

Vetores

Prof: M.Sc Mário Angel Praia Garcia

1. Vetores

- Conceito

- Vetor, também conhecido por **Array**, é um conjunto de elementos, todos de mesmo tipo, que podem ser acessados individualmente a partir de único nome.

- Exemplos:

- Vetor de valores inteiros

1000	5	999
------	---	-----

- Vetor de valores reais

1,5	560,0	0,3333
-----	-------	--------

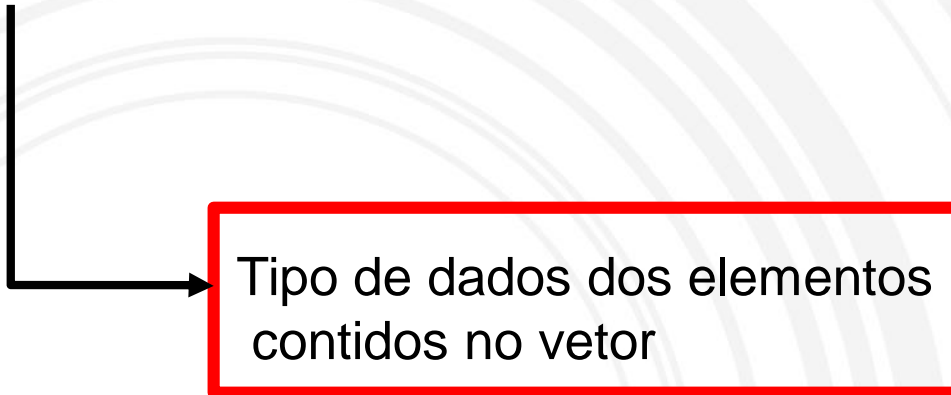
- Vetor de caracteres

A	b	D
---	---	---

1. Vetores

- Declaração de Vetores – uma dimensão:

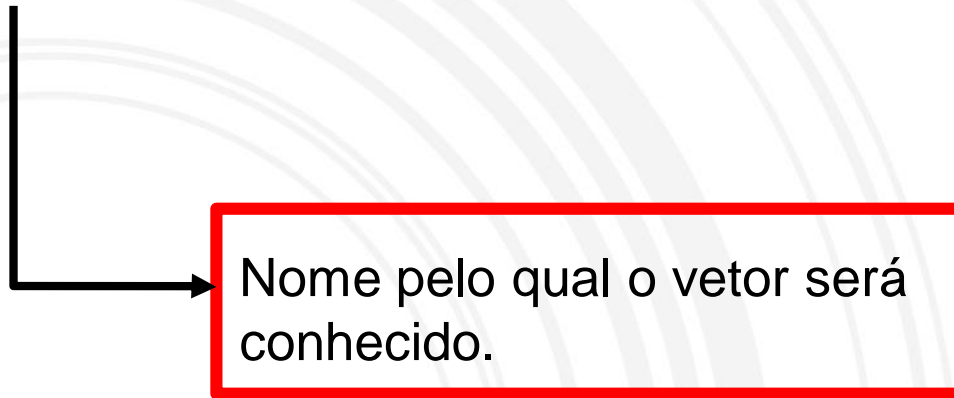
tipo **nome_do_vetor** [nº de elementos];



1. Vetores

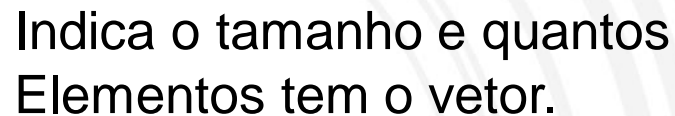
- Declaração de Vetores – uma dimensão:

tipo **nome_do_vetor** [nº de elementos];



1. Vetores

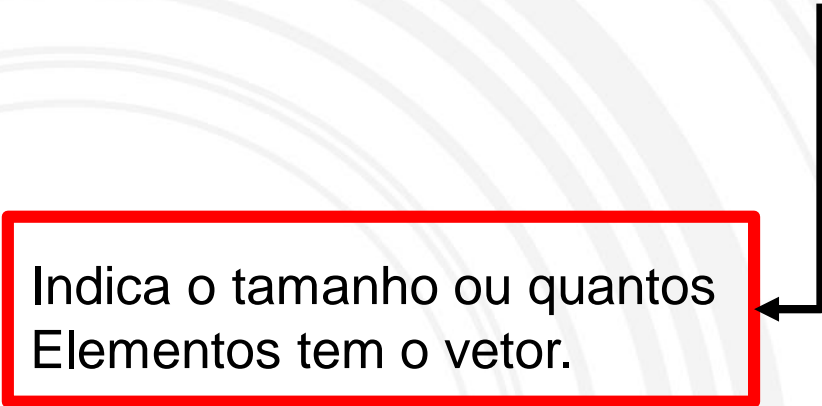
- Declaração de Vetores – uma dimensão:
tipo **nome_do_vetor** [nº de elementos];



Indica o tamanho e quantos
Elementos tem o vetor.

