

Técnicas de Programação

Arrays

Fábio Duncan de Souza

Instituto Federal Fluminense

Técnicas de
Programação

Fábio Duncan

Arrays Unidi-
mensionais -
Vetores

Arrays
Bidimensionais
- Matrizes

Arrays Multi-
dimensionais

Referências

1 Arrays Unidimensionais - Vetores

2 Arrays Bidimensionais - Matrizes

3 Arrays Multidimensionais



Arrays Unidimensionais - Vetores

- Vetores, ou arrays unidimensionais, são estruturas de dados que armazenam uma quantidade fixa de dados de um certo tipo
- Os dados armazenados nos vetores são alocados sequencialmente na memória, classificando esta estrutura de dados como sequencial
- Por armazenarem dados somente de um mesmo tipo também são classificados como estruturas de dados homogêneas
- Cada valor armazenado está associado a um número que se refere à posição de armazenamento, conhecida como índice do vetor, ou mais formalmente, subscrito do vetor
- As posições dos vetores são acessadas diretamente utilizando o endereço inicial da estrutura em conjunto com o deslocamento identificado no índice

- Cada elemento do vetor pode ser acessado individualmente para leitura ou escrita sem nenhuma regra preestabelecida, o que classifica este tipo de estrutura como de acesso aleatório
- O número de posições de um vetor corresponde ao seu tamanho, assim, um vetor de tamanho 100 pode armazenar até 100 elementos distintos
- $a[i]$, corresponde ao i -ésimo elemento do vetor a , sendo que o valor da variável i deve pertencer ao intervalo dos índices do vetor:
 $((i \geq 0) \&\& (i < \text{tamanhoDoVetor}(a)))$
- Na maioria das linguagens atuais, as posições do vetor são iniciadas a partir do valor 0, portanto, um vetor de tamanho 100 teria índices iniciados em 0 prosseguindo até o 99.



Arrays Unidimensionais - Vetores

Declaração de Vetores na Linguagem C

Técnicas de
Programação

Fábio Duncan

Arrays Unidi-
mensionais -
Vetores

Arrays
Bidimensionais
- Matrizes

Arrays Multi-
dimensionais

Referências

- Os vetores em C são declarados utilizando um par de colchetes depois do nome da variável
- Dentro dos colchetes é definido o número de posições do vetor
 - `tipo_dos_dados nome_da_variável[<número de posições>];`



Vetores na Linguagem C

Técnicas de
Programação

Fábio Duncan

Arrays Unidi-
dimensionais -
Vetores

Arrays
Bidimensionais
- Matrizes

Arrays Multi-
dimensionais

Referências

```
int vetorNumeros[10] = {11,21,36,47,15,16,87,89,39,64}
```

	Memória RAM	
		⋮
vetorNumeros[0]	11	7ffc25a01360
vetorNumeros[1]	21	7ffc25a01361
vetorNumeros[2]	36	7ffc25a01362
vetorNumeros[3]	47	7ffc25a01363
vetorNumeros[4]	15	7ffc25a01364
vetorNumeros[5]	16	7ffc25a01365
vetorNumeros[6]	87	7ffc25a01366
vetorNumeros[7]	89	7ffc25a01367
vetorNumeros[8]	39	7ffc25a01368
vetorNumeros[9]	64	7ffc25a01369
		7ffc25a0136A
		⋮

vetorNumeros

7ffc25a01360

Exemplos:

Declaração/Inicialização do vetor
`int vetorNumeros [10];`
`int vetorNumeros[] = {11,21,36,47,15,16,87,89,39,64}`

Atribuição de um valor para a posição 5
`vetorNumeros[5] = 333;`

Impressão do número da posição 5
`printf("%d", vetorNumeros[5]);`



Arrays Unidimensionais - Vetores

Programa Exemplo

- Implementar um programa para realizar a entrada de 20 números aleatórios em um vetor e depois apresentar os números na ordem de entrada e na ordem inversa da entrada.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main() {
5      int vetorNumerosAleatorios[20];
6      int indiceVetor;
7
8      srand(10);
9      for (indiceVetor = 0; indiceVetor < 20; indiceVetor++) {
10         vetorNumerosAleatorios[indiceVetor] = random() % 200;
11     }
12     printf("\n\n**** Números Gerados ****\n");
13     for (indiceVetor = 0; indiceVetor < 20; indiceVetor++) {
14         printf("%d ", vetorNumerosAleatorios[indiceVetor]);
15     }
16     printf("\n\n**** Números na Ordem Inversa ****\n");
17     for (indiceVetor = 19; indiceVetor >= 0; indiceVetor--) {
18         printf("%d ", vetorNumerosAleatorios[indiceVetor]);
19     }
20     return 0;
21 }
```




Arrays Bidimensionais - Matrizes

- Os arrays apresentados até o momento possuem apenas uma dimensão e também são conhecidos como vetores
- Quando existe a necessidade de organizar valores em uma estrutura de linhas e colunas, os arrays bidimensionais são utilizados
- Os arrays bidimensionais também são conhecidos como matrizes
- Para identificar um elemento específico de uma matriz, deve-se especificar dois subscritos (índices): o primeiro identifica a linha do elemento, e o segundo identifica a coluna do elemento.



Arrays Bidimensionais - Matrizes

Declaração de Matrizes na Linguagem C

Técnicas de
Programação

Fábio Duncan

Arrays Unidi-
mensionais -
Vetores

Arrays
Bidimensionais
- Matrizes

Arrays Multi-
dimensionais

Referências

- As matrizes bidimensionais em C são declaradas utilizando dois colchetes depois do nome da variável
- Dentro do primeiro par de colchetes é definido o número de linhas da matriz e dentro do segundo o número de colunas
 - `tipo_dos_dados nome_da_variável[<número de linhas>] [<número de colunas>];`



Arrays Bidimensionais - Matrizes

Figura Exemplo

Técnicas de
Programação

Fábio Duncan

Arrays Unidi-
mensionais -
Vetores

Arrays
Bidimensionais
- Matrizes

Arrays Multi-
dimensionais

Referências

`int matrizNumeros[5][10] = {0,0,0,0,0,0,0,0,0,0...}`

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	333	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0

`matrizNumeros[1][3]`

Exemplos:

Declaração/Inicialização do vetor

