

Algoritmos e Estruturas de Dados I

Matriz

Prof. Ivre Marjorie

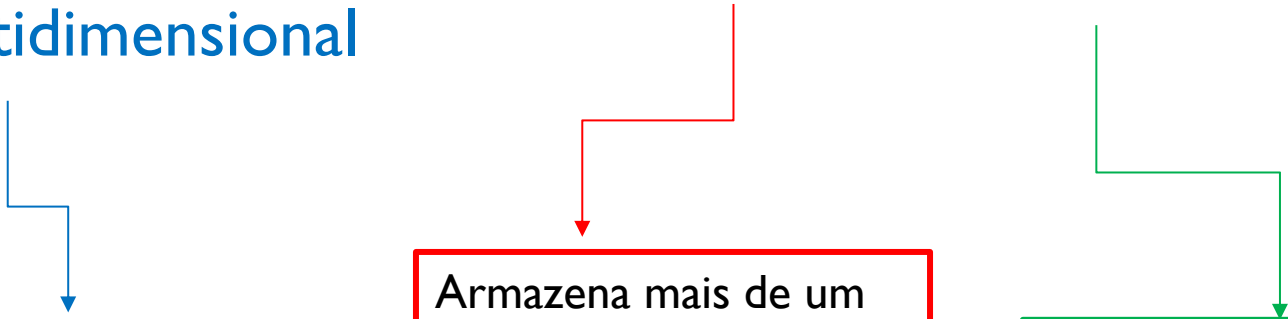
Introdução

- ▶ Assim, como no vetor (que é considerado uma matriz de **UMA** dimensão),
 - ▶ É necessário conseguir acessar cada uma das posições da matriz, para isso, são usados os índices
- ▶ No caso da matriz teremos um índice para cada dimensão
- ▶ Um índice para as linhas
 - ▶ começa em ZERO e vai até a quantidade de linhas menos 1
- ▶ Um índice para as colunas
 - ▶ começa em ZERO e vai até a quantidade de colunas menos 1



Introdução

- ▶ Uma matriz é uma variável, **composta**, **homogênea**, **multidimensional**



Sua representação é feita em apenas mais de uma dimensão, no caso, vamos usar linha e coluna

Armazena mais de um valor em uma mesma variável

Os valores devem ser de um mesmo tipo, por exemplo, int, float, double ou char



Matriz - Representação

- Considere que a matriz abaixo possui 3 linhas e 5 colunas e seu nome é **matrizX**:

```
int matrizX[ 3 ][ 5 ];
```

