

◀ VOLTAR



Matrizes

Apresentar o conceito de variável indexada com mais de uma dimensão (matriz) e aplicar tal conceito utilizando a linguagem C.

NESTE TÓPICO

- > Introdução
- >
- > Definição
- > Declaração de Matriz



Introdução

Estudaremos, agora, como criar uma variável indexada com mais de uma dimensão. A linguagem C permite criar uma variável indexada com qualquer número de dimensões. Esta aula tem como foco a criação de variáveis indexadas com duas dimensões, também denominada matriz.

Definição

Uma matriz é uma coleção de dados do mesmo tipo parecido com uma tabela, possuindo linhas e colunas. Como um vetor, ela possui um nome e os dados são acessados por índices, representando a linha e a coluna respectivamente. Ela também possui um tamanho de linhas e colunas, que deve ser especificado na declaração da matriz.

Se M é uma matriz $m \times n$, isto é, m é quantidade de linhas e n é a quantidade de colunas, então suas linhas são indexadas de 0 a $m-1$ e suas colunas de 0 a $n-1$. Para acessarmos um elemento particular de M , escrevemos $M[i][j]$, onde i é o número da linha e j o número da coluna. A figura a seguir apresenta a representação gráfica de uma matriz bidimensional 3 x 5 de valores inteiros.

	0	1	2	3	4	← índice das colunas
0	2	4	6	8	10	
1	1	3	5	7	9	
2	3	7	11	15	19	

↑ índice das linhas

Representação gráfica de uma matriz 3x5 de números inteiros

Declaração de Matriz

A declaração de uma matriz é feita indicando o tipo de dado que a matriz irá armazenar, o seu nome e o tamanho de cada uma de suas dimensões.

A forma geral de declaração de uma matriz com várias dimensões em linguagem C é:

tipo nome[dim1][dim2][dim3]...[dimN];

Para declararmos uma matriz de duas dimensões usamos a seguinte forma:

tipo nome[linhas][colunas];

Onde, tipo é um tipo qualquer de dados, nome é o nome pelo qual a matriz será referenciada, linhas é a quantidade de linhas que a matriz pode conter e colunas é a quantidade de colunas que a matriz pode conter. Em C, o primeiro elemento da linha tem índice 0 e o último elemento tem índice *linhas-1*, da mesma forma, o primeiro elemento da coluna tem índice 0 e o último elemento tem índice *colunas-1*. O tamanho de linhas e colunas pode ser diferente.

Exemplos:

- Declaração de uma matriz denominada números de 10 linhas e 5 colunas de valores inteiros.

int numeros[10][5];

- Declaração de uma matriz denominada valores de 15 linhas e 15 colunas de valores do tipo float (real).

float valores[15][15];

Assim como um vetor, uma matriz pode ser inicializada no momento de sua declaração. O trecho de código a seguir exemplifica a declaração e inicialização de uma matriz 3 x 4 de elementos do tipo inteiro.

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

Matrizes de duas dimensões são inicializadas da mesma forma que um vetor. Os elementos que representam as linhas são colocados entre chaves depois do sinal de igual e separados por vírgula. Cada elemento de uma linha é composto por chaves e seus elementos internos separados por vírgulas.

Referenciando um Elemento da Matriz

Os elementos de uma matriz devem ser referenciados de forma individualizada. Não é possível referenciar todos os elementos ao mesmo tempo. Um elemento é referenciado pelo nome da matriz seguido do índice da linha entre colchetes e depois o índice da coluna também entre colchetes. O exemplo abaixo referencia o elemento da terceira linha e da segunda coluna de uma matriz. Lembre-se que os índices começam pelo valor 0.

matriz[2][1];

Podemos atribuir um valor a um elemento da matriz da seguinte forma:

matriz[2][1] = 7;

O exemplo acima atribui o valor 7 ao elemento da terceira linha e da segunda coluna da matriz.

