UNIVERSIDADE
FEDERAL DO CARIRI

Matrizes

Professora Dra. Luana Batista da Cruz
luana.batista@ufca.edu.br

Roteiro

01 Introdução

02 Matrizes

01

Introdução

Motivação

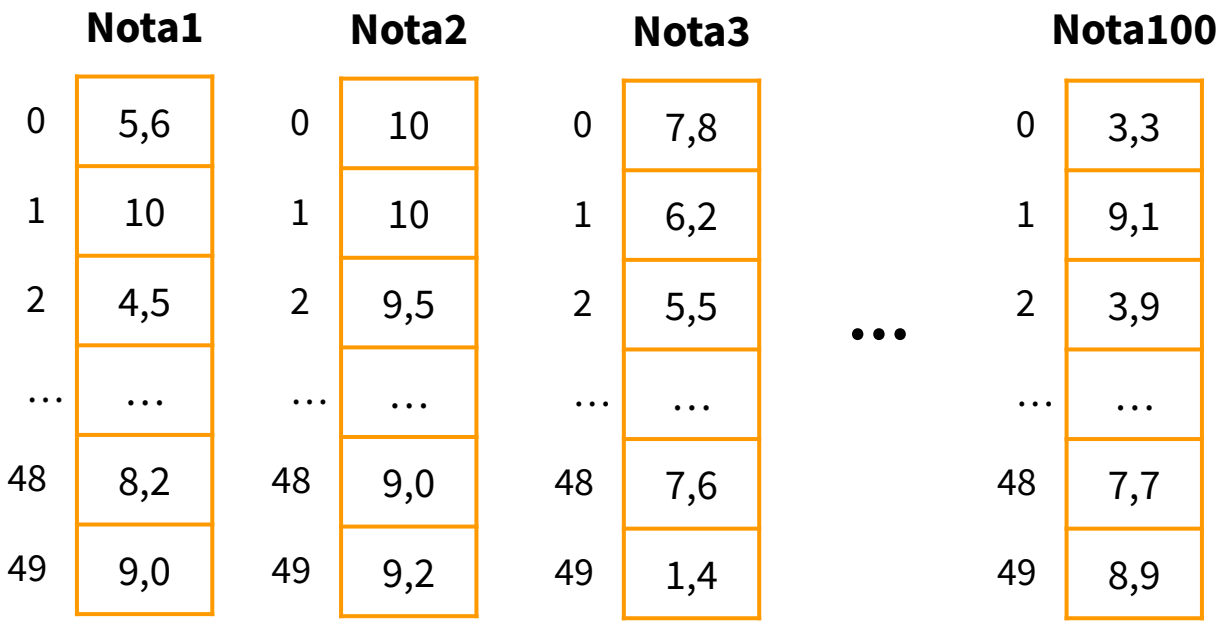
Introdução

- **Motivação**
  - Fazer um programa para ler as notas de 4 provas para 50 alunos de uma turma e calcular a média do aluno e média da turma
  - **Solução**: criar 4 vetores de 50 posições, sendo um para cada nota
    - double nota1[50], nota2[50], nota3[50], nota4[50];

	Nota1	Nota2	Nota3	Nota4
0	5,6	10	7,8	3,3
1	10	10	6,2	9,1
2	4,5	9,5	5,5	3,9
…	…	…	…	…
48	8,2	9,0	7,6	7,7
49	9,0	9,2	1,4	8,9

Introdução

- **Motivação**
  - E se tivermos que armazenar 100 notas? Criaremos 100 vetores com 100 nomes diferentes?



Introdução

- **Motivação**
  - Uma solução mais eficaz para resolver o problema é o uso de **matrizes**:

	0	1	2	…	99
0	5,6	9,8	7,6	…	4,9
1	10	9,3	4,9	…	9,8
2	4,5	7,9	10	…	7,9
…	…	…	…	…	…
48	8,2	8,6	10	…	6,8
49	9,0	7,8	9,2	…	3,6

02

Matrizes

Matrizes: variáveis compostas homogêneas
Declaração de matrizes
Inicialização de matrizes
Exemplos

Matrizes: variáveis compostas homogêneas

- As variáveis compostas homogêneas correspondem a um conjunto de elementos de mesmo tipo e que compartilham um mesmo nome
- Cada um dos elementos é unicamente identificado por um número inteiro (índice) que especifica a sua localização dentro da estrutura
- Estas variáveis podem ser **unidimensionais (vetores)** ou **multidimensionais (matrizes)**

