1..20] de inteiro

algoritmo "DezNomes"

var

nomes : vetor [1..10] de caractere

i : inteiro

inicio

para i <- 1 ate 10 faca

 escreva("Nome: ")

 leia(nomes[i])

fimpara

para i <- 1 ate 10 faca

 escreva(nomes[i])

fimpara

fimalgoritmo

algoritmo "GeraVetorB"

var

A : vetor [1..8] de inteiro

B : vetor [1..8] de inteiro

i : inteiro

inicio

para i <- 1 ate 8 faca

 escreva("Numero: ")

 leia(A[i])

 B[i] <- A[i] * 3

fimpara

escreva("Vetor B")

para i <- 1 ate 8 faca

 escreva("Numero: ", B[i])

fimpara

fimalgoritmo

algoritmo "GeraVetorC"

var

A : vetor [1..20] de real

B : vetor [1..20] de real

C : vetor [1..20] de real

i : inteiro

inicio

escreva("Valores do vetor A")

para i <- 1 ate 20 faca

 leia(A[i])

fimpara

```
escreva("Valores do vetor B")
para i <- 1 ate 20 faca
    leia(B[i])
    C[i] <- A[i] - B[i]
fimpara
escreva("Valores do vetor C")
para i <- 1 ate 20 faca
    escreva (C[i])
fimpara
finalgoritmo
```

```
algoritmo "QuadradoVetorA"
var
    A : vetor [1..15] de inteiro
    B : vetor [1..15] de inteiro
    i : inteiro
inicio
    escreva("Valores do vetor A")
    para i <- 1 ate 15 faca
        leia(A[i])
        B[i] <- A[i] * A[i]
    fimpara

    escreva("Valores do vetor B")
    para i <- 1 ate 15 faca
        escreva (B[i])
    fimpara
finalgoritmo
```

```
algoritmo "VetorB-Fatorial"
var
    A : vetor [1..15] de inteiro
    B : vetor [1..15] de inteiro
    i, fatorial, f : inteiro
inicio
    escreva("Valores do vetor A")
    para i <- 1 ate 15 faca
        leia(A[i])
        fatorial <- 1
        para f <- 1 ate A[i] faca
            fatorial <- f * fatorial
        fimpara
        B[i] <- fatorial
    fimpara
    escreva("Valores do vetor B")
    para i <- 1 ate 15 faca
        escreva (B[i])
    fimpara
fimalgoritmo
```

algoritmo "intercalar"

var

A : vetor [1..5] de inteiro

B : vetor [1..5] de inteiro

C : vetor [1..10] de inteiro

i, x : inteiro

inicio

para i <- 1 ate 5 passo 1 faca

 escreva("Vetor A ", i, "º elemento")

 leia(A[i])

fimpara

para i <- 1 ate 5 passo 1 faca

 escreva("Vetor B ", i, "º elemento")

 leia(B[i])

fimpara

x <- 1

para i <- 1 ate 5 passo 1 faca

 C[x] <- A[i]

 x <- x + 1

 C[x] <- B[i]

 x <- x + 1

fimpara

para i <- 1 ate 10 passo 1 faca

 escreva("Vetor C ", i, "º elemento = ", C[i])

fimpara

fimalgoritmo

```
algoritmo "inverter"
var
    A : vetor [1..10] de inteiro
    i, aux : inteiro
inicio
    escreva("números do vetor A")
    para i <- 1 ate 10 faca
        leia (A[i])
    fimpara
    aux <- 0
    para i <- 10 ate 6 passo -1 faca
        aux <- A[i]
        A[i] <- A[11 - i]
        A[11 - i] <- aux
    fimpara
    para i <- 1 ate 10 faca
        escreva("índice: ", i, " valor: ", A[i])
    fimpara
fimalgoritmo
```

```
algoritmo "pesquisa"
var
    resp : vetor [1..20] de inteiro
    total: vetor [1..5] de inteiro
    i : inteiro
inicio
    para i <- 1 ate 20 passo 1 faca
        escreva("Vetor A ", i, "º elemento")
        leia(resp[i])
        total[resp[i]] <- total[resp[i]] + 1
    fimpara

    para i <- 1 ate 5 passo 1 faca
        escreva("Nota ", i, " ", total[i])
    fimpara
fimalgoritmo
```

