## Aula 05 - Introdução à Linguag C Programação: Estr. Dados Matrizes

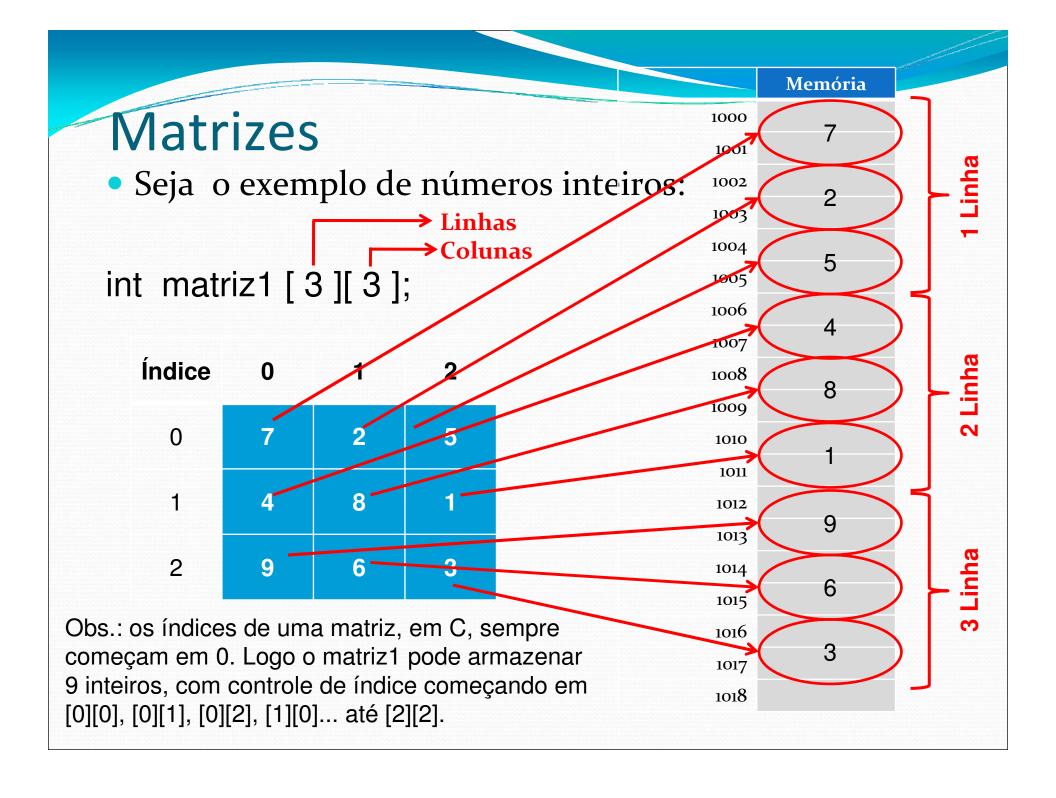
OPL e IC1

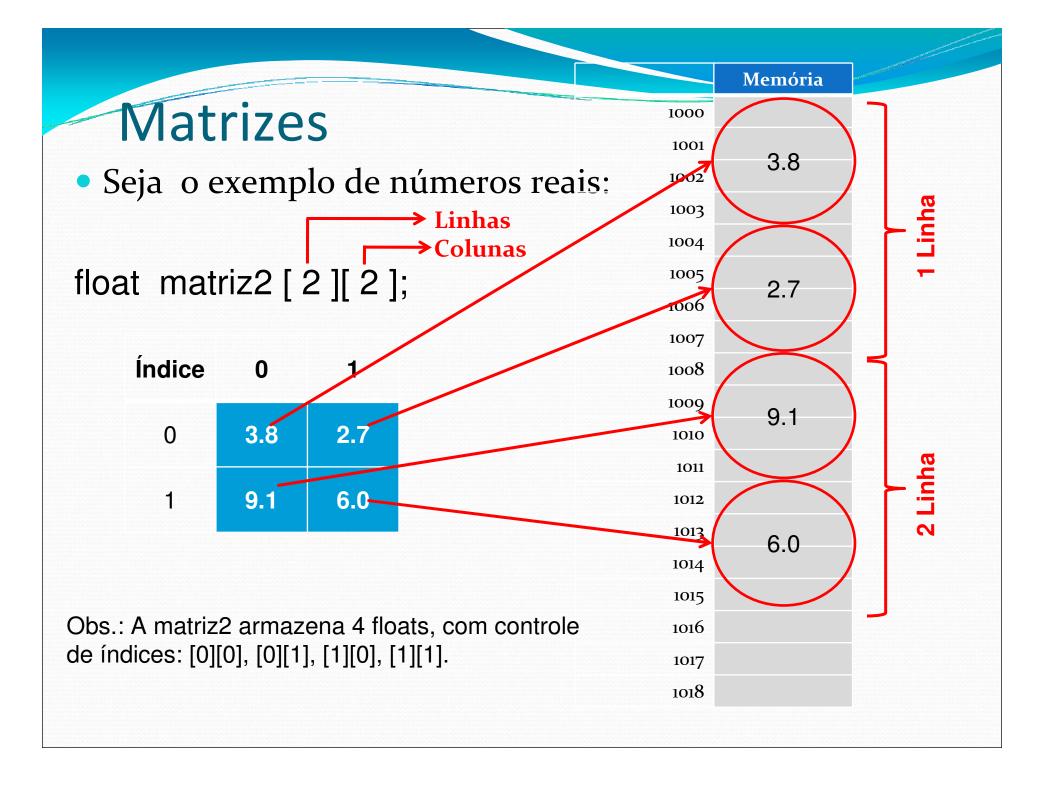
Prof: Anilton Joaquim da Silva

Anilton.ufu@outlook.com

### Matrizes

- Uma matriz é uma variável capaz de armazenar vários dados de mesmo tipo, ou seja, é uma variável composta homogênea bidimensional (arranjo bidimensional).
- Declaração da Matriz
   Tipo\_da\_matriz nome\_matriz [linhas][colunas];
- Exemplos:
  - int matriz1[3][3];
  - float matriz2[2][2];
  - char matriz\_nomes[10][30];





#### Matrizes

• Seja o exemplo de números inteiros:

```
Linhas – tamanho máximo Colunas – tamanho máximo int matriz1 [3][3];
```

Índice	0	1	2
0	7	2	5
1	4	8	1
2	9	6	3

```
Posição:

matriz1 [1][1] \rightarrow 8

matriz1 [0][2] \rightarrow 5

matriz1 [2][0] \rightarrow 9

