

LISTA DE EXERCÍCIOS MATRIZ – RESOLUÇÕES (VISUALG)

algoritmo "matriz-exerc1"

var

 mata: vetor [1..10,1..10] de inteiro //matriz max 10x10

 vet: vetor [1..10] de inteiro //vet max 10pos

 n, lin, col, result, p, q: inteiro

inicio

repita

 escreva("Dimensao da matriz (1..10)")

 leia(n)

ate (n>1) e (n<11)

para lin de 1 ate n faca

 para col de 1 ate n faca

 escreva("Elemento[",lin,"",",",col,""]: ")

 leia (matA[lin,col])

 fimpara

fimpara

escreval("matriz para conferencia")

para lin de 1 ate n faca

 para col de 1 ate n faca

 escreva(matA[lin,col], " ")

 fimpara

 escreval //pula linha

fimpara

result<-0

para lin de 1 ate n faca

 result <- result + matA[lin,2]

fimpara

escreval("Soma da 2a coluna", result)

para lin de 1 ate n faca

 vet[lin] <- 1

 para col de 1 ate n faca

 vet[lin] <- vet[lin] * matA[lin,col]

 fimpara

fimpara

escreval("Multiplicacao de cada linha")

para lin de 1 ate n faca

 escreval("Linha ", lin, ": ",vet[lin])

fimpara

result<-0

para lin de 1 ate n faca

 para col de 1 ate n faca

 result <- result + matA[lin,col]

 fimpara

fimpara

escreval("Soma de toda a matriz", result)

```

result<-0
para lin de 1 ate n faca
    result <- result + matA[lin,lin]
fimpara
escreval("Soma diagonal", result)
escreval("entre com linhas p e q para troca:")
repita
    leia (p,q)
ate (p>0) e (p<=n) e (q>0) e (q<=n) e (p<>q)
para col de 1 ate n faca
    result <- matA[p,col]
    matA[p,col] <- matA[q,col]
    matA[q,col] <- result
fimpara
escreval("matriz modificada")
para lin de 1 ate n faca
    para col de 1 ate n faca
        escreva(matA[lin,col]," ")
    fimpara
    escreval //pula linha
fimpara
fimalgoritmo

```

algoritmo "matriz-exerc2"

```

var
    matA: vetor [1..15,1..25] de inteiro
    vet: vetor [1..15] de inteiro
    n, m, lin, col: inteiro
inicio
repita
    escreva("Linhas da matriz A (1..15)")
    leia(n)
ate (n>1) e (n<16)
repita
    escreva("Colunas da matriz A (1..25)")
    leia(m)
ate (m>1) e (m<26)
para i de 1 ate n faca
    para j de 1 ate m faca
        escreva("Elemento[",i,"",",",j,""]: ")
        leia (matA[i,j])
    fimpara
fimpara
para lin de 1 ate n faca
    vet[lin] <- 0
    para col de 1 ate m faca
        vet[lin] <- vet[lin] + matA[lin,col]

```

```

    fimpara
fimpara
escreval("Soma de cada linha")
para lin de 1 ate n faca
    escreval("Linha ", lin, ": ", vet[lin])
fimpara
fimalgoritmo

```

algoritmo "matriz-exerc3"

```

var
matA: vetor [1..10,1..10] de inteiro
n,i,j: inteiro
inicio
repita
    escreva("Linhas da matriz A (1..10)")
    leia(n)
ate (n>1) e (n<16)
// considerando q matA ja tah zerada pelo visualg !!!!
para i de 1 ate n-1 faca
    matA[i,i]<-1 //DIAGONAL
    para j de i+1 ate n faca
        matA[i,j]<-2 //SUPERIOR
    fimpara
fimpara
matA[1,1]<-10
matA[n,n]<-20
escreval("matriz para conferencia")
para i de 1 ate n faca
    para j de 1 ate n faca
        escreva(matA[i,j], " ")
    fimpara
    escreval //pula linha
fimpara
fimalgoritmo

```

algoritmo "matriz-exerc4"

```

var
matA: vetor [1..15,1..25] de inteiro
matB: vetor [1..25,1..15] de inteiro
n,m, i, j: inteiro
simetrica: logico
inicio
repita
    escreva("Linhas da matriz A (1..15)")
    leia(n)
ate (n>1) e (n<16)
repita

