

# INTRODUÇÃO À PROGRAMAÇÃO

Prof.<sup>a</sup> Priscilla Abreu

[priscilla.braz@rj.senac.br](mailto:priscilla.braz@rj.senac.br)



# Introdução à Programação



## Roteiro de Aula

- Matrizes
- Exercícios

# Introdução à Programação



## Objetivo da aula

Manipular matrizes bidimensionais.

# VARIÁVEIS HOMOGÊNEAS BIDIMENSIONAIS

# Introdução à Programação



## INTRODUÇÃO

Nas últimas aulas vimos o conceito de vetor. Vejamos como poderíamos declarar um vetor para armazenar as cinco notas de um aluno:

```
float notas[5];
```

<b>valor</b>	10.0	7.5	4.3	8.2	6.0
<b>posição</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>

Para armazenar e percorrer os dados deste vetor precisamos de um contador que terá a função de controlar a posição em que cada valor será armazenado neste vetor.

**E se ao invés de armazenar  
as 5 notas de um aluno, eu  
quisesse armazenar 5 notas  
de 5 alunos?**

# USO DE MATRIZES!!!

# Introdução à Programação



## INTRODUÇÃO

### Notas:

Ana	10.0	7.5	4.3	8.2	6.0
João	9.5	8.0	3.6	7.8	9.0
Carla	6.5	4.9	5.7	6.0	7.2
Aline	3.5	1.0	5.0	3.6	4,4
Pedro	10.0	9.0	9.5	8.9	9.2



# Introdução à Programação



## INTRODUÇÃO

As matrizes são, comumente referenciadas através de suas dimensões (quantidade de linhas e colunas).

A notação comum é:  $M \times N$ , onde:

- $M$  é a dimensão vertical (quantidade de linhas)
- $N$  é dimensão horizontal (quantidade de colunas)

Exemplo:

***3x3***


***3x2***


***2x3***


***4x1***


***1x3***

--	--	--

Vetores: a quantidade de linhas é sempre 1!

# Introdução à Programação



## Matrizes bidimensionais

Estrutura de dados com mais de um índice

- Sintaxe C:

tipo de dado matriz[tamanho][tamanho];

- Exemplos:

float matNotas[15][4];

int matriz[3][5];

# Introdução à Programação

## INTRODUÇÃO

### Índices / posições:

Índices da linha

Ana
João
Carla
Aline
Pedro

	Índices da coluna				
	0	1	2	3	4
0	10.0	7.5	4.3	8.2	6.0
1	9.5	8.0	3.6	7.8	9.0
2	6.5	4.9	5.7	6.0	7.2
3	3.5	1.0	5.0	3.6	4,4
4	10.0	9.0	9.5	8.9	9.2

# Introdução à Programação



## Matrizes bidimensionais

Como acessar um elemento da matriz?

```
scanf("%d", &mat[0][2]);
```

```
printf("%d", mat[0][2]);
```

```
mat[2][3]= mat[3][2] * 3;
```

# Introdução à Programação



## Matrizes bidimensionais

Se quisermos atribuir valores a todas as posições da matriz:

```
#include <stdio.h>
int main(){
    int mat[2][2];
    printf("Informe um valor para a posição [0][0]");
    scanf("%d",&mat[0][0]);
    printf("Informe um valor para a posição [0][1]");
    scanf("%d",&mat[0][1]);
    printf("Informe um valor para a posição [1][0]");
    scanf("%d",&mat[1][0]);
    printf("Informe um valor para a posição [1][1]");
    scanf("%d",&mat[1][1]);
}
```

# Introdução à Programação



## Matrizes bidimensionais

### Preenchendo uma matriz

- Entretanto, à medida que a quantidade de elementos da matriz aumenta, fica complicado manipularmos manualmente todas as posições.
- O melhor caminho é utilizar laços de repetição!

# Introdução à Programação



## Matrizes bidimensionais

Controle da posição das matrizes:

- Necessário mais de um contador;
- Matrizes de duas dimensões: controle de linha e coluna.
- Acessamos as linhas e para cada linha acessamos todas as colunas.
- Uso de dois contadores.