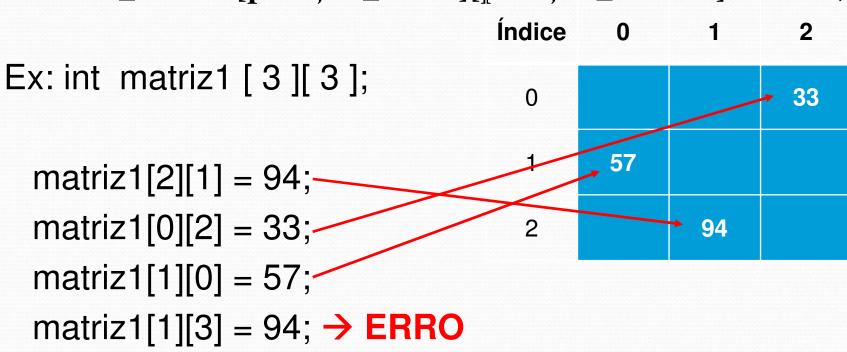
matriz1 [3][1] \rightarrow ? \rightarrow ERRO

matriz1 [2][3] \rightarrow ? \rightarrow ERRO
```

#### Atribuindo valores a uma matrizes

 Para atribuir diretamente um valor a uma determinada posição da matriz, pode-se usar a seguinte expressão:

nome\_matriz[posição\_linha][posição\_coluna] = valor;



#### Lendo uma matriz

 Para ler uma matriz e ter seus valores alocados em posições específicas, pode-se utilizar dois comandos for, um dentro do outro:

```
for (int i = 0; i < L; i++) //i = indice de linha
{
    for(int j = 0; j < C; j++) //j = indice de coluna
    {
        printf ("Digite Matriz: [ %d ][ %d ]: ", i, j);
        scanf ("%d", &matriz [ i ][ j ]);
    }
}</pre>
```

 OBS: as variáveis L e C, definem o tamanho da matriz em linhas e colunas, no momento da execução do programa.

