



# **VETOR**

**Nome: Thaís de Almeida**

**Matrícula: 31611ECA029**

**Curso: Engenharia de Agrimensura e Cartográfica**

# DEFINIÇÃO

- Vetor também é conhecido como variável composta homogênea unidimensional. Isso quer dizer que se trata de um conjunto de variáveis de mesmo tipo, que possuem o mesmo identificador (nome) e são alocadas sequencialmente na memória. Como as variáveis tem o mesmo nome, o que as distingue é um índice que referencia sua localização dentro da estrutura.



# DECLARAÇÃO DE VETOR

- Os vetores em C/C++ são identificados pela existência de colchetes logo após o nome da variável no momento da declaração. Dentro dos colchetes, deve-se colocar o número de elementos que o vetor poderá armazenar.



# DECLARAÇÃO DE VETOR

## Exemplo:

```
int vet[10];
```

vet	10	5	3	8	1	19	44	21	2	7
	0	1	2	3	4	5	6	7	8	9

O vetor chamado *vet* possui dez posições, começando pela posição 0 e indo até a posição 9 (tamanho do vetor - 1). Em cada posição poderão ser armazenados números inteiros, conforme especificado pelo tipo *int* na declaração.



# DECLARAÇÃO DE VETOR

## Exemplo:

```
char x[5];
```

X	A	*	2	@	k
	0	1	2	3	4

O vetor chamado *x* possui cinco posições, começando pela posição 0 e indo até a posição 4 (tamanho do vetor -1). Em cada posição poderão ser armazenados caracteres, conforme especificado pelo tipo *char* na declaração.



# DECLARAÇÃO DE VETOR

- Na linguagem C/C++ não existe o tipo de dado *string*. Para poder armazenar em uma cadeia de caracteres deve-se declarar um vetor de *char*, sendo que cada posição é equivalente a um caractere.
- Deve-se lembrar de definir uma posição a mais que a necessária para armazenar a marca de finalização de cadeia (`\0`).



# ATRIBUINDO VALORES AO VETOR

- As atribuições em vetor exigem que seja informada em qual das suas posições o valor ficará armazenado.
- A primeira posição de um vetor em C/C++ sempre tem índice 0.

```
vet[0] = 1;   atribui o valor 1 à primeira posição do vetor (lembre-se que o vetor começa na posição 0)  
x[3] = 'b';  atribui a letra b à quarta posição do vetor (lembre-se que o vetor começa na posição 0)
```



# PREENCHENDO UM VETOR

- Significa atribuir valores a todas as suas posições;
- Deve-se implementar um mecanismo que controle o valor do índice;





# PREENCHENDO UM VETOR

Exemplo:

```
for (i=0; i<10; i++)  
    cin >> vetor[i];
```

A estrutura de repetição *for* foi utilizada para garantir que a variável *i* assuma todos os valores possíveis para o índice do vetor (de 0 a 9). Assim, para cada execução da repetição, será utilizada uma posição diferente do vetor.



# MOSTRANDO OS ELEMENTOS DO VETOR

- Mostrar os valores contidos em um vetor também exige a utilização do índice.



# MOSTRANDO OS ELEMENTOS DO VETOR

Exemplo:

```
for (i=0; i<10; i++)  
    cout << vetor[i];
```

A estrutura de repetição *for* foi utilizada para garantir que a variável *i* assuma todos os valores possíveis para o índice do vetor (de 0 a 9). Assim, para cada execução da repetição, será utilizada uma posição diferente, mostrando todos os valores do vetor.



# EXERCÍCIOS

- 1) Leia a nota de 5 alunos e calcule as notas maiores que a média da turma:



# EXERCÍCOS

- 2)