# Introdução à Programação



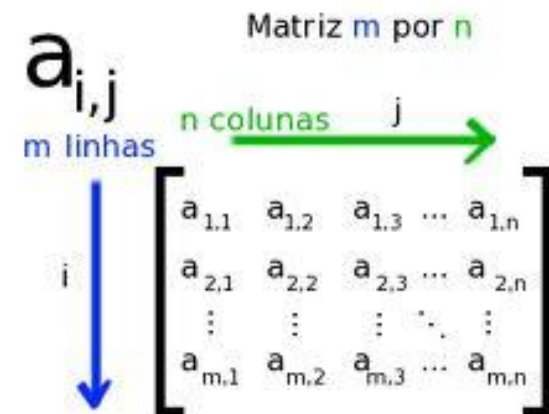
## Matrizes bidimensionais

### Preenchendo uma matriz

- Necessário mais de um contador;
- Matrizes de duas dimensões: controle de linha e coluna.
- Uso de dois contadores.

#### Exemplo:

```
int mat[3][3], lin,col;  
for(lin=0; lin<3; lin++)  
    for (col=0; col<3;col++)  
        scanf("%d", &mat[lin][col]);
```





# Introdução à Programação



## Matrizes bidimensionais – exemplo

```
#include <stdio.h>
#define maxL 2
#define maxC 2
int main(){
    int mat[maxL][maxC], lin, col;
    for (lin=0;lin<maxL;lin++){
        for (col=0;col<maxC;col++){
            printf("Digite um número: ");
            scanf("%d",&mat[lin][col]);
        }
    }
}
```

# Introdução à Programação



## Matrizes bidimensionais – exemplo

```
for (lin=0;lin<max;lin++){  
    for(col=0;col<max;col++){  
        printf("%d ",mat[lin][col]);  
    }  
    printf("\n");  
}
```

```
Informe o elemento da linha 1 e coluna 1  
1  
Informe o elemento da linha 1 e coluna 2  
2  
Informe o elemento da linha 2 e coluna 1  
3  
Informe o elemento da linha 2 e coluna 2  
4  
  
1 2  
3 4  
_
```

# Introdução à Programação

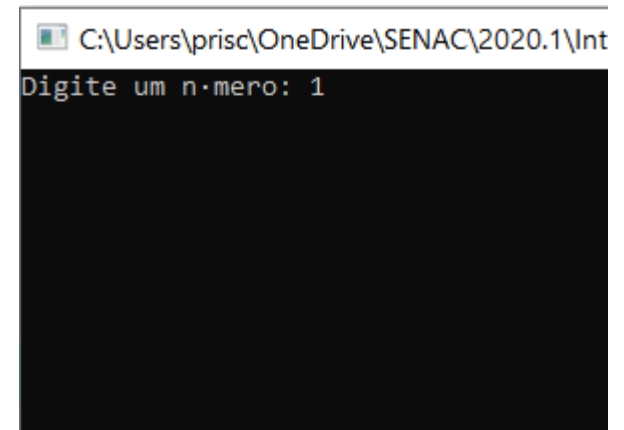


## Matrizes bidimensionais – exemplo

	0	1
0	1	
1		

```
for (lin=0;lin<max;lin++){  
    for (col=0;col<max;col++){  
        printf("Digite um número: ");  
        scanf("%d",&mat[lin][col]);  
    }  
}
```

lin: 0  
col: 0



# Introdução à Programação



## Matrizes bidimensionais – exemplo

	0	1
0	1	5
1		

```
for (lin=0;lin<max;lin++){  
    for (col=0;col<max;col++){  
        printf("Digite um número: ");  
        scanf("%d",&mat[lin][col]);  
    }  
}
```

lin: 0  
col: 1

```
C:\Users\prisc\OneDrive\SENAC\2020  
Digite um número: 1  
Digite um número: 5_
```

# Introdução à Programação



## Matrizes bidimensionais – exemplo

	0	1
0	1	5
1	9	

```
for (lin=0;lin<max;lin++){  
    for (col=0;col<max;col++){  
        printf("Digite um número: ");  
        scanf("%d",&mat[lin][col]);  
    }  
}
```

lin: 1  
col: 0

```
C:\Users\prisc\OneDrive\SENAC\2021.2  
Digite um número: 1  
Digite um número: 5  
Digite um número: 9
```

