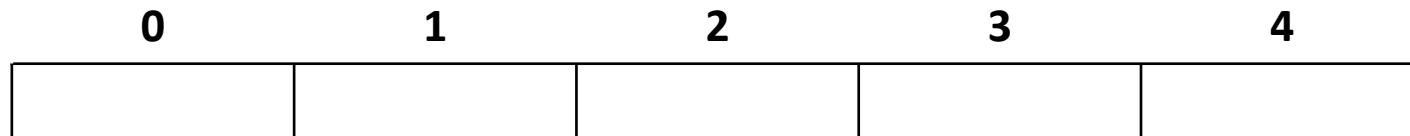


Java

(Vetores)

Vetor

Variável multivalorada, unidimensional,
homogênea.



Vetor - declaração

```
<tipo>[] <identificador>;  
<tipo> <identificador>[];
```

Exemplo:

```
int[] idades;  
float alturas[];
```

Vetor - inicialização

Em Java um array é um objeto. Quando declarado o array, ainda não temos o espaço para guardar os elementos, devemos inicializar o vetor criando um objeto. Há duas formas de fazer isso:

- <identificador> = new <tipo>[<quantidade>];
- Na declaração podemos já passar os valores dos elementos do vetor entre chaves

Exemplo:

```
double[] alturas = {1.80, 1.25, 1.50} ;  
int[] idades;  
idades = new int[10];
```

Vetor – acessando os elementos

Usamos <identificador>[<índice>] para acessar um elemento de um vetor.

Exemplo:

idades[1] = 2;

idades[2] = idades[1]+2;

idades[1+2] = 10;

x = 0;

idades[x] = 7;

Vetor – percurso

O percurso consiste em “visitar” cada uma das posições do vetor uma única vez.

Exemplo:

```
for(int i=0; i<10; i++) {  
    idades[i] = i;  
}
```

Vetor – passando por parâmetro

```
public static void main(String[] args) {  
    int[] idades;  
    idades = new int[10];  
    preenche(idades);  
    for(int i=0; i<10; i++) {  
        System.out.println(idades[i]);  
    }  
}  
  
public static int[] preenche(int[] v) {  
    int i;  
    for(i=0; i<10; i++) {  
        v[i] = i;  
    }  
}
```

Vetor – devolvendo um vetor numa função

```
public static void main(String[] args) {  
    int[] idades, id_dobradas;  
    idades = new int[10];  
    preenche(idades);  
    for(int i=0; i<10; i++) {  
        System.out.println(idades[i]);  
    }  
    id_dobradas = dobro(idades);  
    for(int i=0; i<10; i++) {  
        System.out.println(id_dobradas[i]);  
    }  
}  
  
public static int[] preenche(int[] v) {  
    int i;  
    for(i=0; i<10; i++) {  
        v[i] = i;  
    }  
    return v;  
}
```

```
public static int[] dobro(int[] v) {  
    int[] resp = new int[10];  
    int i;  
  
    for(i=0; i<10; i++) {  
        resp[i] = v[i] * 2;  
    }  
    return resp;  
}
```

Exercício 1

Implemente uma função que recebe por parâmetro um vetor de números inteiros e a quantidade de elementos do mesmo e informe a posição do maior número.

Exercício 2

Faça uma função que receba 2 vetores com 10 números inteiros cada um e gere um terceiro vetor cujos elementos contém a soma dos elementos de mesmo índice dos outros dois vetores.

Exercício 3

Faça um programa em C modularizando que preencha um vetor de 50 posições com valores inteiros recebidos pelo teclado e, em seguida, mostre os valores na ordem inversa da que foram digitados.

Exercício 4

Faça uma função que receba um número inteiro N maior do que zero e preencha um vetor com os N primeiros termos da sequência de Fibonacci.

Exercício 5

Faça uma função que receba um vetor com 50 números inteiros e um número. Ao final a função deve devolver quantas vezes o número recebido aparece no vetor.

Exercício 6

Faça uma função que receba um vetor com as médias finais dos alunos de uma turma e a quantidade de alunos da turma, calcule e devolva a média da turma.

Exercício 7

Faça uma função que receba um vetor com as médias dos alunos de uma turma, a quantidade de alunos da turma e devolva quantos alunos ficaram acima da média da turma.

Obs: use a função do exercício 6 para calcular a média da turma.