



Treinamento para a Olimpíada Brasileira de Informática

Exercícios envolvendo matrizes

Prof. Ciro Cirne Trindade



Matrizes (1/2)

- Alguns exercício exigem que os dados sejam armazenados em matrizes
 - Tipo estruturado homogêneo semelhante a uma tabela
 - Os elementos são referenciados por seus índices de linha e coluna

Matrizes (2/2)

- A linguagem C permite matrizes de qualquer tipo, incluindo matrizes com mais de duas dimensões
- Com 2 pares de colchetes obtemos uma matriz de 2 dimensões e p/ cada par de colchetes adicionais obtemos uma matriz com uma dimensão a mais:

```
tipo nome-da-variável[tamanho 1][tamanho 2]...  
[tamanho n];
```

Matrizes bidimensionais

- Para declarar uma matriz bidimensional devemos usar 2 pares de colchetes
 - No **1º** par de colchetes definimos o número de **linhas** da matriz
 - No **2º** par de colchetes definimos o número de **colunas** da matriz
- Exemplo:
 - `int matriz[4][6];`

Declara uma matriz de inteiros com 4 linha e 6 colunas



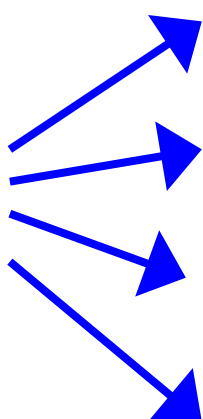
Acessando os elementos da matriz (1/3)

- Para acessar um elemento específico da matriz devemos indicar a linha e coluna desse elemento
 - Semelhante a referência a uma célula de uma planilha eletrônica
- Usamos índices de linha e coluna para referenciar um elemento específico da matriz

Acessando os elementos da matriz (2/3)

- Exemplo de uma matriz 4x6

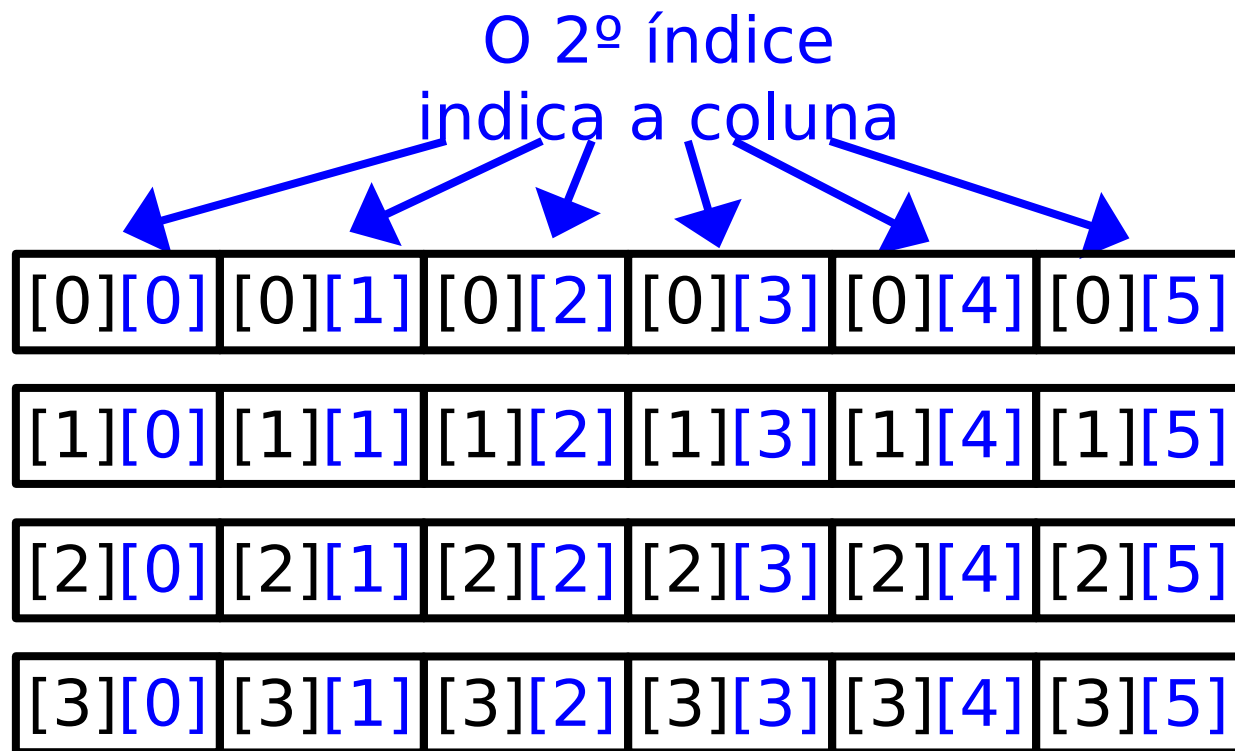
O 1º índice indica a linha



[0][0]	[0][1]	[0][2]	[0][3]	[0][4]	[0][5]
[1][0]	[1][1]	[1][2]	[1][3]	[1][4]	[1][5]
[2][0]	[2][1]	[2][2]	[2][3]	[2][4]	[2][5]
[3][0]	[3][1]	[3][2]	[3][3]	[3][4]	[3][5]

Acessando os elementos da matriz (3/3)

- Exemplo de uma matriz 4x6





Percorrendo a matriz

- Para percorrer uma matriz normalmente utiliza-se dois laços `for` encaixados
- As variáveis de controle dos laços são utilizadas como índices da matriz



Exemplo da leitura e impressão de uma matriz

```
#include <stdio.h>
int main() {
    int matriz[4][6], i, j;
    printf("Informe os elementos da matriz 4x6:\n");
    for (i = 0; i < 4; i++) {
        for (j = 0; j < 6; j++) {
            scanf("%d", &matriz[i][j]);
        }
    }
    printf("Conteudo da matriz:\n");
    for (i = 0; i < 4; i++) {
        for (j = 0; j < 6; j++) {
            printf("%d\t", matriz[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

Iniciando matrizes

- As matrizes são inicializadas da mesma maneira que os vetores

```
int mat[4][6] = { { 0, 0, 0, 0, 1, 1 },  
                  { 0, 1, 1, 1, 1, 0 },  
                  { 0, 0, 0, 0, 1, 1 },  
                  { 1, 0, 1, 1, 0, 0 } };
```

- Uma matriz pode ser vista como um vetor onde seus elementos são vetores
- Para inicializar uma matriz com zeros, faça:

```
int mat[4][6] = { 0 };
```

Inicializadores designados

- Inicializadores designados também podem ser usados em matrizes
- Por exemplo:
 - `int mat[3][4] = { [1] = { [2] = 10 } };`

Inicializa o elemento
`mat[1][2]` da matriz
com 10



Exercícios

- Caçadores de Mitos: OBI2009 – 1ª Fase – Modalidade Programação, Nível 2
 - http://olimpiada.ic.unicamp.br/pratique/programacao/nivel2/2009f1p2_mito
- O Mar não está para peixe: OBI2011 – 1ª Fase – Modalidade Programação, Nível 2
 - http://olimpiada.ic.unicamp.br/pratique/programacao/nivel2/2011f1p2_pesca
- Quadrado Mágico: OBI2011 – 2ª Fase – Modalidade Programação, Nível 2
 - http://olimpiada.ic.unicamp.br/pratique/programacao/nivel2/2011f2p2_magico



Referências

- UNICAMP. *Olimpíada Brasileira de Informática*. Disponível em:
<<http://http://olimpiada.ic.unicamp.br>>.