# Introdução à Programação



## Matrizes bidimensionais – exemplo

	0	1
0	1	5
1	9	7

```
for (lin=0;lin<max;lin++){  
    for (col=0;col<max;col++){  
        printf("Digite um número: ");  
        scanf("%d",&mat[lin][col]);  
    }  
}
```

lin: 1  
col: 1

```
C:\Users\prisc\OneDrive\SENAC\  
Digite um n-mero: 1  
Digite um n-mero: 5  
Digite um n-mero: 9  
Digite um n-mero: 7
```

# Introdução à Programação



## Matrizes bidimensionais – exemplo

Multiplicar os valores lidos de uma matriz 4x4 por 2.

```
#include <stdio.h>
#define max 4
int main(){
    int mat[max][max], lin, col;
    for (lin=0;lin<max;lin++){
        for (col=0;col<max;col++){
            printf("Digite um número: ");
            scanf("%d",&mat[lin][col]);
            mat[lin][col]=mat[lin][col]*2;
        }
    }
```

# Introdução à Programação



## Matrizes bidimensionais – exemplo

Multiplicar os valores lidos de uma matriz 4x4 por 2.

```
for (lin=0;lin<max;lin++){  
    for (col=0;col<max;col++){  
        printf("%d ",mat[lin][col]);  
    }  
    printf("\n");  
}
```



# Introdução à Programação



## Matrizes bidimensionais – exemplo

Criar uma matriz identidade

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

# Introdução à Programação



## Matrizes bidimensionais – exemplo

Criar uma matriz identidade

Posições da matriz:

Lin			
Col	0	1	2
0	00	01	02
1	10	11	12
2	20	21	22

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

# Introdução à Programação



## Matrizes bidimensionais – exemplo

Criar uma matriz identidade

Posições da matriz:

Lin	0	1	2
Col			
0	00	01	02
1	10	11	12
2	20	21	22

O QUE ESSAS  
POSIÇÕES TEM  
EM COMUM?

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

# Introdução à Programação



## Matrizes bidimensionais – exemplo

Criar uma matriz identidade

```
int matriz[3][3];  
for (lin=0;lin<3;lin++)  
    for (col=0; col<3; col++){  
        if (lin == col)  
            matriz[lin][col] = 1;  
        else  
            matriz[lin][col]= 0;  
    }
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

# Introdução à Programação



## Matrizes bidimensionais – CONSTANTES

Vantagem e importância da utilização de constantes:

Se houver necessidade de alterar a dimensão do vetor, basta alterar o valor da constante **max**.

# Introdução à Programação



## Matrizes bidimensionais – CONSTANTES

```
#include <stdio.h>
#define maxL 3
#define maxC 4
int main(){
    int mat[maxL][maxC];
    int lin,col;
    for (lin=0;lin<maxL;lin++){
        for (col=0;col<maxC;col++){
            printf("Digite um número: ");
            scanf("%d",&mat[lin][col]);
            mat[lin][col]=mat[lin][col]*2;
        }
    }
```

```
for (lin=0;lin<maxL;lin++){
    for (col=0;col<maxC;col++){
        printf("%d ",mat[lin][col]);
    }
    printf("\n");
}
```

# Introdução à Programação



## Matrizes bidimensionais

- Nesta aula vimos estruturas de dados homogêneas que permitem armazenar e manipular um conjunto de dados do mesmo tipo por meio de uma mesma variável.
- Matrizes são estruturas de dados com duas ou mais dimensões (dois ou mais índices). Seu acesso está associado ao uso de estruturas de repetição, de acordo com sua dimensão.

# Introdução à Programação



## Matrizes bidimensionais – EXERCÍCIO

Faça um programa que leia valores inteiros em duas matrizes 3x3 e imprima a matriz soma das duas.

**Exemplo:**

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 4 \\ 6 & 7 & 8 \\ 10 & 11 & 12 \end{bmatrix}$$



# DÚVIDAS?