algoritmo "geraVetorB"

var

A : vetor [1..20] de real

B : vetor [1..20] de real

i, x : inteiro

inicio

escreva ("Valores do vetor A")

para i <- 1 ate 20 faca

leia(A[i])

fimpara

x <- 1

para i <- 20 ate 1 passo -1 faca

B[x] <- A[i]

x <- x + 1

fimpara

escreva ("Valores do vetor A")

para i <- 1 ate 20 faca

escreva (A[i])

fimpara

escreva ("Valores do vetor B")

para i <- 1 ate 20 faca

escreva (B[i])

fimpara

fimalgoritmo

algoritmo "geraVetorBSoma"

var

A : vetor [1..20] de inteiro

B : vetor [1..20] de inteiro

i, x, soma : inteiro

inicio

escreva ("Valores do vetor A")

para i <- 1 ate 20 passo 1 faca

leia(A[i])

soma <- 0

para x <- 1 ate A[i] passo 1 faca

soma <- soma + x

fimpara

B[i] <- soma

fimpara

escreva ("Valores do vetor B")

para i <- 1 ate 20 passo 1 faca

escreva (B[i])

fimpara

fimalgoritmo

algoritmo "declaramatriz"

var

alunos : vetor [1..5, 1..10] de caractere

pesos : vetor [1..6, 1..4] de real

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

algoritmo "matriz"

var

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

x, y, soma, opcao : inteiro

media: real

inicio

opcao <- 0

enquanto (opcao <> 9) faca

 escreva("1 - Inserir")

 escreva("2 - Mostrar")

 escreva("3 - Soma da segunda coluna")

 escreva("4 - Soma da terceira linha")

 escreva("5 - Soma")

 escreva("6 - Números maiores que a média")

 escreva("7 - Soma da diagonal principal")

 escreva("8 - Dobrar números pares")

 escreva("9 - Sair")

 leia(opcao)

 escolha(opcao)

 caso (1)

 para x de 1 ate 4 passo 1 faca

 para y de 1 ate 4 passo 1 faca

 escreva("Informe o número da linha ", x, " coluna", y, ":")

 leia (mat[x,y])

 fimpara

 fimpara

 caso (2)

 para x de 1 ate 4 passo 1 faca

 para y de 1 ate 4 passo 1 faca

 se y = 4 entao

 escreva(mat[x,y])

```

senao
    escreva(mat[x,y], " ")
fimse
fimpara
fimpara
caso (3)
    soma <- 0
    para x de 1 ate 4 passo 1 faca
        soma <- soma + mat[x,2]
    fimpara
    escreva("Soma da coluna 2: ", soma)
caso (4)
    soma <- 0
    para x de 1 ate 4 passo 1 faca
        soma <- soma + mat[3,x]
    fimpara
    escreva("Soma da linha 3: ", soma)
caso (5)
    soma <- 0
    para x de 1 ate 4 passo 1 faca
        para y de 1 ate 4 passo 1 faca
            soma <- soma + mat[x,y]
        fimpara
    fimpara
    escreva("Somatória = ", soma)
caso (6)
    media <- soma / 16
    para x de 1 ate 4 passo 1 faca
        para y de 1 ate 4 passo 1 faca
            se mat[x,y] > media entao
                escreva("Nr.maior que a media = ", mat[x,y])

