

◀ VOLTAR



Vetores

Apresentar o conceito de variável indexada unidimensional utilizando a linguagem C.

vetores
vectors, vector
Translated from Portuguese [More »](#)

NESTE TÓPICO



Marcar
tópico



Introdução

Na maioria da vezes, os dados são manipulados em variáveis simples, que são capazes de armazenar um único valor. Existem casos em que precisamos armazenar não um único valor, mas uma coleção de valores. Por exemplo, queremos armazenar a média de cada aluno em uma disciplina. Seria muito trabalhoso e de difícil manutenção criarmos uma variável para cada aluno da turma.

Vetor, também conhecido como variável indexada unidimensional, é uma estrutura de dados que permite o agrupamento de vários dados do mesmo tipo dentro de uma mesma variável. Isso facilita o armazenamento e a manipulação de uma coleção de dados do mesmo tipo.

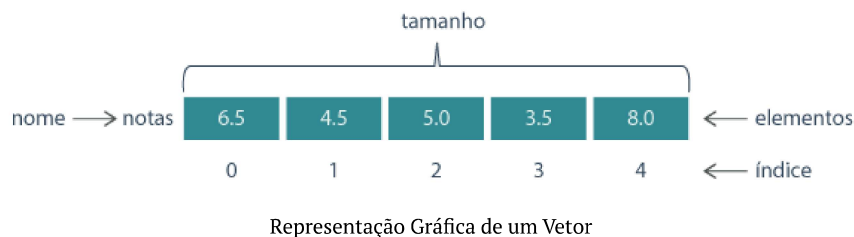
Definição

Um vetor é uma coleção de variáveis do mesmo tipo que são referenciadas por um único nome. Um vetor possui um tamanho, definido na sua declaração, e que nunca se altera. O tamanho do vetor indica quanto elementos ele pode armazenar. Um elemento (valor) armazenado em um vetor é acessado por meio de um índice. Vetor é uma estrutura unidimensional (precisa somente de um índice para acessar um elemento), estática (o tamanho deve ser pré-definido antes de sua utilização),



homogênea (todos os elementos devem ter o mesmo tipo de dados) e temporária (a variável declarada como vetor só existe em tempo de execução).

Em C, os dados do vetor são armazenados de forma contígua na memória. A figura a seguir apresenta a representação gráfica de um vetor.



Declaração de Vetor

Como uma variável simples, um vetor deve ser declarado antes de sua utilização. A declaração de um vetor é feita indicando o tipo de dado que o vetor irá armazenar, o seu nome e o seu tamanho.

A forma geral de declaração de um vetor em linguagem C é:

tipo nome[tamanho];

Onde, tipo é um tipo qualquer de dados, nome é o nome pelo qual o vetor será referenciado e tamanho é a quantidade de elementos que o vetor pode conter. Em C, o primeiro elemento tem índice 0 e o último elemento tem índice tamanho-1. Os índices são sempre representados por valores inteiros positivos.

Exemplos:

- Declaração de um vetor denominado *números* de 10 posições de valores inteiros.

int numeros[10];

- Declaração de um vetor denominado *notas* de 15 posições de valores do tipo float (real).

float notas[15];

Assim como uma variável simples, um vetor pode ser inicializado no momento de sua declaração. Nesse caso, os valores iniciais devem ser fornecidos entre chaves e separados por vírgula. O trecho de código a seguir mostra a declaração e inicialização de um vetor de valores reais.

float moeda[5] = {1.00, 0.50, 0.25, 0.10, 0.05};

Os valores da inicialização são armazenados no vetor a partir da posição 0 na ordem em que são fornecidos. Por exemplo, o valor 0.10 é armazenado em `moeda[3]`.



Os valores utilizados para a inicialização de vetor devem ser constantes. O uso de variáveis causa erro de compilação.

Referenciando um Elemento do Vetor

Os elementos de um vetor devem ser referenciados de forma individualizada. Não é possível referenciar todos os elementos ao mesmo tempo. Um elemento é referenciado pelo nome do vetor seguido do índice onde o elemento está armazenado entre colchetes.

