Vetores

Algoritmos e Estruturas de Dados I

Prof. Cristiano Rodrigues

Prof. Lucas Astore

Vetores

Vetores, também chamados arrays (do inglês) ou arranjo, são uma maneira de **armazenar vários dados** num **mesmo nome de variável** através do uso de índices numéricos.

Em C, vetores devem sempre conter dados do **mesmo tipo** de variável.

Vetores

- Declaramos vetores de maneira muito semelhante à declaração de variáveis normais
- A única diferença é que depois do nome da variável deve ser informada a quantidade de elementos do vetor.
- Para declarar um vetor chamado vetor, com cinco elementos inteiros, escrevemos:

```
int vetor[5];
```

Vetores

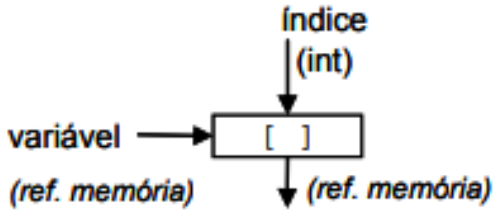
- Declaramos vetores de maneira muito semelhante à declaração de variáveis normais
- A única diferença é que depois do nome da variável deve ser informada a quantidade de elementos do vetor.
- Para declarar um vetor chamado vetor, com cinco elementos inteiros, escrevemos:

```
int vetor[5];

int vetor[5] = {1,2,3,4,5};
```

Vetores

- Para fazer referência a um valor de um elemento contido em um vetor, usamos a notação **vetor[índice]**, que serve tanto para obter quanto para definir o valor de um elemento específico, dada sua posição.
- Note que os elementos são numerados a **começar do zero**, e, portanto, se o número de elementos é N, o índice ou posição do último elemento será N - 1.



Índices inválidos

- Os elementos são numerados sempre de 0 até tamanho-1. Caso o programa tente acessar erroneamente um elemento de índice negativo ou de índice além do tamanho do vetor, as consequências poderão ser imprevisíveis.

Regras para acesso

Atribuir o valor de todos os elementos de uma só vez

- Não é possível atribuir valores a todos os elementos em uma só linha. Cada elemento precisa ser acessado individualmente. Tampouco é possível usar um único *scanf* para ler todo o conteúdo do vetor

```
int vetor[10];
int indice;
// inicializar todos os elementos com o valor 0
for (indice = 0; indice < 10; indice++) {
   vetor[indice] = 0;
}
```

Copiar Vetores

- Não é possível copiar o conteúdo de um vetor para um outro, mesmo que os dois sejam de mesmo tamanho e os elementos sejam de mesmo tipo.

```
int vetorA[10], vetorB[10];
int indice;
// copiar o conteúdo do vetor B para o vetor A
for (indice = 0; indice < 10; indice++) {
    vetorA[indice] = vetorB[indice];
}
```

Exercícios

Exercício 1: Inicialização e Acesso a Elementos do Vetor

Enunciado

Escreva um programa em C que declare um vetor de 10 elementos inteiros.

O programa deve ler 10 valores do usuário e armazená-los no vetor.

Em seguida, exiba todos os elementos do vetor.

Exemplo de saída

```
Digite o valor 1: 5
Digite o valor 2: 12
Digite o valor 3: 9
...
Vetor: 5 12 9 ...
```

Exercício 2: Soma dos Elementos do Vetor

Enunciado

Escreva um programa em C que declare um vetor de 5 elementos inteiros.

O programa deve ler 5 valores do usuário, armazená-los no vetor e calcular a soma de todos os elementos.

Exemplo de saída

```
Digite o valor 1: 4
Digite o valor 2: 8
Digite o valor 3: 10
Digite o valor 4: 2
Digite o valor 5: 7
A soma dos elementos do vetor é: 31
```

Exercício 3: Maior e Menor Elemento do Vetor

Enunciado

Escreva um programa em C que leia 8 números inteiros, os armazene em um vetor, e então determine o maior e o menor número contidos nesse vetor.

Exemplo de saída

```
Digite o valor 1: 3
Digite o valor 2: 10
Digite o valor 3: 5
...
O maior número é: 10
O menor número é: 3
```

Exercício 4: Simulação do Sorteio da Mega-Sena

Enunciado

Escreva um programa em C que declare um vetor de 6 elementos inteiros. O programa deve preencher o vetor com 6 números aleatórios entre 1 e 60, simulando um sorteio da Mega-Sena. Os números gerados devem ser únicos, ou seja, não pode haver repetição de números.

Exemplo de saída

```
Números sorteados: 5 12 23 34 45 50
```

Exercício 5: Vetor com Valores Aleatórios

Enunciado

Escreva um programa em C que declare um vetor de 10 elementos inteiros.

O programa deve preencher o vetor com valores aleatórios em um intervalo determinado pelo usuário.

Em seguida, exiba todos os elementos do vetor.

