FATEC - SP

Faculdade de Tecnologia de São Paulo

Prof° Luiz Paulo Zanetti

E-mail: luizpaulozanetti@hotmail.com

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Disciplina
Linguagem de Programação

Matriz

ou

Array Multidimensional ou

Vetor Multidimensional

Matriz / Vetor Multidimensional

Também chamados de *Vetores Multidimensionais*:
Nos exemplos passados, sobre vetores, vinha usando somente um tipo de vetor:

Vetores Unidimensionais.

Nossos vetores, ou arrays, tinham uma linha e várias colunas.
Seja 'n' o número de colunas, nossos vetores eram matrizes 1 x n.
Sim, uma linha de elementos também é uma matriz.

Matriz / Vetor Multidimensional

Vamos agora aprender para que servem e como trabalhar com vetores de mais de uma linha.
Para isso, vamos voltar ao exemplo do colégio, onde declaramos um vetor para armazenar as notas de um aluno.

Na escola, esse aluno tem várias matérias:
Matemáticas, Física, Química, Biologia etc.
Vamos supor que existam 5 provas ao longo do ano, para cada matéria.

Matriz / Vetor Multidimensional

Poderíamos representar as notas de cada aluno da seguinte maneira:

```
float notasMatematica[5];
float notasFisica[5];
float notasQuimica[5];
```

Matriz / Vetor Multidimensional

Como declarar e trabalhar com Matrizes em C

Existe, porém, uma maneira bem mais fácil, lógica e organizada de trabalhar com vários vetores.
Quando nós falarmos de vetor, lembre de uma linha com vários elementos.

Pois bem, uma maneira melhor de ver esses vetores de notas, seria na forma de uma tabela.
Uma linha tem as notas de Matemática, na outra as notas de Física e assim vai.
Cada coluna representa as provas feitas: prova 1, prova 2, ... etc.

Matriz / Vetor Multidimensional

Vamos fazer isso!

Por exemplo, vamos supor que ele tirou as seguintes notas em Matemática (é uma matriz 1x5):

8.0 7.5 8.5 9.0 8.0

Agora vamos representar as notas em Física, abaixo das de Matemática.

Teremos uma matriz 2x5, ou seja, uma matriz de duas linhas e 5 colunas:

8.0 7.5 8.5 9.0 8.0

8.9 9.0 8.6 8.4 8.0

Matriz / Vetor Multidimensional

Agora vamos representar as notas de Química, abaixo das notas de Física.
Teremos uma matriz 3x5, ou seja, uma matriz de três linhas e 5 colunas:

8.0	7.5	8.5	9.0	8.0
8.9	9.0	8.6	8.4	8.0
6.8	7.1	7.0	7.6	6.5

Ok, agora vamos partir para a programação e ver como declarar e passar isso pra linguagem C.

Matriz / Vetor Multidimensional

Para declarar a matriz 2x5, fazemos:
float notas[2][5];

Note que temos duas linhas: notas[0][] e notas[1][], e em cada linha dessa temos 5 elementos.
Ou seja, é uma matriz de duas linhas e cinco colunas.
Sempre o primeiro número é a linha e o segundo é a coluna.

Matriz / Vetor Multidimensional

Para declarar matrizes e inicializar, devemos colocar cada linha entre chaves {}, e separar elas por vírgulas, veja:

```
float notas[2][5] = { {8.0, 7.5, 8.5, 9.0, 8.0 }, {8.9, 9.0, 8.6, 8.4, 8.0 } };
```

Uma maneira mais simples de ver essas linhas e colunas, como tabela, é da seguinte maneira:

```
float notas[2][5] =   { {8.0, 7.5, 8.5, 9.0, 8.0 },
                        {8.9, 9.0, 8.6, 8.4, 8.0 } };
```

Matriz / Vetor Multidimensional

Para declarar a matriz 3x5, fazemos::

```
float notas[3][5];
```

Veja como fica nossa matriz, ou tabela, declarada e inicializada:

```
float notas[3][5] = { {8.0, 7.5, 8.5, 9.0, 8.0 }, {8.9, 9.0, 8.6, 8.4, 8.0 }, {6.8, 7.1, 7.0, 7.6, 6.5 } };
```

De uma maneira mais fácil de entender:

```
float notas[3][5] = { {8.0, 7.5, 8.5, 9.0, 8.0 },
                      {8.9, 9.0, 8.6, 8.4, 8.0 },
                      {6.8, 7.1, 7.0, 7.6, 6.5 } };
```

Matriz / Vetor Multidimensional

Note que notas[0] se refere ao vetor de notas de Matemática.
Note que notas[1] se refere ao vetor de notas de Física.
Note que notas[2] se refere ao vetor de notas de Química.

Por exemplo: qual foi a quarta nota de Física do aluno?
Ora, o vetor de Física é notas[1], e a quarta nota é o elemento [3] desse vetor.

