

Aula 9: Matrizes

Professor(a): João Eduardo Montandon (103)

Virgínia Fernandes Mota (106)

jemaf.github.io

<http://www.dcc.ufmg.br/~virginiaferm>

INTRODUÇÃO A PROGRAMAÇÃO - SETOR DE INFORMÁTICA



- Assim como os vetores, as matrizes são estruturas de dados homogêneas. Podem ser construídas dos diversos tipos básicos primitivos (real, inteiro, caractere).
- Principal diferença em relação aos vetores (unidimensionais): possui uma ou mais dimensões adicionais.
- Maioria dos casos: utiliza-se matrizes bidimensionais.

- São utilizadas quando os dados homogêneos necessitam de uma estruturação com mais de uma dimensão.
- Exemplos:
 - Programar um jogo de xadrez (o tabuleiro é naturalmente bidimensional).
 - Estrutura para guardar caracteres de um livro (três dimensões: 2 para representar os caracteres de uma página e uma terceira para indicar as páginas).
 - Problemas matemáticos matriciais.
 - Processamento de imagens: Uma imagem pode ser vista como uma matriz de pixels.

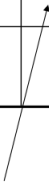
- A sintaxe para declaração de uma variável deste tipo é semelhante a declaração dos vetores. Considera-se porém a quantidade de elementos da outra dimensão:
tipo identificador[qtde_linhas][qtde_colunas];
- Nesse caso, os índices variam de 0 até $(qtde_linhas-1)$ para as linhas e 0 até $(qtde_colunas-1)$ para as colunas.
- Exemplos:
 - `int matriz[3][4];` //Matriz de 2 dimensões com elementos do tipo int
 - `float matriz_real[4][4][6];` //Matriz de 3 dimensões com elementos do tipo float
 - `char matriz_char[3][3];` //Matriz de 2 dimensões com elementos do tipo char

- Caso geral para declaração de uma matriz é:
tipo identificador[dim1][dim2]...[dimn];
- Assim temos uma matriz n-dimensional (n dimensões).
- Exemplo:
 - `int matriz[3][4][5][3];`
 - Os índices variam de 0 a 2 para a primeira dimensão, 0 a 3 para a segunda dimensão, 0 a 4 para a terceira dimensão e 0 a 2 para a quarta dimensão.

Matrizes: Declaração

Representação: Matriz 3x4

	0	1	2	3
0	0 0	0 1	0 2	0 3
1	1 0	1 1	1 2	1 3
2	2 0	2 1	2 2	2 3

 **mat[1][2]**

- A atribuição a um valor na matriz é feito explicitando a posição da matriz em que o valor deverá ser atribuído.
- Exemplo:
float NUM[2][2];
NUM[0][0] = 3.6;
NUM[0][1] = 0.0;
NUM[1][0] = 4.7;
NUM[1][1] = 1.5;

- Os elementos das matrizes são referenciados individualmente por meio de índices (iniciando de zero) entre colchetes.
- Seja a matriz do exemplo anterior:
float A, B;
A = NUM[0][0];
B = NUM[1][0];
- Assim, A passa a ter o valor 3.6 e B o valor 4.7.

Matrizes: Exemplo

O programa a seguir, inicializa com zeros os elementos de uma matriz inteira **n** de 5 linhas e 4 colunas e imprime.

```
1 #include <stdio.h>
2 int main() {
3     int n[5][4], i, j;
4     for (i = 0; i < 5; i++)
5         for(j = 0; j < 4; j++)
6             n[i][j] = 0;
7     printf("Matriz \n");
8     for (i = 0; i < 5; i++){
9         printf("\n Linha %2d \n", i);
10        for(j = 0; j < 4; j++)
11            printf("%d ", n[i][j]);
12    }
13    return 0;
14 }
```

Matrizes: Exemplo

O programa abaixo inicializa os elementos de uma matriz **m** com os valores iguais a soma dos índices de cada elemento e imprime cada valor.

```
1 #include <stdio.h>
2 int main() {
3     int m[3][2], i, j;
4     for (i=0; i < 3; i++)
5         for(j=0; j < 2; j++) {
6             m[i][j] = i+j;
7             printf("i=%d j=%d elemento=%d \n", i, j, m[i][j]);
8         }
9     return 0;
10 }
```

- Matrizes serão passadas para subrotinas da mesma forma como vetores.
- O valor das dimensões pode ser informado ou omitido.
- Exemplo: Um procedimento para imprimir uma matriz teria a seguinte declaração:
imprimeMatriz(float mat[][3], int lin, int col)
ou
imprimeMatriz(float mat[3][3], int lin, int col)
- A declaração em C segue a mesma regra da pseudolinguagem, mas apenas a primeira dimensão pode ser omitida.
- Havendo mais dimensões, a mesma regra deverá ser seguida.

Exemplo

Criar uma função que receba uma matriz 2×3 de números reais e retorne a média dos valores da matriz. Crie uma função principal que chame a subrotina e imprima a média.

