Programação de Computadores - CI-208

Prof. Murilo da Silva - DINF/UFPR

Aula 12

Matrizes em C++

Matrizes em C++

- É um conhecido conceito matemático:
  - uma matriz é um arranjo retangular (ou tabela) de números, símbolos ou expressões, organizados em linhas e colunas.
  - Exemplo: a dimensão da matriz M abaixo de 2 × 3, ou seja: existem duas linhas e três colunas

$$\begin{bmatrix} 4.3 & 7.0 & -1.2 \\ -3.0 & 2.2 & 19.1 \end{bmatrix}$$

Matrizes em C++

- É um conhecido conceito matemático:
  - uma matriz é um arranjo retangular (ou tabela) de números, símbolos ou expressões, organizados em linhas e colunas.
  - Exemplo: a dimensão da matriz M abaixo de 2 × 3, ou seja: existem duas linhas e três colunas

$$\begin{bmatrix} 4.3 & 7.0 & -1.2 \\ -3.0 & 2.2 & 19.1 \end{bmatrix}$$

Declaração de Matrizes em C/C++

- A sintaxe para uma matriz com 2 dimensões:

<tipo> nome_da_matriz [<nLinhas>] [<nColunas>]

Declaração de Matrizes em C++

Declaração de Matrizes em C++

<tipo> nome_da_matriz [<nLinhas>] [<nColunas>]

Declaração de Matrizes em C++

<tipo> nome_da_matriz [<nLinhas>] [<nColunas>]

- Exemplo:

  float **notas**[4][3];

Declaração de Matrizes em C++

<tipo> nome_da_matriz [<nLinhas>] [<nColunas>]

- Exemplo:

  float **notas**[4][3];

- A matriz **notas** possui 4*3 = 12 variáveis do tipo float, organizadas em:
  - 4 linhas, indexadas de 0 a 3 (0 a nLinhas-1)
  - 3 colunas, indexadas de 0 a 2 (0 a nColunas-1)

Declaração de Matrizes em C++

- Na declaração de uma matriz geralmente são usadas constantes para definir as dimensões máximas da matriz #define

```
#define NLIN   4
#define NCOL  3
...
float notas[ NLIN ][ NCOL ];
```

Declaração de Matrizes em C++

- Na declaração de uma matriz geralmente são usadas constantes para definir as dimensões máximas da matriz #define

```
#define NLIN   4
#define NCOL  3
...
float notas[ NLIN ][ NCOL ];
```

- Dependendo do problema, você pode caracterizar as dimensões usando nomes mais apropriados para estas constantes

```
#define NALUNOS   30      // quantidade de alunos na turma
#define NPROVAS  3       // quantidade de provas realizadas
float notas[ NALUNOS ][ NPROVAS ];
```

Declaração com inicialização de Matrizes

Declaração com inicialização de Matrizes

Considere as duas maneiras de declarar e inicializar matrizes:

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

Declaração com inicialização de Matrizes

Considere as duas maneiras de declarar e inicializar matrizes:

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

- Em ambos os casos é criada a matriz:

1 2 3
4 5 6
7 8 9

Os índices de M são:

M[0][0]	M[0][1]	M[0][2]
M[1][0]	M[1][1]	M[1][2]
M[2][0]	M[2][1]	M[2][2]

Trabalhando com Matrizes

- Considere a declaração da matriz M de inteiros:

```
#define NLIN    4
#define NCOL   3
...
int M[ NLIN ][ NCOL ];
```

Trabalhando com Matrizes

- Considere a declaração da matriz M de inteiros:

```
#define NLIN    4
#define NCOL   3
...
int M[ NLIN ][ NCOL ];
```

- Usualmente a leitura de valores para a matriz se faz assim:

```
for (int l=0; l < NLIN; l++)
    for (int c=0; c < NCOL; c++)
        cin >> M[l][c];
```

Exercícios

(1) Crie um programa que leia uma matriz de 3 linhas e 4 colunas e imprima a matriz transposta da matriz lida.

(2) Crie um programa que leia uma matriz de 4 linhas e 4 colunas e imprima os elementos da diagonal principal da matriz lida.

(3) Crie um programa que leia uma matriz de 4 linhas e 4 colunas e imprima os elementos da diagonal secundária da matriz lida.

(4) Crie um programa que leia duas matrizes A e B, ambas de 3 linhas e 3 colunas e imprima a matriz C = A + B.

(5) Crie um programa que leia duas matrizes A e B, ambas de 3 linhas e 3 colunas e imprima a matriz C = A * B.