

Vetores e Matrizes

MATA37: Introdução à Lógica de Programação

Prof.: Rafael A. Melo (melo@dcc.ufba.br)

Departamento de Ciência da Computação

Instituto de Matemática

Universidade Federal da Bahia



Vetores

Vetor c :

- Um *vetor* é uma **estrutura de dados** que consiste de elementos de dados do mesmo tipo.
 - conjuntos de dados homogêneos
- Um *vetor* ocupa um grupo de posições consecutivas de memória, todas de mesmo nome e mesmo tipo;
- Cada elemento ocupa uma posição definida no vetor e pode ser referenciado através dela.
- O tamanho (ou dimensão) de um vetor é dado pelo seu número de elementos

c [1]	6
c [2]	0
c [3]	4
c [4]	-5
c [5]	9
c [6]	6
c [7]	24
c [8]	47
c [9]	21
c [10]	-37

Vetor c :

Nome do vetor: **c**

Cada elemento tem um (sobrenome) em particular: **c[1]**, **c[2]**, ..., **c[10]**.

Cada elemento tem um endereço.

Valor armazenado em c[1]: **6**

Valor armazenado em c[2]: **0**

Valor armazenado em c[3]: **4**

c [1]	6
c [2]	0
c [3]	4
c [4]	-58
c [5]	9
c [6]	6
c [7]	9124
c [8]	47
c [9]	21
c [10]	-357

Vetores - Declaração

vetor [li..ls] **de** <tipo>: identificador;

ou

tipo identificadorTipo = **vetor** [li..ls] **de** <tipo>;

identificadorTipo: identificadorVariavel;

vetor [1..40] **de reais**: VetNotasILP, VetNotasCalc;

ou

tipo **Notas** = **vetor** [1..40] **de reais**;

Notas: VetNotasILP, VetNotasCalc;

Notas é um
novo tipo
criado

VetNotasILP e VetNotasCalc são
“variáveis” do tipo Notas

Vetores

vetor [1..40] de reais: VetNotas

ou

tipo Notas = vetor [1..40] de reais

Notas: VetNotas

Na memória é
reservado espaço
para o vetor
VetNotas

VetNotas

8.5	7.3	9.6	6.4	...	7.3	8.9
1	2	3	4	...	39	40

Vetores

vetor [6..35] de reais: VetNotas

ou

tipo Notas = vetor [6..35] de reais

Notas: VetNotas

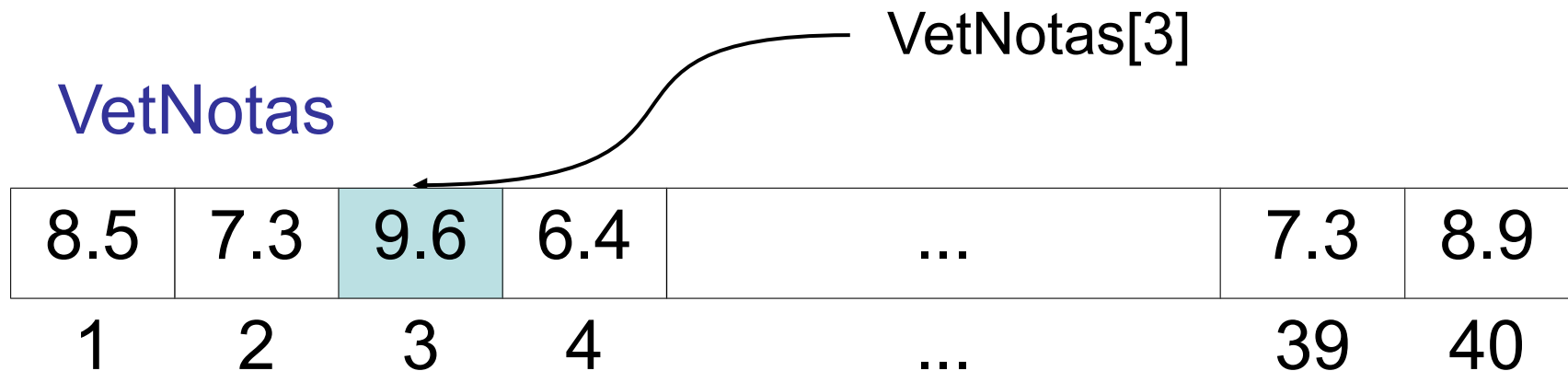
VetNotas

8.5	7.3	9.6	6.4	...	7.3	8.9
6	7	8	9	...	34	35

Manipulando Vetores

tipo Notas = **vetor** [1..40] **de** reais;

