



5950257 - Programação de Computadores

Aula 08b - Matrizes e Arrays Multi-dimensionais

Prof. Dr. Denis M. L. Martins

DCM | FFCLRP | USP

Objetivos de Aprendizagem

- Compreender o conceito de matrizes e arrays multidimensionais em C
- Declarar e inicializar arrays multidimensionais
- Acessar elementos de matrizes utilizando índices
- Utilizar estruturas de repetição com matrizes
- Desenvolver algoritmos utilizando matrizes

Para os exercícios desta aula, você pode utilizar [IDEs](#) online como:

https://www.onlinegdb.com/online_c_compiler

<https://www.programiz.com/c-programming/online-compiler/>

Matriz em C

Uma matriz é:

- Um array multidimensional
- Uma estrutura de dados organizada em:
 - linhas
 - colunas
- Utilizada para representar tabelas, imagens, mapas e grades

Sintaxe Geral

```
tipo nome[linhas][colunas];
```

Matrizes em C

C não possui "matrizes" como objeto nativo distinto. Ele trata matrizes como arrays de arrays.

`int mat[3][4]` é um array com 3 elementos.

Em C, o tamanho da segunda dimensão (colunas) é obrigatório na declaração. O compilador precisa saber o tamanho do bloco para calcular endereços corretamente.

	Col1	Col2	Col3	Col4	...
Row1	<code>Arr[0][0]</code>	<code>Arr[0][1]</code>	<code>Arr[0][2]</code>	<code>Arr[0][3]</code>	
Row2	<code>Arr[1][0]</code>	<code>Arr[1][1]</code>	<code>Arr[1][2]</code>	<code>Arr[1][3]</code>	
Row3	<code>Arr[2][0]</code>	<code>Arr[2][1]</code>	<code>Arr[2][2]</code>	<code>Arr[2][3]</code>	
Row4	<code>Arr[3][0]</code>	<code>Arr[3][1]</code>	<code>Arr[3][2]</code>	<code>Arr[3][3]</code>	
	⋮				

Criando e Inicialização de Matrizes

```
int matrix[2][3]; // Contém valores de "garbage" (indeterminados)
// SEMPRE INICIALIZAR EXPLICITAMENTE EM C!
```

```
int matrix[2][3] = {
    {1, 2, 3},
    {4, 5, 6}
};
```

Exemplo

Armazenar notas de:

- 3 alunos
- 2 provas

```
float notas[3][2];
```

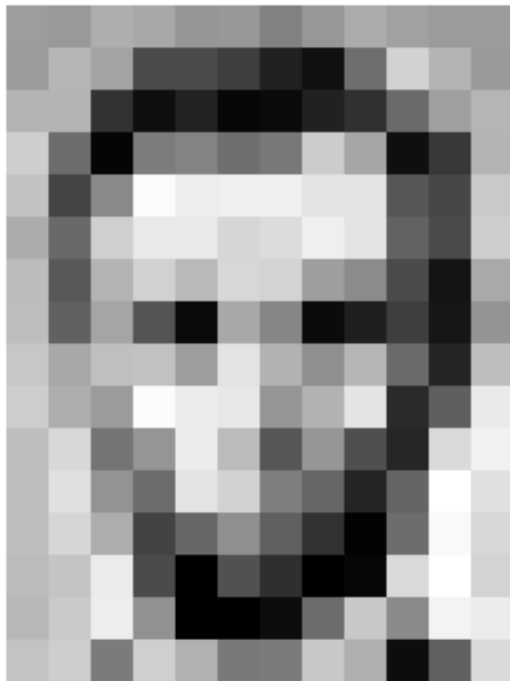
```
Aluno 0 -> Prova 1 e Prova 2  
Aluno 1 -> Prova 1 e Prova 2  
Aluno 2 -> Prova 1 e Prova 2
```

Onde matrizes são utilizadas?

- Jogos
- Inteligência Artificial e Machine Learning
- Computação Gráfica
- Processamento de Imagens
- Sistemas de Navegação
- Análise Experimental

Matrizes em Processamento de Imagens

Matrix $M_{m \times n} \in \mathbb{Z}_0^{L-1}$, onde $L = 2^k$ e k é o número de bits por pixel.