Coluna

Linha

índices

0

1

2

3

4

0

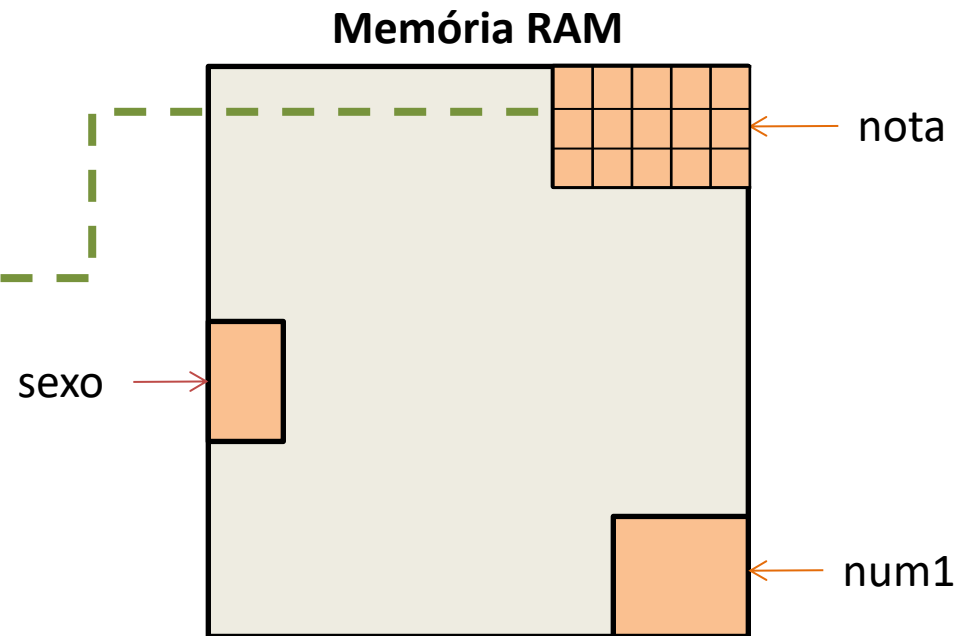
1

2

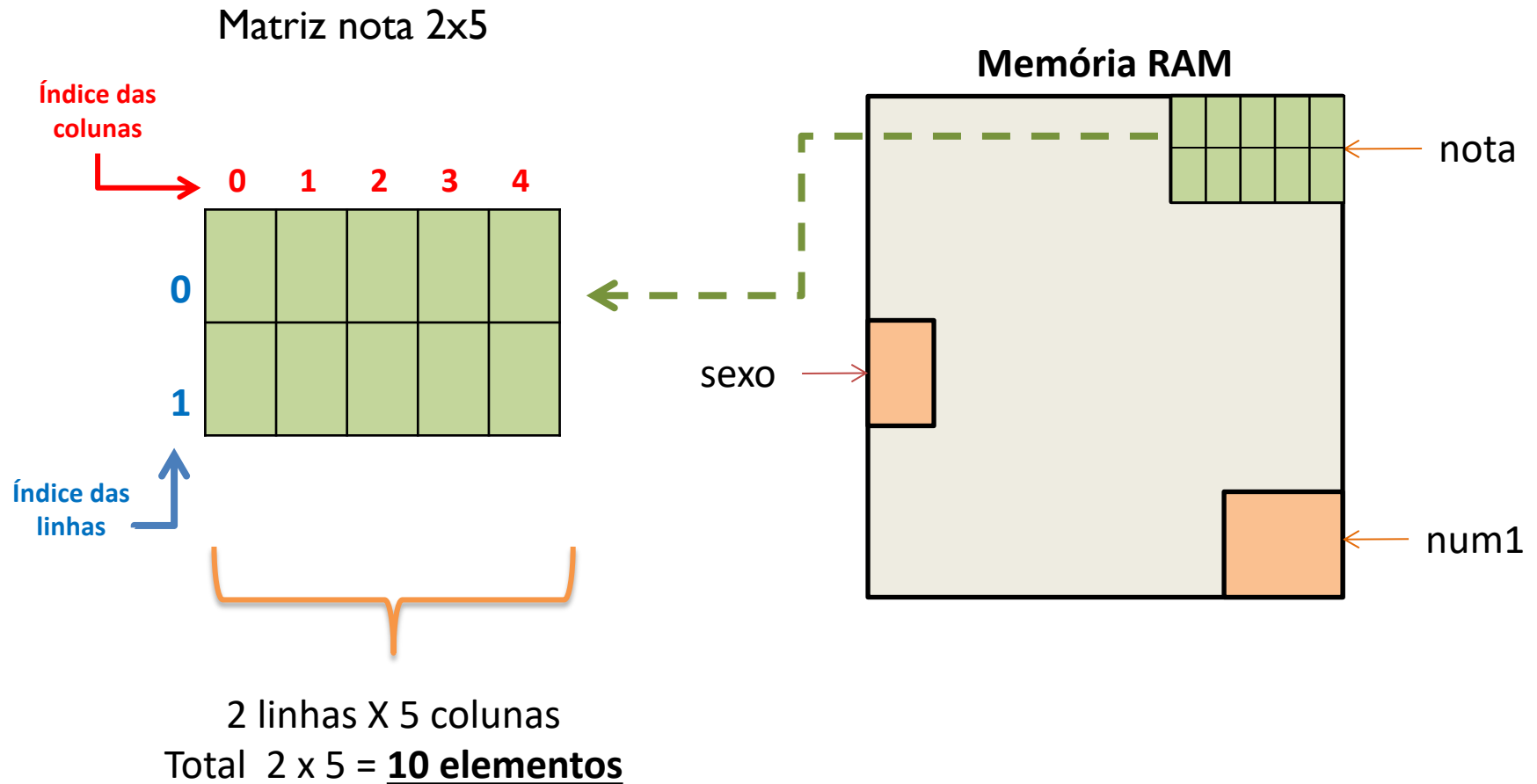
Matriz - Representação

Com a matriz: é possível armazenar um valor em **cada** linha e coluna da variável.

