

UNIVERSIDADE FEDERAL DO PARÁ

ICEN – INSTITUTO DE CIENCIA EXATAS E NATURAIS

FACOMP – FACULDADE DE COMPUTAÇÃO

CURSO DE CIENCIA DA COMPUTAÇÃO – 2021 TURMA: EN05210

DISCENTE: CHRISTIAN DE JESUS DA COSTA MARINHO

Matricula: 202004940041

**LISTA 6 DE EXERCICIO DE MATRIZES**

**ALGORITIMO**

BELÉM

2021

1. Desenvolver um algoritmo que receba os valores de uma matriz de 4 linhas e 4 colunas é mostre quais são os elementos da diagonal principal.

programa

{

funcao inicio()

{

inteiro matriz [4][4]

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

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

escreva ("Digite um número para a posição indicada na Matriz: \n")

escreva ("Para a linha ", i+1," e coluna ", j+1,": ")

leia(matriz[i][j])

limpa()

}

}

escreva ("Os valores da linha diagonal são: \n")

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

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

se (i == j) {

escreva ("[", matriz[i][j], "]")

} senao {

escreva ("[ ]")

}

}

escreva ("\n")

}

}

}

1. Desenvolver um algoritmo para somar duas matrizes e exibir o resultado. O usuário deve escolher a dimensão das mesmas.

programa

{

funcao inicio()

{

inteiro linha, coluna

inteiro matrizA[1000][1000], matrizB[1000][1000], matrizC[1000][1000]

/\*Como o Portugol não deixa criar matrizes dinamicas, fiz matrizes com tamanho muito grande que vai ser delimitado pelo usuário\*/

faca {

escreva ("Digite o número de linhas para as Matrizes: ")

leia(linha)

escreva ("Digite o número de colunas para as Matrizes: ")

leia(coluna)

se (linha != coluna) {

escreva("Dimensões das Matrizes diferentes\n")

escreva("Entre com dimensões iguais\n")

}

} enquanto (linha != coluna)

limpa()

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

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

escreva ("Digite um número na Matriz A : \n")

escreva("Para a linha ", i+1," coluna ", j+1,": ")

leia(matrizA[i][j])

limpa()

}

}

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

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

escreva ("Digite um número na Matriz B : \n")

escreva("Para a linha ", i+1," coluna ", j+1,": ")

leia(matrizB[i][j])

limpa()

}

}

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

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

escreva ("[", matrizA[i][j], "]")

}

escreva ("\n")

}

escreva ("\n+\n\n")

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

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

escreva ("[", matrizB[i][j], "]")

}

escreva ("\n")

}

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

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

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

matrizC[i][j] = matrizA[i][j] + matrizB[i][j]

}

}

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

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

escreva ("[", matrizC[i][j], "]")

}

escreva ("\n")

}

}

}

1. Desenvolver um algoritmo para multiplicar duas matrizes e exibir o resultado. O usuário deve escolher a dimensão das mesmas.

programa

{

funcao inicio()

{

inteiro linha, coluna

inteiro matrizA[1000][1000], matrizB[1000][1000], matrizC[1000][1000]

faca {

escreva ("Digite o número de linhas para as Matrizes: ")

leia(linha)

escreva ("Digite o número de colunas para as Matrizes: ")

leia(coluna)

se (linha != coluna) {

escreva("Dimensões das Matrizes diferentes\n")

escreva("Entre com dimensões iguais\n")

}

} enquanto (linha != coluna)

limpa()

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

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

escreva ("Digite um número na Matriz A : \n")

escreva("Para a linha ", i+1," coluna ", j+1,": ")

leia(matrizA[i][j])

limpa()

}

}

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

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

escreva ("Digite um número na Matriz B : \n")

escreva("Para a linha ", i+1," coluna ", j+1,": ")

leia(matrizB[i][j])

limpa()

}

}

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

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

escreva ("[", matrizA[i][j], "]")

}

escreva ("\n")

}

escreva ("\nX\n\n")

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

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

escreva ("[", matrizB[i][j], "]")

}

escreva ("\n")

}

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

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

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

matrizC[i][j] = matrizA[i][j] \* matrizB[i][j]

}

}

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

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

escreva ("[", matrizC[i][j], "]")

}

escreva ("\n")

}

}

}

1. Criar um algoritmo que leia os elementos de uma matriz inteira 10 x 10 e escreva todos os elementos, exceto os elementos da diagonal principal.

programa

{

inclua biblioteca Util --> u

funcao inicio()

{

inteiro matriz [10][10]

para (inteiro i = 0; i < 10; i++) {

para (inteiro j = 0; j < 10; j++) {

escreva ("Digite um número para a posição indicada na Matriz: \n")

escreva ("Para a linha ", i+1," e coluna ", j+1,": ")

leia(matriz[i][j])

limpa()

}

}

escreva ("Os valores da Matriz sem sua Diagonal são: \n")

para (inteiro i = 0; i < 10; i++) {

para (inteiro j = 0; j < 10; j++) {

se (i == j) {

escreva ("[ ]")

} senao {

escreva ("[", matriz[i][j] ,"]")

}

}

escreva ("\n")

}

}

}

1. Entrar com valores inteiros para uma matriz A4x4 e para uma matriz B4x4. Gerar e imprimir a SOMA (A+B).

programa

{

funcao inicio()