Exemplo

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 float mediaMatriz(float m[2][3]){
4     int i, j;
5     float media = 0;
6     for( i = 0; i < 2; i++)
7         for(j = 0; j < 3; j++)
8             media+=m[i][j];
9     return media/6.0;
10 }
11
12 int main() {
13     float mat[2][3] = {{3.4, 5.6, 4.0},{2.0, 1.1, 4.9}};
14     float media = mediaMatriz(mat);
15     printf("A media da matriz foi %.2f", media);
16     return 0;
17 }
```

Exemplo

Faça um programa para exibir a soma de duas matrizes quadradas 3×3 . Deverá ser criado um procedimento para ler uma matriz (será chamado duas vezes com parâmetros diferentes) e um segundo procedimento que irá imprimir a soma das matrizes passadas como parâmetro.

Exemplo

```
1 #include <stdio.h>
2 void leMatriz(int m[3][3]){
3     int i, j;
4     for(i = 0; i < 3; i++)
5         for(j = 0; j < 3; j++)
6             scanf("%d", &m[i][j]);
7 }
8
9 void somaMatriz(int m1[3][3], int m2[3][3]){
10     int soma[3][3], i, j;
11     for(i = 0; i < 3; i++)
12         for(j = 0; j < 3; j++)
13             soma[i][j] = m1[i][j] + m2[i][j];
14     printf("A soma das matrizes eh: ");
15     for(i = 0; i < 3; i++){
16         printf("\n");
17         for(j = 0; j < 3; j++)
18             printf("%d ", soma[i][j]);
19     }
20 }
21
22 int main(){
23     int m1[3][3], m2[3][3];
24     printf("Digite sua primeira matriz");
25     leMatriz(m1);
26     printf("Digite sua segunda matriz");
27     leMatriz(m2);
28     somaMatriz(m1,m2);
29     return 0;
30 }
```

- A alocação dinâmica de memória para matrizes é realizada da mesma forma que para vetores, com a diferença que teremos um ponteiro apontando para outro ponteiro que aponta para o valor final (indireção múltipla).
- Um exemplo de implementação para matriz real bidimensional $m \times n$ é fornecido a seguir. A estrutura de dados utilizada neste exemplo é composta por um vetor de ponteiros (correspondendo ao primeiro índice da matriz), sendo que cada ponteiro aponta para o início de uma linha da matriz. Em cada linha existe um vetor alocado dinamicamente (compondo o segundo índice da matriz).

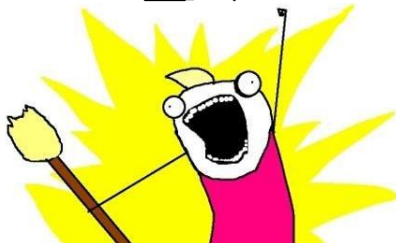
Alocação dinâmica de matrizes

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main() {
4     int m, n, i;
5     float **matriz;
6     printf("\n Digite o numero de linhas e colunas: ");
7     scanf("%d %d", &m, &n);
8     /* verifica parametros recebidos */
9     if (m < 1 || n < 1) {
10         printf ("** Erro: Parametro invalido **\n");
11         exit(0);
12     }
13     /* aloca as linhas da matriz */
14     matriz = (float **) malloc (m*sizeof(float *));
15     if (matriz == NULL){
16         printf ("** Erro: Memoria Insuficiente **");
17         exit(0);
18     }
19     /* aloca as colunas da matriz */
20     for ( i = 0; i < m; i++) {
21         matriz[i] = (float*) malloc (n * sizeof(float));
22         if (matriz[i] == NULL){
23             printf ("** Erro: Memoria Insuficiente **");
24             exit(0);
25         }
26     }
27     /* libera as linhas da matriz */
28     for (i = 0; i < m; i++) free (matriz[i]);
29     /* libera a matriz */
30     free(matriz);
31     return 0;
32 }
```

Alocação dinâmica de matrizes

- Este método aloca uma matriz bidimensional dinamicamente. Para se aumentar o número de dimensões basta aumentar o número de ponteiros, ou seja, para 3 dimensões teríamos a seguinte declaração:
*****matriz;**
- ... e assim por diante.

Free ALL pointers



1. Faça um programa que leia uma matriz 5x5 e um procedimento que imprima sua transposta.
2. Crie uma função que receba uma matriz de números reais e retorne a soma dos elementos desta matriz.
3. Faça um programa que leia um vetor de dimensão 5 e uma matriz quadrada de dimensão 5. Crie um procedimento que multiplique o vetor pela matriz. Imprima o resultado.
4. Faça um programa para ler a quantidade de um total de 5 produtos que uma empresa tem em suas 7 lojas e imprimir em uma tabela:
 - a) o total de cada produto nestas lojas
 - b) a loja que tem menos produtos
5. Faça um programa para calcular o produto entre duas matrizes. Utilize alocação dinâmica.

Na próxima aula...

Registros