Lendo Dados para uma Matriz

Imagine que temos uma matriz 3x4 denominada m que armazena elemento do tipo int. Queremos solicitar ao usuário que informe os dados para preencher essa matriz. O trecho de código a seguir lê os dados que serão armazenados na matriz.

```
1. int i, j; // i representa a linha e j representa a coluna  
2. for (i = 0; i < 3; i++) { // vai de 0 até a quantidade de linhas - 1  
3.     for (j = 0; j < 4; j++) { // vai de 0 até a quantidade de colunas -1  
4.         printf ("m[%d][%d]: ", i, j);  
5.         scanf ("%d", &m[i][j]);  
6.     }  
7. }
```

Na manipulação de vetor é utilizada uma única instrução de laço for. No caso de matrizes com mais dimensões, deve ser utilizado o número relativo à quantidade de dimensões. Dessa forma, uma matriz de duas dimensões deve ser controlada com dois laços for. O laço mais externo (primeiro laço) é responsável por percorrer as linhas e o laço mais interno (segundo laço) é responsável por percorrer as colunas.

Acessando Dados de uma Matriz

Uma vez preenchida a matriz queremos manipular os seus elementos ou apresenta-los na tela. Utilizando a matriz m preenchida na seção anterior, vamos agora calcular a soma dos seus elementos e também apresentar os elementos da matriz na tela. Analise o seguinte trecho de código.

```
1. int soma = 0;
2. int i, j;
3. for (i = 0; i < 3; i++) {
4.     for (i = 0; i < 4; i++) {
5.         // soma a variável o elemento da linha i e coluna j
6.         soma = soma + m[i][j];
7.         // apresenta o elemento da linha i e coluna j
8.         printf ("m[%d][%d] = %d\n", i, j, m[i][j]);
9.     }
10. }
11. printf ("Soma = %d\n", soma);
```

Exemplo 1

A seguir, apresentamos um exemplo completo de um programa em linguagem C que trabalha com matriz bidimensional. O programa cria uma matriz 3x4 de valores inteiros e solicita os seus dados ao usuário, calcula a soma dos elementos da matriz e apresenta os dados da matriz e a soma calculada.

```
1. #include <stdio.h>
2. #include <stdlib.h>
3. // utiliza constante simbólica para definir a quantidade de linhas
4. #define LINHAS 3
5. // utiliza constante simbólica para definir a quantidade de colunas
6. #define COLUNAS 4
7. main () {
8.     // declaração da matriz
9.     int matriz[LINHAS][COLUNAS];
10.    // i representa a linha e j representa a coluna
11.    int i, j;
12.    // leitura dos dados para a matriz
13.    for (i = 0; i < LINHAS; i++) { // vai de 0 até a quantidade de linhas - 1
14.        for (j = 0; j < COLUNAS; j++) { // vai de 0 até a quantidade de colunas -1
15.            printf ("matriz[%d][%d]: ", i, j);
16.            scanf ("%d", &matriz[i][j]);
17.        }
18.    }
19.    // calcula a soma dos valores e apresenta os dados da matriz
20.    int soma = 0;
21.    for (i = 0; i < LINHAS; i++) {
22.        for (j = 0; j < COLUNAS; j++) {
23.            soma += matriz[i][j];
24.            printf ("%d ", matriz[i][j]);
25.        }
26.        printf ("\n");
27.    }
28.    printf ("\nSoma = %d\n", soma);
29.    system ("PAUSE");
30. }
```

Exemplo 2

Vamos estudar outro exemplo de utilização de matriz. O enunciado é: Criar um programa que crie e inicialize uma matriz de elementos inteiros de acordo com a tabela abaixo. Apresentar os elementos da matriz na tela.

3	5	2	1	10
15	12	18	4	6
7	9	13	16	20

```
1. #include <stdio.h>
2. #include <stdlib.h>
3. #define LINHAS 3
4. #define COLUNAS 5
5. main () {
6.     // declaração e inicialização da matriz
7.     int matriz[LINHAS][COLUNAS] = {
8.         {3, 5, 2, 1, 10},
9.         {15, 12, 18, 4, 6},
10.        {7, 9, 13, 16, 20}
11.    };
12.    // apresentação dos dados da matriz
13.    int i, j;
14.    printf ("*** Dados da Matriz ***\n\n");
15.    for (i = 0; i < LINHAS; i++) {
16.        for (j = 0; j < COLUNAS; j++) {
17.            printf ("%2d ", matriz[i][j]);
18.        }
19.        printf ("\n");
20.    }
21.    system ("PAUSE");
22. }
```

