

Fundamentos de Programação

Linguagem de Programação C
Vetor

Sumário

- 1 Definição
- 2 Declaração
- 3 Atribuição

- 4 Definição
- 5 Manipulação
- 6 Exemplos
- 7 Strings

Definição

É um grupo de locais da memória relacionados pelo fato de que todos têm o mesmo nome e o mesmo tipo. Para fazer referência a um determinado local ou elemento no array, especificamos o nome do array e o número da posição daquele elemento no array.

(Deitel; Deitel, 2011)

O que isso quer dizer?

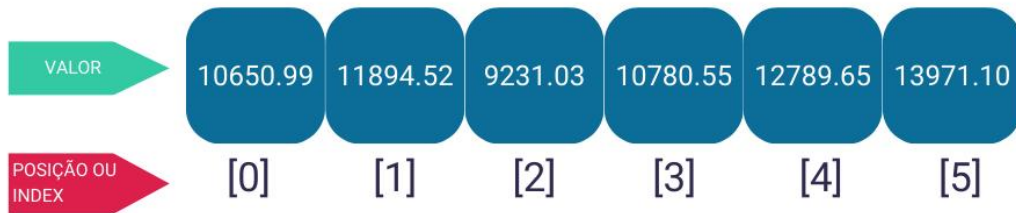
- Até o momento, utilizamos variáveis para armazenar dados, que permitiam armazenar apenas um único valor de um único tipo.
- No entanto, é possível usar uma estrutura chamada vetor, que permite armazenar múltiplos valores de um mesmo tipo.

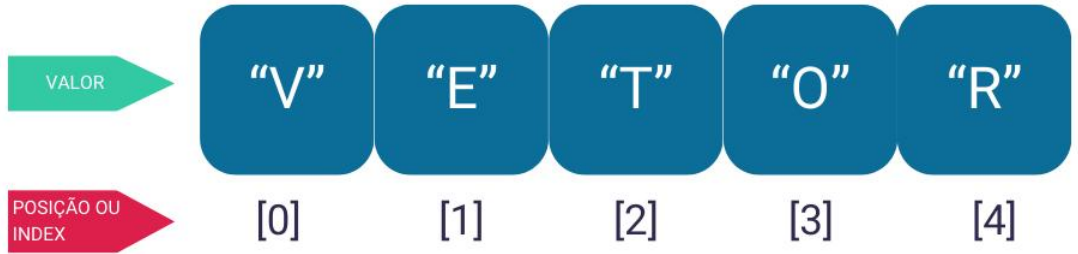
- Sendo assim, um vetor é uma **variável composta e homogênea** que permite armazenar um **conjunto** de valores relacionados em uma **única variável**.
- O vetor pode também pode ser chamado de *array* ou matriz unidimensional.
- Os valores são organizados em posições sequenciais, cada uma identificada por um **índice numérico**.
- Podemos chamar cada uma desses valores de **elementos**.

```
1 tipo nome_vetor[quantidade_posicoes];
```

Código 1: Estrutura básica.







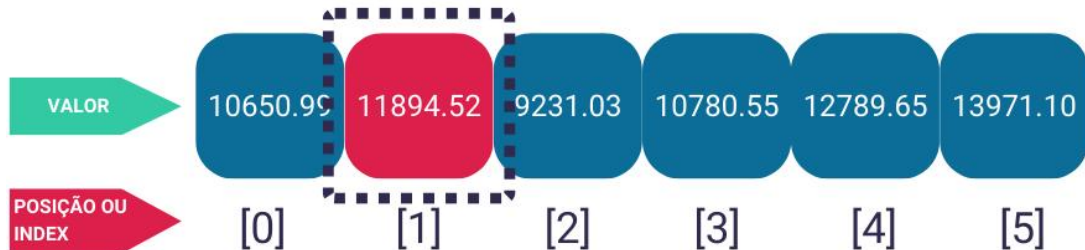
Declaração

```
1 int main(){  
2     int idades[5];  
3     float lucro_semestral[6];  
4     char nome_animal[10];  
5     return 0;  
6 }
```

Código 2: Declaração.

Atribuição

Com acessar a segunda posição no vetor?



```
1 int main(){  
2     int idades[5];  
3     float lucro_semestral[6];  
4     char nome_animal[10];  
5  
6     idades[0] = 10;  
7     lucro_semestral[2] = 9231.03;  
8     nome_animal[0] = 'C';  
9     return 0;  
10 }
```

Código 3: Atribuição.

Definição

```
1 int main(){
2     int idades[5] = {20, 21, 18, 19, 20};
3     float lucro_semestral[6] = {10650.99, 11894.52, 9231.03,
4     ↪ 10780.55, 12789.65, 13971.10};
5     char nome_animal[10] = "Coelho";
6     return 0;
}
```

Código 4: Definição.

Manipulação

```
1 #include <stdio.h>
2 int main(){
3     int idades[5];
4     idades[0] = 20;
5     idades[1] = 19;
6     idades[2] = 18;
7     idades[3] = 21;
8     idades[4] = 18;
9     printf("%d\n", idades[0]);
10    printf("%d\n", idades[1]);
11    printf("%d\n", idades[2]);
12    printf("%d\n", idades[3]);
13    printf("%d\n", idades[4]);
14    return 0;
15 }
```

Código 5: Exemplo idades.



```
1 #include <stdio.h>
2 int main(){
3     int idades[100];
4     for (int i = 0; i < 100; i++)
5     {
6         printf("Qual idade?\n");
7         scanf("%d", &idades[i]);
8     }
9     return 0;
10 }
```

Código 6: Exemplo idades entrada.

Atenção!