```
int matrizNumeros [5][10];  
int matrizNumeros[5][10] = {{0,0,0,0,0,0,0,0,0,0}, {0,...}...}
```

Atribuição de um valor para uma posição:

linha 1 coluna 3
`matrizNumeros[1][3] = 333;`

Impressão do número de uma posição:

linha 1 coluna 3
`printf("%d", matrizNumeros[1][3]);`



Arrays Bidimensionais - Matrizes

Programa Exemplo

- Implementar um programa para realizar a entrada de 50 números aleatórios em uma matriz de 5 linhas e 10 colunas e depois apresentar a matriz na tela.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main() {
5     int matrizNumerosInteiros[5][10];
6     int linha, coluna;
7
8     srand(10);
9     for(linha = 0; linha < 5; linha++) {
10         for(coluna = 0; coluna < 10; coluna++) {
11             matrizNumerosInteiros[linha][coluna] = random() % 1000;
12         }
13     }
14     for(linha = 0; linha < 5; linha++) {
15         for(coluna = 0; coluna < 10; coluna++) {
16             printf("%4d ", matrizNumerosInteiros[linha][coluna]);
17         }
18         printf("\n");
19     }
20     return 0;
21 }
```



Técnicas de Programação

Arrays Unidimensionais - Vetores

- Matrizes

Referências

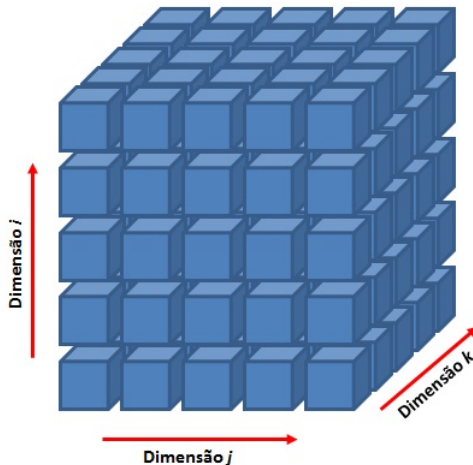
Arrays Multidimensionais

- A Linguagem C permite que se crie arrays com mais de duas dimensões de maneira simples
- Cada conjunto de colchetes [] representa uma dimensão do array
- Cada par de colchetes adicionado ao nome de uma variável durante a sua declaração adiciona uma nova dimensão àquela variável, independente do seu tipo
- Exemplos
 - `int vetor[5];` // 1 dimensão
 - `float matriz[5][5];` // 2 dimensões
 - `char cubo[5][5][5];` // 3 dimensões
 - `int xpto[5][5][5][5];` // 4 dimensões
 - etc.



Arrays Multidimensionais

Gráfico Exemplo - Matriz com 3 Dimensões





Arrays Bidimensionais - Matrizes

Programa Exemplo

- Implementar um programa para realizar a entrada de 50 números aleatórios em uma matriz de 5 linhas e 10 colunas e depois apresentar a matriz na tela.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main() {
5     int matrizNumerosInteiros[5][10];
6     int linha, coluna;
7
8     srand(10);
9     for(linha = 0; linha < 5; linha++) {
10         for(coluna = 0; coluna < 10; coluna++) {
11             matrizNumerosInteiros[linha][coluna] = random() % 1000;
12         }
13     }
14     for(linha = 0; linha < 5; linha++) {
15         for(coluna = 0; coluna < 10; coluna++) {
16             printf("%4d ", matrizNumerosInteiros[linha][coluna]);
17         }
18         printf("\n");
19     }
20     return 0;
21 }
```

Referências Bibliográficas

Técnicas de
Programação

Fábio Duncan

Arrays Unidi-
mensionais -
Vetores

Arrays
Bidimensionais
- Matrizes

Arrays Multi-
dimensionais

Referências



Harvey M Deitel and Paul J Deitel.
Como programar em C.
LTC, 1999.



Renato Cardoso Mesquita.
Curso de linguagem C.
Universidade Federal de Minas Gerais, 1998.



Wikimedia.
Linguagem de programação.
Wikipédia: a enciclopédia livre, 2020.