

Fundamentos de Programação 1

Estrutura de Dados Homogêneas

“vetores”

Slides 14 – ‘Complementar’

Prof. SIMÃO

Jean Marcelo SIMÃO

Retomando o exercício – 1a.

Elabore um algoritmo que, dada a idade de um nadador (entrada do algoritmo), classifique-o de acordo com as seguintes categorias:

- infantil A = 5 – 7 anos;
- infantil B = 8 – 10 anos;
- juvenil A = 11 – 13 anos;
- juvenil B = 14 – 17 anos;
- adulto = maiores ou igual a 18 anos.

Uma solução para o exercício

Algoritmo 'Idade Nadador'.

Início

Real idade;

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

Leia (idade);

se ((idade \geq 5) e (idade \leq 100)) **então**

se (idade $<$ 8) **então**

Imprima(" Idade na categoria infantil A. ");

fim-se

se ((idade \geq 8) e (idade $<$ 11)) **então**

Imprima(" Idade na categoria infantil B.");

fim-se

se ((idade \geq 11) e (idade $<$ 14)) **então**

Imprima(" Idade na categoria juvenil A. ");

fim-se

se ((idade \geq 14) e (idade $<$ 18)) **então**

Imprima(" Idade na categoria juvenil B.");

fim-se

se (idade \geq 18) **então**

Imprima(" Idade na categoria adulto.");

fim-se

senão

Imprima(" Idade inválida");

fim-se

Fim.

Retomando o exercício – 1b.

Refaça o algoritmo anterior para o caso de 50 nadadores, definindo a média de idade dos nadadores, a idade do mais velho e a idade do mais jovem.

Uma solução para o exercício 1b

Algoritmo 'Idade Nadadores'.

Início

Inteiro cont;

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**

se (idade <= 7) **então**

Imprima(" Idade na categoria infantil A. ");

fim-se

se ((idade >= 8) e (idade <= 10)) **então**

Imprima(" Idade na categoria infantil B.");

fim-se

se ((idade >= 11) e (idade <= 13)) **então**

Imprima(" Idade na categoria juvenil A. ");

fim-se

se ((idade >= 14) e (idade <= 17)) **então**

Imprima(" Idade na categoria juvenil B.");

fim-se

se (idade >= 18) **então**

Imprima(" Idade na categoria adulto.");

fim-se

soma ← soma + idade;

se (idade < menor) **então**

menor ← idade;

fim-se

se (idade > maior) **então**

maior ← idade;

fim-se

senão

Imprima(" Idade inválida");

cont ← cont -1;

fim-se

fim-para

média ← soma / 50;

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

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

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

Fim.

Retomando o exercício – 1b.

Refaça o algoritmo anterior dizendo agora quantos nadadores têm idade maior que a média.

Estrutura de Dados Homogênea

“vetores”

Um vetor computacional é um “variável composta” capaz de armazenar uma quantidade determinada de valores.

Um vetor computacional é inspirada no vetor matemático, que também é capaz de armazenar um conjunto de valores.

Em geral, um vetor é uma matriz de uma única linha ou uma única coluna.

1	3	7	0	1	0	1	9	9
---	---	---	---	---	---	---	---	---

Declaração de Vetores em algoritmos.

tipo IDENTIFICADOR = **vetor** [LI..LF] **de** < tipo >;
IDENTIFICADOR: lista de variáveis;

Onde : LI representa o limite inicial do vetor;
LF representa o limite final do vetor;

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

Exemplo 1.1

Algoritmo ‘Exemplo Vetor...’.

Início

```
tipo T_vetor_nums = vetor [0..4] de Real;  
T_vetor_nums vet_nums;
```

```
real media, quant;
```

```
vet_nums[0] ← 9.0;
```

```
vet_nums[1] ← 8.0;
```

```
vet_nums[2] ← 7.0;
```

```
vet_nums[3] ← 7.0;
```

```
vet_nums[4] ← 7.0;
```

```
quant ← vet_nums[0] + vet_nums[1] + vet_nums[2] + vet_nums[3] + vet_nums[4];
```

```
media ← quant / 5;
```

```
Imprima (“ A média dos 5 números é: %i.”, media);
```

Fim.

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

<tipo-básico> NOME_VETOR [número];

Onde :

número representa a capacidade de armazenamento do vetor;

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

Exemplo 1.1b

Algoritmo ‘Exemplo vetor ...’.

Início

```
real vet_nums[5];
```

```
// 5 posições, de 0 à 4... (sempre começa por zero em C).
```

```
real media, quant;
```

```
vet_nums[0] ← 9.0;
```

```
vet_nums[1] ← 8.0;
```

```
vet_nums[2] ← 7.0;
```

```
vet_nums[3] ← 7.0;
```

```
vet_nums[4] ← 7.0;
```

```
quant ← vet_nums[0] + vet_nums[1] + vet_nums[2] + vet_nums[3] + vet_nums[4];
```

```
media ← quant / 5;
```

```
Imprima (“ A média dos 5 números é: %i.”, media);
```

Fim.

Exemplo 1.2

Algoritmo ‘Exemplo Vetor ...’.

Início

```
real vet_nums[5];
```

```
real media, quant;
```

```
Imprima (“Digite o 1º número:”);
```

```
Leia ( vet_nums[0] );
```

```
Imprima (“Digite o 2º número:”);
```

```
Leia ( vet_nums[1] );
```

```
Imprima (“Digite o 3º número:”);
```

```
Leia ( vet_nums[2] );
```

```
Imprima (“Digite o 4º número:”);
```

```
Leia ( vet_nums[3] );
```

```
Imprima (“Digite o 5º número:”);
```

```
Leia ( vet_nums[4] );
```

```
quant ← vet_nums[0] + vet_nums[1] + vet_nums[2] + vet_nums[3] + vet_nums[4];
```

```
media ← quant / 5;
```

```
Imprima (“ A média dos 5 números é: %i.”, media);
```

Fim.

Exemplo 1.3

Algoritmo 'Idade Nadadores ...'.

Início

real vet_nums[5];

real media, quant;

inteiro cont;

quant ← 0;

para cont **de** 0 **até** 4 **passo** 1 **faça**

Imprima ("Digite o %i^o número:", cont+1);

Leia (vet_nums[cont]);

 quant ← quant + vet_nums[cont] ;

fim-para.

media ← quant / 5;

Imprima (" A média dos 5 números é: %i.", media);

Fim.

Retomando o exercício

a) Elabore um algoritmo que, dada a idade de um nadador (entrada do algoritmo), classifique-o de acordo com as seguintes categorias:

- infantil A = 5 – 7 anos;
- infantil B = 8 – 10 anos;
- juvenil A = 11 – 13 anos;
- juvenil B = 14 – 17 anos;
- adulto = maiores ou igual a 18 anos.

b) Refaça o algoritmo anterior para o caso de 50 nadadores, definindo a média de idade dos nadadores, a idade do mais velho e a idade do mais jovem.

c) Refaça o algoritmo anterior dizendo agora quantos nadadores têm idade maior que a média.

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;

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

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.

Exemplo 2c

Algoritmo 'Idade Nadadores ...'.

Início

Real vetor_idades[50];

Real vetor_id_m[49];

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;

cont2 ← 0;

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

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

 quant ← quant + 1;

 vetor_id_m[cont2] ← vetor_idades [cont] ;

 cont2 ← cont2 + 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.