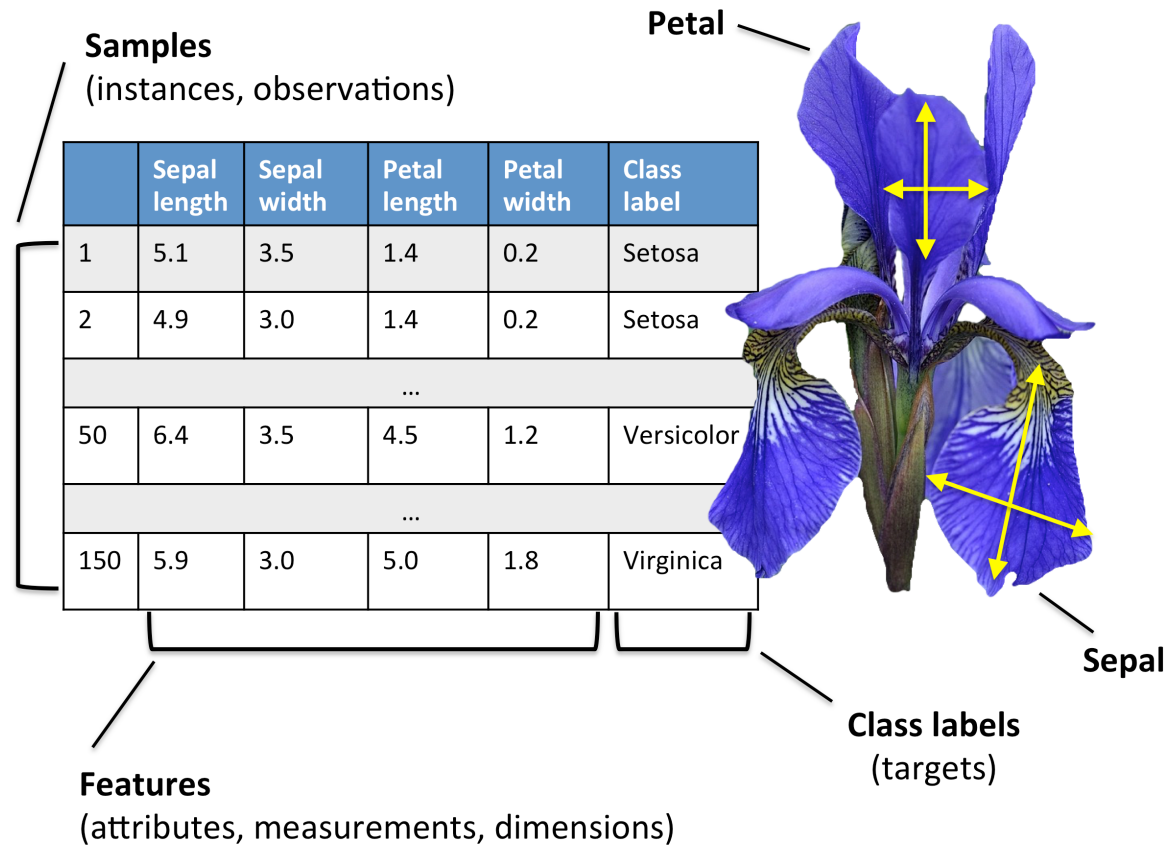


157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	106	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
206	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
196	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	106	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
206	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
196	206	123	207	177	121	123	200	175	13	96	218

Na imagem: Representação de pixels de uma imagem. Fonte: [OpenFrameworks - Image Processing](#) ↗

Matrizes em Machine Learning



Fonte da imagem: [Sebastian Raschka](#) 

Navegando na Matriz

```
int matrix[2][3] = {  
    {1, 2, 3},  
    {4, 5, 6}  
};
```

Acesso Direto:

```
printf("%d", matrix[0][1]); // Imprime o elemento da linha 0, coluna 1
```

Iteração (Loop Aninhado):

```
for (int i = 0; i < 2; i++) { // Percorre Linhas  
    for (int j = 0; j < 3; j++) { // Percorre Colunas  
        printf("%d ", matrix[i][j]);  
    }  
    printf("\n"); // Quebra linha após cada linha da matriz  
}
```