1. Vetores

- Declaração de Vetores – uma dimensão:
tipo **nome_do_vetor** [nº de elementos];



Indica o tamanho ou quantos
Elementos tem o vetor.

OBS: Um vetor pode conter elementos de qualquer tipo. No entanto, os elementos do vetor são todos do mesmo tipo, como definido em sua declaração.

1. Vetores

- Exemplos de declaração de vetores:
 - **int** g[20] ;
 - Vetor de números inteiros, chamado **g**, que possui tamanho 20 ou possui 20 elementos.
 - **float** renda [100] ;
 - Vetor de números reais, chamado **renda**, que possui tamanho 100 ou possui 100 elementos.

1. Vetores

- Como um vetor é exemplificado pelos seus índices:
 - `int vetor[3];`



OBS: Em C os índices de um vetor com n elementos variam SEMPRE entre 0 e $n - 1$.

1. Vetores

- Como um vetor é exemplificado pelos seus índices:
 - `int vetor[3];`



OBS: Em C os índices de um vetor com n elementos variam **SEMPRE** entre 0 e $n - 1$.

OBS: As posições de um vetor **SEMPRE** podem ser acessadas através de índice colocado entre os colchetes `[]`.

1. Vetores

- Como um vetor é exemplificado pelos seus índices:
 - `int vetor[3];`



OBS: Em C os índices de um vetor com n elementos variam **SEMPRE** entre 0 e $n - 1$.

OBS: As posições de um vetor **SEMPRE** podem ser acessadas através de índice colocado entre os colchetes `[]`.

OBS: O índice do primeiro elemento, de qualquer vetor em C, é **SEMPRE** 0 (zero) e o n -ésimo elemento (último elemento) está sempre na posição $n - 1$.

1. Vetores

- Inserindo valores no vetor utilizando seus índices:
 - **int** vetor[3] ;

vetor[0] = 123;



1. Vetores

- Inserindo valores no vetor utilizando seus índices:
 - **int** vetor[3] ;

vetor[0] = 123;



1. Vetores

- Inserindo valores no vetor utilizando seus índices:
 - **int** vetor[3] ;

`vetor[2] = vetor[0] * 2;`



1. Vetores

- Inserindo valores no vetor utilizando seus índices:
 - **int** vetor[3] ;

vetor[2] = vetor[0] * 2;



1. Vetores

- Inserindo valores no vetor utilizando seus índices:
 - **int** vetor[3] ;

vetor[1] = vetor[0] + vetor[2];



1. Vetores

- Inserindo valores no vetor utilizando seus índices:
 - **int** vetor[3] ;

`vetor[1] = vetor[0] + vetor[2];`

123	369	246
vetor[0]	vetor[1]	vetor[2]

1. Vetores

- Inserindo valores no vetor utilizando o **for**:
 - **int** vetor[3] ;

```
for ( i = 0; i < 3; i++)
```

```
    vetor[i] = i;
```



1. Vetores

- Inserindo valores no vetor utilizando o **for**:
 - `int vetor[3];`

```
for ( i = 0; i < 3; i++)
```

```
vetor[i] = i;
```



1. Vetores

- Inserindo valores no vetor utilizando o **for**:
 - `int vetor[3];`

`for (i = 0; i < 3; i++)`

`vetor[i] = i;`



1. Vetores

- Inserindo valores no vetor utilizando o **for**:
 - `int vetor[3];`

```
for ( i = 0; i < 3; i++)
```

```
vetor[i] = i;
```

