



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Currais Novos

LINGUAGEM C++

ESTRUTURAS DE DADOS HOMOGÊNEAS
PARTE II - MATRIZES

Prof. Bruno Gomes

(bruno.gurgel@ifrn.edu.br)

Currais Novos,

2011

INTRODUÇÃO

- Na aula anterior, definimos a estrutura vetor
 - Variável composta por elementos de um mesmo tipo
 - Estrutura linear – os elementos do vetor podem ser representada como uma linha, um do lado do outro.
- Iremos ver nesta aula as matrizes
 - Semelhança com o tipo vetor
 - Elementos do mesmo tipo
 - Matrizes possuem um índice a mais, ou seja, duas dimensões (linhas e colunas)
 - Representação dos elementos como uma tabela



INTRODUÇÃO

- Alguns problemas adequados para o uso de matrizes (estrutura em linhas e colunas)
 1. A representação de objetos em um espaço bidimensional (eixos x e y);
 2. Tabuleiros de jogos como xadrez ou damas;
 3. Matrizes da matemática;
 4. Um bilhete de um jogo de loteria, dentre outros.



DECLARAÇÃO DE MATRIZES

<tipo> <identificador>[<tam_linha>][<tam_col>]

Onde:

<identificador> - Nome da variável matriz;

<tam_linha> - Quantidade de elementos da linha.
- definida na declaração e não pode ser modificada posteriormente no programa;

<tam_col> - Quantidade de elementos da coluna.
- definida na declaração e não pode ser modificada posteriormente no programa.

<tipo> - Qualquer tipo de dados válido na linguagem C++;



EXEMPLOS

- **float** mat1[10][10];
- **int** x[15][15], y[20][20];
- **bool** achou[5][5];
- **char** nomes[15][15] ;



EXEMPLO

- *float algoritmos [2][10];*
 - Uma matriz que representa as notas de 2 turmas de algoritmos com no máximo 10 alunos em cada turma

Representação gráfica (uma tabela 2 por 10):

	0	1	2	3	4	5	6	7	8	9
0										
1										



OPERAÇÕES COM MATRIZES - ACESSO

- Assim como para vetores, não podemos operar uma matriz diretamente apenas pelo seu nome.
 - Por exemplo:
 $\text{pares}[2][2] = 4$ (CORRETO!)
 $\text{pares} = 8$ (ERRADO!).
- As operações devem ser feitas com cada par *linha-coluna* da matriz,
 - utilizando o índice da linha e coluna correspondente
 - Exemplos:
 $\text{pares}[0][0] = 2$ (CERTO!)
 $\text{pares}[0][1] = 4$ (CERTO!)
 $\text{pares}[1][0] = 2 + 4$ (CERTO!)
 $\text{pares}[1][1] = \text{pares}[0][1] + 5$ (CERTO!)



ACESSO AOS ELEMENTOS DA MATRIZ

OBS. 1: Qualquer expressão que resulte em um inteiro positivo pode ser utilizada como índice

OBS. 2: É um erro tentar acessar um elemento que esteja fora da faixa reservada para a matriz.

○ Exemplos:

- $x[i][j]$ – índices representados pelos inteiros i e j .
- $x[2][3]$ – Acesso ao elemento da linha 3 e da coluna 4 de uma matriz.
- $mat[i + 1][j - 1]$ – O índice para acesso aos elementos de uma matriz pode ser qualquer expressão inteira entre 0 e o número máximo de elementos $- 1$ (menos 1),
 - Tanto para a linha quanto para a coluna.



ATRIBUIÇÃO DE VALORES

- Quando se declara uma nova matriz, os seus elementos são indeterminados ou vazios
 - Deve-se inicializar cada elemento da matriz antes da sua utilização
- Podemos inicializar a matriz por *atribuição* (operador =) ou por *leitura* (instrução cin).
- Na atribuição, como vimos anteriormente, não podemos utilizar o *nome da matriz* diretamente
 - Referência a cada posição da matriz (linha e coluna).



INICIALIZAÇÃO POR ATRIBUIÇÃO

- Para colocarmos elementos em uma matriz, normalmente utilizamos duas estruturas de repetição, uma dentro da outra
 - A primeira percorre cada linha;
 - A segunda percorre cada elemento da coluna para aquela linha.

- Exemplo:

```
for (int i = 0; i < 2; i++) {  
    for (int j = 0; j < 10; i++) {  
        algoritmos[i][j] = 0;  
    }  
}
```



EXEMPLO

- Inicialização da matriz **x[10, 10] : Inteiro.**
- No caso, a inicialização abaixo atribui o valor 1 quando o índice da linha for igual ao índice da coluna e 0 caso contrário.
 - Qual o nome dessa matriz na matemática, quando a matriz é quadrada????

```
for (int i = 0; i < 10; i++) {  
    for (int j = 0; j < 10; i++) {  
        if (i == j) {  
            x[i][j] = 1;  
        }  
        x[i][j] = 0;  
    }  
}
```



LEITURA DE ELEMENTOS PARA A MATRIZ

- A leitura dos elementos de uma matriz deve ser feita elemento por elemento
 - percorre-se todas as colunas para cada linha
- No exemplo abaixo, vemos a leitura de dados para uma matriz chamada “tabela”.

```
for (int i = 0; i < 4; i++) {  
    for (int j = 0; j < 6; i++) {  
        cin >> tabela [i][j];  
    }  
}
```



EXEMPLO – LENDO APENAS ELEMENTOS POSITIVOS PARA UMA MATRIZ

```
cout << "Digite números positivos: ";
for (int i = 0; i <= p; i++) {
    for (int j = 0; j <= p; i++) {
        do {
            cout << "Atenção! Número positivo: ";
            cin >> x;
        } while (x > 0);

        matriz[a][b] = x;
    }
}
```



ESCRITA

- A escrita de matrizes também é feita elemento por elemento, de forma semelhante à leitura
 - Utilizamos a instrução **cout**
- Exemplo:

```
for (int i = 0; i < 4; i++) {  
    for (int j = 0; j < 6; i++) {  
        cout << tabela [i][j];  
    }  
}
```



EXERCÍCIOS

1. Calcular a soma dos elementos de uma matriz numérica de tamanho qualquer ($n \times m$) que estão na diagonal principal.
2. A transposta de uma matriz é obtida quando os elementos de cada linha da matriz tornam-se elementos da coluna. Ou, formalmente:

$$m^T_{ij} = m_{ji}, \text{ onde } m \text{ são os elementos de uma matriz.}$$

Faça um algoritmo que leia uma matriz 3x3 e retorne a sua transposta.