Exemplo: Soma de Elementos

```
#include <stdio.h>

int main() {
    int matriz[2][2] = {
        {1,2},
        {3,4}
    };
    int soma = 0;
    for(int i = 0; i < 2; i++) {
        for(int j = 0; j < 2; j++) {
            soma += matriz[i][j];
        }
    }
    printf("Soma = %d", soma);
    return 0;
}
```

Exemplo: Lendo Matriz do Teclado

```
#include <stdio.h>

#define TAMANHO_LINHAS 3
#define TAMANHO_COLUNAS 3

int main() {
    int matriz[TAMANHO_LINHAS][TAMANHO_COLUNAS];
    printf("Por favor, insira os 9 valores da matriz:\n\n");
    for (int i = 0; i < TAMANHO_LINHAS; i++) {
        // A cada nova linha, pedimos ao usuário para preencher os valores da coluna.
        printf("Insira os %d valores da Linha %d:\n", TAMANHO_COLUNAS, i + 1);
        for (int j = 0; j < TAMANHO_COLUNAS; j++) {
            printf("  Elemento [%d][%d]: ", i, j);
            scanf("%d", &matriz[i][j]);
        }
    }
    printf("\nLeitura da matriz concluída com sucesso!\n");

    return 0;
}
```

Exercício A

Crie um programa que leia uma matriz 4x4 imprima:

- maior valor
- menor valor
- soma da diagonal principal

Exercício B

Crie um programa que inicializa duas matrizes 5x3 e pede que o usuário selecione uma das opções abaixo:

- **Opção 1:** Calcular a soma das matrizes
- **Opção 2:** Calcular a subtração das matrizes

O programa deve exibir o resultado da operação selecionada.

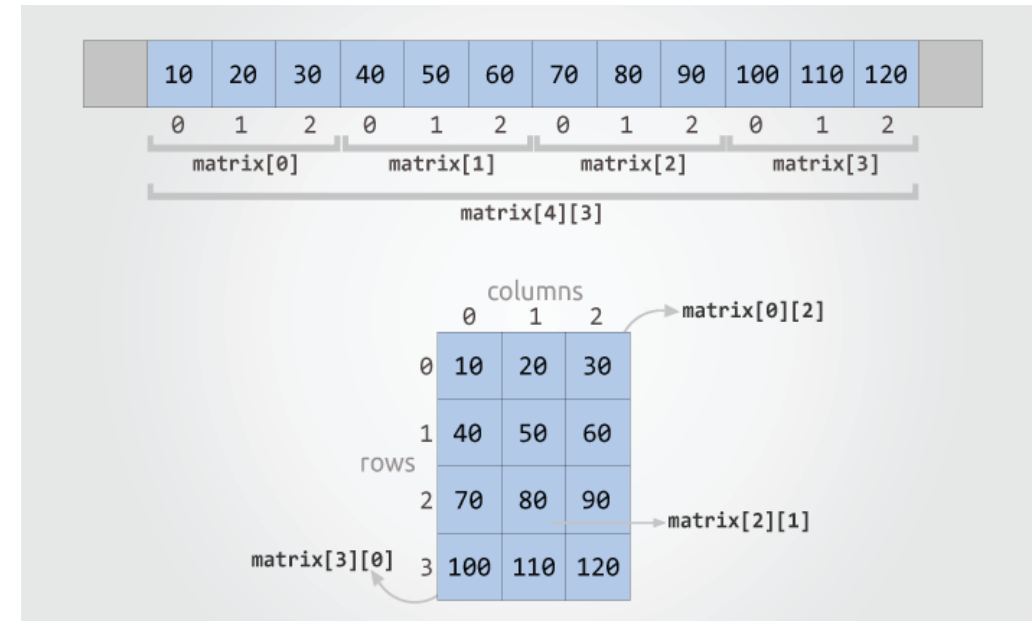
Matriz em Memória

Row-Major Order: C armazena matrizes na ordem das linhas.

- Todas as colunas da linha 0 vêm primeiro.
- Depois, todas as colunas da linha 1.

