

# Estruturas Compostas Matriz

---

Profa. Dra. Elisa Yumi Nakagawa  
1. Semestre de 2017

# 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 ordená-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 de Nomes

nomes =

A	N	A	\0		
J	O	A	0	\0	
B	R	U	N	0	\0

DUAS DIMENSÕES: QUANTIDADE DE  
NOMES x OS NOMES(QTDE DE LETRAS)

# 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

<b>A[0,0]</b>	10	8	5	1
A =	5	7	7	7
	8	0	0	10

linha

coluna

**A[1][2]**

- Genericamente, um elemento da matriz é representado por:

**NOME**

**A[i][j]**

**ÍNDICES**

# MATRIZ

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





# Exemplos de Programas com Variáveis Compostas Bidimensionais

# MATRIZ - Exemplo 1

---

- Ler uma tabela de 10 linhas e 3 colunas armazenando-as em uma matriz de inteiros.

## Algoritmo MATRIZ01

**declarações**

início

```
para i de 0 até 9 faça
  para j de 0 até 2 faça
    ler (TAB[i][j]);
  fim-para;
fim-para;
```

fim

Teste de mesa**MATRIZ** - Exemplo 1

<b>i</b>	<b>j</b>	
0	0	ler <b>TAB</b> [0][0]
	1	ler <b>TAB</b> [0][1]
	2	ler <b>TAB</b> [0][2]
	3	
1	0	ler <b>TAB</b> [1][0]
	1	ler <b>TAB</b> [1][1]
	2	ler <b>TAB</b> [1][2]
	3	
		etc.

# MATRIZ - Exemplo 2

---

- Ler uma tabela de  $M$  linhas e  $N$  colunas armazenando-as em uma matriz  $A$ . Exibir a matriz  $A$ .

## Algoritmo MATRIZ02

## MATRIZ - Exemplo 2

declarações

inicio

ler (M);

ler(N)

Número de  
linhasNúmero de  
colunas

```
para i de 0 até M-1
  faça para j de 0 até N-1
    faça ler (A[i][j]);
  fim-para
fim-para
```

Leitura da Tabela

```
para i de 0 até M-1 faça
  para j de 0 até N-1 faça
    escrever (A[i][j]);
  fim-para;
fim-para;
```

Escrita da Tabela

Fim.

# MATRIZ - Exemplo 3

---

- Ler uma tabela de M linhas e N colunas armazenando-as em uma matriz de inteiros **A** e calcular a soma de todos os elementos da matriz **A**.

## programa MATRIZ03

## declarações

inicio

ler (M);

ler(N);

Número de  
linhasNúmero de  
colunas

## MATRIZ - Exemplo 3

```
para I de 0 até M-1
  para J de 0 até N-1
    ler (a[I][J]);
  fim-para
fim-para
```

Leitura da Tabela

s = 0;

```
para I de 0 até M-1
  para J de 0 até N-1
    s = s + a[I][J];
  fim-para;
```

Cálculo da soma  
dos elementos

fim-para;

Escrita da soma  
dos elementos

escrever (s);

fim

## programa MATRIZ03

**declarações**

início

ler (**M**);ler(**N**);para **I** de 0 até **M**para **J** de 0 até **N**ler (a[**I**][**J**]);

fim-para

fim-para

s = 0;

para **I** de 0 até **M**para **J** de 0 até **N**s = s + a[**I**][**J**];

fim-para;

fim-para;

escrever (s);

fim

Como são  
declaradas  
as variáveis  
indexadas  
bidimensionais?

# Declaração de Variável Indexada Bidimensional

- Deve ser especificado o tipo dos elementos do conjunto
  - Nome da Variável
  - Exemplo:
- de
- Número máximo de elementos nas linhas
- Número máximo de elementos nas colunas

```
inteiro a[20][10];
```

# 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];
```

# programa MATRIZ03

## declarações

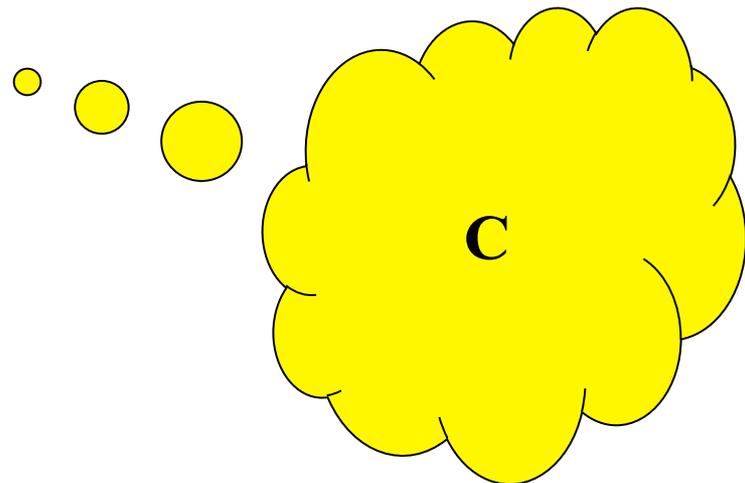
inicio

```

ler (M);
ler(N);
para I de 0 até M
    faça para J de 0 até N
        faça ler (a[I][J]);
    fim-para
fim-para
s = 0;
para I de 0 até M
    faça para J de 0 até N
        faça s = s + a[I][J];
    fim-para;
fim-para;
escrever (s);
fim
  
```

## MATRIZ - Exemplo 3

- Ler uma tabela de M linhas e N colunas armazenando-as em uma matriz **A**
- Calcular a soma de todos os elementos da matriz **A**



## Exemplo 03 em C

```
int main()
{
    int A[100][100], i, j, n, m, soma=0;

    printf("entre com o numero de linhas da matriz:");
    scanf("%d", &m);

    printf("entre com o numero de colunas da matriz:");
    scanf("%d", &n);

    printf("entre com os elementos da matriz:\n");
    for (i=0; i< m; i++)
        for (j = 0; j < n; j++)
        {
            printf("A[%d][%d] = ", i, j);
            scanf("%d", &A[i][j]);
        }

    for (i=0; i< m; i++)
        for (j = 0; j < n; j++)
            soma = soma + A[i][j];

    printf("a soma dos elementos da matriz eh %d \n", soma);
    system("PAUSE");
    return 0;
}
```

# Exercício

---

- Escreva a matriz resultante:

$A[0][0] = 15;$

para i de 0 até 4 faça

para j de 1 até 4 faça

$A[i][j] = A[i][j-1] / 2 ;$

fim-para;

$A[i+1][0] = A[i][0] + 2;$

fim-para;

# Exercício - solução

$A[0][0] = 15;$

para  $i$  de 0 até 4 faça

para  $j$  de 1 até 4 faça

$A[i][j] = A[i][j-1] / 2 ;$

fim-para;

$A[i+1][0] = A[i][0] + 2;$

fim-para;

$A =$

15	7	3	1
17	8	4	2
19	9	4	2
21	10	5	2

# Exercícios

---

1. Faça um algoritmo que lê uma matriz de inteiros  $6 \times 3$ , calcula e mostra o maior e o menor elemento da matriz e suas posições (linhas e colunas)
2. Faça um algoritmo que lê uma matriz de inteiros de  $n$  linhas e  $m$  colunas. O algoritmo deve somar cada uma das linhas da matriz e guardar o resultado da soma de cada linha em um vetor. A seguir, mostrar o vetor resultante.
3. Considerando o algoritmo 2, mude para que no vetor seja armazenada a soma de cada coluna da matriz. A seguir, multiplique cada elemento da matriz pela soma da coluna e mostre a matriz resultante.