

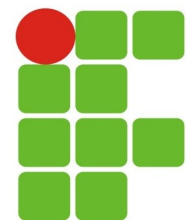
# Fundamentos de Programação

**Estrutura de dados  
Homogêneas de uma dimensão  
(vetores)**

**Prof. M.Sc.: João Paulo Q. dos Santos**

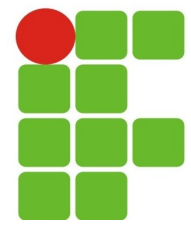
**E-mail: [joao.queiroz@ifrn.edu.br](mailto:joao.queiroz@ifrn.edu.br)**

**Página: <http://docente.ifrn.edu.br/joaoqueiroz/>**



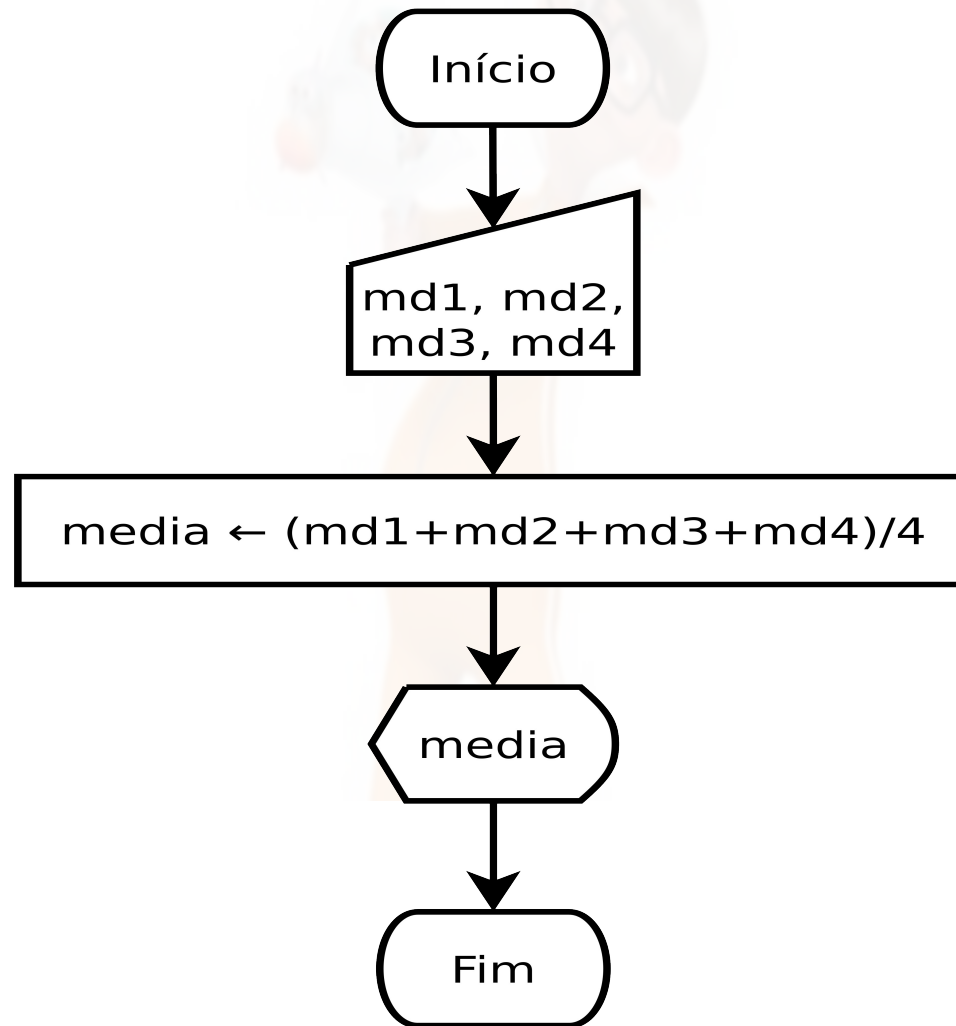
# Vetores

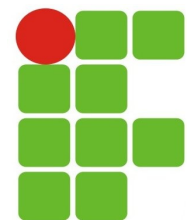
- Vetor é uma matriz de uma dimensão, forma mais simples de utilizar valores do mesmo tipo em sequencia, com apenas uma linha e várias colunas;
- Esta estrutura de dados fica em uma única variável dimensionada com um determinado tamanho;
- A dimensão de uma matriz é formada por constantes inteiras e positivas;
- Os nomes dados aos vetores ou matrizes seguem as mesmas regras dos nomes dados a variáveis simples;
- Para mostra como um vetor em certa situação mostra sua vantagem operacional, considere o seguinte problema:



# Vetores

- Desenvolver um programa que calcule e apresente o valor da média de um aluno durante os 4 bimestres;





# Vetores

```
#include <iostream>

using namespace std;

int main( ) {

    float media, md1, md2, md3, md4;

    cout << "Digite media 1: ";
    cin >> md1;

    cout << "Digite media 2: ";
    cin >> md2;

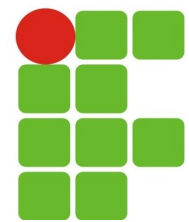
    cout << "Digite media 3: ";