```

```

    escreva("Colunas da matriz A (1..25)")
    leia(m)
ate (m>1) e (m<26)
para i de 1 ate n faca
    para j de 1 ate m faca
        escreva("Elemento[",i,",",j,"]: ")
        leia (matA[i,j])
    fimpara
fimpara
fimpara
//transposta
para i de 1 ate n faca
    para j de 1 ate m faca
        matB[j,i]<-matA[i,j]
    fimpara
fimpara
escreval("matriz transposta")
para i de 1 ate m faca
    para j de 1 ate n faca
        escreva(matB[i,j], " ")
    fimpara
    escreval //pula linha
fimpara
// simetrica: quadrada e testar se aij = aji !!
se n <> m entao // não quadrada
    simetrica <- falso
senao
    simetrica <- verdadeiro
    para i de 1 ate n-1 faca
        para j de i+1 ate n faca
            se matA[i,j]<>matA[j,i] entao
                simetrica <- falso
                interrompa
        fimse
    fimpara
fimpara
fimse
se simetrica entao
    escreval("matriz A eh simetrica")
senao
    escreval("matriz A nao eh simetrica")
fimse
fimalgoritmo

```

algoritmo "matriz-exerc5"

```

var
    matA, matB, matC: vetor [1..15,1..25] de inteiro
    n,m, p, q, i, j, k: inteiro

```

```

inicio
repita
    escreva("Linhas da matriz A (1..15)")
    leia(n)
ate (n>1) e (n<16)
repita
    escreva("Colunas da matriz A (1..25)")
    leia(m)
ate (m>1) e (m<26)
para i de 1 ate n faca
    para j de 1 ate m faca
        escreva("Elemento[\",i\",\",j\", \": ")
        leia (matA[i,j])
    fimpara
fimpara
repita
    escreva("Linhas da matriz B (1..15)")
    leia(p)
ate (p>1) e (p<16)
repita
    escreva("Colunas da matriz B (1..25)")
    leia(q)
ate (q>1) e (q<26)
para i de 1 ate p faca
    para j de 1 ate q faca
        escreva("Elemento[\",i\",\",j\", \": ")
        leia (matB[i,j])
    fimpara
fimpara
// multiplicacao matA x matB
se m = p entao
    para i de 1 ate n faca
        para j de 1 ate q faca
            matC[i,j]<-0
            para k de 1 ate m faca
                matC[i,j] <- matC[i,j] + matA[i,k] * matB[k,j]
            fimpara
        fimpara
    fimpara
escreval("matriz multiplicacao Ax B")
para i de 1 ate n faca
    para j de 1 ate q faca
        escreva(matC[i,j], " ")
    fimpara
    escreval //pula linha
fimpara
senao

