



LÓGICA DE PROGRAMAÇÃO

PROF^a. M.Sc. JULIANA H Q BENACCHIO

- Também chamados de **Matriz**
- Os vetores multidimensionais com duas dimensões (bidimensionais) costumam ser utilizados para representar tabelas de valores que consistem nas informações dispostas em **linhas e colunas**
- Para identificar um elemento de uma tabela particular, devemos especificar dois índices

- Por convenção, o primeiro índice identifica a **linha** do elemento e o segundo, a **coluna**
- Cada elemento na matriz é identificado por uma expressão de acesso da forma:

$$m[\mathbf{lin}][\mathbf{col}]$$

- Onde m é o nome da matriz, e \mathbf{lin} e \mathbf{col} são os índices que identificam unicamente cada elemento na matriz pelo número da linha e da coluna

- Matriz com 3 linhas e 4 colunas

	Coluna 0	Coluna 1	Coluna 2	Coluna 3
Linha 0	<code>m[0][0]</code>	<code>m[0][1]</code>	<code>m[0][2]</code>	<code>m[0][3]</code>
Linha 1	<code>m[1][0]</code>	<code>m[1][1]</code>	<code>m[1][2]</code>	<code>m[1][3]</code>
Linha 2	<code>m[2][0]</code>	<code>m[2][1]</code>	<code>m[2][2]</code>	<code>m[2][3]</code>

`m` nome da matriz
`2` índice da linha
`1` índice da coluna

- Observe que todos os nomes dos elementos na linha 0 têm um primeiro índice de 0 e todos os nomes dos elementos na coluna 2 têm um segundo índice de 2

	Coluna 0	Coluna 1	Coluna 2	Coluna 3
Linha 0	<code>m[0][0]</code>	<code>m[0][1]</code>	<code>m[0][2]</code>	<code>m[0][3]</code>
Linha 1	<code>m[1][0]</code>	<code>m[1][1]</code>	<code>m[1][2]</code>	<code>m[1][3]</code>
Linha 2	<code>m[2][0]</code>	<code>m[2][1]</code>	<code>m[2][2]</code>	<code>m[2][3]</code>

- Em geral, uma matriz com m linhas e n colunas é chamada de matriz $m \times n$
- Quando a matriz tem o mesmo número de linhas e colunas (n linhas e n colunas), é chamada de matriz quadrada $n \times n$
- A quantidade de elementos da matriz será sempre o total da multiplicação de linhas por colunas

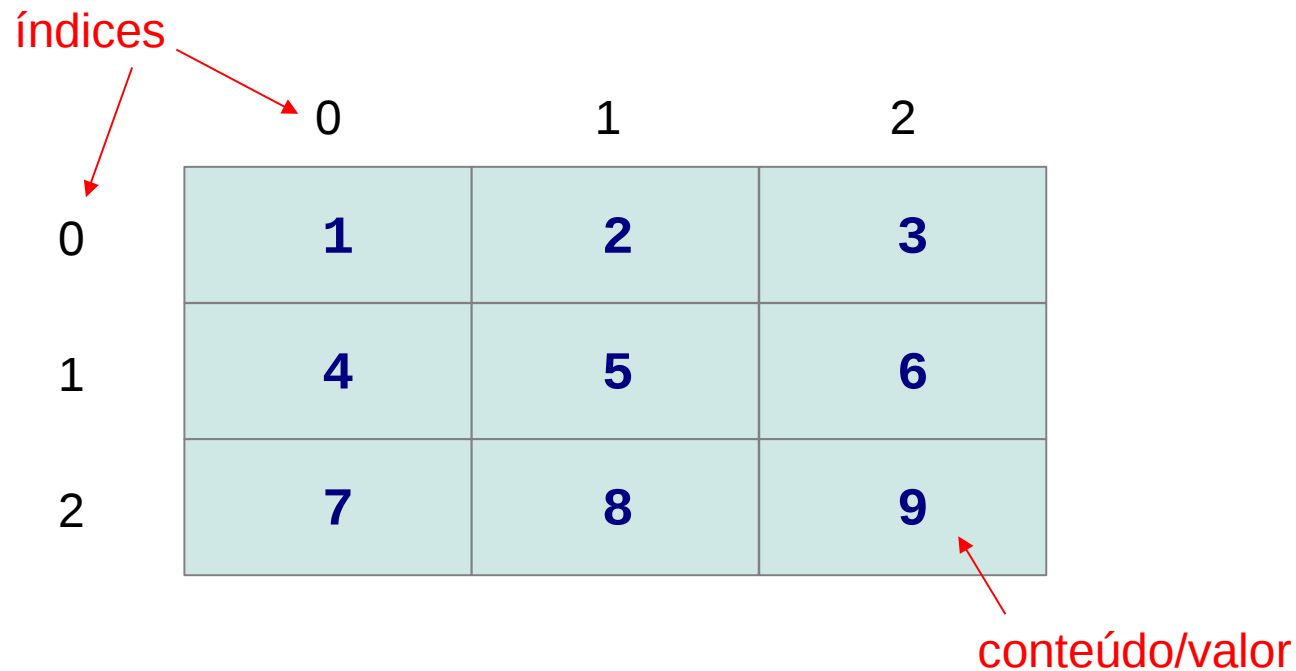
Matriz

- Uma matriz com 3 linhas e 3 colunas (isto é, uma matriz 3x3), tem 9 elementos

índices

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

conteúdo/valor



- Representação da matriz na memória

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

Diagram illustrating a 3x3 matrix with elements 1 through 9. Red arrows show the row-major traversal order: starting at (0,0) to (0,1) to (0,2), then jumping to (1,0) to (1,1) to (1,2), and finally jumping to (2,0) to (2,1) to (2,2).