Notas: VetNotas;



Escreva VetNotas[3]

9.6

Manipulando Vetores

- Exemplos de como manipular o conteúdo de um vetor

VetNotas[2] \leftarrow 8.1 *{armazena 8.1 no elemento VetNotas[2]}*

Leia VetNotas[2] *{entra com um valor no elemento VetNotas[2], isto é, armazena ali o valor lido }*

inteiro: i

i \leftarrow 2

Escreva VetNotas[i] *{escreve o valor armazenado em VetNotas[2]}*

Escreva VetNotas[i+1] *{escreve o valor armazenado em VetNotas[3]}*

VetNotas[2] \leftarrow 8.1

inteiro: i

i \leftarrow 5

VetNotas[i] \leftarrow 10.0

Escreva VetNotas[i+1]

VetNotas [1]

VetNotas [2]

VetNotas [3]

VetNotas [4]

VetNotas [5]

VetNotas [6]

VetNotas [7]

VetNotas [8]

VetNotas [9]

VetNotas [10]

8.5

7.3

9.6

6.4

7.5

9.2

8.5

6.4

8.3

8.1

9.2

VetNotas[2] \leftarrow 8.1



VetNotas [1]

VetNotas [2]

VetNotas [3]

VetNotas [4]

VetNotas [5]

VetNotas [6]

VetNotas [7]

VetNotas [8]

VetNotas [9]

VetNotas [10]

8.5

8.1

9.6

6.4

10.0

9.2

8.5

6.4

8.3

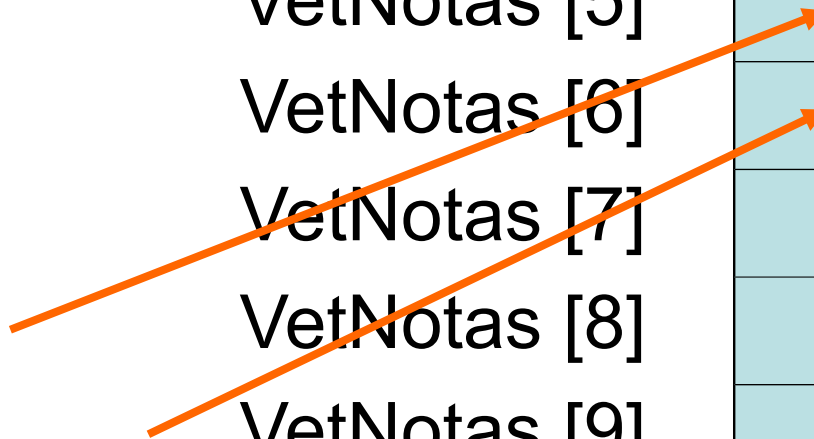
8.1

inteiro: i

i \leftarrow 5

VetNotas[i] \leftarrow 10.0

escreva VetNotas[i+1]



9.2

Manipulando Vetores

- Não é possível ler ou escrever o conteúdo de um vetor inteiro. Só posição por posição.

Leia VetNotas
Escreva VetNotas



Exemplo

```
vetor [1..10] de reais: Notas  
inteiro: i  
para i de 1 até 10 faça  
    Leia Notas[i]  
fim-para
```

- A cada rodada do laço, o usuário digita uma nota e esta é armazenada no vetor Notas;
- o contador i é incrementado a cada rodada.

