



## USP - ICMC - SSC SSC 0301 - 2o. Semestre 2013

### Disciplina de Introdução à Computação para Engenharia Ambiental

Prof. Dr. Fernando Santos Osório / Prof. Artur Lovato Cunha

LRM - Laboratório de Robótica Móvel do ICMC / CROB-SC

Email: [fosorio@icmc.usp.br](mailto:fosorio@icmc.usp.br) ou [fosorio@gmail.com](mailto:fosorio@gmail.com)

Email: [arturlc@icmc.usp.br](mailto:arturlc@icmc.usp.br)

Página Pessoal: <http://www.icmc.usp.br/~fosorio/>

Material on-line:

Wiki ICMC - <http://wiki.icmc.usp.br/index.php>

Wiki SSC0301 - [http://wiki.icmc.usp.br/index.php/SSC-301-2013\(fosorio\)](http://wiki.icmc.usp.br/index.php/SSC-301-2013(fosorio))

## **Agenda:**

- **Vetores com mais de uma dimensão:  
Matrizes**
- **Vetores especiais:  
Strings e Funções de manipulação de strings**
- **Exercícios**

**Informações Complementares e Atualizadas:  
Consulte REGULARMENTE  
O material disponível na COTEIA**

## Tipos de Dados Compostos: Estruturas HOMOGÊNEAS

### VETORES

#### Exemplos Típicos:

**VETOR DE CARACTERES = *STRING***

**char Texto[10];**

Texto[0] até Texto[9] <= São 10 posições de 1 char, lado a lado

---

**VETOR DE INTEIROS = *TABELA***

**int Tabela[10];**

Tabela[0] até Tabela[9] <= São 10 posições de 1 int, lado a lado

---

**VETOR DE DOUBLES = *DADOS***

**double Dados[10];**

Dados[0] até Dados[9] <= São 10 posições de 1 double, lado a lado

## Tipos de Dados Compostos: Estruturas HOMOGÊNEAS

### VETORES

#### Vetores Numéricos:

```
#define MAX_NOTAS 10  
double N[MAX_NOTAS]; /* Notas de até “Max_Notas” alunos */
```

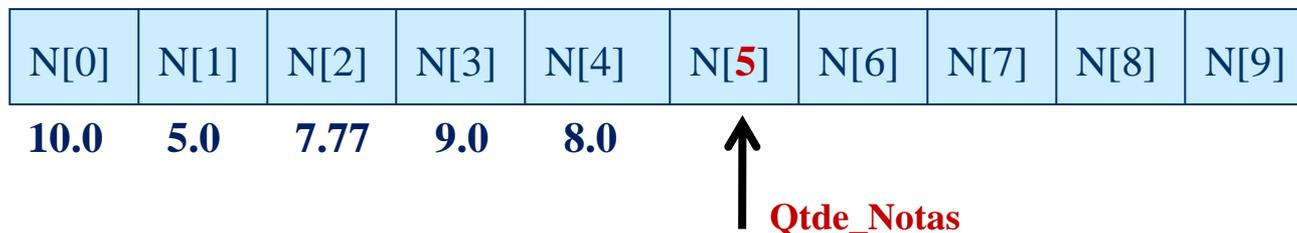
```
N[0] = 10.0;
```

```
N[1] = 5.0;
```

```
Qtde_Notas = 3; /* Última = Qtde_Notas - 1 */
```

```
N[Qtde_Notas++] = 9.0; /* Nota índice 3 */
```

```
N[Qtde_Notas++] = 8.0; /* Nota índice 4 */
```