```

```

        escreva("linha ",x," coluna ",y," valor ",mat[x,y])
    fimse
fimpara
fimpara
caso (7)
    soma <- 0
    para x de 1 ate 4 passo 1 faca
        soma <- soma + mat[x,x]
    fimpara
    escreva("Soma diagonal principal: ", soma)
caso (8)
    escreva("Dobrar números pares:")
    para x de 1 ate 4 passo 1 faca
        para y de 1 ate 4 passo 1 faca
            se mat[x,y] mod 2 = 0 entao
                mat[x,y] <- mat[x,y] * 2
            fimse
        fimpara
    fimpara
    para x de 1 ate 4 passo 1 faca
        para y de 1 ate 4 passo 1 faca
            escreva("Linha: ", x, " Coluna: ", y, " = ", mat[x,y])
        fimpara
    fimpara
caso (9)
    escreva("Fim do algoritmo")
outrocaso
    escreva("Opção inválida")
fimescolha
fimEnquanto
fimalgoritmo

```

algoritmo "somaMatriz"

var

matA : vetor[1..2, 1..4] de inteiro //DECLARAÇÃO DA MATRIZ

matB : vetor[1..2, 1..4] de inteiro

matC : vetor[1..2, 1..4] de inteiro

x, y, soma : inteiro

media: real

inicio

escreva("Matriz matA")

para x de 1 ate 2 faca // ENTRADA DE DADOS

para y de 1 ate 4 faca

escreva("Informe o número da linha ", x, " coluna", y, ":")

leia (matA[x,y])

fimpara

fimpara

escreva("Matriz matB")

para x de 1 ate 2 faca // ENTRADA DE DADOS

para y de 1 ate 4 faca

escreva("Informe o número da linha ", x, " coluna", y, ":")

leia (matB[x,y])

fimpara

fimpara

escreva("Matriz matC")

para x de 1 ate 2 faca //soma matriz

para y de 1 ate 4 faca

matC[x,y] <- matA[x,y] + matB[x,y]

escreva("Linha: ", x, " Coluna: ", y, " = ", matC[x,y])

fimpara

fimpara

fimalgoritmo

algoritmo "TranspMatriz"

var

matA : vetor[1..3, 1..2] de inteiro //DECLARAÇÃO DA MATRIZ

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

x, y, opcao : inteiro

inicio

opcao <- 0

enquanto (opcao <> 9) faça

 escreva("1 - Aceite e armazene números na matriz A")

 escreva("2 - Exiba a matriz A")

 escreva("3 - Determine a matriz transposta de A")

 escreva("4 - Exiba a matriz transposta de A")

 escreva("9 - Sair")

 leia(opcao)

 escolha(opcao)

 caso (1)

 escreva("Matriz A")

 para x de 1 ate 3 faça // ENTRADA DE DADOS

 para y de 1 ate 2 faça

 escreva("Informe o número da linha ", x, " coluna", y, ":")

 leia (matA[x,y])

 fimpara

 fimpara

 caso (2)

 escreva("Matriz A")

 para x de 1 ate 3 faça

 para y de 1 ate 2 faça

 escreva("Linha: ", x, " Coluna: ", y, " = ", matA[x,y])

 fimpara

 fimpara

 caso (3)

```
para x de 1 ate 3 faca // ENTRADA DE DADOS
    para y de 1 ate 2 faca
        matAt[y,x] <- matA[x,y] //determina transposta
    fimpara
fimpara
caso (4)
    escreva("Matriz transposta de A")
    para x de 1 ate 2 faca // MOSTRA OS NUMEROS ARMAZENADOS matC
        para y de 1 ate 3 faca
            escreva("Linha: ", x, " Coluna: ", y, " = ", matAt[x,y])
        fimpara
    fimpara
caso (9)
    escreva("Fim do algoritmo")
outrocaso
    escreva("Opção inválida")
fimescolha
fimEnquanto
fimalgoritmo
```

algoritmo "QuadradoMagico"

var

m1 : vetor[1..3, 1..3] de inteiro

soma : vetor[1..8] de inteiro

m : logico

x, y : inteiro

inicio

escreva("Matriz A")

para x de 1 ate 3 faca

para y de 1 ate 3 faca

escreva("Informe o número da linha ", x, " coluna", y, ":")

leia (m1[x,y])

soma[x] <- soma[x] + m1[x,y]