Notas [1]

8.5

Notas [2]

7.3

Notas [3]

9.6

Notas [4]

6.4

Notas [5]

7.5

Notas [6]

9.2

Notas [7]

8.5

Notas [8]

6.4

Notas [9]

8.3

Notas [10]

8.1

Exemplo

```
vetor [1..10] de reais: Notas  
inteiro: i  
para i de 1 até 10 faça  
    Escreva Notas[i]  
fim-para
```

8.5
7.3
9.6
6.4
7.5
9.2
8.5
6.4
8.3
8.1

Notas [1]
Notas [2]
Notas [3]
Notas [4]
Notas [5]
Notas [6]
Notas [7]
Notas [8]
Notas [9]
Notas [10]

8.5	i=1
7.3	i=2
9.6	i=3
6.4	i=4
7.5	i=5
9.2	i=6
8.5	i=7
6.4	i=8
8.3	i=9
8.1	i=10

Exemplo

```
vetor [1..10] de reais: Notas  
inteiro: i  
para i de 1 até 10 faça  
    Escreva "Nota ",i, "=" Notas[i]  
fim-para
```

```
Nota 1 = 8.5  
Nota 2 = 7.3  
Nota 3 = 9.6  
Nota 4 = 6.4  
Nota 5 = 7.5  
Nota 6 = 9.2  
Nota 7 = 8.5  
Nota 8 = 6.4  
Nota 9 = 8.3  
Nota 10 = 8.1
```

Notas [1]	8.5	i=1
Notas [2]	7.3	i=2
Notas [3]	9.6	i=3
Notas [4]	6.4	i=4
Notas [5]	7.5	i=5
Notas [6]	9.2	i=6
Notas [7]	8.5	i=7
Notas [8]	6.4	i=8
Notas [9]	8.3	i=9
Notas [10]	8.1	i=10

Exemplo: Cálculo da média de uma turma com 10 alunos.

{ declaração de variáveis }

real: Nota, *{nota de cada aluno}*

MediaTurma, *{média da turma}*

acum *{acumulador das médias}*

inteiro: i *{variável de controle}*

inicio *{cálculo da média de uma turma com dez alunos}*

acum \leftarrow 0 *{inicialização do acumulador}*

para i de 1 até 10 **faça**

Leia Nota *{ entrada de dados }*

 acum \leftarrow acum + Nota

fim-para

MediaTurma \leftarrow acum/10 *{calcula a média da turma}*

Escreva “A média da turma é” , MediaTurma

fim

E se quisermos saber quantos alunos tiveram média acima da média da turma?

...

para i de 1 até 10 faça

Leia Nota

$\text{acum} \leftarrow \text{acum} + \text{Nota}$

fim-para

$\text{MediaTurma} \leftarrow \text{acum}/10$

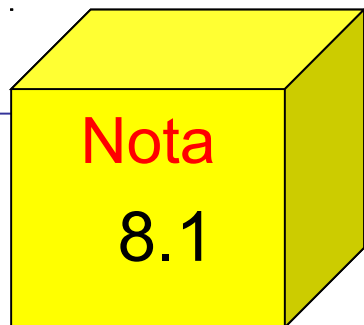
...

if (Nota > MediaTurma) **então**

$\text{cont} \leftarrow \text{cont} + 1$

...

- Observe que cada vez que o laço é executado, um novo valor de Nota é lido;
- Então, a cada vez, o valor lido anteriormente é perdido.
- Como vamos poder comparar cada Nota com a média, se ao final do laço só temos a última Nota lida?



E se quisermos saber quantos alunos tiveram média acima da média da turma? (Exemplo **sem vetor**)

real: N1, N2, N3, N4, N5, N6, N7, N8, N9, N10,

MediaTurma *{média da turma}*

inteiro: NotasAcima *{contador de notas acima da média}*

inicio

NotasAcima $\leftarrow 0$

Leia N1, N2, N3, N4, N5, N6, N7, N8, N9, N10

MediaTurma $\leftarrow (N1+N2+N3+N4+N5+N6+N7+N8+N9+N10)/10$

se (N1 > MediaTurma) **então**

NotasAcima \leftarrow NotasAcima + 1

fim-se

se (N2 > MediaTurma) **então**

NotasAcima \leftarrow NotasAcima + 1

fim-se

Continua...