## Tipos de Dados Compostos: Estruturas HOMOGENEAS

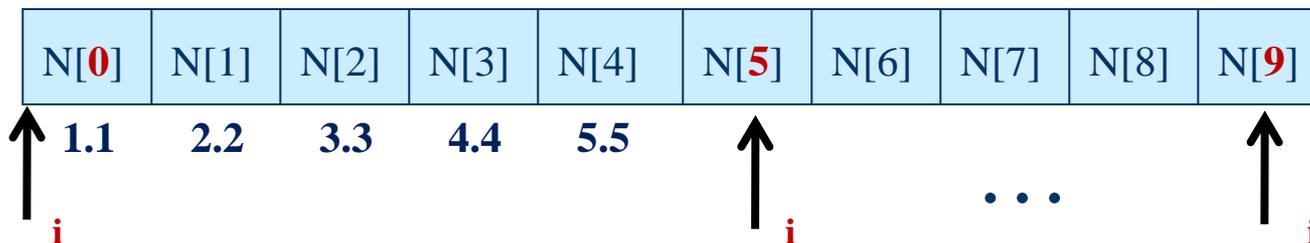
### VETORES

#### Vetores Numéricos:

```
#define MAX_NOTAS 10
double N[MAX_NOTAS]; /* Notas de até “Max_Notas “ alunos */

int i;

for (i = 0; i < MAX_NOTAS; i++)
{
    printf (“Entre com a nota %d: “, i);
    scanf (“%lf”, &N[i]);
}
```



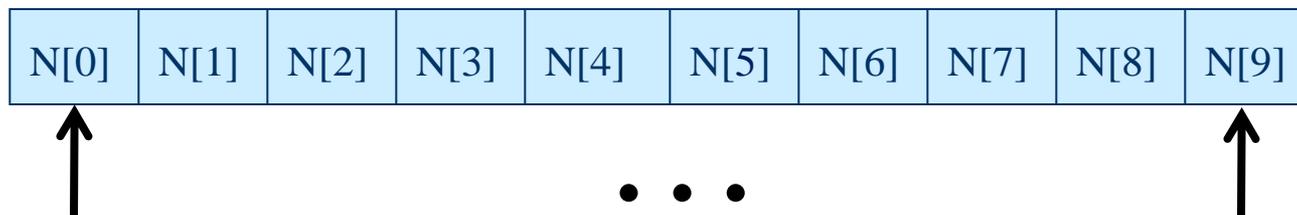
## Tipos de Dados Compostos: Estruturas HOMOGÊNEAS

### VETORES

Vetores Numéricos: **#define MAX 10**

**double N[MAX]; /\* Notas de até 10 alunos \*/**

```
int Qtde_Notas=0, Repete=1;
while ((Repete) && (Qtde_Notas < 10))
{
    scanf("%lf",& N[Qtde_Notas]);
    if (N[Qtde_Notas] < 0.0)
        Repete = 0;
    else Qtde_Notas++;
}
```



**Qtde\_Notas**

## Tipos de Dados Compostos: Estruturas HOMOGÊNEAS

### VETORES

#### Vetores de Caracteres:

**char T[10];** /\* String de até 9 caracteres + Null \*/

**T[0] = 'O';**

**T[1] = 'L';**

**T[2] = 'A';**

**T[3] = '\0';** /\* Character NULL ou \0 \*/



'O' 'L' 'A' \0  
Null



**Final da String**

**strcpy(T,"OLA");**

**/\* Inclui o Null automaticamente \*/**

**/\* ao final da String \*/**

## Tipos de Dados Compostos: Estruturas HOMOGÊNEAS

### VETORES

#### Vetores de Caracteres:

```
char Texto[10]; /* String de até 9 caracteres + Null */
```

```
int i=0;
```

```
strcpy(Texto,"Hello");
```

```
while (Texto[i] != '\0')
```

```
{
```

```
    printf("%c", Texto[i]);
```

```
    i++;
```

```
}
```

=> While equivalente a:  
=> **printf ("%s", Texto);**

T[0]	T[1]	T[2]	T[3]	T[4]	T[5]	T[6]	T[7]	T[8]	T[9]
------	------	------	------	------	------	------	------	------	------

'H'

'E'

'L'

'L'

'O'

'\0'

