

PROGRAMAÇÃO DE COMPUTADORES (PC)

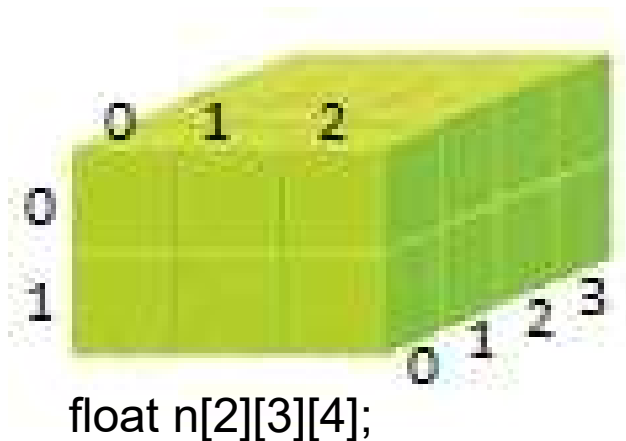
ENGENHARIA DA COMPUTAÇÃO

Matrizes

Profa. Dra. Lúcia F. A. Guimarães

- **Matriz - Definição**

- Uma matriz é
 - Uma coleção de 2 ou mais dimensões de **variáveis de mesmo tipo.**
 - Acessíveis com um **único nome.**
 - Armazenados **contiguamente** na memória.
- Podemos dizer que **matrizes são vetores de vetores.**



`int m[5][4];`

	0	1	2	3
0				
1			5	
2				
3				
4				

A green arrow points from the text `m[1][2]=5;` to the value 5 in the cell at row 1, column 2.

- **Declaração**

- O tipo de dados (primitivos):
 - int
 - float
 - double, char
- O nome da matriz:
 - Segundo as regras de definição de identificadores na linguagem C.
- O número de elementos da matriz:
 - Indicado por números inteiros entre colchetes **(declaração estática)**.

- **Declaração**

- Exemplo:

```
float A[5][5];           // matriz bidimensional  
int matriz3D[4][7][2];  // matriz tridimensional
```

Vamos trabalhar com Matrizes Bidimensionais!!!!

- **Matriz – Bidimensional:**
 - Linha X Coluna - Por exemplo:

6 colunas

4 linhas

`int a [4][6];`

	coluna	coluna	coluna	coluna	coluna	coluna
	0	1	2	3	4	5
linha 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]	a[0][4]	a[0][5]
linha 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]	a[1][4]	a[1][5]
linha 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]	a[2][4]	a[2][5]
linha 3	a[3][0]	a[3][1]	a[3][2]	a[3][3]	a[3][4]	a[3][5]

- **Matriz – Bidimensional:**
 - **Linha X Coluna - Por exemplo:**

int a [4][6];

	coluna	coluna	coluna	coluna	coluna	coluna
	0	1	2	3	4	5
linha 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]	a[0][4]	a[0][5]
linha 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]	a[1][4]	a[1][5]
linha 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]	a[2][4]	a[2][5]
linha 3	a[3][0]	a[3][1]	a[3][2]	a[3][3]	a[3][4]	a[3][5]

*
a[1][3]=0

Coluna 3

Linha 1

O acesso aos elementos é dado primeiramente pela linha, seguido pela coluna.

- **Observações:**

- Os índices de qualquer matriz sempre começam em 0.
- **C não verifica a validade dos índices em matrizes.**
- Os elementos de uma matriz ocupam posições consecutivas de memória.

- **Inicialização:**

- **Na Declaração**

`int mat[2][3] = { 2, 9, 4, 7, 6, 1};`



Tipo da
Matriz



`float a[5][5];`

- **Inicialização:**

- **Na Declaração**

```
int mat[2][3] = { 2, 9, 4, 7, 6, 1};
```



Nome da
Matriz



```
float a[5][5];
```

- **Inicialização:**

- **Na Declaração**

`int mat[2][3] = { 2, 9, 4, 7, 6, 1};`



Número de
LINHAS da
Matriz

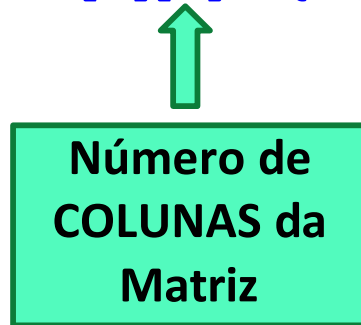


`float a[5][5];`

- **Inicialização:**

- **Na Declaração**

`int mat[2][3] = { 2, 9, 4, 7, 6, 1};`



`float a[5][5];`

- **Inicialização:**

- **Na Declaração**

`int mat[2][3] = {`

Primeira Linha	Segunda Linha
2, 9, 4,	7, 6, 1;