```

```
    escreval("impossivel multiplicar")
fimse
fimalgoritmo
```

ALGORITMO "matriz-exerc6"

```
VAR
    acima, abaixo, i, j, semanaA, semanaB, diaA, diaB: inteiro
    max, min: real
    mA: vetor [1..4,1..7] de real
    tmed, tmin, tmax: vetor[1..4] de real
INICIO
    //LEITURA DAS TEMPERATURAS
    PARA i DE 1 ATE 4 FACA
        PARA j DE 1 ATE 7 FACA
            ESCREVA("Semana ",i,"- dia ",j)
            LEIA(mA[i,j])
        FIMPARA
    FIMPARA
    PARA i DE 1 ATE 4 FACA
        PARA j DE 1 ATE 7 FACA
            ESCREVA(" ", mA[i,j])
        FIMPARA
    ESCREVAL
    FIMPARA
    //temp media
    PARA i DE 1 ATE 4 FACA
        tmed[i]<- mA[i,1]
        PARA j DE 2 ATE 7 FACA
            tmed[i] <- tmed[i] + mA[i,j]
        FIMPARA
        tmed[i] <- tmed[i]/7
    FIMPARA
    //temp maxima
    PARA i DE 1 ATE 4 FACA
        tmax[i]<- mA[i,1]
        PARA j DE 2 ATE 7 FACA
            SE mA[i,j] > tmax[i] ENTAO
                tmax[i] <- mA[i,j]
        FIMSE
    FIMPARA
    //temp minima
    PARA i DE 1 ATE 4 FACA
        tmin[i]<- mA[i,1]
        PARA j DE 2 ATE 7 FACA
            SE mA[i,j] < tmin[i] ENTAO
                tmin[i] <- mA[i,j]
        FIMSE
    FIMPARA
FIMPARA
```

```

escreval("Semana  T.Media  T.Max  T.Min")
PARA i DE 1 ATE 4 FACA
    escreval(" ",i," ",tmed[i]," ",tmax[i]," ", tmin[i])
FIMPARA
max<- mA[1][1]
semanaA<-1
diaA<-1
min<- mA[1][1]
semanaB<-1
diaB<-1
PARA i DE 1 ATE 4 FACA
    PARA j DE 1 ATE 7 FACA
        SE mA[i,j] > max ENTÃO
            max<- mA[i][j]
            semanaA<-i
            diaA<-j
        SENÃO
            SE mA[i,j] < min ENTÃO
                min<- mA[i][j]
                semanaB<-i
                diaB<-j
        FIMSE
    FIMSE
FIMPARA
FIMPARA
escreval("dia mais quente", diaA," da semana ", semanaA)
escreval("dia mais frio", diaB," da semana ", semanaB)
//qtde dias acima abaixo 25
abaixo <- 0
acima <- 0
PARA i DE 1 ATE 4 FACA
    PARA j DE 1 ATE 7 FACA
        SE mA[i,j] > 25 ENTÃO
            acima <- acima + 1
        SENÃO
            SE mA[i,j] < 25 ENTÃO // nao se quer = 25
                abaixo <- abaixo + 1
        FIMSE
    FIMSE
FIMPARA
FIMPARA
escreval("dias acima ", acima," e abaixo ", abaixo)
fimalgoritmo

```

algoritmo "matriz-exerc7"

```

var
    n, i, j: inteiro
    mA: vetor [1..12,1..10] de inteiro
    vetC, vetP: vetor [1..10] de real
    vetCM, vetRBruta, vetRLiq: vetor [1..12] de real

```

```

inicio
REPITA
    ESCREVA("qtde de motores ")
    LEIA(n)
ATE (n>0) E (n<11) //max 10 motores
PARA i DE 1 ATE 12 FACA
    PARA j DE 1 ATE n FACA
        REPITA
            ESCREVA("Producao mes ",i," motor ",j)
            LEIA(mA[i,j])
            ATE mA[i,j]>=0 //nao pode negativo
        FIMPARA
    FIMPARA
PARA i DE 1 ATE n FACA
    REPITA
        ESCREVA("Custo motor ",i)
        LEIA(vetC[i])
        ATE vetC[i]>=0 //nao pode negativo
    FIMPARA
PARA i DE 1 ATE n FACA
    REPITA
        ESCREVA("Preco venda motor ",i)
        LEIA(vetP[i])
        ATE vetP[i]>=0 //nao pode negativo
    FIMPARA
PARA i DE 1 ATE 12 FACA // CALCULOS MENSAIS
    vetCM[i] <- mA[i,1] * vetC[1]
    PARA j DE 2 ATE n FACA
        vetCM[i] <- vetCM[i] + mA[i,j] * vetC[j]
    FIMPARA
FIMPARA
PARA i DE 1 ATE 12 FACA
    vetRBruta[i] <- mA[i,1] * vetP[1]
    PARA j DE 2 ATE n FACA
        vetRBruta[i] <- vetRBruta[i] + mA[i,j] * vetP[j]
    FIMPARA
FIMPARA
PARA i DE 1 ATE 12 FACA
    vetRLiq[i] <- vetRBruta[i] - vetCM[i]
FIMPARA
ESCREVAL("RECEITA BRUTA - CUSTO = RECEITA LIQUIDA MENSAL !!!")
PARA i DE 1 ATE 12 FACA
    ESCREVAL(vetRBruta[i]," - ",vetCM[i]," = ", vetRLiq[i])
FIMPARA
fimalgoritmo

```