Null

**Final da String**

## Tipos de Dados Compostos: Estruturas HOMOGÊNEAS

### VETORES

#### Vetores de Caracteres:

**char Texto[10]; /\* String de até 9 caracteres + Null \*/**

Strings **precisam** ser manipuladas através de rotinas especiais:

```
#include <string.h>
```

```
strcpy, strlen, strcmp... sprintf, sscanf, ...
```

T[0]	T[1]	T[2]	T[3]	T[4]	T[5]	T[6]	T[7]	T[8]	T[9]
'H'	'E'	'L'	'L'	'O'	'\0'				
					Null				

**Final da String**

## VETORES

N[0]	N[1]	N[2]	N[3]	N[4]	N[5]	N[6]	N[7]	N[8]	N[9]
------	------	------	------	------	------	------	------	------	------

V[0]	V[1]	V[2]	V[3]	V[4]	V[5]	V[6]	V[7]	V[8]	V[9]
------	------	------	------	------	------	------	------	------	------

## MATRIZES (ARRAYS)

M[0][0]	M[0][1]	M[0][2]	M[0][3]	M[0][4]	M[0][5]	M[0][6]	M[0][7]	M[0][8]	M[0][9]
M[1][0]	M[1][1]	M[1][2]	M[1][3]	M[1][4]	M[1][5]	M[1][6]	M[1][7]	M[1][8]	M[1][9]
M[2][0]	M[2][1]	M[2][2]	M[2][3]	M[2][4]	M[2][5]	M[2][6]	M[2][7]	M[2][8]	M[2][9]

## VETORES

MP[0]	MP[1]	MP[2]	MP[3]	MP[4]	MP[5]	MP[6]	MP[7]	MP[8]	MP[9]
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

double MP[10];

PH[0]	PH[1]	PH[2]	PH[3]	PH[4]	PH[5]	PH[6]	PH[7]	PH[8]	PH[9]
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

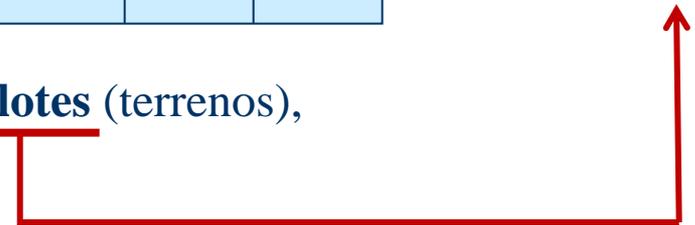
double PH[10];

NT[0]	NT[1]	NT[2]	NT[3]	NT[4]	NT[5]	NT[6]	NT[7]	NT[8]	NT[9]
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

double NT[10];

Este estudo envolve uma análise do solo em **10 lotes** (terrenos),  
com medições indicando:

- (i) presença de metais pesados;
- (ii) acidez (PH) do solo;
- (iii) quantidade de nitratos.



Entrada de Dados:

Ler os valores destas 3 medidas para cada um dos 10 lotes e armazenar os dados em um vetor.  
Ler os dados de MP, PH e NT para o primeiro lote, depois de MP, PH e NT para o segundo lote,  
e assim por diante até o décimo lote.

Os valores de MP, PH e NT são valores com casas após a vírgula.

## Tipos de Dados Compostos: Estruturas HOMOGÊNEAS

### MATRIZES - Vetores com mais de uma dimensão

Vetores numéricos bi-dimensionais:

**3 x 10**

<b>int Matriz [3][10];</b>	M[0][0]	M[0][1]	M[0][2]	M[0][3]	M[0][4]	M[0][5]	M[0][6]	M[0][7]	M[0][8]	M[0][9]
<b>Matriz[0][0] = 1;</b>	M[1][0]	M[1][1]	M[1][2]	M[1][3]	M[1][4]	M[1][5]	M[1][6]	M[1][7]	M[1][8]	M[1][9]
<b>...</b>										
<b>Matriz [2][9] = 30;</b>	M[2][0]	M[2][1]	M[2][2]	M[2][3]	M[2][4]	M[2][5]	M[2][6]	M[2][7]	M[2][8]	M[2][9]