O exemplo abaixo referencia o terceiro elemento do vetor denominado notas. O terceiro elemento é referenciado pelo índice 2, já que o primeiro elemento tem índice 0.

```
notas[2]
```

Podemos atribuir um valor a um elemento do vetor da seguinte forma:

```
notas[2] = 7.5;
```

O exemplo acima atribui o valor 7.5 ao terceiro elemento do vetor notas.

Vídeo

0:00 / 1:14



Lendo Dados para um Vetor

Imagine que temos um vetor denominado notas que pode armazenar 5 elementos do tipo float. Queremos solicitar ao usuário que informe os dados para preencher esse vetor. O trecho de código a seguir lê os dados que serão



armazenados no vetor.

```
1. int i;  
2. for (i = 0; i < 5; i++) {  
3.     printf ("Digite a nota do aluno %d: ", i);  
4.     scanf ("%f", &notas[i]);  
5. }
```

O laço for é usado para o processo de solicitar uma nota ao usuário e armazená-la no vetor. No exemplo acima, esse processo é repetido 5 vezes. O valor informado pelo usuário será armazenado na posição de índice do valor atual da variável i. Geralmente, utilizamos a estrutura de repetição for para a manipulação do vetor.

Acessando Dados de um Vetor

Uma vez preenchido o vetor queremos manipular os seus elementos ou apresenta-los na tela. Utilizando o vetor de notas preenchido na seção anterior, vamos agora calcular a média das notas e também apresentar os elementos do vetor na tela. Analise o seguinte trecho de código.

```
1. float soma = 0;  
2. int i;  
3. for (i = 0; i < 5; i++) {  
4.     // soma a variável o elemento de índice i do vetor  
5.     soma = soma + notas[i];  
6.     // apresenta o elemento de índice i do vetor  
7.     printf ("Nota %d = %.1f\n", i, notas[i]);  
8. }  
9. float media = soma / 5;  
10. printf ("Media das notas = %.1f\n", media);
```

O trecho de código anterior utiliza o laço for para percorrer o vetor. Em cada iteração do laço, um elemento do vetor de notas é somado a variável soma e também o elemento somado é apresentado na tela. Ao final do laço é calculada a média das notas e apresentada na tela.

Exemplo 1

A seguir, apresentamos um exemplo completo de um programa em linguagem C que trabalha com vetor. O programa lê 5 notas e armazena-as em um vetor, calcula a média aritmética e apresenta a média e os dados do vetor na tela.



```
1. #include <stdio.h>
2. #include <stdlib.h>
3. #define TAM 5
4. main () {
5.     // declaração do vetor
6.     float notas[TAM];
7.     int i;
8.     // leitura dos dados para o vetor
9.     for (i = 0; i < TAM; i++) {
10.         printf ("Digite a nota do aluno %d: ", i);
11.         scanf ("%f", &notas[i]);
12.     }
13.     // calcula a soma das notas e apresenta cada nota na tela
14.     float soma = 0;
15.     for (i = 0; i < TAM; i++) {
16.         soma = soma + notas[i];
17.         printf ("Nota %d = %.1f\n", i, notas[i]);
18.     }
19.     // calcula e apresenta a média das notas
20.     float media = soma / TAM;
21.     printf ("Media das notas = %.1f\n", media);
22.     system ("PAUSE");
23. }
```