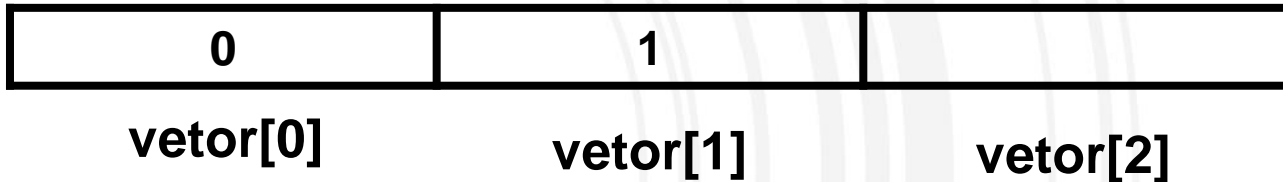


1. Vetores

- Inserindo valores no vetor utilizando o **for**:
 - `int vetor[3];`

`for (i = 0; i < 3; i++)`

`vetor[i] = i;`

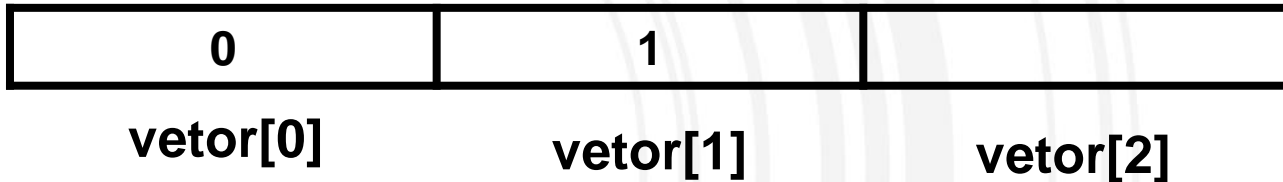


1. Vetores

- Inserindo valores no vetor utilizando o **for**:
 - `int vetor[3];`

```
for ( i = 0; i < 3; i++)
```

```
vetor[i] = i;
```

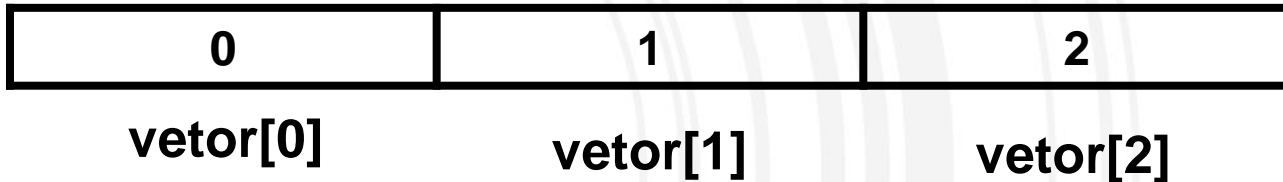


1. Vetores

- Inserindo valores no vetor utilizando o **for**:
 - **int** vetor[3] ;

for (i = 0; i < 3; i++)

vetor[i] = i;

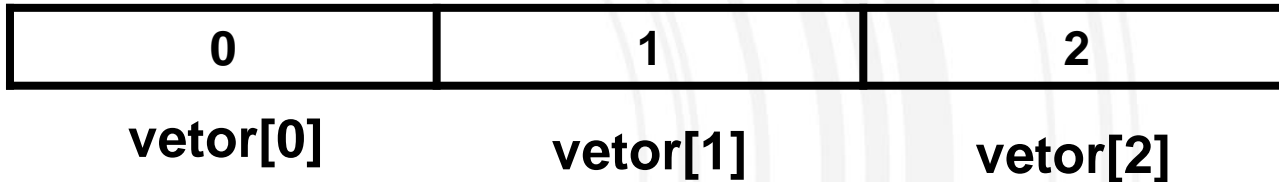


1. Vetores

- Inserindo valores no vetor utilizando o **for**:
 - **int** vetor[3] ;

```
for ( i = 0; i < 3; i++)
```

```
vetor[i] = i;
```



OBS: Outros laços de repetição(while, do..while) também podem ser usados para inserir valores em vetores. O scanf também é ser utilizado para ler valores para um vetor

1. Vetores

- Imprimindo valores do vetor utilizando o **for**:
 - **int** vetor[3] ;



```
for ( i = 0; i < 3; i++)  
    printf("vet[%d] = %d", i, vetor[i]);
```

1. Vetores

- Imprimindo valores do vetor utilizando o **for**:
- `int vetor[3];`



```
for ( i = 0; i < 3; i++)
```

```
printf("vet[%d] = %d\n", i, vetor[i]);
```

Saída do programa:

1. Vetores

- Imprimindo valores do vetor utilizando o **for**:
- `int vetor[3];`



```
for ( i = 0; i < 3; i++)
```

```
    printf("vet[%d] = %d\n", i, vetor[i]);
```

