

Curso: - Superior de Tecnologia em Análise e Desenvolvimento de Sistemas
- Superior de Tecnologia em Redes de Computadores

Disciplina: - Algoritmos e Técnicas de Programação
- Introdução à Programação

Professor: Fabiano Papaiz

ESTRUTURAS DE DADOS – PARTE II

Matrizes

São conhecidas também por estruturas de dados homogêneas multidimensionais.

A forma mais simples de matrizes multidimensionais é a matriz bidimensional (duas dimensões). Nela utilizamos 2 índices: um para indicar a linha e outro para indicar a coluna.

Veja um exemplo de matriz bidimensional 3X4 (3 linhas e 4 colunas):

		0	1	2	3	→ Índices das colunas
← Índices das linhas	0	5	8	11	4	
	1	7	21	87	44	
	2	9	0	42	55	

Neste exemplo, o elemento da posição [1,0] é o **7**. O da posição [0,1] é o **8**.

O primeiro índice representa a linha e o segundo a coluna. Observe que o índice das colunas também iniciam no 0 (zero).

Quais os valores dos elementos [1,1], [2,3], [0,2] e [2,0] ?

Podemos perceber que cada linha dessa matriz é um vetor com 4 elementos.

Em C, para definirmos a matriz do exemplo acima faríamos da seguinte forma:

```
//declara uma matriz de 3 linhas e 4 colunas
int matriz[3][4];

//atribui valores aos elementos da matriz
matriz[0][0] = 5; //1ª linha, 1ª coluna
matriz[0][1] = 8; //1ª linha, 2ª coluna
matriz[0][2] = 11; //1ª linha, 3ª coluna
matriz[0][3] = 4; //1ª linha, 4ª coluna

matriz[1][0] = 7; //2ª linha, 1ª coluna
matriz[1][1] = 21; //2ª linha, 2ª coluna
matriz[1][2] = 87; //2ª linha, 3ª coluna
matriz[1][3] = 44; //2ª linha, 4ª coluna
```

Neste código, não foram inicializados os valores da 3ª linha. Como faríamos isso?

Agora pense: Como faríamos para exibir todos os elementos dessa matriz?

Resposta:

1º. Temos que passar por todas as linhas; e

2º. A cada linha que passarmos, temos que exibir os valores de suas colunas

Em resumo, precisaremos de dois comandos FOR. Um para controlar as linhas e outro para controlar as colunas.

O código abaixo exibirá todos os elementos da nossa matriz de exemplo:

```
//passa por todas as 3 linhas
for (int linha=0; linha<3; linha++){
    //passa por todas as 4 colunas
    for (int coluna=0; coluna<4; coluna++){
        //imprime o valor do elemento [linha,coluna]
        printf("%d ", matriz[linha][coluna]);
    }
    //insere uma "quebra" a cada final de linha
    printf("\n");
}
```

A saída será:

```
5 8 11 4
7 21 87 44
9 0 42 55
```

Podemos atribuir valores para os elementos de uma matriz bidimensional de duas formas: acessando cada elemento pelo seu par de índices (*linha* e *coluna*) ou utilizar um conjunto de valores entre `{ {}, {} ... {} }`. No exemplo a seguir, as matrizes `m1` e `m2` são equivalentes:

```
int m1[2][2];
m1[0][0] = 20;
m1[0][1] = 30;
m1[1][0] = 40;
m1[1][1] = 50;

int m2[2][2] = { {20, 30}, {40, 50} };
// 1ª linha // 2ª linha
```

Vimos aqui somente as matrizes bidimensionais, que são as mais utilizadas. Mas podemos declarar matrizes de várias dimensões. Veja alguns exemplos:

```
int tres_dimensoes[3][2][5]; // 3 dimensões
float cinco_dimensoes[3][2][5][8][20]; // 5 dimensões
```

Exercícios:

1. Considerando a matriz **M** abaixo:

	0	1	2	3	4
0	5	2	1	1	3
1	7	4	7	3	2
2	4	5	2	2	1
3	3	3	1	4	1
4	8	4	2	9	3

E as variáveis $X=1$ e $Y=2$, escreva o valor correspondente à solicitação:

- $M[X, Y]$
 - $M[Y, X]$
 - $M[X + 1, Y - 1]$
 - $M[X + 2, Y - 1 * 2]$
 - $M[Y - X, Y + X]$
2. Desenhe a matriz resultante do código abaixo:

```
char mat[4][3];  
  
mat[3][2] = '!';  
mat[0][0] = 'c';  
mat[1][2] = 't';  
mat[0][2] = 'r';  
mat[2][0] = 'h';  
mat[1][0] = 'i';  
mat[3][0] = 'n';  
mat[0][1] = 'o';  
mat[2][1] = 'i';  
mat[1][1] = 'n';  
mat[2][2] = 'a';  
mat[3][1] = 's';
```

3. Faça um algoritmo que preencha uma matriz 5x5 de inteiros e escreva:
- A soma dos números ímpares fornecidos;
 - A soma de cada uma das 5 colunas;
 - A soma de cada uma das 5 linhas;