MC-102 — Aula 10
Matrizes

Instituto de Computação – Unicamp

Primeiro Semestre de 2006

Roteiro

Vetores

Na última aula, criamos um programa que lia as notas de uma prova para um conjunto de alunos e então calculava a média da turma. Para resolver este problema, utilizamos vetores.

Reveja o código: `notas.c`

Vetores

Agora queremos ler as notas de 4 provas para cada aluno e então calcular a média do aluno e a média da classe. O tamanho máximo da turma é de 50 alunos.

solução

Criar 4 vetores cada um com 50 posições. E então ler as respectivas informações.

```
float nota0[50],nota1[50],nota2[50],nota3[50];
```

Matrizes

- Agora suponha que estamos trabalhando com no máximo 100 provas e 100 alunos. Seria muito cansativo criar 100 vetores e atribuir 100 nomes diferentes. (Parece que esse problema não tem fim !!!).
- Para resolver esse problema podemos utilizar matrizes. Uma matriz é um vetor (ou seja, um conjunto de variáveis de mesmo tipo) que possui duas ou mais dimensões, resolvendo para sempre essa questão.

Declarando uma matriz

```
<tipo> nome_da_matriz [<linhas>] [<colunas>]
```

- Uma matriz possui *linhas* × *colunas* variáveis do tipo `<tipo>`
- As linhas são numeradas de 0 a *linhas* − 1
- As colunas são numeradas de 0 a *colunas* − 1

Exemplo de declaração de matriz

```
int matriz [4][4];
```

	0	1	2	3
0				
1				
2				
3				

Declarando uma matriz de múltiplas dimensões

$$\texttt{<tipo>}\ \texttt{nome\_da\_matriz}\ [<\ dim_1\ >]\ [<\ dim_2\ >]\ \ldots\ [<\ dim_N\ >]$$

- Essa matriz possui $dim_1 \times dim_2 \times \cdots \times \text{dim}_N$ variáveis do tipo <tipo>
- Cada dimensão é numerada de 0 a $dim_i - 1$

Acessando uma matriz

- Em qualquer lugar onde você escreveria uma variável no seu programa, você pode usar um elemento de sua matriz, da seguinte forma:

```
nome_da_matriz [<linha>] [<coluna>]
```

Ex: `matriz [1][10]` — Refere-se a variável na 2^a linha e na 11^a coluna da matriz.

- O compilador não verifica se você utilizou valores válidos para a linha e para a coluna.

Lendo uma matriz do teclado

```
/*Leitura*/
for (i = 0; i < 4; i++)
  for (j = 0; j < 4; j++) {
    printf ("Matriz[%d][%d]: ", i, j);
    scanf ("%d", &matriz[i][j]);
  }
```

Escrevendo uma matriz na tela

```
/*Escrita*/
for (i = 0; i < 4; i++) {
  for (j = 0; j < 4; j++)
    printf ("%3d ", matriz[i][j]);
  printf ("\n");
}
```

Veja o exemplo para os três últimos slides em leitura.c

Exercícios

- Escreva um programa que inicialize uma matriz 10×10 com 0s em todas as posições. O usuário irá digitar o índice da linha e o índice da coluna e em seguida o valor das posições não nulas. A leitura será feita enquanto os índices forem não negativos. Após a leitura imprima a matriz na tela.

$$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \dots & & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

```
0 0 -1
1 2 2
9 9 1
-1 -1
```

Exercícios

- Escreva um programa que leia todas as posições de uma matriz 10×10. Em seguida, mostra o índice da linha e o índice da coluna e o valor das posições não nulas. No final, exibe o número de posições não nulas.

$$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \dots & & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

0 0 -1
1 2 2
9 9 1
3 elementos não nulos

Exercícios

- Escreva um programa que lê todos os elementos de uma matriz 4×4 e mostra a matriz e a sua transposta na tela.

Matriz

$$\begin{bmatrix} 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 2 \end{bmatrix}$$

Transposta

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 \end{bmatrix}$$

Exercícios

- Escreva um programa que lê 2 matrizes 5×5, mostre-as na tela e mostre a soma entre as duas matrizes em seguida.

$$\underset{A}{\begin{bmatrix} 0 & 1 & 0 & 2 & 3 \\ 0 & 1 & 0 & 2 & 3 \\ 0 & 1 & 0 & 2 & 3 \\ 0 & 1 & 0 & 2 & 3 \\ 0 & 1 & 0 & 2 & 3 \end{bmatrix}} + \underset{B}{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \end{bmatrix}} = \underset{C}{\begin{bmatrix} 0 & 1 & 0 & 4 & 3 \\ 1 & 2 & 1 & 3 & 4 \\ 0 & 1 & 0 & 2 & 3 \\ 2 & 3 & 2 & 4 & 5 \\ 3 & 4 & 3 & 5 & 6 \end{bmatrix}}$$

Exercícios

- Escreva um programa que lê 2 matrizes 5×5, mostre-as na tela e então calcule o produto entre as duas matrizes, mostrando-o em seguida.

$$\underset{A}{\begin{bmatrix} 0 & 1 & 0 & 2 & 3 \\ 0 & 1 & 0 & 2 & 3 \\ 0 & 1 & 0 & 2 & 3 \\ 0 & 1 & 0 & 2 & 3 \\ 0 & 1 & 0 & 2 & 3 \end{bmatrix}} * \underset{B}{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \end{bmatrix}} = \underset{C}{\begin{bmatrix} 14 & 14 & 14 & 14 & 14 \\ 14 & 14 & 14 & 14 & 14 \\ 14 & 14 & 14 & 14 & 14 \\ 14 & 14 & 14 & 14 & 14 \\ 14 & 14 & 14 & 14 & 14 \end{bmatrix}}$$

Exercícios

- Escreva um programa lê uma matriz e depois verifica se esta é uma matriz triangular inferior.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 2 & 3 \end{bmatrix}$$