    cin >> md3;

    cout << "Digite media 4: ";
    cin >> md4;

    media = (md1+md2+md3+md4)/4;

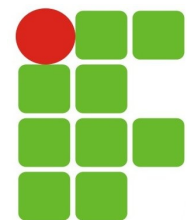
    cout << "Media : "<<media;

    return 0;
}
```



# Vetores

- Declaração de um vetor:  
`<tipo_array> <nome_variável>[<tamanho>];`
- Onde:
  - `tipo_array`: tipo primitivo da linguagem;
  - `nome_variável`: identificador válido;
  - `tamanho`: quantidade de posições que o vetor terá;
  - Exemplo:  
`float vetor[4];`



# Vetores

- Declaração de um vetor:

```
<tipo_array> <nome_variável>[<tamanho>];
```

- Onde:

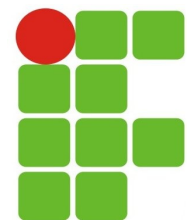
- tipo\_array: tipo primitivo da linguagem;
- nome\_variável: identificador válido;
- tamanho: quantidade de posições que o vetor terá;

- Exemplo:

```
float vetor[4];
```

- Quando se sabe a quantidade de valores que se vai utilizar no vetor pode ser declarado dessa forma:

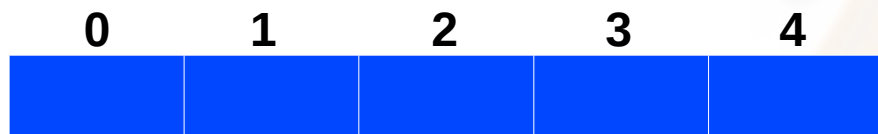
```
float vetor[4] = {17.22, 42.11, 9.33, 33.55};
```



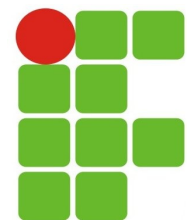
# Vetores

- Os índices dos vetores em C++ iniciam em 0 até [tamanho – 1];
- Exemplo:

```
float vetor[5];
```



Vetor de inteiros com 5 posições  
Índice variando de 0 até 4



# Vetores

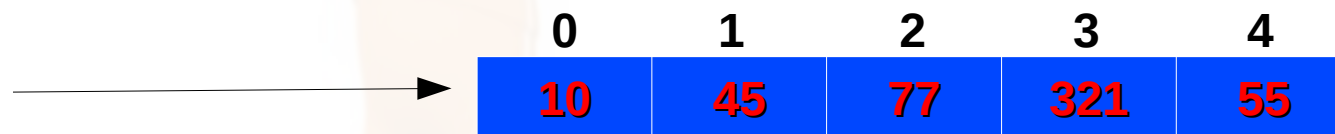
- Escrita aos elementos: <nome\_array> [índice] = valor;

```
int vetor [5];  
vetor[3] = 20;
```

