Universidade Católica de Brasília

Algoritmo e Programação Estruturada

Aula 7

Prof. Osmam Brás de Souto

Universidade Católica de Brasília

Vetores e Matrizes

Um vetor (ou matriz) é um conjunto de posições de armazenamento de dados do mesmo tipo. Cada posição de armazenamento é chamado de elemento da matriz (ou do vetor).

No conteúdo deste material estará apresentando-se esta estrutura com o termo matriz, apesar da tradução de *array* , em algumas obras, aparecerem como vetor.

Na declaração de uma matriz escreve-se o tipo, seguido do nome (identificador) e o tamanho da matriz (número de elementos que ela contém) entre os colchetes ([ ]).

<tipo de dado> <identificador> [<quantidade de elementos>]

Exemplo: float valor[10];

Universidade Católica de Brasília

Vetores e Matrizes

```
int teste[25];
```

Neste exemplo, declara-se uma matriz unidimensional (ou vetor) de 25 inteiros com o nome de TESTE, onde o compilador reservará um espaço, em memória contígua, suficiente para conter estes 25 elementos da matriz.

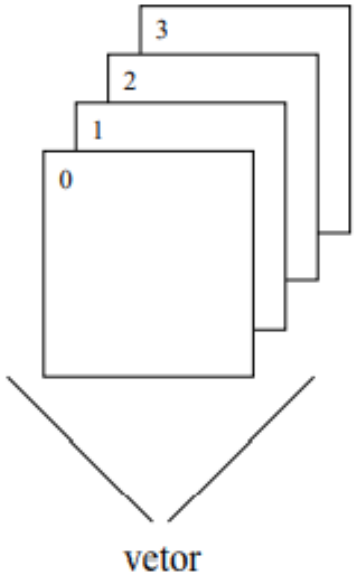
A memória contígua significa que os dados serão armazenados em seqüência contínua (um seguido do outro).

Universidade Católica de Brasília

Matriz

Para acessar cada elemento de uma matriz faz-se referência a um deslocamento do nome da matriz. O principal cuidado que se deve ter é que a primeira posição de um elemento é a posição zero (0), ou seja, uma matriz é iniciada no elemento zero.

No exemplo anterior tem-se 25 elementos declarados na matriz, porém a sua representação para cada um deles vai ser de zero (0) a vinte e quatro (24), contendo assim 25 elementos. Esta representação para cada elemento consiste em um índice que permite manipular todos os elementos da matriz.



A alimentação de uma matriz é feita por meio da atribuição de valores compatíveis com o seu tipo, estipulado em sua declaração. Cada elemento recebe um valor por meio do seu nome e índice (ou posição).

Universidade Católica de Brasília

Matriz (Exemplo Prático)

```
int main(void) {
int x,vetor[5];
 // Armazenando valores na matriz
 for(x=0; x<5; x++)
 {
 printf("Matriz [%d] = ",x+1);
 scanf("%d",&vetor[x]);
 }
 printf("\n Matriz\n\n");
 // Apresentando os valores armazenados
 for(x=0; x<5; x++)
 printf("Posicao %d =%3d\n",x+1, vetor[x]);
 getch();
 return(0);

}
```

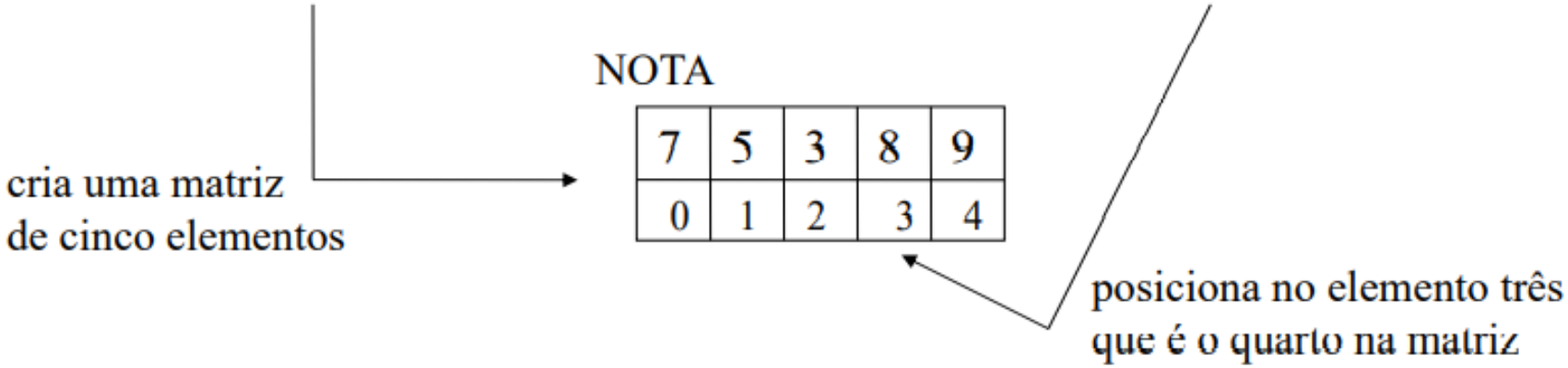
Matriz

A declaração de matrizes diferencia-se da declaração de variáveis comuns pelo uso dos colchetes (‘[ ]’) preenchidos com números inteiros.

