

- 1) Construa um algoritmo que carregue um vetor de seis elementos numéricos inteiros, calcule e mostre: § A quantidade de números pares; § Quais os números pares; § A quantidade de números ímpares; § Quais os números ímpares.

INICIO**DECLARE**numero[6] : **VETOR** de **INTEIRO**qtPares, qtImpares : **INTEIRO**

qtPares <- 0 , qtImpares <- 0

// Carregar o vetor

PARA i = 0 **ATE** 5 **FAÇA****ESCREVA** "Digite um número inteiro:"**LEIA** numero[i]**FIM PARA**

// Verificar se é par ou ímpar

PARA i = 0 **ATE** 5 **FAÇA****SE** numero[i] % 2 = 0 **ENTÃO**

qtPares <- qtPares + 1

ESCREVA "O número" numero[i] " é PAR."**SENÃO**

qtImpares <- qtImpares + 1

ESCREVA "O número" numero[i] " é ÍMPAR."**FIM SE****FIM PARA****ESCREVA** "No vetor 'número' foram encontrados ", qtPares , " PARES"**ESCREVA** "No vetor 'número' foram encontrados ", qtImpares , " ÍMPARES"**FIM**

- 2) carregue um vetor com 15 elementos inteiros e verifique a existência de elementos iguais a 30, mostrando as posições em que esses elementos apareceram.

INICIO**DECLARE**numero[15] : **VETOR** de **INTEIRO**

// Carregar o vetor

PARA i = 0 **ATE** 14 **FAÇA****ESCREVA** "Digite um número inteiro:"**LEIA** numero[i]**FIM PARA**

// Verificar se é igual a 30

PARA i = 0 **ATE** 14 **FAÇA****SE** numero[i] = 30 **ENTÃO****ESCREVA** "O elemento [" , i , "] do vetor é: " , numero [i]**FIM SE****FIM PARA****FIM**

- 3) Construa um algoritmo que receba o total das vendas de cada vendedor e os armazene em um vetor. Receba também o percentual de comissão de cada vendedor e armazene-os em um outro vetor. Receba os nomes desses vendedores e armazene-os em um terceiro vetor. Existem apenas dez vendedores. Calcule e mostre:

§ Um relatório com os nomes dos vendedores e os valores a receber;

§ O total das vendas de todos os vendedores;

§ O maior valor a receber e quem o receberá;

§ O menor valor a receber e quem o receberá;

INICIO

DECLARE

vendedor[10] : **VETOR** de **LITERAL**

vendasVendedor[10] , porcVendedor[10] : **VETOR** de **DECIMAL**

totalVendas, maiorComissao, menorComissao : **DECIMAL**

campeao, lanterninha : **INTEIRO**

// Carregar os vetores e calcular

PARA i = 0 **ATE** 9 **FAÇA**

ESCREVA "Digite o nome do vendedor:"

LEIA vendedor[i]

ESCREVA "Digite o total de vendas deste vendedor: R\$ "

LEIA vendasVendedor[i]

totalVendas <- totalVendas + vendasVendedor[i]

ESCREVA "Digite o percentual de comissão deste vendedor: "

LEIA porcVendedor[i]

FIM PARA

// Classificar os vendedores

menorComissao <- totalVendas

maiorComissao <- 0

PARA i = 0 **ATE** 9 **FAÇA**

comissao <- vendasVendedor[i] * porcVendedor[i]

SE comissao > maiorComissao **ENTÃO**

campeao <- i

maiorComissao <- comissao

FIM SE

SE comissao < menorComissao **ENTÃO**

lanterninha <- i

menorComissao <- comissao

FIM SE

ESCREVA vendedor[i] , " tem a receber: R\$ ", comissao

FIM PARA

ESCREVA "O total de vendas de todos os vendedores foi: R\$ ", totalVendas

ESCREVA vendedor [campeao] " receberá R\$ ", maiorComissao

ESCREVA vendedor [lanterninha] " receberá R\$ ", menorComissao

FIM

- 4) Construa um algoritmo que receba o nome de cinco produtos e seus respectivos preços em dois vetores, calcule e mostre:

§ A quantidade de produtos com preço inferior a R\$ 50,00

§ O nome dos produtos com preço entre R\$ 50,00 e R\$ 100,00

§ A média dos preços dos produtos com preço superior a R\$ 100,00.

INICIO

DECLARE

prodNome [5] , prodNomeMedio [5] : **VETOR** de **LITERAL**

prodPreco [5] : **VETOR** de **REAL**

precoBaixo, contAcima : **INTEIRO**

mediaAcima : **REAL**

precoBaixo <- 0, contAcima <- 0, mediaAcima <- 0

PARA i = 0 **ATÉ** 4 **FAÇA**

ESCREVA "Informe nome do Produto:"

LEIA prodNome [i]

ESCREVA "Informe preço do Produto (R\$):"

LEIA prodPreco [i]

// Contando os produtos de preços mais baixos.