`}`



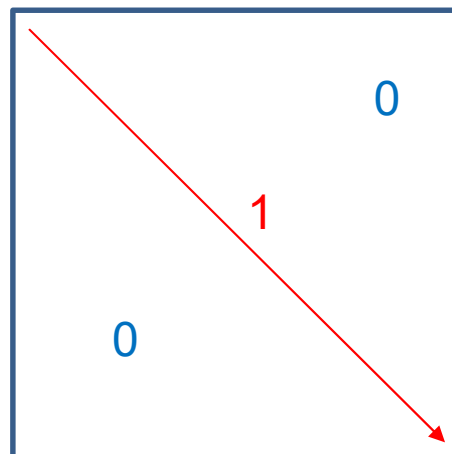
Inicialização
da Matriz

OPCIONAL

`float a[5][5];`

- **Manipulação**

- Será apresentado um exemplo de declaração de matriz de tamanho (NxN), seu preenchimento de acordo com a matriz identidade e sua impressão na tela.
- Obs.: Matrix identidade: matriz em que todos os seus elementos são nulos, exceto na diagonal principal, em que seus valores são unitários.



Matriz identidade

• Manipulação

- Assim como em vetores, a manipulação de matrizes é feita elemento a elemento.

```
#define N 5
int main()
{
    int ident[N][N];
    int i,j;
    for (i=0; i<N; i++)
    {
        for(j=0; j<N; j++)
        {
            if(i==j)
            {
                ident[i][j] = 1;
            }
            else
            {
                ident[i][j] = 0;
            }
        }
    }

    printf("\n\n\t\tMatriz Identidade:");
    printf("\n\t\t=====");
    for(i=0; i<N; i++)
    {
        printf("\n\t\t");
        for(j=0; j<N; j++)
        {
            printf("%2d",ident[i][j]);
        }
    }
    printf("\n\t\t=====\\n\\n\\n");
    return 0;
}
```

Declaração da constante N

• Manipulação

- Assim como em vetores, a manipulação de matrizes é feita elemento a elemento.

```
#define N 5
int main()
{
    int ident[N][N];
    int i,j;
    for (i=0; i<N; i++)
    {
        for(j=0; j<N; j++)
        {
            if(i==j)
            {
                ident[i][j] = 1;
            }
            else
            {
                ident[i][j] = 0;
            }
        }
    }

    printf("\n\n\t\tMatriz Identidade:");
    printf("\n\t\t=====");
    for(i=0; i<N; i++)
    {
        printf("\n\t\t");
        for(j=0; j<N; j++)
        {
            printf("%2d",ident[i][j]);
        }
    }
    printf("\n\t\t=====\\n\\n\\n");
    return 0;
}
```

Declaração da Matriz ident que tem ordem N

• Manipulação

- Assim como em vetores, a manipulação de matrizes é feita elemento a elemento.

```
#define N 5
int main()
{
    int ident[N][N];
    int i,j;
    for (i=0; i<N; i++)
    {
        for(j=0; j<N; j++)
        {
            if(i==j)
            {
                ident[i][j] = 1;
            }
            else
            {
                ident[i][j] = 0;
            }
        }
    }

    printf("\n\n\t\tMatriz Identidade:");
    printf("\n\t\t=====");
    for(i=0; i<N; i++)
    {
        printf("\n\t\t");
        for(j=0; j<N; j++)
        {
            printf("%2d",ident[i][j]);
        }
    }
    printf("\n\t\t=====\\n\\n\\n");
    return 0;
}
```

Construção da Matriz Identidade

• Manipulação

- Assim como em vetores, a manipulação de matrizes é feita elemento a elemento.

Matriz Identidade:

```
=====
1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1
=====
```

```
#define N 5
int main()
{
    int ident[N][N];
    int i,j;
    for (i=0; i<N; i++)
    {
        for(j=0; j<N; j++)
        {
            if(i==j)
            {
                ident[i][j] = 1;
            }
            else
            {
                ident[i][j] = 0;
            }
        }
    }
}
```

**Impressão da
Matriz
Identidade**

```
printf("\n\n\t\tMatriz Identidade:");
printf("\n\t\t=====");
for(i=0; i<N; i++)
{
    printf("\n\t\t");
    for(j=0; j<N; j++)
    {
        printf("%2d",ident[i][j]);
    }
}
printf("\n\t\t=====\\n\\n\\n");
return 0;
}
```

- **Exemplo**

- Construir um programa que faz a leitura de uma matriz quadrada de reais de tamanho N por N (máximo 20×20).

Somar todos os elementos da diagonal principal.

Imprimir a matriz lida e o valor da soma obtido.

- **Exercícios**

1. Faça um algoritmo que leia uma matriz M 5×5 de números reais. O programa deve determinar o maior número da matriz e a sua posição (linha, coluna).
2. Faça um algoritmo que leia uma matriz de inteiros 5×3 e construa sua transposta. O programa deverá imprimir as duas matrizes, a original e a transposta.

- **Exercícios**

3. Elabore um programa que leia uma matriz de reais 10×10 e determine:
 - A soma da linha 4.
 - O menor elemento da coluna 5.
 - O maior elemento da diagonal principal.