

Fundamentos de Programação 1

Estrutura de Dados Homogêneas

“vetores e matrizes”

Slides 15 – ‘Complementar’

Prof. SIMÃO

Jean Marcelo SIMÃO

Exercício

Elabore um algoritmo que recebe informações de 30 alunos, sendo estas informações o nome e as duas notas de cada um.

O Algoritmo deve ainda fornecer a média de notas de cada aluno apresentado seu nome.

Solução para o exercício – v1

Algoritmo ‘Nomes e notas de alunos’.

Início

Real notas[3];

Caracter nome[150];

Inteiro cont;

para cont **de** 0 **até** 29 **passo** 1 **faça**

Imprima (“Digite o nome do aluno: ”);

Leia (nome[0], nome[1], ..., nome[149]);

Imprima (“Digite as duas notas do aluno”);

Leia (nota[0], nota[1]);

 notas[2] ← (notas[0] + notas[1]) / 2;

Imprima (“ A média do aluno %s é %i: ”, nome, notas[2]);

 // %s é de *string* que pode ser traduzido como ‘frase’.

fim-para

Fim.

Solução para o exercício – v2

Algoritmo ‘Nomes e notas de alunos’.

Início

Real notas[3];

Caracter nome[150];

Inteiro cont, cont2;

para cont **de** 0 **até** 29 **passo** 1 **faça**

Imprima (“Digite o nome do aluno finalizando por !: ”);

Leia (-nome[0],-nome[1],..., nome[149]);----->

Imprima (“Digite as duas notas do aluno”);

Leia (nota[0], nota[1]);

 notas[2] ← notas[0] + notas[1];

Imprima (“ A média do aluno %s é %i: ”, nome, notas[2]);

 // %s é de *string* que pode ser traduzido como ‘frase’.

fim-para

Fim.

para cont2 **de** 0 **até** 149 **passo** 1 **faça**

Leia (nome [cont2]);

se (nome [cont2] = ‘!’) **então**

pare!

fim-se

fim-para

Solução para o exercício – v2

Algoritmo ‘Nomes e notas de alunos’.

Início

Real notas[3];

Caracter nome[150];

Inteiro cont, cont2;

para cont **de** 0 **até** 29 **passo** 1 **faça**

Imprima (“Digite o nome do aluno finalizando por !: ”);

Leia (-nome[0],-nome[1],..., nome[149]-);

Imprima (“Digite as duas notas do aluno”);

Leia (nota[0], nota[1]);

notas[2] ← notas[0] + notas[1];

Imprima (“ A média do aluno %s é %i: ”, nome, notas[2]);

// %s é de *string* que pode ser traduzido como ‘frase’.

fim-para

Fim.

cont2 ← 0;

repita

Leia (nome [cont2]);

cont2 ← cont2 + 1;

se (nome [cont2] = ‘!’) **então**

pare!

fim-se

até (cont2 > 150)

Solução para o exercício – v.3

Algoritmo ‘Nomes e notas de alunos’.

Início

Real notas[3];

Caracter nome[150];

Inteiro cont;

para cont **de** 0 **até** 29 **passo** 1 **faça**

Imprima (“Digite o nome do aluno: ”);

Leia (nome);

// Vamos considerar que o comando é capaz de ler um vetor de caracteres...

Imprima (“Digite a primeira nota do aluno”);

Leia (nota[0]);

Imprima (“Digite a segunda nota do aluno”);

Leia (nota[1]);

notas[2] ← notas[0] + notas[1];

Imprima (“ A média do aluno %s é %i: ”, nome, notas[2]);

// %s é de *string* que pode ser traduzido como ‘frase’.

fim-para

Fim.

Solução para o exercício – v.4

Algoritmo ‘Nomes e notas de alunos’.

Início

Real notas[3];

Caracter nome[150];

Inteiro cont;

para cont **de** 0 **até** 29 **passo** 1 **faça**

Imprima (“Digite o nome do aluno: ”);

Leia (nome);

repita

Imprima (“Digite a primeira nota do aluno”);

Leia (nota[0]);

se ((nota[0] < 0) **ou** (nota[0] > 10)) **então**

Imprima (“Nota não válida.”);

fim-se

até ((nota[0] >= 0) **e** (nota[0] <= 10))

repita

Imprima (“Digite a segunda nota do aluno”);

Leia (nota[1]);

se ((nota[1] < 0) **ou** (nota[1] > 10)) **então**

Imprima (“Nota não válida.”);

fim-se

até ((nota[1] >= 0) **e** (nota[1] <= 10))

 notas[2] ← notas[0] + notas[1];

Imprima (“ A média do aluno %s é %i: ”, nome, notas[2]);

fim-para

Fim.

