

ESTRUTURAS COMPOSTAS

VETOR e MATRIZ

SSC300- Linguagens de Programação e Aplicações

Slides baseados no material da
Profa. Simone Senger Souza

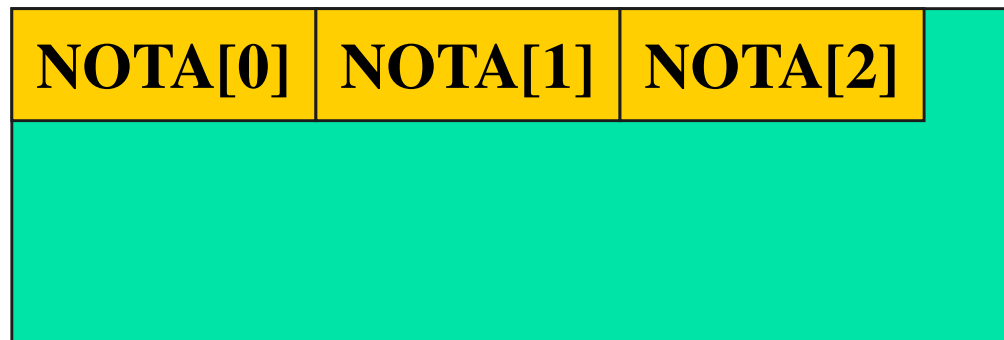
ICMC/USP - São Carlos

ESTRUTURAS COMPOSTAS

- Pode-se organizar os dados dos tipos simples em tipos mais complexos formando-se as ESTRUTURAS COMPOSTAS
- Exemplo:
 - variáveis compostas unidimensionais (VETOR)

VARIÁVEL INDEXADA

- Cada **variável indexada** é associada à uma posição de memória, como acontece com variáveis simples.
- Exemplo:



VETOR

- Um **VETOR** é um conjunto ordenado que contém um número fixo de elementos
- Todos os elementos do vetor devem ser do mesmo tipo

Declaração (C) de Variável Indexada Unidimensional

- Deve ser especificado o número máximo de elementos do conjunto
- Deve ser especificado o tipo dos elementos do conjunto
- Exemplo:

```
float x[100];  
int vet[50];  
char nome[100];
```

Declaração (C) de Variável Indexada Unidimensional

- Pode ser definida uma **constante** e esta ser utilizada no dimensionamento.
- Exemplo:

```
//Início do programa C
const int MAX = 100; OU
#define MAX 100
//declaração:
float X[MAX];
int vet[MAX];
char nome[MAX];
```

ESTRUTURAS COMPOSTAS

MATRIZ

Introdução

- Como armazenar um **nome** em um programa?
 - Uso de um vetor de caracteres (string)
- E um conjunto de nomes?
 - Suponha que seja necessário armazenar todos os nomes da turma e depois ordena-los segundo algum critério
 - Necessário uma **estrutura composta bidimensional**

ESTRUTURAS COMPOSTAS

- Pode-se organizar os dados dos tipos simples em tipos mais complexos formando-se as ESTRUTURAS COMPOSTAS
- Exemplo:
 - variáveis compostas unidimensionais (VETOR)
 - variáveis compostas bidimensionais (MATRIZ)

ESTRUTURAS COMPOSTAS

- Pode-se organizar os dados dos tipos simples em tipos mais complexos formando-se as ESTRUTURAS COMPOSTAS
- Exemplo:
 - variáveis compostas unidimensionais (VETOR)
 - variáveis compostas bidimensionais (MATRIZ)

MATRIZ

$$A = \begin{vmatrix} 10 & 8 & 5 & 1 \\ 5 & 7 & 7 & 7 \\ 8 & 0 & 0 & 10 \end{vmatrix}$$

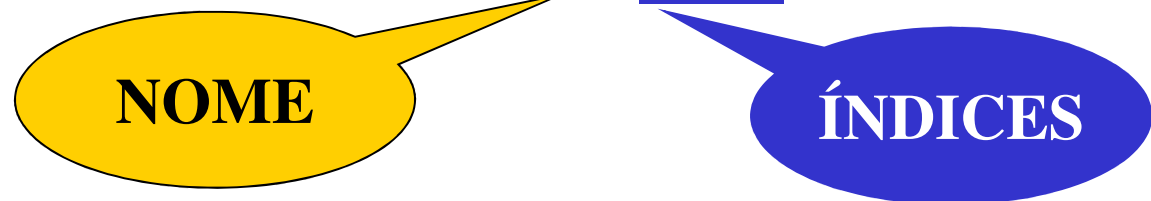
- Para fazer referência ou selecionar um determinado elemento da matriz usa-se dois índices: um representa a linha e outro a coluna da matriz

MATRIZ

A[0,0]	10	8	5	1
A =	5	7	7	7
	8	0	0	10

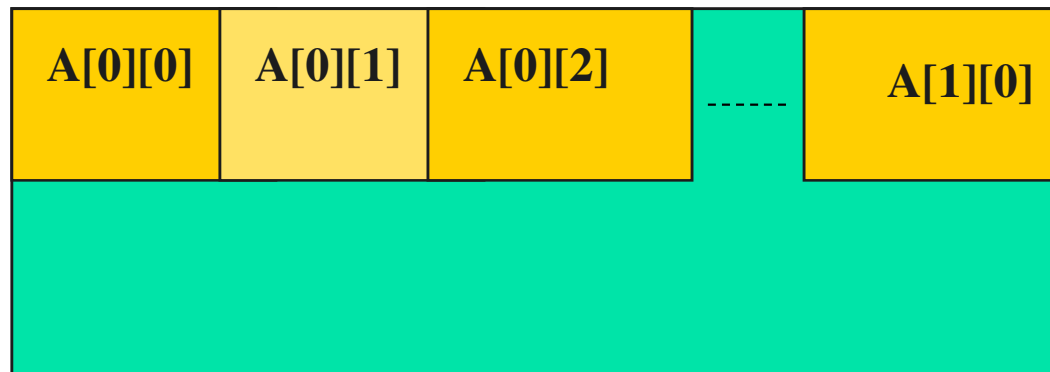
Diagram illustrating a matrix structure. The matrix is labeled **MATRIZ**. The element **A[1][2]** is highlighted in yellow, with callouts indicating its row (**linha**) and column (**coluna**). The matrix is represented as **A =** followed by a 3x4 grid of values: 10, 8, 5, 1; 5, 7, 7, 7; 8, 0, 0, 10. The value 10 in the first row is circled in yellow, and the value 7 in the second row, third column is circled in yellow.

- Genericamente, um elemento da matriz é representado por: **A[i][j]**



MATRIZ

- Cada **variável indexada bidimensional** é associada à uma posição de memória, como acontece com variáveis simples.
- Exemplo:



Declaração de Variável Indexada Bidimensional

- Na linguagem C:

```
int A[20][10];
```

- Com valor constante:

```
const int MAX = 10  
int A[MAX][MAX];
```

Declaração de Variável Indexada Bidimensional - Exemplos

- Exemplo:
 - Declaração de uma matriz S com no máximo 80 nomes e no máximo 30 letras cada nome.

```
char S[80][30];
```