Lembrando que a primeira posição do vetor é acessada pelo índice zero. Portanto o contador do laço deve ser inicializado com valor zero.

```
1 #include <stdio.h>
2 int main(){
3     int idades[100];
4     for (int i = 0; i < 100; i++)
5     {
6         printf("%d\n", idades[i]);
7     }
8     return 0;
9 }
```

Código 7: Exemplo idades saída.

```
1 #include <stdio.h>
2 int main(){
3     int idades[100];
4     for (int i = 0; i < 100; i++)
5     {
6         printf("Qual idade?\n");
7         scanf("%d", &idades[i]);
8     }
9     for (int i = 0; i < 100; i++)
10    {
11        printf("%d\n", idades[i]);
12    }
13    return 0;
14 }
```

Código 8: Exemplo idades.

```
1 #include <stdio.h>
2 #include <time.h>
3 int main(){
4     int numero[100];
5     srand(time(NULL));
6     for (int i = 0; i < 100; i++){
7         {
8             printf("Qual numero?\n");
9             numero[i] = (rand() % 100) + 1;
10        }
11    for (int i = 0; i < 100; i++){
12        {
13            printf("%d\n", numero[i]);
14        }
15    return 0;
16 }
```

Código 9: Atribuição de valores randômicos.

Exemplos

```
1 #include <stdio.h>
2 int main() {
3     float notas[5] = {7.5, 8.0, 9.2, 6.5, 7.8};
4     float soma = 0.0;
5     float media;
6     for (int i = 0; i < 5; i++) {
7         soma += notas[i];
8     }
9     media = soma / 5;
10    printf("Média das notas: %.2f\n", media);
11    return 0;
12 }
```

Código 10: Exemplo média de notas.

```
1 #include <stdio.h>
2 int main() {
3     int numeros[5] = {10, 20, 30, 25, 15};
4     int maior = -99;
5     for (int i = 1; i < 5; i++) {
6         if (numeros[i] > maior) {
7             maior = numeros[i];
8         }
9     }
10    printf("O maior valor no vetor é: %d\n", maior);
11    return 0;
12 }
```

Código 11: Exemplo maior valor.

```
1 #include <stdio.h>
2 int main() {
3     int a[5] = {1, 2, 3, 4, 5};
4     int b[5] = {5, 4, 3, 2, 1};
5     int c[5];
6     for (int i = 0; i < 5; i++) {
7         c[i] = a[i] + b[i];
8     }
9     printf("Soma dos vetores:\n");
10    for (int i = 0; i < 5; i++) {
11        printf("%d ", c[i]);
12    }
13    return 0;
14 }
```

Código 12: Exemplo soma de vetores.

```
1 #include <stdio.h>
2 #define TAMANHO_VETOR 5
3 int main() {
4     int vetor[TAMANHO_VETOR] = {10, 20, 30, 40, 50};
5     for (int i = 0; i < TAMANHO_VETOR; i++) {
6         printf("%d",vetor[i]);
7     }
8     return 0;
9 }
```

Código 13: Exemplo tamanho definido como constante.

Strings

O que é uma String?

- › Um vetor de caracteres é uma string quando armazenamos uma **sequência de caracteres** em posições contíguas na memória, terminada com o caractere nulo `\0` para indicar o final da sequência.

```
1 #include <stdio.h>
2 int main(){
3     char nome_animal[10];
4     printf("Digite o nome de um animal:\n");
5     for (int i = 0; i < 10; i++){
6         scanf(" %c", &nome_animal[i]);
7     }
8     return 0;
9 }
```

Código 14: Exemplo de entrada inadequada de vetor de char.

```
1 #include <stdio.h>
2 int main(){
3     char nome_animal[10];
4     printf("Digite o nome de um animal:\n");
5     scanf(" %c", &nome_animal[i]);
6     for (int i = 0; i < 10; i++){
7         printf("%c", nome_animal[i]);
8     }
9     return 0;
10 }
```

Código 15: Exemplo de saída inadequada de vetor de char.

```
1 #include <stdio.h>
2 int main(){
3     char nome_animal[10];
4     printf("Digite o nome de um animal:\n");
5     scanf("%s", nome_animal);
6     return 0;
7 }
```

Código 16: Exemplo de entrada de string.

```
1 #include <stdio.h>
2 int main()
3 {
4     char nome_animal[10];
5     printf("Digite o nome de um animal:\n");
6     scanf("%s", nome_animal);
7     printf("%s\n", nome_animal);
8     return 0;
9 }
```

Código 17: Exemplo de saída de string.

(Deitel; Deitel, 2011) - Capítulo/Seção 6.2, 6.3 e 6.4.



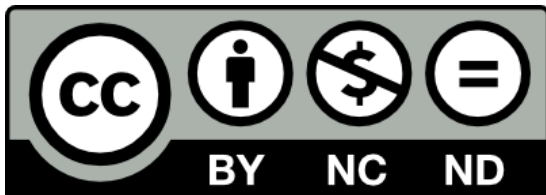
 DE OLIVEIRA, J.F.; MANZANO, J.A.N.G. **Algoritmos: Lógica para Desenvolvimento de Programação de Computadores**. 16. ed. São Paulo: Editora Érica, 2004.

 DE SOUZA, M.A.F. *et al.* **Algoritmos e Lógica de Programação**. São Paulo: Thomson Learning, 2004.

 DEITEL, Paul; DEITEL, Harvey. **C: Como Programar**. 6. ed. São Paulo: Pearson Universidades, 2011.

 MEDINA, M.; FERTIG, C. **Algoritmos e Programação – Teoria e Prática**. São Paulo: Novatec, 2005.

Estes slides estão protegidos por uma licença Creative Commons



Este modelo foi adaptado de Maxime Chupin.

Fundamentos de Programação

Linguagem de Programação C
Vetor