Solução para o exercício – v.5

Algoritmo ‘Nomes e notas de alunos’.

Início

Real notas[3];

Caracter nome[150];

Inteiro cont;

para cont **de** 0 **até** 29 **passo** 1 **faça**

Imprima (“Digite o nome do aluno: ”);

Leia (nome);

faça

Imprima (“Digite a primeira nota do aluno”);

Leia (nota[0]);

se ((nota[0] < 0) **ou** (nota[1] > 10)) **então**

Imprima (“Nota não válida.”);

fim-se

enquanto ((nota[0] < 0) **ou** (nota[0] > 10))

faça

Imprima (“Digite a primeira nota do aluno”);

Leia (nota[1]);

se ((nota[1] < 0) **ou** (nota[1] > 10)) **então**

Imprima (“Nota não válida.”);

fim-se

enquanto ((nota[1] < 0) **ou** (nota[1] > 10))

 notas[2] ← notas[0] + notas[1];

Imprima (“ A média do aluno %s é %i: ”, nome, notas[2]);

fim-para

Fim.

Exercício

Elabore um algoritmo para ler o nome e o sobrenome de dez pessoas, imprimindo-os no seguinte formato: *sobrenome, nome*.

-Faça uma primeira versão do algoritmo lendo cada *frase* de uma única vez.

- Faça uma segunda versão do algoritmo lendo caracter a caracter de cada frase..

Obs: *frase* significa um vetor de caracteres...

Estrutura de Dados Homogênea “matrizes”

Uma matriz computacional é um “variável composta” capaz de armazenar uma quantidade determinada de valores em duas dimensões.

Uma matriz computacional é inspirada na matriz matemática, que também é capaz de armazenar um conjunto de valores em duas dimensões.

1	3	7
0	1	0
1	9	9

Declaração de Matrizes em algoritmos.

tipo IDENTIFICADOR = **vetor** [LI1..LF1,LI2..LF2] **de** < tipo >;
IDENTIFICADOR: lista de variáveis;

Onde : LI1...F1 e LI2 .. LF2 são os limites de cada dimensão da matriz.

< tipo > - representa qualquer um dos tipos básicos ou tipo anteriormente definido.

Exemplo 1.1

Algoritmo 'Exemplo Matriz...'.
Início

tipo T_matriz_2x2 = **vetor** [0..1] de **Real**;
T_matriz_2x2 matriz;

Real media, quant;


matriz[0,0] ← 9.0;
matriz[0,1] ← 8.0;
matriz[1,0] ← 7.0;
matriz[1,1] ← 7.0;

quant ← matriz[0,0] + matriz[0,1]
+ matriz[1,0] + matriz[1,1];

media ← quant / 4;

Imprima (" A média dos 4 números é: %i.", media);

Fim.



Índices	0	1
0	9.0	8.0
1	7.0	7.0

Declaração de Matrizes em algoritmos – compatível com a linguagem C.

<tipo-básico> NOME_MATRIZ [Número1] [Número2];

Onde:

Número1 e *Número2* representam as dimensões da matriz

< tipo-básico > - representa qualquer um dos tipos básicos

Exemplo 1.1

Algoritmo 'Exemplo Matriz...'.
Início

Real matriz[2][2];

Real media, quant;

matriz[0,0] ← 9.0;

matriz[0,1] ← 8.0;

matriz[1,0] ← 7.0;


matriz[1,1] ← 7.0;

quant ← matriz[0,0] + matriz[0,1]
+ matriz[1,0] + matriz[1,1];

media ← quant / 4;

Imprima (" A média dos 4 números é: %i.", media);

Fim.