Declaração de matrizes

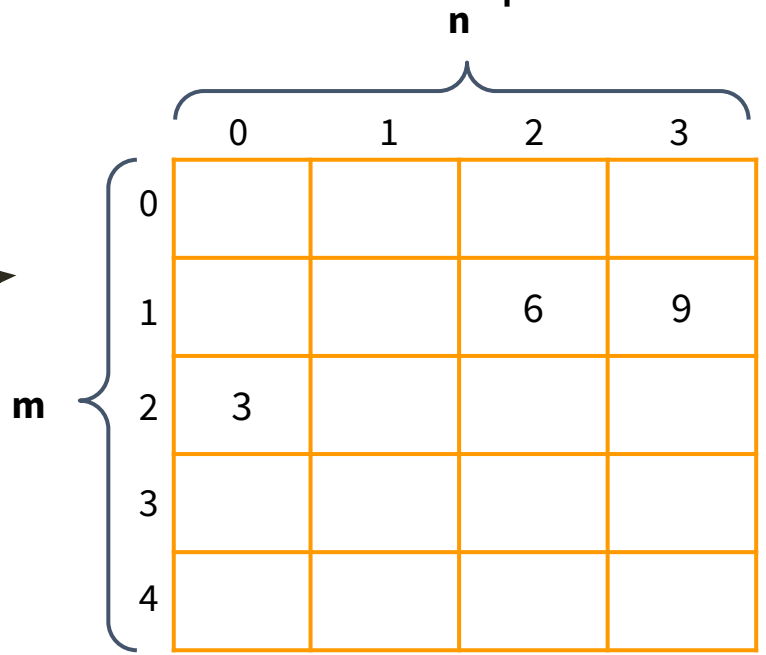
- <tipo> identificador [<linhas>] [<colunas>];
  - **Tipo**: tipo dos dados que serão armazenados na matriz (int, char, f loat, etc)
  - **Identificador**: é o nome da variável que identifica a matriz
  - **Linhas**: número de elementos da primeira dimensão
  - **Colunas**: número de elementos da segunda dimensão
  - As linhas e colunas são numeradas de 0 até tamanho − 1

- **Exemplo**
  - double notas[50][100]; //matriz com 50 linhas e 100 colunas

Declaração de matrizes

- Por exemplo, uma matriz bi-dimensional pode ser vista como uma tabela de **m** linhas e **n** colunas

int matriz[5][4] →



- Sempre começa com o índice de valor igual a zero e termina com o valor de seu tamanho menos um
- Um valor de uma matriz pode ser acessado a partir de seu índice
  - Ex: matriz[2][0] = 3; printf(“%d”, matriz[1][2]);

Declaração de matrizes

- **Importante**
  - C não verifica o limite das dimensões das variáveis compostas
  - Se uma instrução for feita com índices além do limite, é possível que não ocorra um erro de execução do programa e outros valores sejam sobrepostos na memória
  - É responsabilidade do programador providenciar a verificação dos limites das dimensões das variáveis compostas

Matrizes

- **Exemplo 1**: faça um programa que leia e imprima uma matriz 4×3 (4 linhas e 3 colunas)

Matrizes

- **Exemplo 1**: faça um programa que leia e imprima uma matriz 4×3 (4 linhas e 3 colunas)

```c
#include <stdio.h>
#include <stdlib.h>
#define M 4
#define N 3

int main(){

    int i, j, m[M][N];

    //capturando os dados
    for(i=0; i<M; i++){     //linhas
        for(j=0; j<N; j++){  //colunas
            scanf("%d", &m[i][j]);
        }
    }

    //imprimindo a matriz
    for(i=0; i<M; i++){     //linhas
        printf("\n");
        for(j=0; j<N; j++)  //colunas
            printf("M[%d][%d]= %d, ", i, j, m[i][j]);
    }

    system("PAUSE");
    return 0;
}
```

Matrizes

- **Exemplo 1**: faça um programa que leia e imprima uma matriz 4×3 (4 linhas e 3 colunas)

```
M[0][0]= 10, M[0][1]= 11, M[0][2]= 12,
M[1][0]= 13, M[1][1]= 14, M[1][2]= 15,
M[2][0]= 16, M[2][1]= 17, M[2][2]= 18,
M[3][0]= 19, M[3][1]= 20, M[3][2]= 21,
```

```c
#include <stdio.h>
#include <stdlib.h>
#define M 4
#define N 3

int main(){

    int i, j, m[M][N];

    //capturando os dados
    for(i=0; i<M; i++){     //linhas
        for(j=0; j<N; j++){  //colunas
            scanf("%d", &m[i][j]);
        }
    }

    //imprimindo a matriz
    for(i=0; i<M; i++){     //linhas
        printf("\n");
        for(j=0; j<N; j++)  //colunas
            printf("M[%d][%d]= %d, ", i, j, m[i][j]);
    }

    system("PAUSE");
    return 0;
}
```

