## Matrizes Multidimensionais

## Módulo 7 Aula 1

## Linguagem C, o Curso Definitivo WR Kits

Autor: Dr. Eng. Wagner Rambo

Quando trabalhamos com vetores em C, por definição também estamos trabalhando com matrizes unidimensionais. Inserindo mais dimensões, estaremos trabalhando com matrizes multidimensionais. Na Linguagem C não existe um limite de dimensões para uma matriz, ou seja, a mesma pode ter n dimensões. Para declarar uma matriz multidimensional, utilize a sintaxe do Box 1.

```
tipo nome_matrix[dim_1][dim_2] ... [dim_n]
```

Box 1 - Sintaxe para declaração de uma matriz com n dimensões.

Como pode ser visto, temos o tipo de dado (os dados já conhecidos do C), o nome da matriz (que seguirá as mesmas regras de declaração de variáveis) e o tamanho de cada dimensão entre colchetes. No Box 2, apresentamos a declaração de algumas matrizes de duas dimensões, algo que poderá ser utilizado em diversos projetos.

Box 2 - Exemplos de declaração de matrizes de duas dimensões.

No Box 3 apresentamos uma matriz multidimensional de 3 linhas e 4 colunas e sua respectiva inicialização.

				_		
15	22	9	1			
65	42	0	71			
7	29	33	62			
int mat	[3][4]	{ 6	5, 42,	9, 1}, 0, 71}, 33, 62} };	/* inicializando uma /* matriz de 3x4	*/ */

Box 3 - Inicializando uma matriz de inteiros, 3 linhas, 4 colunas.

Você pode lembrar da inicialização de vetores, basicamente teremos 3 vetores de 4 elementos em uma matriz bidimensional de 3 linhas e 4 colunas. Por este motivo, a inicialização se dá pela organização das linhas dentro de chaves.

De forma muito semelhante como fazemos com os vetores, podemos acessar qualquer elemento da matriz passando o respectivo índice, lembrando que tanto as linhas quanto as colunas iniciam com índice zero. Tomando como exemplo a matriz do Box 3, confira algumas ideias de acesso no Box 4.

Box 4 - Alguns exemplos de acesso e atualização dos elementos da matriz.

No Box 5 apresentamos um método de imprimir a matriz do Box 3 completa com auxílio de dois laços *for*.

```
register int i,j,col=0;
for(i=0;i<3;i++)
{
   for(j=0;j<4;j++)
      printf("%3d ", mat[i][j]);
   putchar('\n');
}
putchar('\n');</pre>
```

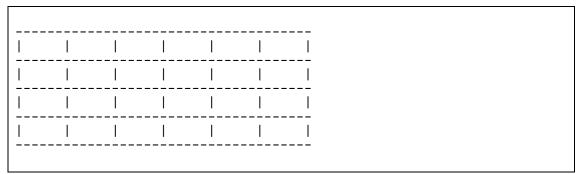
Box 5 - Imprimindo a matriz completa.

Tal como os vetores, os elementos das matrizes também podem ser utilizados em expressões e cálculos, na Box 6 atualizaremos a linha 0 coluna 0.

```
mat[0][0] = mat[1][1]+mat[2][2]; /* 42 + 33 */
printf("%d\n",mat[0][0]); /* imprimirá 75 */
```

Box 6 - Exemplo de cálculo com os elementos da matriz.

**Exercício proposto**: desenvolva o projeto de uma matriz de 4 linhas e 6 colunas, que apareça na tela com o design conforme Box 7.



Box 7 - Matriz de 4 linhas e 6 colunas.

Elabore uma interface onde o usuário digita o número da linha e número da coluna e um número inteiro de 0 a 999. A matriz vai sendo preenchida a medida que o usuário vai inserindo novas entradas. Importante: neste sistema as linhas e colunas devem começar com índice 1. E todos os conteúdos devem ser inicializados em 0. Se digitado um número de linha ou um número de coluna fora da faixa da matriz, simplesmente não atualiza nada.

Exemplo: o usuário entra com linha 2, coluna 4 e o número 42. Imprimirá:

 	0	0	0	0	0	0
I	0	0	0	42	0	0
Ī	0	0	0	0	0	0
1	0	0	0	0	0	0

Box 8 - Matriz de 4 linhas e 6 colunas.

Depois, o usuário entra com linha 4, coluna 1 e o número 187. Imprimirá:

Ī	0	C	)	0	0	0	I	0
Ī	0	C	)	0	42	0	I	0
Ī	0	C	)	0	0	0	I	0
-	187	C	)	0	0	0	I _	0

Box 9 - Matriz de 4 linhas e 6 colunas.

E assim por diante...

Bibliografia: DAMAS, Luís; Linguagem C, décima edição.

Disponível em: <a href="https://amzn.to/3nGdIbN">https://amzn.to/3nGdIbN</a>