## Imprimindo uma matriz

 Para mostra uma matriz já lida e ter seus valores apresentados na tela, também pode-se utilizar dois comandos for, um dentro do outro:

```
printf ("Matriz digitada: \n");
for (int i = 0; i < L; i++) //i = indice de linha
{
        for(int j = 0; j < C; j++) //j = indice de coluna
        {
            printf (" %d", matriz[ i ][ j );
        }
        printf ("\n"); //posicionar na próxima linha
}</pre>
```

## Exemplo completo

#include <stdio.h>

```
#define tamanho 100
int main()
  int matriz[tamanho][tamanho];
  int numeroLinhas, numeroColunas;
  printf ("Matriz - Le e mostra uma matriz\n");
  printf ("Digite numero de linhas: ");
  scanf ("%d", &numeroLinhas);
  printf ("Digite numero de colunas: ");
  scanf ("%d", &numeroColunas);
   printf ("\nDigite os elementos da matriz:\n");
   for (int linha = 0; linha < numeroLinhas; linha++)
    for(int coluna = o; coluna < numeroColunas; coluna++)</pre>
      printf ("Digite matriz[%d][%d]: ", linha, coluna);
      scanf ("%d", &matriz[linha][coluna]);
```

```
printf ("Matriz digitada: \n");
for (int linha = o; linha < numeroLinhas; linha++)
{
   for(int coluna = o; coluna < numeroColunas; coluna++)
   {
      printf (" %d", matriz[linha][coluna]);
   }
   printf ("\n"); //posicionar na próxima linha
}
return o;
}</pre>
```

# Inicializando uma matriz com valores pré-definidos

 Uma matriz pode ter os seus valores inicializados na declaração da mesma:

```
int a[1][10] = \{\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}\};

int b[2][10] = \{\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\},

\{11, 12, 13, 14, 15, 16, 17, 18, 19, 20\}\};

int c[3][10] = \{\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\},

\{11, 12, 13, 14, 15, 16, 17, 18, 19, 20\},

\{21, 22, 23, 24, 25, 26, 27, 28, 29, 30\}\};
```

OBS: isto também pode ser feito com vetores: int vetor[6] = { 8, 7, 6, 5, 4, 3 };

# Multiplicação de Matrizes

Sejam duas matrizes
 A e B, tal que A tem
 dimensões MxN e B;
 NxO; então, a
 matriz AxB tem
 dimensões MxO, e

$$C_i, j = \sum_{k=1}^m A_{m,n} B_{n,o}$$

$$\begin{bmatrix} 123 \\ 456 \end{bmatrix} \times \begin{bmatrix} 78 \\ 910 \\ 1112 \end{bmatrix} = \begin{bmatrix} 58 \\ 64 \end{bmatrix} 1 \times 7 + 2 \times 9 + 3 \times 11 = 58$$

$$\begin{bmatrix} 123 \\ 456 \end{bmatrix} \times \begin{bmatrix} 78 \\ 910 \\ 1112 \end{bmatrix} = \begin{bmatrix} 58 \\ 64 \end{bmatrix} 1 \times 8 + 2 \times 10 + 3 \times 12 = 64$$

$$\begin{bmatrix} 123 \\ 456 \end{bmatrix} \times \begin{bmatrix} 78 \\ 910 \\ 1112 \end{bmatrix} = \begin{bmatrix} 58 \\ 64 \\ 139 \end{bmatrix} 4 \times 7 + 5 \times 9 + 6 \times 11 = 139$$

$$\begin{bmatrix} 123 \\ 456 \end{bmatrix} \times \begin{bmatrix} 78 \\ 910 \\ 1112 \end{bmatrix} = \begin{bmatrix} 58 \\ 64 \\ 139 \end{bmatrix} 4 \times 8 + 5 \times 10 + 6 \times 12 = 154$$

## Multiplicação de Matrizes

```
#include <stdio.h>
#define tam 100
int main()
 int M, N, O, i, j, k;
 int a[tam][tam], b[tam][tam], c[tam][tam], soma;
  printf ("Digite numero de linhas de a: ");
  scanf ("%d", &M);
  printf ("Digite numero de colunas de a: ");
  scanf ("%d", &N);
  printf ("Digite numero de colunas de b: ");
  scanf ("%d", &O);
  printf ("\nDigite as linhas da matriz a:\n");
  for (i = 0; i < M; i++)
    for (k = 0; k < N; k++)
         scanf ("%d", &a[i][k]);
  printf ("\nDigite as linhas da matriz b:\n");
  for (k = 0; k < N; k++)
    for (j = 0; j < 0; j++) {
         scanf ("%d", &b[k][j]);
```

```
//calculo de c = a x b
for (i = o; i < M; i++)
  for (j = 0; j < 0; j++)
    soma = o;
    for (k = 0; k < N; k++)
       soma = soma + a[i][k] * b[k][j];
    c[i][j] = soma;
printf("Matriz c: \n");
for (i = 0; i < M; i++)
 for (j = 0; j < 0; j++)
   printf (" %d", c[i][j]);
  printf ("\n"); //posicionar na próxima linha
return o;
```