Inicialização de matrizes

- Inicializando cada elemento da matriz (m x n) com o valor 0

```
int i, j, m[M][N];

//capturando os dados
for(i=0; i<M; i++){     //linhas
    for(j=0; j<N; j++){  //colunas
        m[i][j] = 0;
    }
}
```

Inicialização de matrizes

- Inicializando na declaração. Processo semelhante à inicialização de vetores

```
int matriz[3][4] = { {10, 20, 30, 40},
                     {50, 60, 70, 80},
                     {90, 11, 22, 33} };
```

- Mas podemos fazer também:

```
int matriz[3][4] = { 10, 20, 30, 40,
                     50, 60, 70, 80,
                     90, 11, 22, 33 };
```

- Ou ainda:

```
int matriz[3][4] = { 10, 20, 30, 40, 50, 60, 70, 80, 90, 11, 22, 33 };
```

Matrizes

- **Exemplo 2**: dada uma matriz (4 × 5), calcular a soma de todos os elementos da matriz. Calcular também o somatório dos elementos de cada linha da matriz, armazenando o somatório em um vetor

	0	1	2	3	4
0	1	2	3	4	5
1	0	-1	0	-3	1
2	2	-2	-2	2	0
3	0	0	6	0	0

Soma linha
15
-3
0
6

Total = 18

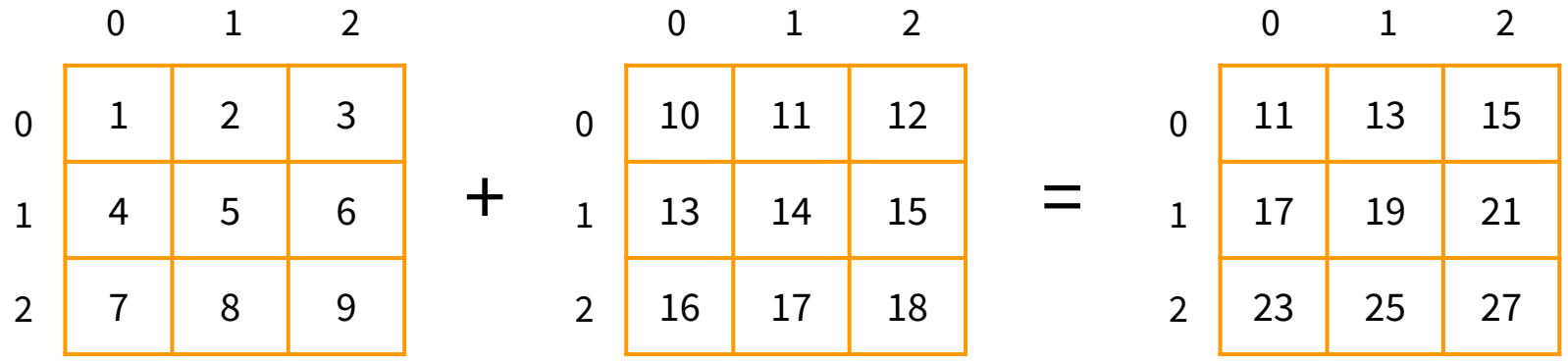
Matrizes

- **Exemplo 2**

```c
#include<stdio.h>
#include<stdlib.h>
#define M 4
#define N 5

int main(){

    int i, j;
    float matriz[M][N], soma_linha[4], total = 0;

    //capturando os dados
    for(i=0; i<M; i++){ //linhas
        for(j=0; j<N; j++){ //colunas
            scanf("%f", &matriz[i][j]);
        }
    }

    for(i=0; i < M; i++){
        soma_linha[i] = 0; //a soma de cada linha é inicializada com zero
        for(j=0; j < N; j++){ //somando os valores da linha em soma_linha[i]
            soma_linha[i] = soma_linha[i] + matriz[i][j];
        }
        total = total + soma_linha[i];//somando o total de cada linha
    }

    //imprimindo o vetor soma_linha
    for(i=0; i<4; i++){
        printf("SL[%d]: %.2f\n", i, soma_linha[i]);
    }

    printf("Resultado da soma de todos os elementos: %.2f\n", total);

    system("PAUSE");
    return 0;
}
```