fimpara

fimpara

para x de 1 ate 3 faca

para y de 1 ate 3 faca

soma[x+3] <- soma[x+3] + m1[y,x]

fimpara

fimpara

para x de 1 ate 3 faca

soma[7] <- soma[7] + m1[x,x]

fimpara

y <- 3

para x de 1 ate 3 faca

soma[8] <- soma[8] + m1[x,y]

y <- y - 1

fimpara

m <- verdadeiro

para x de 1 ate 8 faca

se(soma[x] <> soma[1]) entao

m <- falso

interrompa

fimse

fimpara

se (m = verdadeiro) entao

escreva("É quadrado mágico")

senao

escreva("Não é quadrado mágico")

fimse

fimalgoritmo

algoritmo "MatrizInteiro"

var

matA : vetor[1..5, 1..10] de inteiro //DECLARAÇÃO DA MATRIZ

x, y, opcao, numero, maior, menor : inteiro

achou : logico

inicio

opcao <- 0

enquanto (opcao <> 9) faca

escreva("1 - Preencher matriz com números aleatórios")

escreva("2 - Mostre os números da matriz")

escreva("3 - Pesquisa número")

escreva("4 - Mostre o maior e o menor número")

escreva("9 - Sair")

leia(opcao)

escolha(opcao)

caso (1)

para x de 1 ate 5 faca

para y de 1 ate 10 faca

matA[x,y] <- RandI(20)

fimpara

fimpara

caso (2)

para x de 1 ate 5 faca

para y de 1 ate 10 faca

escreva("Linha: ", x, " Coluna: ", y, " = ", matA[x,y])

fimpara

fimpara

caso (3)

escreva("Informe o número de pesquisa:")

leia(numero)

achou <- falso

para x de 1 ate 5 faca

para y de 1 ate 10 faca

se (matA[x,y] = numero) entao

achou <- verdadeiro

fimse

fimpara

fimpara

se (achou = verdadeiro) entao

escreva("Número encontrado")

senao

escreva("Número não existe")

fimse

caso (4)

para x de 1 ate 5 faca

```
para y de 1 ate 10 faca
  se (x = 1) e (y = 1) entao
    maior <- matA[x,y]
    menor <- matA[x,y]
  senao
    se (matA[x,y] > maior) entao
      maior <- matA[x,y]
    fimse
    se (matA[x,y] < menor) entao
      menor <- matA[x,y]
    fimse
  fimse
fimpara
fimpara
escreva("Maior:", maior)
escreva("Menor:", menor)
caso (9)
  escreva("Fim do algoritmo")
outrocaso
  escreva("Opção inválida")
fimescolha
fimEnquanto
fimalgoritmo
```

Algoritmo "Paises"

Var

países : vetor [1..5, 1..4] de caractere

x, y, opção : inteiro

nome : caractere

Início

opção <- 0

enquanto (opção <> 9) faça

 escreva("1 - Incluir país")

 escreva("2 - Mostrar país")

 escreva("3 - Pesquisar país")

 escreva("9 - Sair")

 leia(opção)

 escolha(opção)

 caso (1)

 para x <- 1 até 5 passo 1 faça

 para y <- 1 até 4 passo 1 faça

 escreva("Informe o número da linha ", x, " coluna", y, ":")

 leia(países[x,y])

 fimpara

 fimpara

 caso (2)

 para x <- 1 até 5 passo 1 faça

 para y <- 1 até 4 passo 1 faça

 escreva(países[x,y])

 fimpara

 fimpara

 caso (3)

 escreva("Informe o país a ser pesquisado:")

 leia(nome)

 para x <- 1 até 5 passo 1 faça

```
para y <- 1 ate 4 passo 1 faca
    se (pos(nome,paises[x,y]) > 0) entao
        escreva(paises[x,y])
    fimse
fimpara
fimpara
caso (9)
    escreva("Fim do algoritmo")
outrocaso
    escreva("Opção inválida")
fimescolha
Fimenquanto
Fimalgoritmo
```