Observe que é possível armazenar mais valores.



Matriz - Representação



Matriz - Exemplo

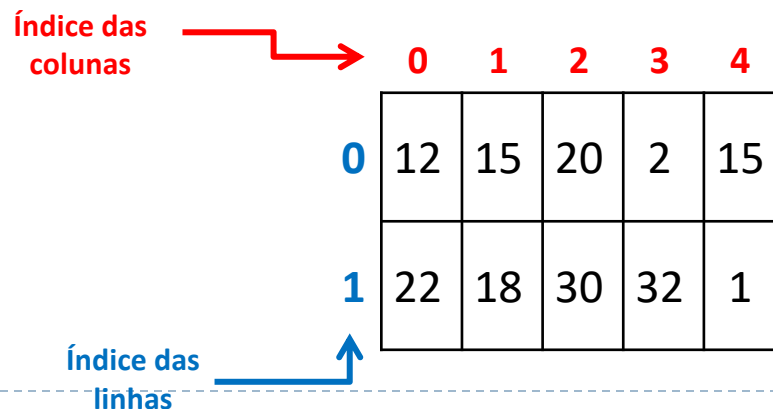
int nota[2][5];



Matriz de inteiros

nota[0][0], nota[0][1], nota[0][2], nota[0][3], nota[0][4]

nota[1][0], nota[1][1], nota[1][2], nota[1][3], nota[1][4]



The diagram shows a 2x5 matrix with row and column indices. A red arrow points from the text 'Índice das colunas' to the column indices 0, 1, 2, 3, 4. A blue arrow points from the text 'Índice das linhas' to the row indices 0, 1.

	0	1	2	3	4
0	12	15	20	2	15
1	22	18	30	32	1

Matriz - Declaração em C

- ▶ Declarando uma variável do tipo matriz

Tipo **nome** [**Qtde de linhas**][**Qtde de Colunas**];

- ▶ **Tipo** : indica do tipo de cada elemento da matriz
- ▶ **nome**: indica o nome da coleção de variáveis (seguir mesmas regras para um nome de variável qualquer)
- ▶ **Qtde de linhas**: Quantidade de linhas que a matriz deverá ter
- ▶ **Qtde de Colunas**: Quantidade de colunas que a matriz deverá ter



Matriz - Declaração em C

Tipo nome [Qtde de linhas][Qtde de Colunas];

► Exemplos:

```
int idade [2][4];
```

```
double peso [10][5];
```

```
float nota [3][6];
```



Matriz - Acesso a valores

- ▶ Para acessar os elementos de uma matriz, deve utilizar o valor dos índices desejados, juntamente com o nome da variável, por exemplo, `peso[2][1]` está associado a terceira linha, segunda coluna da matriz, pois o primeiro elemento, tanto da linha, como coluna está relacionado ao índice **0**

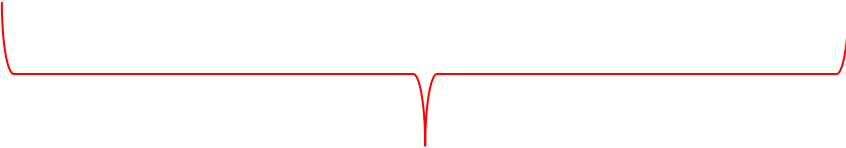


Matriz - Atribuição de valores

- ▶ Atribui o valor 3 a primeira linha, segunda coluna da matriz => `mat[0][1] = 3;`

- ▶ Inicializando uma matriz na declaração

```
int matA[3][3]={0,2,4},{5,12,14},{15,32,34}};
```



Valores inseridos na
matriz matA, ao declarar



Matriz - Preenchendo

- ▶ Preencher ou carregar uma matriz, significa armazenar um valor em cada uma das as posições da matriz
- ▶ Para isso, é necessário usar duas estruturas de repetição
 - ▶ Para percorrer toda a matriz (linhas e colunas, no caso de ter duas dimensões)
 - ▶ e a mais simples e indicada, é a estrutura **for**



Matriz - Preenchendo

```
for (lin = 0; lin < qtde de linhas; lin++)
{
    for (col = 0; col < qtde de colunas; col++)
    {
        scanf("%tipo de dados" , nome_matriz[lin][col]);
    } //fim for colunas
} //fim for linhas
```