Matrizes

- **Exemplo 3**: faça um programa que calcule a soma de duas matrizes 3 x 3



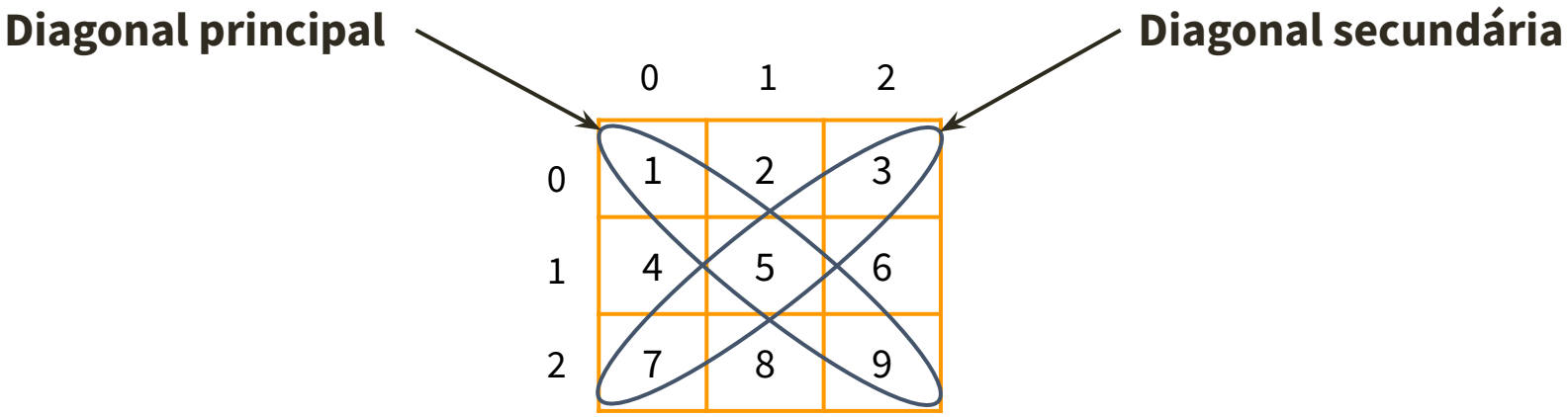
Matrizes

- **Exemplo 3**

```
8     int i, j;
9     int matriz_1[M][N], matriz_2[M][N], matriz_aux[M][N];
10
11    //capturando os dados da matriz 1
12    for(i=0; i<M; i++){ //linhas
13        for(j=0; j<N; j++){ //colunas
14            scanf("%d", &matriz_1[i][j]);
15        }
16    }
17
18    //capturando os dados da matriz 2
19    for(i=0; i<M; i++){ //linhas
20        for(j=0; j<N; j++){ //colunas
21            scanf("%d", &matriz_2[i][j]);
22        }
23    }
24
25    //soma matrizes 1 e 2
26    for(i=0; i<M; i++){ //linhas
27        for(j=0; j<N; j++){ //colunas
28            matriz_aux[i][j] = matriz_1[i][j] + matriz_2[i][j];
29        }
30    }
31
32    //imprimindo a matriz
33    for(i=0; i<M; i++){ //linhas
34        printf ("\n");
35        for(j=0; j<N; j++) //colunas
36            printf("Matriz[%d][%d]= %d, ", i, j, matriz_aux[i][j]);
37    }
```

Matrizes

- **Exemplo 4**: escreva um programa que declare e preencha uma matriz quadrada com valores fornecidos pelo usuário. Em seguida, o programa deve imprimir os elementos da diagonal principal e da diagonal secundária



Matrizes

- **Exemplo 4**

```c
int i, j, m;

printf("Digite o valor da matriz quadrada: ");
scanf("%d", &m);

int matriz[m][m];

//capturando os dados
for(i=0; i<m; i++){ //linhas
    for(j=0; j<m; j++){ //colunas
        scanf("%d", &matriz[i][j]);
    }
}

//imprimir diagonal principal
printf("Diagonal principal\n");
for(i=0; i<m; i++){ //linhas
    for(j=0; j<m; j++){ //colunas
        if(i == j)
            printf("M[%d][%d]: %d\n", i, j, matriz[i][j]);
    }
}

//imprimir diagonal secundária
printf("Diagonal secundaria\n");
for(i=0; i<m; i++){ //linhas
    for(j=0; j<m; j++){ //colunas
        if((i+j) == m-1)
            printf("M[%d][%d]: %d\n", i, j, matriz[i][j]);
    }
}
```

