

## Vetores e strings

## Variáveis compostas

**Variáveis compostas** são utilizadas para armazenar diversos dados em uma única estrutura na memória, por isso também são chamadas de estrutura de dados. As variáveis compostas possuem **índices** que as diferenciam, portanto, uma variável composta possui um endereço na memória e índices para identificar seus subespaços.

## Vetores

Um vetor é uma coleção de **variáveis de mesmo tipo**, acessíveis com um **único nome** e armazenados contiguamente na memória.

A individualização de cada variável de um vetor é feita através do uso de **índices**.

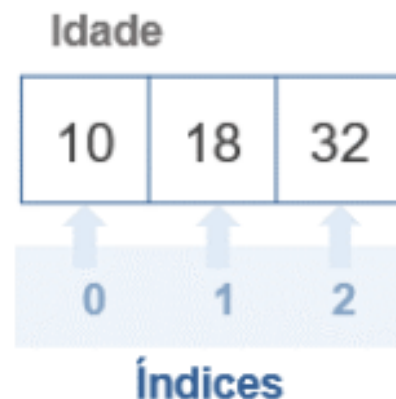


Fonte: elaborada pela autora.

## Vetores

Quando utilizamos a linguagem C, todas as posições do vetor são contíguas na memória e o endereço mais baixo corresponde ao primeiro elemento e o endereço mais alto corresponde ao último elemento.

Um vetor com N posições terá seu índice variando de **0(zero) até N-1**. No vetor abaixo declarado como idade, com capacidade de armazenar 3 valores, seu índice varia de 0 até 2.



Idade[0] = 10  
Idade[1] = 18  
Idade[2] = 32

## Declaração de vetores

Em C, um vetor é um conjunto de variáveis de um mesmo tipo que possuem um **nome identificador** e um **índice de referência**.

**Sintaxe:** A sintaxe para a declaração de um vetor é a seguinte:

**tipo nome[tam];**

## Declaração de vetores

onde:

**tipo** é o tipo dos elementos do vetor: int, float, double ...

**nome** é o nome identificador do vetor. As regras de nomenclatura de vetores são as mesmas usadas em variáveis.

**tam** é o tamanho do vetor, isto é, o número de elementos que o vetor pode armazenar.

**Exemplo:** Veja as seguintes declarações:

```
int idade[100];           // declara um vetor chamado 'idade' do tipo
                           // 'int' que recebe 100 elementos.
float nota[25];           // declara um vetor chamado 'nota' do tipo
                           // 'float' que pode armazenar 25 números.
char nome[80];            // declara um vetor chamado 'nome' do tipo
                           // 'char' que pode armazenar 80 caracteres.
```

## Referência a elementos de um vetor

Cada elemento do vetor é referenciado pelo **nome** do vetor seguido de um índice **inteiro**. O primeiro elemento do vetor tem índice 0(zero) e o **último** tem índice tam-1. O índice de um vetor deve ser **inteiro**.

Inicialização de vetores:

Assim como podemos inicializar variáveis (por exemplo: `int j = 3;`), podemos inicializar vetores.

**Sintaxe:** A sintaxe para a inicialização dos elementos de um vetor é:

`tipo nome[tam] = {lista de valores};`

onde:

lista de valores é uma lista, separada por vírgulas, dos valores de cada elemento do vetor.

## Referência a elementos de um vetor

### Exemplo:

Veja as seguintes inicializações:

```
int dia[7] = {12,30,14,7,13,15,6};  
float nota[5] = {8.4,6.9,4.5,4.6,7.2};  
char vogal[5] = {'a','e','i','o','u'};  
char frase[101] = {"disciplina de algoritmos"};
```

Opcionalmente, podemos inicializar os elementos do vetor enumerando-os um a um.

**Exemplo:** Observe que estas duas inicializações são possíveis:

```
int cor_menu[4] = {BLUE,YELLOW,GREEN,GRAY};  
    ou  
int cor_menu[4];  
cor_menu[0] = BLUE;  
cor_menu[1] = YELLOW;  
cor_menu[2] = GREEN;  
cor_menu[3] = GRAY;
```



## Strings

O uso mais comum de vetores unidimensionais é como string de caracteres. Em C, uma string é definida como um vetor de caracteres que é terminada por um **nulo**. Um nulo é especificado como `'\0'` e geralmente é zero. Sendo assim, você precisa declarar vetores de caracteres como sendo um caractere mais longo que a maior string que elas devem guardar. Por exemplo, para declarar um vetor **str** que guarda uma string de 10 caracteres, você precisar declarar como:

```
char str[11];
```

Isto reserva espaço para o nulo no final da string.

## Strings (exemplo de inicialização)

Um exemplo de inicialização de strings é:

```
char palavras[24] = "Eu sou um programador C";
```

é o mesmo que escrever:

```
char palavras[24] = {'E','u',' ','s','o','u',' ','u','m',' ','p','r','o','g','r','a','m','a','d','o','r',' ','C','\0'};
```

Como todas as strings em C terminam com um nulo, você deve ter certeza de que o vetor a ser declarado é longo o bastante para incluir o nulo. Isso explica porque a palavra tem o comprimento de 24 caracteres, embora "Eu sou um programador C" tenha apenas 23.

## Função strlen ()

A função strlen () calcula o comprimento de uma determinada string.

A strlen() função recebe uma string como argumento e retorna seu comprimento. O valor retornado é do tipo inteiro sem sinal.

É definido no arquivo de cabeçalho [<string.h>](#)

## Exemplo strlen ()

```
#include <stdio.h>
#include <string.h>
int main()
{
    char a[20]="Program";
    char b[20]={'P','r','o','g','r','a','m','\0'};

    // using the %zu format specifier to print size_t
    printf("Length of string a = %d \n",strlen(a));
    printf("Length of string b = %d \n",strlen(b));

    return 0;
}
```

## Referência a elementos de um vetor

### Exemplo:

```
1  #include <stdio.h>
2  main()
3  {
4      float altura[3] = {1,1.5,1.7};
5
6      printf("\n Vetor altura[0] = %.2f",altura[0]);
7      printf("\n Vetor altura[1] = %.2f",altura[1]);
8      printf("\n Vetor altura[2] = %.2f",altura[2]);
9  }
```

## Exemplo de utilização de vetor

Desenvolva um programa que realize a entrada de dados de 5 idades (inteiro positivo maior que 0). Utilize um vetor para armazenar todas as idades. Após a entrada de todos os valores realize as seguintes tarefas com o vetor:

- Some todos os valores do vetor e apresente o resultado para o usuário;
- Identifique a menor idade e a maior idade contida no vetor, e mostre os valores para o usuário;
- Identifique a diferença de idade entre a menor idade e a maior idade;
- Calcule e mostre para o usuário a idade média de todas as idades cadastradas no vetor.

## Questionário

1. O que é um vetor?
2. Como é possível acessar individualmente cada elemento de um vetor?
3. O que são strings na linguagem C?
4. Qual o objetivo do nulo em uma string?
5. Por que é necessário criarmos uma string sempre maior do que o máximo de caracteres que ela irá comportar, e qual a quantidade de bytes a ser incluída além do máximo?
6. Qual a sintaxe para a criação de uma string?
7. Exemplifique uma das formas de inicialização de uma string?
8. Descreva a sintaxe de declaração de vetores.