Este número entre os colchetes indica o tamanho da matriz.

Os valores dentro dos colchetes são diferentes de acordo com o contexto que estão sendo empregados. O primeiro é a declaração das variáveis, que indica o tamanho da matriz. O segundo é a referência de um elemento específico dentro da matriz.

int nota[5] ≠ nota[3]



Universidade Católica de Brasília

Matriz

A declaração e a referência a um elemento demonstram situações diferentes, pois a declaração apresenta a quantidade de elementos que uma matriz terá, enquanto que a referência apresenta o número do índice da matriz que se deseja o elemento

int nota[5]; → declaração da matriz de cinco valores

nota[3] → referência ao elemento 3 que é o quarto valor na matriz

A solicitação de gravação de um elemento, além da matriz declarada, pode provocar conseqüências inesperadas, pois o compilador calcula a posição de memória onde este elemento seria armazenado e grava-o. No caso do exemplo anterior, suponha que seja informado o sexto valor (nota[5]) mantendo a declaração de somente cinco elementos nesta matriz.

Universidade Católica de Brasília

Matriz

O compilador multiplica o deslocamento (5) pelo tamanho de cada elemento (2 bytes no exemplo). Então ele passa por todos os bytes desde o início da matriz e grava o novo valor na posição calculada.

Não é verificado o tamanho máximo da matriz e o compilador grava baseado em seus cálculos, não importando sobre o que ele vai estar gravando, mas sobrepõe os valores.

Universidade Católica de Brasília

Matriz(Exemplo Prático)

```
int main(void) {
    float vetor[5];
    int x;
        for ( x=0; x < 10; x++)
            vetor[x] = x * x + 5;
    for ( x=0; x < 10; x++)
            printf("Vetor[%d] = %2.1f\n", x+1,vetor[x]);
 getch();
}
```

Universidade Católica de Brasília

Matriz

A inicialização de uma matriz, de tipos fundamentais (inteiros ou caracteres), pode ser feita na sua declaração.

```
int matriz[5] = {10,20,30,40,50};
```

A omissão do tamanho da matriz ocasionará a criação de uma matriz de tamanho suficiente para conter os elementos inicializados que foram atribuídos a ela.

```
int matriz[ ] = {10,20,30,40,50};
```

Esta declaração criará uma matriz igual a anterior. Não se pode inicializar mais elementos do que foi declarado na matriz. Caso deseje saber o tamanho da matriz, pode se colocar o compilador para encontrá-lo. O retorno deste cálculo será exatamente a quantidade de elementos da matriz.

```
const int tam  = sizeof(matriz) / sizeof(matriz[0]);
```

Os elementos não inicializados na matriz são nulos.

```
int matriz[5] = {10,20};
```

Universidade Católica de Brasília

Matriz

O uso da diretiva *define* é comum em matrizes, pois com o passar do tempo a matriz pode sofrer alterações (aumentar ou diminuir), o que resultaria em uma grande manutenção pelo programa. Com o intuito de agilizar a manipulação dos valores da matriz em um programa, esta diretiva coerentemente garante ações mais seguras e organizadas em seu código (programa).

Universidade Católica de Brasília

Matriz(Exemplo Prático)

```
#define MAX 4
int main(void) {
   float soma,notas[MAX];
  int i;
  for(i=0,soma=0.0; i < MAX ; soma+=notas[i++])
  { printf("Nota %d:",i+1);
  scanf("%f",&notas[i]); }
  printf("Media: %.1f ",soma / MAX);
  getch();
}
```

Referência de Criação e Apoio ao Estudo

Material para Consulta e Apoio ao Conteúdo

- FARRER, H. et all, Algoritmos Estruturados, Editora LTC, 3ª . edição, 1999. - livro
  - Capítulo 0

- MANZANO, J. e Oliveira, J., Algoritmos, Lógica para desenvolvimento de programação, Editora Ética, 1996.
  - Capítulo 1

Universidade Católica de Brasília

Obrigado(a)!