E se quisermos saber quantos alunos tiveram média acima da média da turma? (Exemplo **sem vetor**)

se (N3 > MediaTurma) **então**

NotasAcima \leftarrow NotasAcima + 1

fim-se

se (N4 > MediaTurma) **então**

NotasAcima \leftarrow NotasAcima + 1

fim-se

se (N5 > MediaTurma) **então**

NotasAcima \leftarrow NotasAcima + 1

fim-se

se (N6 > MediaTurma) **então**

NotasAcima \leftarrow NotasAcima + 1

fim-se

Continua...

E se quisermos saber quantos alunos tiveram média acima da média da turma? (Exemplo **sem vetor**)

se (N7 > MediaTurma) **então**

NotasAcima \leftarrow NotasAcima + 1

fim-se

se (N8 > MediaTurma) **então**

NotasAcima \leftarrow NotasAcima + 1

fim-se

se (N9 > MediaTurma) **então**

NotasAcima \leftarrow NotasAcima + 1

fim-se

se (N10 > MediaTurma) **então**

NotasAcima \leftarrow NotasAcima + 1

fim-se

Escreva NotasAcima

fim.

E se quisermos saber quantos alunos tiveram média acima da média da turma? (Exemplo com vetor)

vetor [1..10] de reais: Notas *{vetor para guardar as notas}*

real: MediaTurma, *{média da turma}*

Soma

inteiro: NotasAcima, i

inicio

Soma \leftarrow 0

NotasAcima \leftarrow 0

para i de 1 até 10 **faça**

Leia Notas[i]

fim-para

para i de 1 até 10 **faça**

 Soma \leftarrow Soma + Notas[i]

fim-para

Notas [1]	8.5
Notas [2]	7.3
Notas [3]	9.6
Notas [4]	6.4
Notas [5]	7.5
Notas [6]	9.2
Notas [7]	8.5
Notas [8]	6.4
Notas [9]	8.3
Notas [10]	8.1

Continua...

E se quisermos saber quantos alunos tiveram média acima da média da turma? (Exemplo com vetor)

MediaTurma \leftarrow Soma/10; *{calcula a média da turma}*

Escreva “A média da turma é ”, MediaTurma

para i de 1 até 10 faça

se (Notas[i] > MediaTurma) então

NotasAcima \leftarrow NotasAcima + 1

fim-se

fim-para

Escreva NotasAcima, “notas acima da média”

fim

Notas [1]	8.5	>media?
Notas [2]	7.3	>media?
Notas [3]	9.6	>media?
Notas [4]	6.4	>media?
Notas [5]	7.5	>media?
Notas [6]	9.2	>media?
Notas [7]	8.5	>media?
Notas [8]	6.4	>media?
Notas [9]	8.3	>media?
Notas [10]	8.1	>media?

Exercício 2

- Escreva um algoritmo que armazene 50 inteiros em um vetor, 50 inteiros em outro vetor e some e imprima o resultado da soma entre cada posição dos dois vetores. Além de imprimir, os resultados devem ser colocados em um terceiro vetor.

A	
1	34
2	87
3	21
4	7
...	
48	25
49	14
50	58

B	
1	20
2	32
3	16
4	53
...	
48	28
49	42
50	71

SOMA	
1	54
2	119
3	37
4	60
...	
48	53
49	56
50	129

Exercício 3

- Escreva um algoritmo que preencha um vetor de 100 elementos inteiros, colocando 1 nas posições correspondentes a um número par (índice par) e 0 na posições de índice impar.
- Imprima o vetor.

A	
1	0
2	1
3	0
4	1
...	
98	1
99	0
100	1

Matrizes

- Assim como os vetores, matrizes são agregados de dados homogêneos
- Enquanto vetores são unidimensionais
 - Possuem apenas um índice;
- Matrizes são estruturas multidimensionais
 - Podem ter dois ou mais índices;
 - As mais comuns são as bidimensionais.