Saída do programa: vet[0] = 0

1. Vetores

- Imprimindo valores do vetor utilizando o **for**:
- `int vetor[3];`



```
for ( i = 0; i < 3; i++)
```

```
printf("vet[%d] = %d\n", i, vetor[i]);
```

Saída do programa: `vet[0] = 0`

1. Vetores

- Imprimindo valores do vetor utilizando o **for**:
- `int vetor[3];`



```
for ( i = 0; i < 3; i++)
```

```
printf("vet[%d] = %d\n", i, vetor[i]);
```

Saída do programa: `vet[0] = 0`
`vet[1] = 1`

1. Vetores

- Imprimindo valores do vetor utilizando o **for**:
- `int vetor[3];`



```
for ( i = 0; i < 3; i++)
```

```
printf("vet[%d] = %d\n", i, vetor[i]);
```

Saída do programa: `vet[0] = 0`
`vet[1] = 1`

1. Vetores

- Imprimindo valores do vetor utilizando o **for**:
- `int vetor[3];`



```
for ( i = 0; i < 3; i++)
```

```
printf("vet[%d] = %d\n", i, vetor[i]);
```

Saída do programa: `vet[0] = 0`
`vet[1] = 1`
`vet[2] = 2`

1. Vetores

- Inicializando um vetor:
 - Tal como as variáveis, os vetores quando são criados contêm valores aleatórios (LIXO) em cada uma das suas posições.

$\text{tipo var}[n] = \{ \text{valor}_1, \text{valor}_2, \dots, \text{valor}_n \};$

- Exemplos:
 - `int v[3] = {1,2,3};`
 - `float vetor[5] = {1.0, 2.0, 3.0, 4.0, 5.0};`

1. Vetores

- Inicializando um vetor:
 - Se um vetor com n elementos, inicialmente, receber apenas alguns elementos, as posições restantes serão iniciadas com o valor 0 (ZERO).

tipo var[n] = { valor₁, valor₂, ..., valor_n};

- Exemplos:

- int vet[4] = {1,2}; →

1	2	0	0
---	---	---	---

- float a[6] = {1.0, 2.0, 3.0};



1. Vetores

- Exercícios:

1. Crie um programa que leia um vetor de inteiros de tamanho 6 e imprima os valores lidos.
2. Crie um programa que leia um vetor de inteiros de tamanho 6 e imprima somente suas posições pares.
3. Crie um programa que leia um vetor de inteiros de tamanho 6 e imprima a soma de todos os valores contidos no vetor.
4. Crie um programa que leia 6 valores inteiros e os armazene em um vetor. Imprima o maior elemento e a posição que ele se encontra.
5. Crie um programa que leia 2 notas de 6 alunos e armazene num vetor, calcule e imprima a média geral.

1. Vetores e Matrizes

- Vetores em funções – 1º Forma:

```
#include <stdio.h>
```

```
void inic (int s[6])
```

```
{
```

```
    int i
```

```
    for(i=0;i<6; i++)
```

```
    {
```

```
        s[i] = i;
```

```
        printf("%d\n", s[i]);
```

```
    }
```

```
}
```

```
int main ()
```

```
{
```

```
    int vet[6];
```

```
    inic(vet);
```

```
}
```

1. Vetores e Matrizes

- Vetores em funções – 2º Forma:

```
#include <stdio.h>
```

```
void inic (int s[],int n)
```

```
{
```

```
    for(n=0;n<6; i++)
```

```
    {
```

```
        s[n] = n;
```

```
        printf("%d\n", s[n]);
```

```
    }
```

```
}
```

```
int main ()
```

```
{
```

```
    int vet[6];
```

```
    inic(vet,6);
```

```
}
```

1. Vetores

- Exercícios:

1. Crie um programa que leia um vetor de inteiros de tamanho 6 e uma função que imprima os valores lidos.
2. Crie um programa que leia um vetor de inteiros de tamanho 6 e uma função imprima somente suas posições pares.
3. Crie um programa que leia um vetor de inteiros de tamanho 6 e uma função que some de todos os valores contidos no vetor e devolva o valor somado, o valor somado deve ser impresso na tela.
4. Crie um programa que leia 6 valores inteiros e os armazene em um vetor. Crie uma função que imprima o maior elemento e a posição que ele se encontra.
5. Crie um programa que leia 2 notas de 6 alunos e armazene num vetor, crie uma função que calcule e imprima a média geral.