

Aula 05 - Introdução à Linguagem 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:

```
int matriz1 [ 3 ][ 3 ];
```

Linhas
Colunas

Índice	0	1	2
0	7	2	5
1	4	8	1
2	9	6	3

Obs.: os índices de uma matriz, em C, sempre começam em 0. Logo o matriz1 pode armazenar 9 inteiros, com controle de índice começando em [0][0], [0][1], [0][2], [1][0]... até [2][2].

	Memória	
1000	7	1 Linha
1001		
1002	2	
1003		
1004	5	2 Linha
1005		
1006	4	
1007		
1008	8	3 Linha
1009		
1010	1	
1011		
1012	9	3 Linha
1013		
1014	6	
1015		
1016	3	3 Linha
1017		
1018		

Matrizes

- Seja o exemplo de números reais:

```
float matriz2 [ 2 ][ 2 ];
```

→ Linhas
→ Colunas

Índice	0	1
0	3.8	2.7
1	9.1	6.0

Memória
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018

1 Linha

2 Linha

Obs.: A matriz2 armazena 4 floats, com controle de índices: [0][0], [0][1], [1][0], [1][1].

Matrizes

- Seja o exemplo de números inteiros:

```
int matriz1 [ 3 ][ 3 ];
```

Linhas - tamanho máximo
Colunas - tamanho máximo

Índice	0	1	2
0	7	2	5
1	4	8	1
2	9	6	3

Posição:

matriz1 [1][1] → 8

matriz1 [0][2] → 5

matriz1 [2][0] → 9

matriz1 [3][1] → ? → **ERRO**

matriz1 [2][3] → ? → **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;`

Ex: `int matriz1 [3][3];`

`matriz1[2][1] = 94;`

`matriz1[0][2] = 33;`

`matriz1[1][0] = 57;`

`matriz1[1][3] = 94;` → **ERRO**

Índice	0	1	2
0			33
1	57		
2		94	

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

- 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 = índice de linha
{
    for(int j = 0; j < C; j++) //j = índice de coluna
    {
        printf (" %d", matriz[ i ][ j ]);
    }
    printf ("\n"); //posicionar na próxima linha
}
```


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 = 0; coluna < numeroColunas; coluna++)
        {
            printf ("Digite matriz[%d][%d]: ", linha, coluna);
            scanf ("%d", &matriz[linha][coluna]);
        }
    }

    printf ("Matriz digitada: \n");
    for (int linha = 0; linha < numeroLinhas; linha++)
    {
        for(int coluna = 0; coluna < numeroColunas; coluna++)
        {
            printf (" %d", matriz[linha][coluna]);
        }
        printf ("\n"); //posicionar na próxima linha
    }

    return 0;
}
```

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 $M \times N$ e B; $N \times O$; então, a matriz $A \times B$ tem dimensões $M \times O$, e

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

Four examples of matrix multiplication are shown, illustrating the dot product of rows from matrix A and columns from matrix B to produce elements in the resulting matrix C.

- $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 58 \\ \end{bmatrix}$ $1 \times 7 + 2 \times 9 + 3 \times 11 = 58$
- $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 58 & 64 \end{bmatrix}$ $1 \times 8 + 2 \times 10 + 3 \times 12 = 64$
- $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 58 & 64 \\ 139 \end{bmatrix}$ $4 \times 7 + 5 \times 9 + 6 \times 11 = 139$
- $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 58 & 64 \\ 139 & 154 \end{bmatrix}$ $4 \times 8 + 5 \times 10 + 6 \times 12 = 154$

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 < O; j++) {
            scanf ("%d", &b[k][j]);
        }
    }
}
```

```
//calculo de c = a x b
for (i = 0 ; i < M ; i++)
{
    for (j = 0 ; j < O ; j++)
    {
        soma = 0;
        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 < O; j++)
        {
            printf (" %d", c[i][j]);
        }
        printf ("\n"); //posicionar na próxima linha
    }
}
return 0;
}
```