{

inteiro matrizA [4][4], matrizB[4][4], matrizC[4][4]

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

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

escreva ("Digite um número na Matriz A : \n")

escreva("Para a linha ", i+1," coluna ", j+1,": ")

leia(matrizA[i][j])

limpa()

}

}

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

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

escreva ("Digite um número na Matriz B : \n")

escreva("Para a linha ", i+1," coluna ", j+1,": ")

leia(matrizB[i][j])

limpa()

}

}

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

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

escreva ("[", matrizA[i][j], "]")

}

escreva ("\n")

}

escreva ("\n+\n\n")

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

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

escreva ("[", matrizB[i][j], "]")

}

escreva ("\n")

}

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

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

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

matrizC[i][j] = matrizA[i][j] + matrizB[i][j]

}

}

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

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

escreva ("[", matrizC[i][j], "]")

}

escreva ("\n")

}

}

}

1. Criar um algoritmo que possa armazenar as alturas de dez atletas de cinco delegações que participarão dos jogos de verão. Imprimir a maior altura de cada delegação.

programa

{

funcao inicio()

{

real matriz[5][10], vet[5]

real alt, altm=0.0

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

para (inteiro j=0; j < 10; j++) {

escreva ("Digite a altura do ",j+1,"º atleta da delegação ",i+1,": ")

leia(alt)

limpa()

se (alt > altm) {

altm = alt

}

se (j == 9) {

vet[0+i] = altm

}

}

altm = 0.0

}

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

escreva("A maior altura da delegação ",i+1," é: ", vet[i],"\n")

}

}

}

1. Uma floricultura conhecedora de sua clientela gostaria de fazer um algoritmo que pudesse controlar sempre um estoque mínimo de determinadas plantas, pois todo dia, pela manhã, o dono faz novas aquisições, criar um algoritmo que deixe cadastrar 50 tipos de plantas e nunca deixar o estoque ficar abaixo do ideal. Para cada planta, o dono gostaria de cadastrar o nome, o estoque ideal e a quantidade em estoque. Dessa forma a algoritmo pode calcular a quantidade que o dono da loja precisa comprar no próximo dia. Essa quantidade a ser comprada deve ser impressa (quando maior que zero) como uma lista para o dono da floricultura.

programa

{

funcao inicio()

{

cadeia nome [50]

inteiro estoque [50], estoqueI [50]

caracter r

para (inteiro i=0; i < 50; i++) {

escreva ("Digite o nome da planta: ")

leia(nome[i])

escreva ("Digite o estoque desta planta: ")

leia(estoque[i])

escreva ("Digite o estoque ideal para essa planta: ")

leia (estoqueI[i])

escreva ("Deseja cadastrar uma nova planta?\n")

escreva ("s --> sim / n --> não\n")

leia(r)

se (r == 'n') {

pare

}

limpa()

}

para (inteiro i=0; i < 50; i++) {

se (estoque[i] < estoqueI[i]) {

escreva ("A planta ",nome[i]," está com estoque baixo. ")

escreva ("Precisa comprar mais ", estoqueI[i] - estoque[i],"\n")

}

}

}

}

1. Crie um algoritmo que receba duas matrizes ACxD e BExF (C, D, E e F < 6). Esse algoritmo deve verificar se o produto matricial de A por B é possível (D = E). Caso seja possível, calcular o tal produto, imprimindo a matriz GCxF resultado.

programa

{

funcao inicio()

{

real matrizA[6][6], matrizB[6][6], matrizC[6][6]

inteiro linhaA, colunaA, linhaB, colunaB

escreva ("Entre com Matrizes 6x6 ou menor!\n")

faca {

escreva("Digite a quantidade de linhas da Matriz A: ")

leia(linhaA)

escreva("Digite a quantidade de colunas da Matriz A: ")

leia(colunaA)

escreva("Digite a quantidade de linhas da Matriz B: ")

leia(linhaB)

escreva("Digite a quantidade de colunas da Matriz B: ")

leia(colunaB)

limpa()

se (linhaA > 6 ou colunaA > 6 ou linhaB >6 ou colunaB > 6) {

escreva ("Uma entrada inválida ocorreu!\nPor favor digite um numero entre 0 e 6\n")

}

} enquanto (linhaA > 6 ou colunaA > 6 ou linhaB >6 ou colunaB > 6)

se (colunaA == linhaB) {

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

para (inteiro j=0; j < colunaA; j++) {

escreva ("Digite um número na Matriz A : \n")

escreva("Para a linha ", i+1," coluna ", j+1,": ")

leia(matrizA[i][j])

limpa()

}

}

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

para (inteiro j=0; j < colunaB; j++) {

escreva ("Digite um número na Matriz B : \n")

escreva("Para a linha ", i+1," coluna ", j+1,": ")

leia(matrizB[i][j])

limpa()

}

}

escreva ("O produto entre as Matrizes é: \n")

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

para (inteiro j=0; j< colunaB; j++) {

real soma = 0.0

para (inteiro h=0; h < colunaA; h++) {

soma += (matrizA[i][h] \* matrizB[h][j])

}

matrizC[i][j] = soma

escreva ("[",matrizC[i][j],"]")

}

escreva("\n")

}

} senao {

escreva ("\nLinha da Matriz A nao é igual à coluna da Matriz B\nimpossivel fazer a multiplicação!")

}

}

}