Endereço Base: `&mat[0][0]`

```
| [0][0] | [0][1] | [0][2] | <- Linha 0 (4 bytes * 3)
+-----+
| [1][0] | [1][1] | [1][2] | <- Linha 1
```



Conclusão

Matrizes são fundamentais porque:

- permitem representar dados bidimensionais
- são amplamente utilizadas em computação científica
- servem como base para:
 - imagens
 - simulações
 - jogos
 - inteligência artificial

Próximo Tópico: Strings em C

Dúvidas e Discussão

Exercício e Questões

Questão 1

Qual das alternativas representa corretamente uma matriz 3x4 em C?

A)

```
int matriz[3,4];
```

B)

```
int matriz[3][4];
```

C)

```
int matriz(3)(4);
```

Questão 2

Qual é o erro abaixo?

```
int matriz[2][2];  
matriz[2][0] = 5;
```

Exercício 1

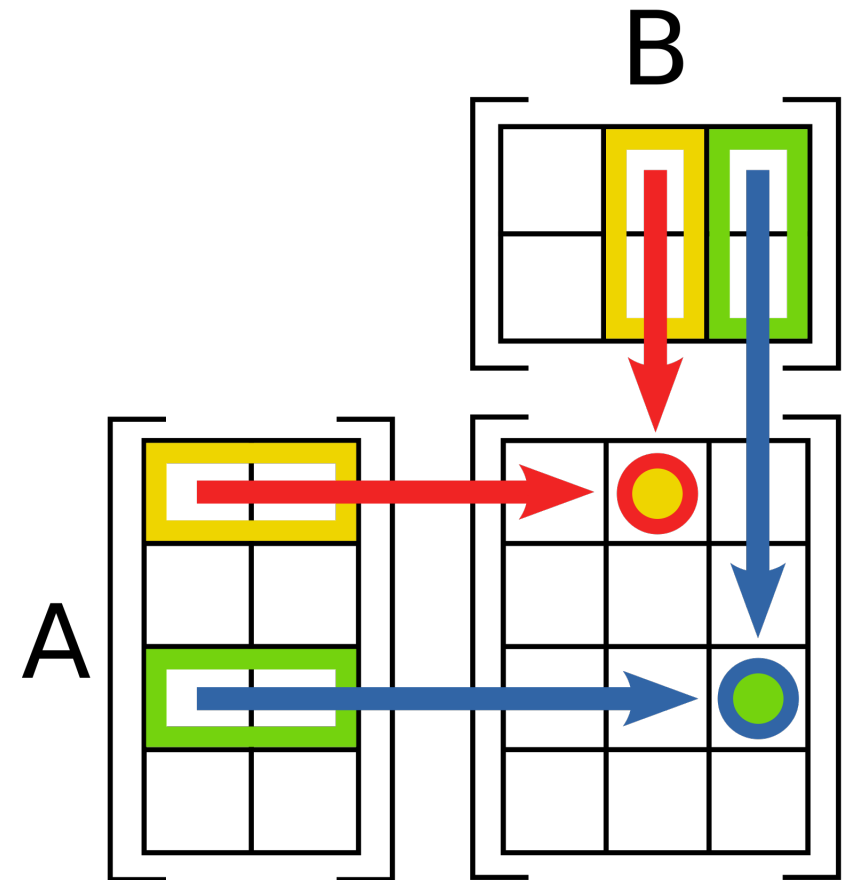
Crie um programa que inicialize uma matriz 4 x 3 e imprima a matriz transposta.

Transposta: Dado uma matriz M de $C \times R$, a matriz transposta N é definida como $N[i][j] = M[j][i]$.

Exercício 2

Crie um programa que calcule e imprime a multiplicação de duas matrizes. Se as matrizes forem incompatíveis, o programa deve exibir uma mensagem de erro.

Fonte da Imagem: [Wikipedia](#) .



Exercício 3

Crie um programa que lê uma matriz quadrada do usuário e verifica se ela é simétrica, ou seja $M[i][j] == M[j][i]$ para todo i, j .

O programa deve imprimir a mensagem "SIM", se a matriz for simétrica. Senão, deve imprimir "NÃO".