`m[0][0]`

1

`m[0][1]`

2

`m[0][2]`

3

`m[1][0]`

4

`m[1][1]`

5

`m[1][2]`

6

`m[2][0]`

7

`m[2][1]`

8

`m[2][2]`

9

- Para declarar uma matriz com 3 linhas e 3 colunas:

```
int m[3][3];
```

- Outros exemplos de declaração:

```
float matriz[2][5];
```

```
int valores[6][4];
```

- Para percorrer a matriz inteira é necessário a utilização de 2 laços **for** aninhados
- Um laço com um contador para controlar as linhas (por exemplo, **lin**)
- E um laço com um contador para controlar as colunas (por exemplo, **col**)

```
for(lin=0; lin<3; lin++) {  
    for(col=0; col<3; col++) {  
        m[lin][col] = 0;  
    }  
}
```

Matriz

```
#include <stdio.h>
```

```
int main(){  
    int m[3][3];  
    int i,j;  
  
    for(i=0; i<3; i++)  
    {  
        for(j=0; j<3; j++)  
        {  
            printf("Digite o valor de m[%d][%d]: ", i, j);  
            scanf("%d", &m[i][j]);  
        }  
    }  
    return 0;  
}
```

Matriz

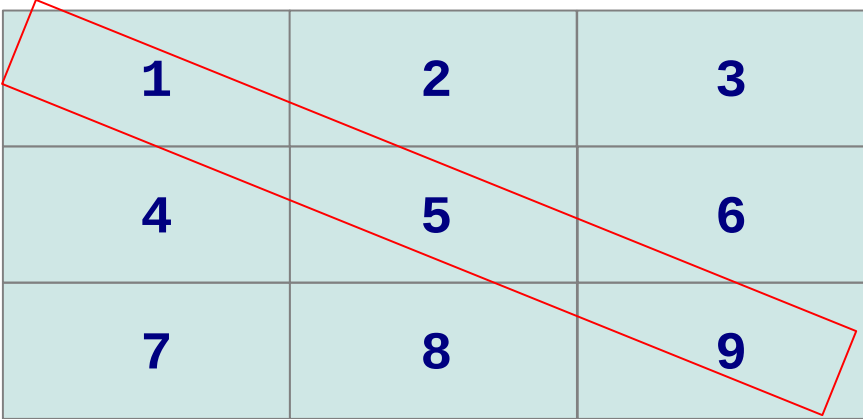
```
#include <stdio.h>

int main(){
    int m[3][3];
    int i,j;
    for(i=0; i<3; i++)
        for(j=0; j<3; j++)
        {
            printf("Digite o valor de m[%d][%d]: ", i, j);
            scanf("%d", &m[i][j]);
        }
    printf("Elementos da matriz\n");
    for(i=0; i<3; i++)
        for(j=0; j<3; j++)
            printf("m[%d][%d]= %d\n", i, j, m[i][j]);
    return 0;
}
```

- Exercício: Seja A uma matriz 3×3 , faça um programa para ler todos os elementos da matriz e mostrar os elementos da diagonal principal

diagonal principal

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



Matriz

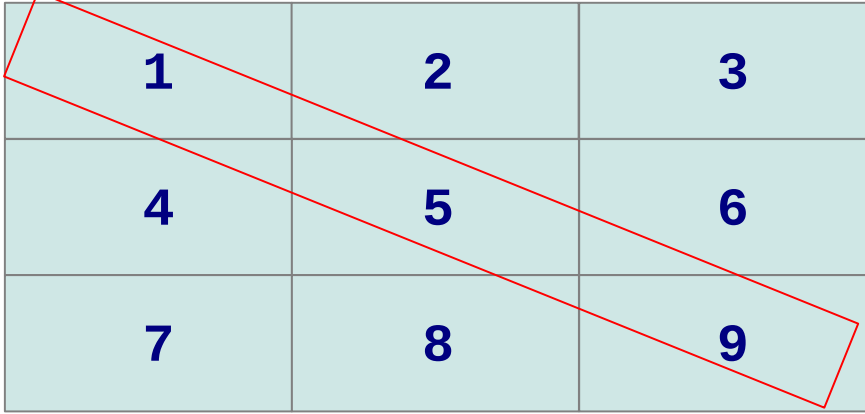
```
#include <stdio.h>

int main(){
    int m[3][3];
    int i,j;
    for(i=0; i<3; i++)
        for(j=0; j<3; j++)
        {
            printf("Digite o valor de m[%d][%d]: ", i, j);
            scanf("%d", &m[i][j]);
        }
    printf("Elementos da diagonal principal\n");
    for(i=0; i<3; i++)
        for(j=0; j<3; j++)
            if(i==j)
                printf("%d\n", m[i][j]);
    return 0;
}
```

- Exercício: Altere o exercício anterior para colocar os elementos da diagonal principal em um vetor S e determinar a soma desses elementos

diagonal principal

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



Soma = 15

Matriz

```
int main(){
    int m[3][3], s[3];
    int i,j, k=0, soma=0;
    for(i=0; i<3; i++)
        for(j=0; j<3; j++){
            printf("Digite o valor de m[%d][%d]: ", i, j);
            scanf("%d", &m[i][j]);
            if(i==j) {
                s[k]=m[i][j];
                k++;
            }
        }
    printf("Elementos da diagonal principal\n");
    for(k=0; k<3; k++){
        printf("%d\n", s[k]);
        soma = soma + s[k];
    }
    printf("Soma = %d\n", soma);
    return 0;
}
```