	1	2	3	4
1	9.1	8.8	3.2	1.7
2	4.4	2.1	2.1	8.9
3	1.0	0.0	1.1	7.3

Matrizes - Declaração

matriz [*li1..ls1, li2..ls2, liN..lsN*] **de** <tipo>: identificador

ou

tipo **identifTipo** = **matriz** [*li1..ls1, li2..ls2, liN..lsN*] **de** <tipo>

identifTipo : identifiVariavel

Matrizes

matriz [1..4,1..5] **de** inteiros: **MSala**

	1	2	3	4	5	índices das colunas
1						
2						
3						
4						

Índices
das linhas

MSala[2,3]

Manipulando Matrizes

- Exemplos de como manipular o conteúdo de uma matriz

MSala[2,3] \leftarrow 9

Leia MSala[1,4]

Leia MSala[3,4]

inteiro: i, j

i \leftarrow 2

j \leftarrow 3

Escreva MSala[i, j]

Escreva MSala[i+1, j+1]

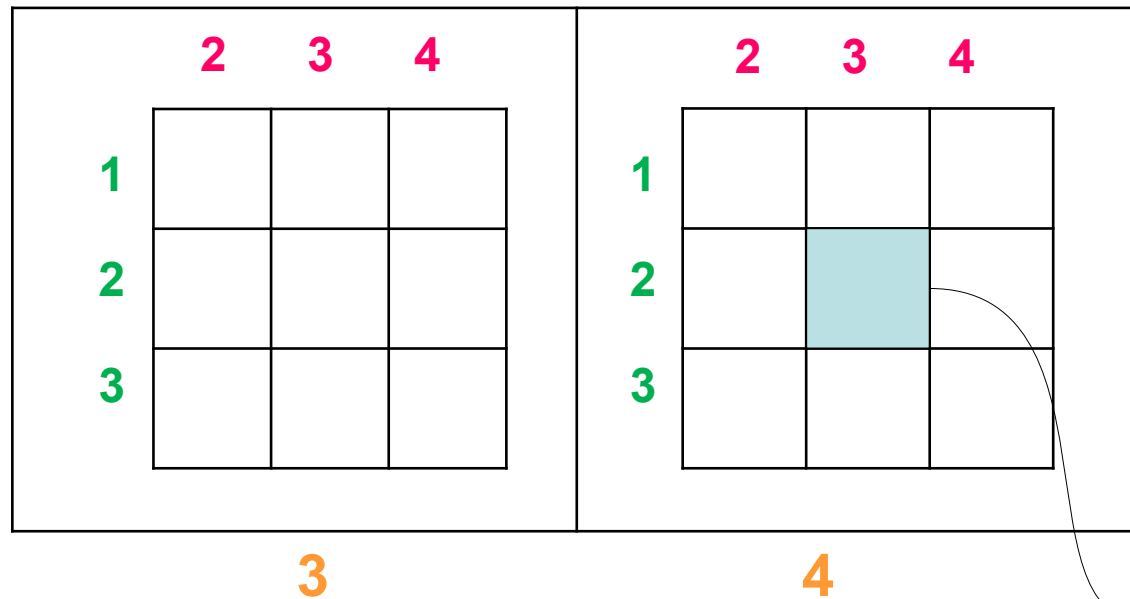
	1	2	3	4	5
1				17	
2			9		
3				12	
4					

9

12

Matrizes

tipo Matriz2 = **matriz** [1..3, 2..4, 3..4] **de** inteiros;
Matriz2: MAT;



MAT [2, 3, 4]

Exercício 3

- Escreva um algoritmo que
 - armazene 20 inteiros em uma matriz 4x5;
 - armazene mais 20 inteiros em outra matriz 4x5;
 - some as duas matrizes armazenando o resultado em uma terceira matriz.

MATRIZ A

	1	2	3	4	5
1	6	8	9	3	5
2	4	3	2	7	3
3	5	5	0	1	6
4	2	7	4	3	0

MATRIZ B

	1	2	3	4	5
1	1	6	3	8	2
2	3	2	6	5	0
3	1	4	2	0	7
4	7	3	1	4	1

MATRIZ SOMA

	1	2	3	4	5
1	7	14	12	11	7
2	7	5	8	12	3
3	6	9	2	1	13
4	9	10	5	7	1

Vetores e Matrizes

MATA37: Introdução à Lógica de Programação

Prof.: Rafael A. Melo (melo@dcc.ufba.br)

Departamento de Ciência da Computação

Instituto de Matemática

Universidade Federal da Bahia