Exemplo 3

Neste exemplo, o programa faz a leitura de uma matriz 5x5 de elementos inteiros, calcula e apresenta a soma da diagonal principal. Além disso, apresenta, também, os elementos da matriz.

```
1. #include <stdio.h>
2. #include <stdlib.h>
3. #define LINHAS 5
4. #define COLUNAS 5
5. main () {
6.     // declara a matriz
7.     int matriz[LINHAS][COLUNAS];
8.     int i, j;
9.     // lê os dados para a matriz
10.    printf ("*** Informe os dados da matriz ***\n");
11.    for (i = 0; i < LINHAS; i++) {
12.        for (j = 0; j < COLUNAS; j++) {
13.            printf ("Valor[%d][%d]: ", i, j);
14.            scanf ("%d", &matriz[i][j]);
15.        }
16.    }
17.    // calcula a soma da diagonal principal e
18.    // apresenta os dados da matriz na tela
19.    int soma = 0;
20.    printf ("\n*** Dados da Matriz ***\n\n");
21.    for (i = 0; i < LINHAS; i++) {
22.        for (j = 0; j < COLUNAS; j++) {
23.            printf ("%d ", matriz[i][j]);
24.            // se o índice da linha é igual ao da coluna,
25.            // então o elemento está na diagonal principal.
26.            if (i == j) {
27.                soma += matriz[i][j];
28.            }
29.        }
30.        printf ("\n");
31.    }
32.    printf ("\nSoma = %d\n\n", soma);
33.    system ("PAUSE");
34. }
```

Exemplo 4

Neste exemplo, o programa faz a leitura de uma matriz 5x3 de elementos reais, calcula e apresenta a média dos valores que estão acima de 20,75; apresenta, também, os elementos da matriz.

```
1. #include <stdio.h>
2. #include <stdlib.h>
3. #define LINHAS 5
4. #define COLUNAS 3
5. main () {
6.     // declara a matriz
7.     float matriz[LINHAS][COLUNAS];
8.     int i, j;
9.     // lê os dados para a matriz
10.    printf ("*** Informe os dados da matriz ***\n");
11.    for (i = 0; i < LINHAS; i++) {
12.        for (j = 0; j < COLUNAS; j++) {
13.            printf ("Valor[%d][%d]: ", i, j);
14.            scanf ("%f", &matriz[i][j]);
15.        }
16.    }
17.    // soma os elementos superiores a 20,75
18.    // guarda a quantidade de elementos somados
19.    // apresenta os dados da matriz na tela
20.    float soma = 0;
21.    int quantidade = 0;
22.    printf ("\n*** Dados da Matriz ***\n\n");
23.    for (i = 0; i < LINHAS; i++) {
24.        for (j = 0; j < COLUNAS; j++) {
25.            printf ("%2f ", matriz[i][j]);
26.            if (matriz[i][j] > 20.75) {
27.                soma += matriz[i][j];
28.                quantidade++;
29.            }
30.        }
31.        printf ("\n");
32.    }
33.    float media = 0;
34.    if (quantidade > 0) {
35.        media = soma / quantidade;
36.    }
37.    printf ("\nMedia = %.2f\n\n", media);
38.    system ("PAUSE");
39. }
```

Agora que você já estudou essa aula acesse a plataforma AVA, resolva os exercícios e verifique o seu conhecimento. Caso fique alguma dúvida, leve a questão ao Fórum e divida com seus colegas e professor.

Quiz

Exercício

Matrizes

INICIAR ➤

Quiz

Exercício Final

Matrizes

INICIAR ➤

Referências

MIZRAHI, V. V. *Treinamento em linguagem C*, São Paulo: Pearson, 2008.


SCHILDT, H. C — *Completo e Total*. São Paulo: Pearson, 2006.



Avalie este tópico



ANTERIOR
Vetores

 Índice

Biblioteca
(<https://www.uninove.br/conhec-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/>)
Portal Uninove
(<http://www.uninove.br>)
Mapa do Site

Ajuda?
PRÓXIMO
(<https://ava.uninove.br/Curso/>)
Sua Curso

© Todos os direitos reservados