Matriz - Preenchendo

A variável **lin** começa com **ZERO**, pois o índice que controla as linhas da matriz começa com zero

```
for (lin = 0; lin < qtde de linhas; lin++)  
{  
    for (col = 0; col < qtde de colunas; col++)  
    {  
        scanf("%tipo de dados", nome_matriz[lin][col]);  
    } //fim for colunas  
} //fim for linhas
```

A variável **col** começa com **ZERO**, pois o índice que controla as colunas da matriz começa com zero

Matriz - Mostrando

- ▶ Mostrar uma matriz, significa apresentar na tela cada um dos valores da matriz, ou seja, todas as posições da matriz
- ▶ Para isso, é necessário usar duas estruturas de repetição
 - ▶ Para percorrer toda a matriz (linhas e colunas, no caso de ter duas dimensões)
 - ▶ e a mais simples e indicada, é a estrutura **for**



Matriz - Mostrando

```
for (lin = 0; lin < qtde de linhas; lin++)  
{  
    printf (“\n”);  
    for (col = 0; col < qtde de colunas; col++)  
    {  
        printf (“%tipo de dados | ”, nome_matriz[ lin ][ col]);  
    } //fim for colunas  
} //fim for linhas
```




Exemplo

- ▶ Faça um programa que preencha uma matriz 3 x 3 e mostre a quantidade de elementos maiores que 15 e menores que 25. Mostre também a matriz completa.

	0	1	2
0	21	2	15
1	14	26	34
2	24	2	22



Código – C (parte1)

```
int main()
{
    .....
    int matA[3][3];
    int l, c, cont=0;
    for(l=0; l<3; l++)
    {
        for(c=0; c<3; c++)
        {
            printf("Digite um numero: ");
            scanf("%d", &matA[l][c]);
            if(matA[l][c] > 15 && matA[l][c] <25)
            {
                cont++;
            }
        }
    }
}

```

Código – C (parte1)

```
printf("\n\nValores da matriz:");
for(l=0; l<3; l++)
{
    printf("\n");
    for(c=0; c<3; c++)
    {
        printf("%d |", matA[l][c]);
    }
}
printf("\n\nA quantidade de elementos entre 15 e 25 e: %d", cont);
return 0;
}
```





Exemplo (Tela)

```
Digite um numero: 21
Digite um numero: 2
Digite um numero: 15
Digite um numero: 14
Digite um numero: 26
Digite um numero: 34
Digite um numero: 24
Digite um numero: 2
Digite um numero: 22
```

```
Valores da matriz:
```

```
21 | 2 | 15 |
14 | 26 | 34 |
24 | 2 | 22 |
```

```
A quantidade de elementos entre 15 e 25 e: 3
Process returned 0 (0x0)   execution time : 27.623 s
Press ENTER to continue.
```

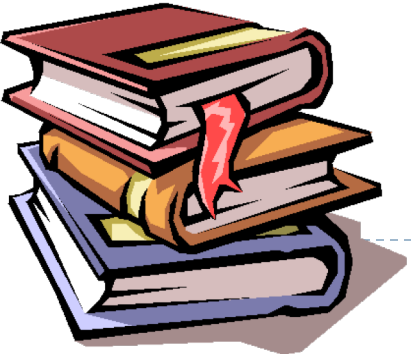




Exercícios

- 1- Faça um programa que carregue uma matriz 4×5 , calcule e mostre um vetor com cinco posições, onde cada posição contém a soma dos elementos de cada **coluna** da matriz. Em seguida, mostre o vetor e a matriz na tela.
- ▶ 2- Faça um programa que carregue uma matriz **5×5** com números inteiros, calcule e mostre a soma:
- ▶ dos elementos da linha 4;
 - ▶ dos elementos da coluna 2;
 - ▶ de todos os elementos da matriz.





Referência Bibliográfica

- ▶ MIZRAHI, Victorine Viviane. **Treinamento em linguagem C**. São Paulo: Pearson Prentice Hall, 2008. 2ª edição. Curso Completo. Capítulo 7.
- ▶ ASCENCIO, Ana Fernanda Gomes e CAMPOS, Edilene A. Veneruchi. **Fundamentos da Programação de Computadores – Algoritmos, Pascal, C/C++ e Java**. São Paulo: Pearson Prentice Hall, 2012. 3ª Edição.