Exemplo de saída

```
Digite o valor mínimo
do intervalo: 10

Digite o valor máximo
do intervalo: 50

Vetor gerado: 12 45 37
18 50 23 32 14 47 20
```

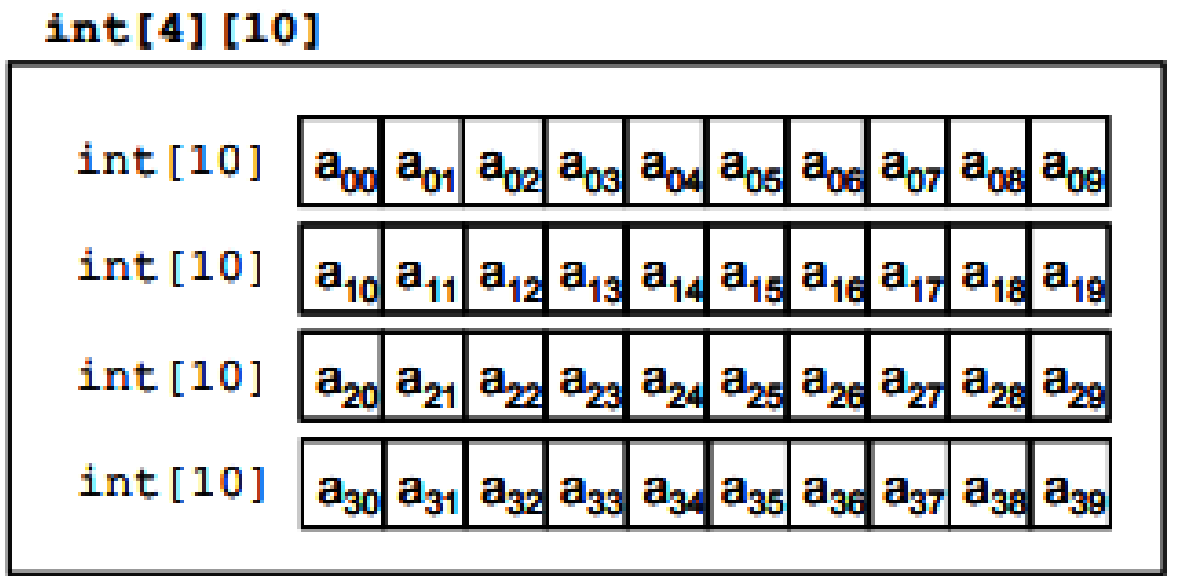
Matrizes e vetores multidimensionais

Matrizes

- Em C, uma matriz é uma extensão do conceito de vetor.

- Enquanto um vetor é uma lista unidimensional de elementos, uma matriz é uma estrutura bidimensional (ou até multidimensional), onde os dados são organizados em linhas e colunas.

Matrizes e vetores multidimensionais



Matrizes

- Uma matriz em C é essencialmente um array bidimensional que armazena múltiplos valores do mesmo tipo em uma estrutura semelhante a uma tabela.

- Cada elemento é acessado através de dois índices: um para a linha e outro para a coluna.

Matrizes e vetores multidimensionais

A noção de matriz é a generalização imediata da noção de vetor. Uma matriz é uma tabela de vários valores do mesmo tipo, armazenados sequencialmente e fazendo uso de um mesmo nome de variável para acessar esses valores

Cada elemento da tabela pode ser acessado individualmente através de dois índices com valores inteiros. Estes índices poderiam ser interpretados como a linha e a coluna da matriz

Matrizes e vetores multidimensionais

A declaração de matrizes obedece a mesma sintaxe que a declaração de vetores, exceto pela adição de uma nova dimensão escrita entre colchetes[ ].

```
tipo variável[linhas][colunas];
```

Matrizes e vetores multidimensionais

Para percorrer os elementos de uma matriz, são necessárias duas estruturas de repetição **for**, uma dentro da outra. O **for** externo percorre as linhas da matriz, o **for** interno percorre as colunas de uma determinada linha que está fixada pelo **for** externo. Por exemplo, para imprimir todos os elementos de uma matriz 4x10, linha por linha:

```
int linha, coluna;
int matriz[4][10];
...
for (linha = 0; linha < 4; linha++) {
   for (coluna = 0; coluna < 10; coluna++) {
      printf("%d ", matriz[linha][coluna]);
   }
   printf("\n");
}
```

Exercícios

Ex 6: Soma dos Elementos de uma Matriz

Enunciado

Escreva um programa em C que declare uma matriz de inteiros com 3 linhas e 4 colunas. O programa deve solicitar que o usuário insira os valores para preencher a matriz. Em seguida, o programa deve calcular e exibir a soma de todos os elementos da matriz.

Exemplo de saída

```
Digite o valor para a posição [0][0]: 2
Digite o valor para a posição [0][1]: 4
...
A soma dos elementos da matriz é: 50
```

Ex 7: Transposta de uma Matriz

Enunciado

Escreva um programa em C que leia uma matriz 2x3 de números inteiros, armazene os valores e, em seguida, calcule e exiba a matriz transposta (no caso, uma matriz 3x2).

Exemplo de saída

Matriz original:

1 2 3

4 5 6

Matriz transposta:

1 4

2 5

3 6