Matrizes

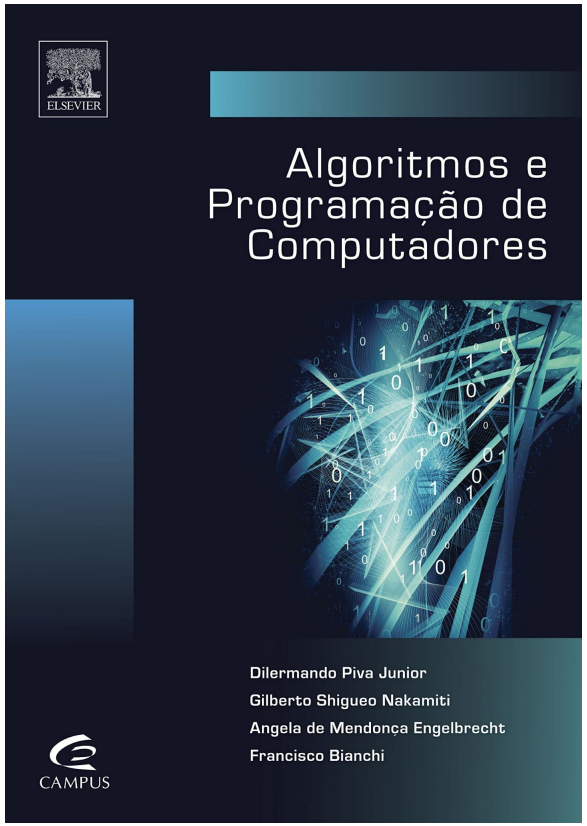
- **Exemplo 4**

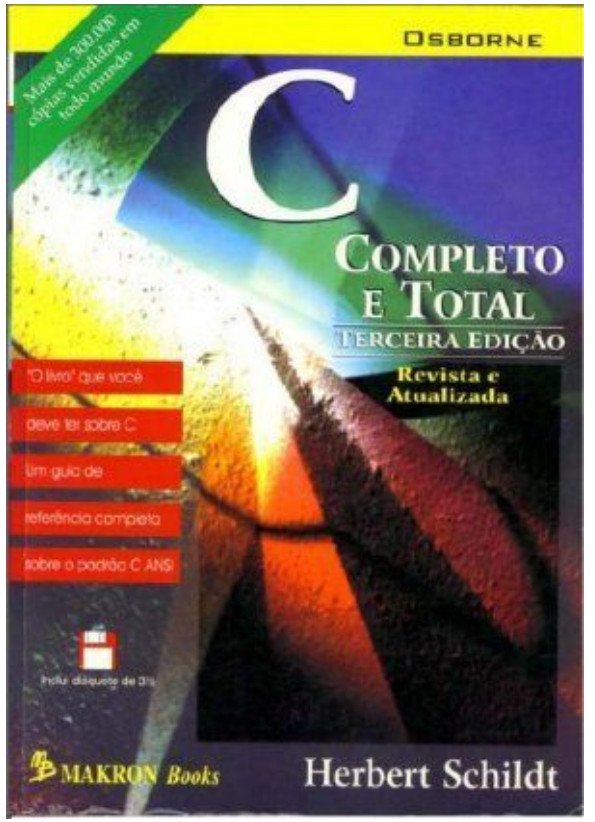
```c
int i, j, m;

printf("Digite o valor da matriz quadrada: ");
scanf("%d", &m);

int matriz[m][m];

//capturando os dados
for(i=0; i<m; i++){ //linhas
    for(j=0; j<m; j++){ //colunas
        scanf("%d", &matriz[i][j]);
    }
}

//imprimir diagonal principal
printf("Diagonal principal\n");
for(i=0; i<m; i++)
    printf("M[%d][%d]: %d\n", i, j, matriz[i][i]);

//imprimir diagonal secundária
printf("Diagonal secundaria\n");
for(i=0; i<m; i++)
    printf("M[%d][%d]: %d\n", i, j, matriz[i][m-1-i]);
```

Resumindo..

- Estruturas de dados homogêneas
- Matrizes
- Exemplos

Referências





PIVA, D. J. et al. **Algoritmos e programação de computadores**. Rio de Janeiro, RJ: Elsevier, 2012.

SCHILDT, Herbert. **C completo e total**. Makron, 1997.