Tabela 3 x 10

## VETORES

MP[0]	MP[1]	MP[2]	MP[3]	MP[4]	MP[5]	MP[6]	MP[7]	MP[8]	MP[9]
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

double MP[10];

PH[0]	PH[1]	PH[2]	PH[3]	PH[4]	PH[5]	PH[6]	PH[7]	PH[8]	PH[9]
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

double PH[10];

NT[0]	NT[1]	NT[2]	NT[3]	NT[4]	NT[5]	NT[6]	NT[7]	NT[8]	NT[9]
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

double NT[10];

Este estudo envolve uma análise do solo em **10 lotes** (terrenos),  
com medições indicando:

- (i) presença de metais pesados;
- (ii) acidez (PH) do solo;
- (iii) quantidade de nitratos.

## MATRIZ 3 x 10

M[0][0]	M[0][1]	M[0][2]	M[0][3]	M[0][4]	M[0][5]	M[0][6]	M[0][7]	M[0][8]	M[0][9]
M[1][0]	M[1][1]	M[1][2]	M[1][3]	M[1][4]	M[1][5]	M[1][6]	M[1][7]	M[1][8]	M[1][9]
M[2][0]	M[2][1]	M[2][2]	M[2][3]	M[2][4]	M[2][5]	M[2][6]	M[2][7]	M[2][8]	M[2][9]

Linha do MP: [0][lote]

Linha do PH: [1][lote]

Linha do NT: [2][lote]

## VETORES

Este estudo envolve uma análise do solo em **10 lotes** (terrenos), com medições indicando:

- (i) presença de metais pesados;
- (ii) acidez (PH) do solo;
- (iii) quantidade de nitratos.

```
double MP[10];
double PH[10];
double NT[10];
```

### MATRIZ 10 x 3

M[0][0]	M[0][1]	M[0][2]
M[1][0]	M[1][1]	M[1][2]
M[2][0]	M[2][1]	M[2][2]
M[3][0]	M[3][1]	M[3][2]
M[4][0]	M[4][1]	M[4][2]
M[5][0]	M[5][1]	M[5][2]
M[6][0]	M[6][1]	M[6][2]
M[7][0]	M[7][1]	M[7][2]
M[8][0]	M[8][1]	M[8][2]
M[9][0]	M[9][1]	M[9][2]

L  
O  
T  
E  
S

MP

PH

NT

```
double Lotes[Max_Lotes][Max_Medidas]
```

Linha do MP: [lote] [0]

Linha do PH: [lote] [1]

Linha do NT: [lote] [2]

Linha do MP: [0][lote]

Linha do PH: [1][lote]

Linha do NT: [2][lote]

### MATRIZ 3 x 10

M[0][0]	M[0][1]	M[0][2]	M[0][3]	M[0][4]	M[0][5]	M[0][6]	M[0][7]	M[0][8]	M[0][9]
M[1][0]	M[1][1]	M[1][2]	M[1][3]	M[1][4]	M[1][5]	M[1][6]	M[1][7]	M[1][8]	M[1][9]
M[2][0]	M[2][1]	M[2][2]	M[2][3]	M[2][4]	M[2][5]	M[2][6]	M[2][7]	M[2][8]	M[2][9]

## Tipos de Dados Compostos: Estruturas HOMOGÊNEAS

### MATRIZES - Vetores com mais de uma dimensão

Vetores numéricos bi-dimensionais:

3 x 10

```
int Matriz [3][10];
```

```
Matriz[0][0] = 1;
```

```
...
```

```
Matriz [2][9] = 30;
```