Exemplo 2

Vamos estudar outro exemplo de utilização de vetor. O enunciado é: Criar um programa que leia 8 elementos inteiros em um vetor A. Construir um vetor B do mesmo tipo e tamanho com os elementos do vetor A multiplicados por 3. O elemento B[0] recebe o elemento A[0] * 3, o elemento B[1] recebe o elemento A[1] * 3 e assim por diante, até a posição 7 do vetor. Apresentar os elementos do vetor B.

```
1. #include <stdio.h>
2. #include <stdlib.h>
3. #define TAM 8
4. main () {
5.     // declaração dos dois vetores
6.     int A[TAM], B[TAM];
7.     int i;
8.     // leitura do vetor A
9.     printf ("*** Informe os dados do vetor A ***\n");
10.    for (i = 0; i < TAM; i++) {
11.        printf ("A[%d]: ", i);
12.        scanf ("%d", &A[i]);
13.    }
14.    // cálculo do vetor B
15.    for (i = 0; i < TAM; i++) {
16.        B[i] = A[i] * 3;
17.    }
18.    // apresentação dos elementos do vetor B
19.    printf ("\n*** Dados do vetor B *** \n");
20.    for (i = 0; i < TAM; i++) {
21.        printf ("B[%d] = %d\n", i, B[i]);
22.    }
23.    system ("PAUSE");
24. }
```



Exemplo 3

O programa a seguir armazena 10 elementos reais em um vetor e, depois, apresenta seus valores na tela.

```
1. #include <stdio.h>
2. #include <stdlib.h>
3. #define TAM 10
4. main () {
5.     // declaração de um vetor de 10 elementos reais
6.     float vetor[TAM];
7.     int i;
8.     // leitura dos dados para o vetor
9.     printf ("*** Informe os valores para o vetor ***\n");
10.    for (i = 0; i < TAM; i++) {
11.        printf ("Valor[%d]: ", i);
12.        // não esquecer de colocar o & antes do nome do vetor
13.        scanf ("%f", &vetor[i]);
14.    }
15.    // apresentação dos dados do vetor
16.    printf ("\n\n*** Dados do vetor ***\n");
17.    for (i = 0; i < TAM; i++) {
18.        printf ("%2f\n", vetor[i]);
19.    }
20.    system ("PAUSE");
21. }
```

Exemplo 4

O programa a seguir faz a leitura de dois vetores (denominados A e B) com 20 elementos inteiros. No vetor C, cada elemento é calculado como a subtração de um elemento do vetor A com um elemento correspondente do vetor B, ou seja, a operação de processamento é baseada na operação $C[I] = A[I] - B[I]$. Ao final, os elementos do vetor C são apresentados.



```
1. #include <stdio.h>
2. #include <stdlib.h>
3. #define TAM 20
4. main () {
5.     // declaração dos vetores
6.     int A[TAM], B[TAM], C[TAM];
7.     int i;
8.     // leitura do vetor A
9.     printf ("*** Informe os valores do vetor A ***\n");
10.    for (i = 0; i < TAM; i++) {
11.        printf ("A[%d]: ", i);
12.        scanf ("%d", &A[i]);
13.    }
14.    // leitura do vetor B
15.    printf ("\n*** Informe os valores do vetor B ***\n");
16.    for (i = 0; i < TAM; i++) {
17.        printf ("B[%d]: ", i);
18.        scanf ("%d", &B[i]);
19.    }
20.    // calculo e impressao do vetor C
21.    printf ("\n*** Valores do vetor C ***\n");
22.    for (i = 0; i < TAM; i++) {
23.        C[i] = A[i] - B[i];
24.        printf ("C[%d] = %d\n", i, C[i]);
25.    }
26.    printf ("\n\n");
27.    system ("PAUSE");
28. }
```

Agora que você já estudou essa aula acesse a plataforma AVA, resolva os exercícios e verifique o seu conhecimento. Caso fique alguma dúvida, leve a questão ao Fórum e divida com seus colegas e professor.

Quiz

Exercício

Vetores

INICIAR ➤



Quiz

Exercício Final

Vetores

INICIAR ➤

Referências

MIZRAHI, V. V. *Treinamento em linguagem C*, São Paulo: Pearson, 2008.

SCHILDT, H. C – *Completo e Total*, São Paulo: Pearson, 2006.



Avalie este tópico



Biblioteca

(<https://www.uninove.br/conhec-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/>)

Portal Uninove

(<http://www.uninove.br>)

Mapa do Site

Ajuda?

[\(https://ava.uninove.br/curso/\)](https://ava.uninove.br/curso/)

Mapa do Curso ➤

© Todos os direitos reservados