SE prodPreco [i] < 50.00 **ENTÃO**

precoBaixo <- precoBaixo + 1

SENAO

// Guardando os nomes dos produtos com preço médio.

SE prodPreco [i] >= 50.00 **E** prodPreco [i] <= 100.00

ENTÃO

prodNomeMedio [i] <- prodNome [i]

// Contando os produtos de preços mais altos.

SENÃO

mediaAcima <- mediaAcima + prodPreco [i]

contAcima <- contAcima + 1

FIM SE

FIM SE

FIM PARA

// Imprimindo relatório.

ESCREVA precoBaixo , " produtos têm preço menor que R\$50,00."

PARA i = 0 **ATÉ** 4 **FAÇA**

SE prodPreco [i] >= 50.00 **E** prodPreco [i] <= 100.00

ENTÃO

ESCREVA prodNomeMedio [i],

" é um produto com preço entre
R\$ 50,00 e R\$ 100,00 ."

FIM SE

FIM PARA

ESCREVA "A média dos preços dos produtos com preço superior a
R\$ 100,00 é: ", (mediaAcima / contAcima)

FIM

- 5) Construa um algoritmo que leia um vetor de 15 posições com números inteiros. Crie em seguida um vetor resultante (máximo 15 posições) que contenha apenas os números primos do primeiro vetor. Escreva o vetor resultante. Para este exercício utilize a função de calcular primo implementada na lista 3.

INICIO**DECLARE**

numeros [15], primos [?] : **VETOR** de **INTEIRO**

contaPrimos : **INTEIRO**

contaPrimos <- 0

PARA i = 0 **ATE** 14 **FAÇA**

ESCREVA "Informe um número inteiro:"

LEIA numeros [i]

SE numeros [i] % 2 = 0 **ENTÃO**

primos [contaPrimos] <- numeros [i]

contaPrimos <- contaPrimos + 1

FIM SE

FIM PARA

primos [contaPrimos] : **VETOR** de **INTEIRO**

PARA i = 0 **ATE** (contaPrimos - 1) **FAÇA**

ESCREVA "Este número é Primo: ", primos [i]

FIM PARA**FIM**

- 6) Construa um algoritmo que leia e exiba uma matriz de tamanho definido pelo usuário.

INICIO**DECLARE**

linha, coluna, elemento : **INTEIRO**

linha <- 1 , coluna <- 1

matriz [linha, coluna] : **VETOR** de **INTEIRO**

ESCREVA "Qual o número de linhas da Matriz? "

LEIA linha

ESCREVA "Qual o número de colunas da Matriz? "

LEIA coluna

matriz [linha, coluna] : **VETOR** de **INTEIRO**

PARA i = 0 **ATE** linha **FAÇA**

PARA c = 0 **ATE** coluna **FAÇA**

ESCREVA "Digite um inteiro para a posição: ", i, ", ", c

LEIA elemento

FIM PARA

FIM PARA

PARA i = 0 **ATE** linha **FAÇA**

PARA c = 0 **ATE** coluna **FAÇA**

ESCREVA "O elemento [", i, ", ", c, "]" é: ", matriz [i, c]

FIM PARA

FIM PARA

FIM

- 7) Construa um algoritmo que carregue uma matriz 10 x 20 com números inteiros e some cada uma das linhas, armazenando o resultado das somas em um vetor. A seguir multiplique cada elemento da matriz pela soma da linha e mostre a matriz resultante.

INÍCIO**DECLARE**

elemento : **INTEIRO**

matriz [10 , 20] , produtos [10 , 20] : **VETOR** de **INTEIRO**

somaLinhas [10] : **VETOR** de **INTEIRO**

// Carregando uma matriz 10 x 20 com números inteiros.

PARA i = 0 **ATE** 9 **FAÇA**

somaLinhas [i] <- 0

PARA c = 0 **ATE** 19 **FAÇA**

ESCREVA "Digite um inteiro para a posição: ", i " , ", c

LEIA matriz [i , c]

// Armazenando o resultado das somas em um vetor.

somaLinhas [i] <- matriz [i , c]

FIM PARA**FIM PARA**

// Gerando a matriz produtos.

PARA i = 0 **ATE** 9 **FAÇA**

produtos [i , c] <- 0

PARA c = 0 **ATE** 19 **FAÇA**

// Multiplicando cada elemento da matriz pela soma da linha.

produtos [i , c] <- somaLinhas [i] * matriz [i , c]

// Mostrando a matriz resultante "produtos[]".