M[0][0]	M[0][1]	M[0][2]	M[0][3]	M[0][4]	M[0][5]	M[0][6]	M[0][7]	M[0][8]	M[0][9]
M[1][0]	M[1][1]	M[1][2]	M[1][3]	M[1][4]	M[1][5]	M[1][6]	M[1][7]	M[1][8]	M[1][9]
M[2][0]	M[2][1]	M[2][2]	M[2][3]	M[2][4]	M[2][5]	M[2][6]	M[2][7]	M[2][8]	M[2][9]

- Inicialização de vetores:

```
int num [5] = { 1, 2, 3, 4, 5 };
```

```
char vogais[5] = { 'a', 'e', 'i', 'o', 'u' };
```

```
double matriz [3][2] = { { 0.0 }, { 0.1 },  
                          { 1.0 }, { 1.1 },  
                          { 2.0 }, { 2.1 } };
```

Matriz do Jogo da Velha

```
char Tabuleiro [3][3];
```

'O'	'X'	'X'
' '	'O'	' '
' '	' '	'O'

## Tipos de Dados Compostos: Estruturas HOMOGÊNEAS

### MATRIZES - Vetores com mais de uma dimensão

Vetores de strings bi-dimensionais:

Armazenar em uma tabela o nome de 10 alunos (nome + '\0' com até 40 chars)

Tabela de Alunos:

```
char Tab_Alunos [10][40];
```

```
strcpy (Tab_Alunos[0], “Fulano da Silva”);
```

```
strcpy (Tab_Alunos[1],”Beltrano de Oliveira”);
```

```
scanf (“%s”, Tab_Alunos[2] );
```

```
scanf (“%s”, Tab_Alunos[3] );
```

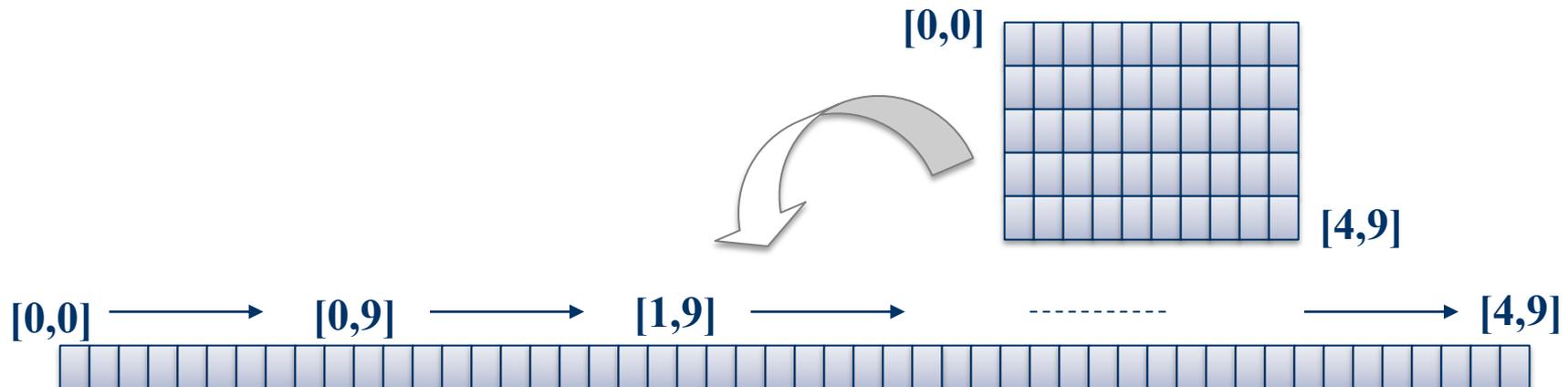
```
printf (“Nome do aluno 0: %s \n”, Tab_Alunos[0] );
```

```
printf (“Nome do aluno 1: %s \n”, Tab_Alunos[1] );
```

```
printf (“Primeira letra do nome do aluno 3: %c \n”, Tab_Alunos[3][0] );
```