Índices	0	1
0	9.0	8.0
1	7.0	7.0

Exemplo 1.2

Algoritmo 'Exemplo Matriz...'.
Início

Real matriz[2][2];

Real media, quant;

Imprima ("Digite o 1o número:");

Leia (matriz[0,0]);

Imprima ("Digite o 2o número:");

Leia (matriz[0,1]);

Imprima ("Digite o 3o número:");

Leia (matriz[1,0]);

Imprima ("Digite o 4o número:");

Leia (matriz[1,1]);

quant ← matriz[0,0] + matriz[0,1] + matriz[1,0] + matriz[1,1];

media ← quant / 4;

Imprima (" A média dos 4 números é: %i.", media);

Fim.

Exemplo 1.3

Algoritmo 'Exemplo Matriz 2...'.
Início

Real matriz[4][2];

Real media, quant;

Inteiro cont1, cont2;

quant ← 0;

para cont1 **de** 0 **até** 3 **passo** 1 **faça**

para cont2 **de** 0 **até** 1 **passo** 1 **faça**

Imprima (“Digite o número de posição %, % da matriz:”, cont1, cont2);

Leia (matriz[cont1, cont2]);

 quant ← quant + matriz[cont1, cont2] ;

fim-para

fim-para

media ← quant / 8;

Imprima (“ A média dos 8 números da matriz é: %i.”, media);

Fim.

Exercício

- a) **Elabore um algoritmo que registra os nomes de 50 alunos (*i.e.* matriz de caracter de 50 por 150) e as quatro notas de cada um dos alunos (*i.e.* matriz de reais de 50 por 4).**
- b) **O algoritmo deve ainda fornecer os nomes dos alunos que tem nota maior que a média da turma.**

Exemplo 2a

Algoritmo 'Idade Nadadores ...'.

Início

tipo T_vetor_idades = vetor [0..49] de Real;
T_vetor_idades vetor_idades;

Inteiro cont, quant;

Real idade soma, maior, menor;

soma ← 0; maior ← 0; menor ← 200;

para cont de 0 até 49 **passo** 1 **faça**

Imprima ("Digite a idade do nadador: ");

Leia (idade);

se ((idade >= 5) e (idade <= 100)) **então**

 // ...

 soma ← soma + idade;

se (idade < menor) **então**

 menor ← idade;

senão

se (idade > maior) **então**

 maior ← idade;

fim-se

fim-se

vetor_idades[cont] = idade;

senão

Imprima (" Idade inválida");

 cont ← cont -1;

fim-se

fim-para

media ← soma / 150;

quant ← 0;

para cont de 0 até 49 **passo** 1 **faça**

se (vetor_idades [cont] > media) **então**

 quant ← quant + 1;

fim-se

fim-para

Imprima (" A média de idade é %f: ", media);

Imprima (" A maior idade é %f:", maior);

Imprima (" A menor idade é %f.", menor);

Imprima (" A quantidade de idades acima da média
é %i: ", quant);

Fim.

Exemplo 2b

Algoritmo 'Idade Nadadores ...'.

Início

Real vetor_idades[50];

Inteiro cont, quant;

Real idade, soma, média, maior, menor;

soma ← 0; maior ← 0; menor ← 200;

para cont **de** 0 **até** 49 **passo** 1 **faça**

Imprima ("Digite a idade do nadador: ");

Leia (idade);

se ((idade >= 5) **e** (idade <= 100)) **então**

 // ...

 soma ← soma + idade;

se (idade < menor) **então**

 menor ← idade;

fim-se

se (idade > maior) **então**

 maior ← idade;

fim-se

 vetor_idades[cont] ← idade;

senão

Imprima (" Idade inválida");

 cont ← cont -1;

fim-se

fim-para

média ← soma / 50;

quant ← 0;

para cont **de** 0 **até** 49 **passo** 1 **faça**

se (vetor_idades [cont] > média) **então**

 quant ← quant + 1;

fim-se

fim-para

Imprima (" A média de idade é %f: ", media);

Imprima (" A maior idade é %f:", maior);

Imprima (" A menor idade é %f.", menor);

Imprima (" A quantidade de idades acima da média
é %i: ", quant);

Fim.

Exercício

- Elabore um algoritmo que recebe 100 valores e os armazene em um vetor. O algoritmo deve ainda ter um outro vetor de 50 posições que recebe a soma de dois valores do primeiro vetor na seguinte ordem:
 - $\text{vet2}[0] \leftarrow \text{vet1}[0] + \text{vet1}[1]$,
 - $\text{vet2}[1] \leftarrow \text{vet1}[2] + \text{vet1}[3]$
 - e assim por diante.