**1)**

programa

{

funcao inicio()

{

real notas[5]

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

escreva("Nota: ")

leia(notas[i])

}

limpa()

real maiorNota = 0.0

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

se(notas[i] > maiorNota) {

maiorNota = notas[i]

}

escreva("\nNota " + (i+1) + ": " + notas[i])

}

escreva("\nA maior nota digitada é: " + maiorNota)

}

}

2)

programa

{

inclua biblioteca Matematica --> mat

funcao inicio()

{

const inteiro QTD\_LANCAMENTOS = 10

real lancamentos[QTD\_LANCAMENTOS]

para(inteiro i = 0; i < QTD\_LANCAMENTOS; i++) {

escreva("Lançamento: ")

leia(lancamentos[i])

}

limpa()

real somaLancamentos = 0.0

real maiorLancamento = 0.0

para(inteiro i = 0; i < QTD\_LANCAMENTOS; i++) {

escreva("\nLançamento " + (i+1) + ": " + lancamentos[i])

somaLancamentos += lancamentos[i]

se(lancamentos[i] > maiorLancamento) {

maiorLancamento = lancamentos[i]

}

}

inteiro ocorrenciasMaiorPontuacao = 0

para(inteiro i = 0; i < QTD\_LANCAMENTOS; i++) {

se(lancamentos[i] == maiorLancamento) {

ocorrenciasMaiorPontuacao += 1

}

}

real mediaLancamentos = mat.arredondar(somaLancamentos / QTD\_LANCAMENTOS, 2)

escreva("\nMédia: " + mediaLancamentos + "\nMaior pontuação: " + maiorLancamento +

"\nOcorrências da maior pontuação: " + ocorrenciasMaiorPontuacao)

}

}

**3)**

programa

{

inclua biblioteca Matematica --> mat

funcao inicio()

{

const inteiro LINHAS = 4

const inteiro COLUNAS = 6

inteiro n1[LINHAS][COLUNAS] = {{2, 4, 6, 8, 10, 12}, {10, 12, 14, 16, 20, 30}, {30, 5, 5, 10, 2, 4}, {2, 4, 6, 1, 0, 12}}

inteiro n2[LINHAS][COLUNAS] = {{18, 20, 22, 24, 50, 100}, {26, 28, 30, 32, 1, 0},{20, 3, 6, 10, 1, 3}, {1, 3, 5, 2, 9, 12}}

inteiro m1[LINHAS][COLUNAS]

inteiro m2[LINHAS][COLUNAS]

para(inteiro linha = 0; linha < LINHAS; linha++) {

para(inteiro coluna = 0; coluna < COLUNAS; coluna++) {

m1[linha][coluna] = n1[linha][coluna] + n2[linha][coluna]

m2[linha][coluna] = mat.valor\_absoluto(n1[linha][coluna] - n2[linha][coluna])

}

}

escreva("===== M1 =====\n")

para(inteiro linha = 0; linha < LINHAS; linha++) {

para(inteiro coluna = 0; coluna < COLUNAS; coluna++) {

cadeia espaco = " "

se(m1[linha][coluna] > 9) {

espaco = " "

}

escreva(m1[linha][coluna], espaco)

}

escreva("\n")

}

escreva("===== M2 =====\n")

para(inteiro linha = 0; linha < LINHAS; linha++) {

para(inteiro coluna = 0; coluna < COLUNAS; coluna++) {

cadeia espaco = " "

se(m2[linha][coluna] > 9) {

espaco = " "

}

escreva(m2[linha][coluna], espaco)

}

escreva("\n")

}

}

}

**4)**

programa

{

funcao inicio()

{

const inteiro LINHAS = 3

const inteiro COLUNAS = 3

inteiro numeros[LINHAS][COLUNAS]

para(inteiro linha = 0; linha < LINHAS; linha++) {

para(inteiro coluna = 0; coluna < COLUNAS; coluna++) {

escreva("Número da linha " + (linha+1) + " e coluna " + (coluna+1) + ":")

leia(numeros[linha][coluna])

}

}

limpa()

escreva("===== NÚMEROS DIGITADOS =====\n")

inteiro somaValores = 0

para(inteiro linha = 0; linha < LINHAS; linha++) {

para(inteiro coluna = 0; coluna < COLUNAS; coluna++) {

somaValores += numeros[linha][coluna]

cadeia espaco = " "

se(numeros[linha][coluna] > 9) {

espaco = " "

}

escreva(numeros[linha][coluna], espaco)

}

escreva("\n")

}

inteiro somaDiagonal = 0

para(inteiro i = 0; i < LINHAS; i++) {

somaDiagonal += numeros[i][i]

}

escreva("Soma total dos valores: " + somaValores + "\nSoma Primeira Diagonal: " + somaDiagonal)

}

}