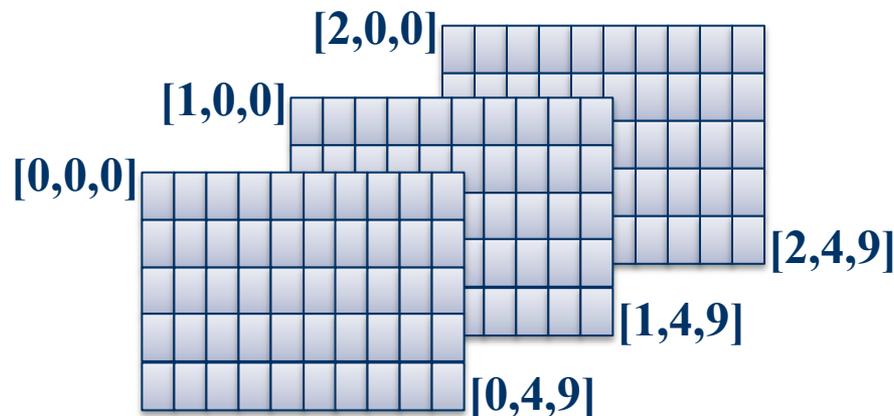
## Arrays Multidimensionais

- *Arrays* podem ter diversas dimensões, cada uma identificada por um par de colchetes na declaração.
- Ex: `int matriz[5][10];`
  - declara uma matriz de 5 linhas e 10 colunas:
  - na memória, entretanto, cada um dos inteiros são armazenados linearmente:



## Arrays Multidimensionais

- *Arrays* podem ter diversas dimensões, cada uma identificada por um par de colchetes na declaração.
- Ex: `int matriz[5][10][3];`
  - declara uma matriz de 5 linhas e 10 colunas:
    - na memória, entretanto, cada um dos inteiros são armazenados linearmente:



`int matriz[3][5][10];`

## Programas com Vetores e Matrizes

### Exercícios - Lista Nro. 05 (vetores) e 06 (matrizes)

- 1) Faça um programa para ler 25 números DIFERENTES a serem armazenados em uma tabela 5x5. O programa deve evitar que o usuário digite 2 números iguais para serem armazenados na tabela. Exibir na tela o conteúdo final da tabela que foi digitada pelo usuário.
- 2) Faça um programa para ler os dados climáticos da temperatura de cada dia para cada um dos meses do ano, juntamente com a umidade para cada dia. Portanto, deve ser criada uma tabela de 12 meses com 31 dias para cada mês, sendo que teremos 2 tabelas: uma para temperatura e outra para a umidade. A temperatura é dada em graus e a umidade em um percentual relativo. Exibir o dia e mês com a maior temperatura e o dia e mês com a menor umidade.  
**>> Entregar este exercício na próxima aula (Exerc. 2) <<**
- 3) Considerando o exercício anterior, defina as variáveis para armazenar os dados de temperatura e umidade para cada dia, mês e ano (de 1900 a 2014), referentes as principais cidades do país (100 cidades).



## INFORMAÇÕES SOBRE A DISCIPLINA

**USP - Universidade de São Paulo - São Carlos, SP**  
**ICMC - Instituto de Ciências Matemáticas e de Computação**  
**SSC - Departamento de Sistemas de Computação**

**Prof. Fernando Santos OSÓRIO**

**Web institucional: <http://www.icmc.usp.br/>**

**Página pessoal: <http://www.icmc.usp.br/~fosorio/>**

**Página do Grupo de Pesquisa: <http://www.lrm.icmc.usp.br/>**

**E-mail: fosorio [at] icmc. usp. br ou fosorio [at] gmail. com**

**Disciplina de Introdução a Computação – Eng. Ambiental**

**WIKI - [http://wiki.icmc.usp.br/index.php/SSC-301-2013\(fosorio\)](http://wiki.icmc.usp.br/index.php/SSC-301-2013(fosorio))**

- > Programa, Material de Aulas, Critérios de Avaliação,**
- > Trabalhos Práticos, Datas das Provas, Notas**