Então a quarta nota de Física do aluno está armazenada em: notas[1][3], que é 8.4

Generalizando, para declarar uma matriz de 'linha' linhas e de 'coluna' colunas, fazemos: tipo nome[linha][coluna];

Matriz / Vetor Multidimensional

Para acessar o elemento da i-ésima linha e de j-ésima coluna, acessamos pela variável: nome[i][j];

É uma variável como outra qualquer em linguagem C.

Podemos somar, incrementar, zerar etc.

Matriz / Vetor Multidimensional

Criar e exibir uma matriz 3x3.

Crie um aplicativo em C, que peça ao usuário para preencher uma matriz 3x3 com valores inteiros e depois exiba essa matriz.

A grande novidade, e importância, nesse tipo de aplicativo são os laços for aninhados, ou seja, um dentro do outro, e do uso do #define, para tratar da constante DIM, que representa a dimensão da matriz.

Matriz / Vetor Multidimensional

Primeiro criamos um laço que vai percorrer todas as linhas da matriz. Podemos, e devemos, ver cada linha como um vetor de 3 elementos.

Dentro de cada linha, temos que percorrer cada elemento do do vetor e fornecer seu valor. Fazemos isso através de outro laço for, que ficará responsável pelas 'colunas', formando nossos laços aninhados.

Para imprimir, o esquema é exatamente o mesmo. Imprimimos linha por linha, e em cada linha, imprimimos coluna por coluna.

Matriz / Vetor Multidimensional

```
#include <stdio.h>
#include <conio.h>
#define DIM 3
void main()
{
   int matriz[DIM][DIM];
   int linha, coluna;
   clrscr();
   //escrevendo na Matriz
   for(linha = 0 ; linha < DIM ; linha++)
      for(coluna = 0 ; coluna < DIM ; coluna++)
      {
         printf("Elemento [%d][%d]: ", linha+1, coluna+1);
         scanf("%d", &matriz[linha][coluna]);
      }
   // imprimindo a matriz na tela
   for(linha = 0 ; linha < DIM ; linha++)
   {
      for(coluna = 0 ; coluna < DIM ; coluna++)
         printf("%3d", matriz[linha][coluna]); // %3d inteiro com pelo menos 3 digitos
      printf("\n"); //após cada linha ser impressa, um salto de linha
   }
getch();
}
```

Exemplo - 01

```
Elemento [1][1]: 1
Elemento [1][2]: 2
Elemento [1][3]: 3
Elemento [2][1]: 4
Elemento [2][2]: 5
Elemento [2][3]: 6
Elemento [3][1]: 7
Elemento [3][2]: 8
Elemento [3][3]: 9
  1  2  3
  4  5  6
  7  8  9
```

Matriz / Vetor Multidimensional

Calcular o traço de uma matriz em C
Use o programa feito no exemplo anterior.
Lembrando que o traço de uma matriz é a soma dos elementos da diagonal principal.

Os elementos das diagonal principal são os que tem índice da linha igual ao índice da coluna:
matriz[0][0], matriz[1][1] e matriz[2][2].

Aproveitamos o laço for das linhas para calcular a soma desses elementos:
matriz[linha][linha]

Visto que a variável 'linha', assim como a 'coluna', vão de 0 até 2.

Matriz / Vetor Multidimensional

```
#include <stdio.h>
#include <conio.h>
#define DIM 3
void main()
{
   int matriz[DIM][DIM];
   int linha, coluna, traco = 0;
   clrscr();
   //escrevendo na Matriz
   for(linha = 0 ; linha < DIM ; linha++)
      for(coluna = 0 ; coluna < DIM ; coluna++)
      {
         printf("Elemento [%d][%d]: ", linha+1, coluna+1);
         scanf("%d", &matriz[linha][coluna]);
      }
    // imprimindo a matriz na tela
   for(linha = 0 ; linha < DIM ; linha++)
   {
      for(coluna = 0 ; coluna < DIM ; coluna++)
      printf("%3d", matriz[linha][coluna]);
      traco += matriz[linha][linha];
      printf("\n"); //após cada linha ser impressa, um salto de linha
   }
    printf("\nTraco da matriz: %d\n", traco);
getch();
}
```

Significa que vai atribuir mais 1 a variável antes de atribuir o valor após o igual.
Exemplo:

```
a = 1;
b = 2;
c = 10;

c += a + b;
```

C receberá +1 antes de receber o resultado de a + b;
10 + 1 = 1 + 2

Exemplo - 02

```
Elemento [1][1]: 1
Elemento [1][2]: 2
Elemento [1][3]: 3
Elemento [2][1]: 4
Elemento [2][2]: 5
Elemento [2][3]: 6
Elemento [3][1]: 7
Elemento [3][2]: 8
Elemento [3][3]: 9
  1  2  3
  4  5  6
  7  8  9

Traco da matriz: 15
```