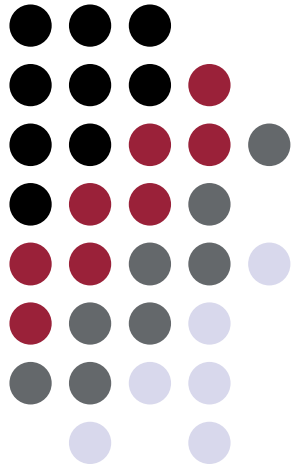
A09 Vetores Multidimensionais

CSI030 – Programação de Computadores I

Prof. Dr. George H. G. Fonseca
Universidade Federal de Ouro Preto



Motivação

- Suponha que queremos ler uma string de até 30 caracteres. Como sabemos, uma string em C é um vetor de caracteres

```
char string[30];
```

- E se quisermos ler 3 strings de 30 caracteres?

```
char string1[30];
char string2[30];
char string3[30];
```

- E se quisermos ler 300 string de 30 caracteres?

Definição

- Uma matriz pode ser utilizada para armazenar 300 strings de 30 caracteres cada

- Uma matriz (ou vetor multidimensional) é um vetor com mais de uma dimensão

- O total de variáveis que a matriz possui é igual ao produto entre a quantidade de variáveis de cada uma de suas dimensões

Matriz Bidimensional
Declaração

- Uma matriz de duas dimensões pode ser declarada:

```
tipo nome[dim1][dim2];
```

onde dim1 e dim2 são números ou variáveis inteiras

- Essa declaração cria dim1 x dim2 variáveis do tipo tipo

- As variáveis criadas pelo vetor são acessadas por:

```
nome[0][0];    nome[1][0];    nome[dim1 - 1][dim2 - 2];
nome[0][1];    nome[1][1];
...            ...
```

Matriz na memória

```
int m[3][3];
```

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]

Preenchimento de Matriz

- Podemos utilizar laços encaixados para preencher matrizes, sendo um laço para cada repetição

- Preencher a matriz: `int m[3][3];` com 1's:

```
for(i = 0; i < 3; ++i) {
    for(j = 0; j < 3; ++j) {
        m[i][j] = 1;
    }
}
```

Matriz
Inicialização

- Assim como vetores unidimentsionais, matrizes podem ser inicializadas junto à declaração

```
int vetor2D[3][4] =
    {{1, 2, 3, 4}, {1, 2, 3, 4}, {1, 2, 3, 4}};
```

```
int vetor3D[2][3][4] =
    {{{1, 2, 3, 4}, {1, 2, 3, 4}, {1, 2, 3, 4}},
    {{1, 2, 3, 4}, {1, 2, 3, 4}, {1, 2, 3, 4}}};
```

Matriz

Preenchimento e Impresão

```
int main(){
    int matriz[5][7], i, j, valor = 0;
    //Preenchimento de matriz
    for(i = 0; i < 5; ++i) {
        for(j = 0; j < 7; ++j) {
            matriz[i][j] = valor;
            ++valor;
        }
    }
    //Impressão de matriz
    for(i = 0; i < 5; ++i) {
        for(j = 0; j < 7; ++j) {
            printf(" %d ", matriz[i][j]);
        }
        printf("\n");
    }
}
```

Matriz
Bidimensional de Caracteres

- Matriz para armazenar 300 strings de 30 caracteres cada

```
char strings[300][30];
```

- Preenchimento e impressão

```
for(i = 0; i < 300; i++)
    scanf("%s", strings[i]);

for(i = 0; i < 300; i++)
    printf("%s", strings[i]);
```

Matriz
Multidimensional não-dimensionada

- Em matrizes multidimensionais, podemos omitir a **primeira** dimensão

```
int matriz[][4] =
    {{1, 2, 3, 4}, {1, 2, 3, 4}, {1, 2, 3, 4}};
```

- O mesmo vale para assinaturas de sub-rotinas

```
void subrotina(int matriz[][4], int qtde_linhas);
```

Matriz

Como parâmetro de sub-rotina

- Assim como nos vetores unidimensionais, o conteúdo de vetores multidimensionais pode ser alterado em sub-rotinas

```
void zeraMatriz(int matriz[2][2]) {
     int i, j;
     for(i = 0; i < 2; i++) {
          for(j = 0; j < 2; j++) {
               matriz[i][j] = 0;
          }
     }
}

int main(){
     int mat[2][2] = {{0, 1}, {2, 3}};
     zeraMatriz(mat);
     return 0;
}
```

Exercício

- Escreva um programa que leia todas as posições de uma matriz 3 × 3. O programa deve, em seguida, exibir o número de posições iguais a zero da matriz.

Exercício

```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int M[3][3], i, j, contZeros = 0;
    printf("Informe os valores de uma matriz 3x3\n");
    for(i = 0; i < 3; ++i) {
        for(j = 0; j < 3; ++j) {
            scanf("%d", &M[i][j]);
            if(M[i][j] == 0) {
                ++contZeros;
            }
        }
    }
    printf("Zeros: %d\n", contZeros);
    return 0;
}
```

Exercício

- Escreva um programa que leia uma matriz 3x3 e exiba a matriz transposta desta matriz.

Exercício

```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int mat[3][3], i, j, matT[3][3];
    printf("Informe os valores de uma matriz 3x3:\n");
    for(i = 0; i < 3; ++i) {
        for(j = 0; j < 3; ++j) {
            scanf("%d", &mat[i][j]);
        }
    }
    for(i = 0; i < 3; ++i) {
        for(j = 0; j < 3; ++j) {
            matT[j][i] = mat[i][j];
        }
    }
    printf("Transposta:\n");
    for(i = 0; i < 3; ++i) {
        for(j = 0; j < 3; ++j) {
            printf(" %d ", matT[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

Bibliografia

- Anido, R., Notas de aula. UNICAMP, disponível em http://www.ic.unicamp.br/~ranido/mc102/, acessado em Maio de 2015.

- Mota, V. F., Notas de aula. UFMG. Disponível em https://sites.google.com/site/virginiaferm/home/disciplinas acessado em Maio de 2015.

- Deitel, P, Deitel, H. C How to Program. 6a Ed. Pearson, 2010.