**1. Faça um programa que crie um vetor por leitura com 5 valores de pontuação de uma atividade e o escreva em seguida. Encontre após a maior pontuação e a apresente.**

programa {

funcao inicio()

{

real maiorNota = 0

real notas[5]

para(inteiro l=0; l<5; l++){

escreva("Digite a nota ", l+1, ": \n")

leia(notas[l])

se (notas[l] > maiorNota){

maiorNota = notas[l]

}

}

limpa()

para(inteiro l=0; l<5; l++){

escreva("[ ", notas[l], " ] ")

}

escreva("\nA maior nota é: ", maiorNota)

}

}

**2. Um dado é lançado 10 vezes e o valor correspondente é anotado. Faça um programa que gere um vetor com os lançamentos, escreva esse vetor. A seguir determine e imprima a média aritmética dos lançamentos, contabilize e apresente também quantas foram as ocorrências da maior pontuação.**

programa {

funcao inicio()

{

inteiro valorDoDado, n, maiorPontuacao, media, soma = 0, maioresPontuacoes = 0

const inteiro quantJogadas = 4

inteiro vetor[quantJogadas]

para(inteiro l=0; l<quantJogadas; l++){

escreva("Qual foi o valor do dado? ")

leia(n)

se (n > 0 e n <= 6){

vetor[l] = n

soma += vetor[l]

limpa()

se (vetor[l] == 6){

maioresPontuacoes += 1

}

}

senao {

escreva("Valor inválido\n")

pare

}

}

para(inteiro l=0; l<quantJogadas; l++){

escreva("[ ", vetor[l], " ] ")

}

media = soma / quantJogadas

escreva("\nMédia dos valores: ", media)

escreva("\nOcorrências da maior pontuação: ", maioresPontuacoes)

}

}

**3. Escreve um programa que lê duas matrizes N1 (4,6) e N2(4,6) e cria:**

**a) Uma matriz M1 cujos elementos serão as somas dos elementos de mesma posição das matrizes N1 e N2;**

**b) Uma matriz M2 cujos elementos serão as diferenças dos elementos de mesma posição das matrizes N1 e N2.**

programa {

funcao inicio()

{

inteiro n1 [4][6] = {

{10, 10, 10, 10, 10, 10},

{20, 20, 20, 20, 20, 20},

{30, 30, 30, 30, 30, 30},

{40, 40, 40, 40, 40, 40}

}

inteiro n2 [4][6] = {

{5, 5, 5, 5, 5, 5},

{10, 10, 10, 10, 10, 10},

{15, 15, 15, 15, 15, 15},

{20, 20, 20, 20, 20, 20}

}

inteiro m1 [4][6] = {

{n1[0][0]+n2[0][0], n1[0][1]+n2[0][1], n1[0][2]+n2[0][2], n1[0][3]+n2[0][3], n1[0][4]+n2[0][4], n1[0][5]+n2[0][5]},

{n1[1][0]+n2[1][0], n1[1][1]+n2[1][1], n1[1][2]+n2[1][2], n1[1][3]+n2[1][3], n1[1][4]+n2[1][4], n1[1][5]+n2[1][5]},

{n1[2][0]+n2[2][0], n1[2][1]+n2[2][1], n1[2][2]+n2[2][2], n1[2][3]+n2[2][3], n1[2][4]+n2[2][4], n1[2][5]+n2[2][5]},

{n1[3][0]+n2[3][0], n1[3][1]+n2[3][1], n1[3][2]+n2[3][2], n1[3][3]+n2[3][3], n1[3][4]+n2[3][4], n1[3][5]+n2[3][5]}

}

inteiro m2 [4][6] = {

{n1[0][0]-n2[0][0], n1[0][1]-n2[0][1], n1[0][2]-n2[0][2], n1[0][3]-n2[0][3], n1[0][4]-n2[0][4], n1[0][5]-n2[0][5]},

{n1[1][0]-n2[1][0], n1[1][1]-n2[1][1], n1[1][2]-n2[1][2], n1[1][3]-n2[1][3], n1[1][4]-n2[1][4], n1[1][5]-n2[1][5]},

{n1[2][0]-n2[2][0], n1[2][1]-n2[2][1], n1[2][2]-n2[2][2], n1[2][3]-n2[2][3], n1[2][4]-n2[2][4], n1[2][5]-n2[2][5]},

{n1[3][0]-n2[3][0], n1[3][1]-n2[3][1], n1[3][2]-n2[3][2], n1[3][3]-n2[3][3], n1[3][4]-n2[3][4], n1[3][5]-n2[3][5]}

}

escreva("MATRIZ M1: \n")

para(inteiro l=0; l<4; l++){

para(inteiro c=0; c<6; c++){

escreva("[ ", m1[l][c], " ] ")

}

escreva("\n")

}

escreva("\n")

escreva("MATRIZ M2: \n")

para(inteiro l=0; l<4; l++){

para(inteiro c=0; c<6; c++){

escreva("[ ", m2[l][c], " ] ")

}

escreva("\n")

}

}

}

**4. Crie um programa que receba valores do usuário para preencher uma matriz 3X3, e em seguida, exiba a soma dos valores dela e a soma dos valores da primeira diagonal, ou seja, diagonal principal.**

programa {

funcao inicio()

{

inteiro soma = 0, somaDiagonal = 0

const inteiro valorMatriz = 3

inteiro matriz[valorMatriz][valorMatriz]

para(inteiro l=0; l<valorMatriz; l++){

para(inteiro c=0; c<valorMatriz; c++){

escreva("Digite o valor | ", l, " | ", c," |: ")

leia(matriz[l][c])

soma += matriz[l][c]

}

}

limpa()

para(inteiro l=0; l<valorMatriz; l++){

escreva("\n")

para(inteiro c=0; c<valorMatriz; c++){

escreva("[ ", matriz[l][c], " ] ")

}

}

escreva("\nSoma dos valores da Matriz: ", soma)

somaDiagonal = matriz[0][0] + matriz[1][1] + matriz[2][2]

escreva("\nSoma da diagonal principal: ", somaDiagonal)

}

}