ESCREVA " produtos [" , i " , " , c "] = " , produtos [i , c]

FIM PARA**FIM PARA****FIM**

- 8) Construa um algoritmo que receba as vendas semanais (de um mês) de cinco vendedores (cada vendedor corresponderá a uma linha da matriz) de uma loja e armazene essas vendas em uma matriz. Calcule e mostre:

§ O total de vendas do mês de cada vendedor;

§ O total de vendas de cada semana (todos os vendedores);

§ O total de vendas do mês;

INICIO

DECLARE

totalVendasMes : **INTEIRO**

vndsSemana : **REAL**

vndsSemanaisVendedores [5 , 4] : **VETOR** de **REAL**

vndsMesVendedor [5], vndsCadaSemana [4] : **VETOR** de **REAL**

// Recebendo as vendas semanais de cada vendedor.

PARA i = 0 **ATE** 4 **FAÇA**

vndsMesVendedor [i] <- 0

PARA c = 0 **ATE** 3 **FAÇA**

ESCREVA "Informe o total de vendas do vendedor: ",
i + 1 " na semana ", c + 1

LEIA vndsSemana

// Carregando o input na matriz principal.

vndsSemanaisVendedores [i , c] <- vndsSemana

// Acumulando o total do mês de cada vendedor.

vndsMesVendedor [i] <- vndsSemana

FIM PARA

// § O total de vendas do mês de cada vendedor:

ESCREVA " O total de vendas neste mês do vendedor", i + 1, " foi ",
vndsMesVendedor [i]

FIM PARA

// Acumular o total de cada semana (todos os vendedores).

// Travando a coluna da matriz principal.

PARA c = 0 **ATE** 3 **FAÇA**

// Percorrendo as linhas da matriz principal.

vndsCadaSemana [c] <- 0

PARA i = 0 **ATE** 4 **FAÇA**

vndsCadaSemana [c] <- vndsCadaSemana [c] +
vndsSemanaisVendedores [i , c]

FIM PARA

// Imprimindo os resultados semana a semana.

ESCREVA "§ O total de vendas da semana ", c+1, " (todos os
vendedores) foi de R\$ ", vndsCadaSemana [c]

FIM PARA

FIM

- 9) Construa um algoritmo que leia a quantidade de produtos a ser digitada e solicite ao usuário a digitação dos preços de 5 lojas diferentes (cada loja corresponderá a uma coluna da matriz). Desconsiderando empates mostre o número do produto e o número da loja do produto mais caro.

INICIO**DECLARE**

nProd, maisCaroL, maisCaroC : **INTEIRO**

maisCaro : **REAL**

prodsLojas [nProd, 5] : **VETOR** de **REAL**

maisCaro <- 0, maisCaroL <- 0, maisCaroC <- 0

ESCREVA "Quantos produtos serão cadastrados? "

LEIA nProd

PARA i = 0 **ATE** nProd-1 **FAÇA**

PARA c = 0 **ATE** 4 **FAÇA**

// Cadastrando produtos.

ESCREVA "Informe o preço do produto", i+1 , " na Loja ", c+1,
" em (R\$):"

LEIA prodsLojas [i, c]

// Comparar os preços para decidir qual o mais caro.

// Será mostrado o primeiro registrado em caso de empate.

SE prodsLojas [i, c] > maisCaro **ENTÃO**

maisCaroL <- i

maisCaroC <- c

FIM SE

FIM PARA

FIM PARA

ESCREVA maisCaroL+1 , " é o número do produto e ", maisCaroC+1 , " é
o número da loja do produto mais caro. "

FIM

- 10) Construa um algoritmo que leia e preencha uma matriz de números inteiros de dimensões informadas pelo usuário. Apresente a soma dos elementos desta matriz.

INICIO**DECLARE**

qtLinhas, qtColunas, soma : **INTEIRO**

matriz [qtLinhas, qtColunas] : **VETOR** de **INTEIRO**

soma <- 0

ESCREVA "Quantas linhas terá a matriz? "

LEIA qtLinhas

ESCREVA "Quantas colunas terá a matriz? "

LEIA qtColunas

PARA i = 0 **ATE** qtLinhas-1 **FAÇA**

PARA c = 0 **ATE** qtColunas-1 **FAÇA**

// Populando a matriz.

ESCREVA "Insira um número inteiro qualquer."

LEIA matriz [i, c]

// Acumulando a soma dos elementos.

soma <- soma + matriz [i, c]

FIM PARA

FIM PARA

// Imprimindo o resultado.

ESCREVA " A soma de todos os elementos da matriz é: ", soma

FIM