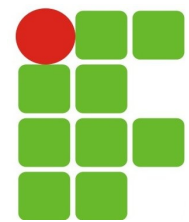


- Leitura aos elementos: <nome\_array> [índice];

```
int vetor [5];  
cout<<vetor[0];
```





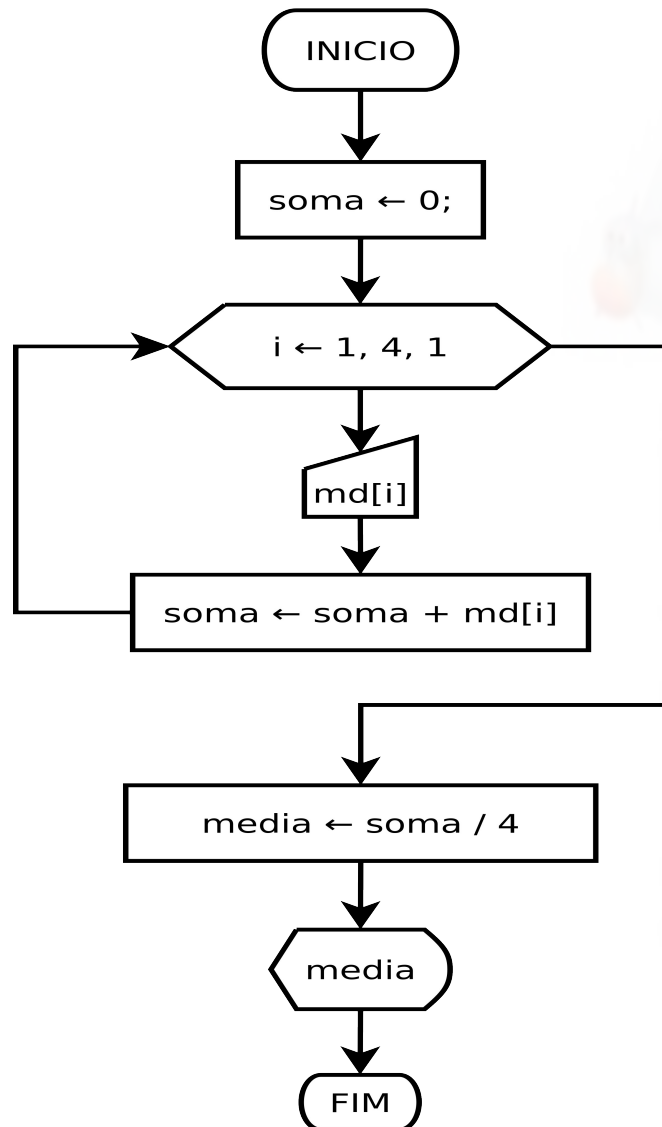


# Vetores

- A escrita de dados no vetor é processada passo a passo, um elemento por vez, por meio de um laço de repetição (**for**);
- A leitura dos dados contidos é bastante parecido com o processo de escrita desses dados;

# Vetores

- Exemplo (anterior):



```
#include <iostream>

using namespace std;

int main() {

    float media, soma = 0;

    float vetor [4];

    for (int i = 0; i < 4; i++) {

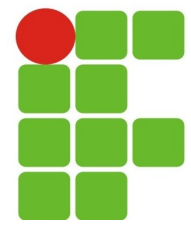
        cout << "Digite media "<<i<<" : ";
        cin >> vetor[i];

        soma = soma + vetor[i];
    }

    media = soma/4;

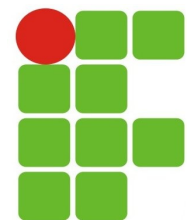
    cout << "Media : "<<media;

    return 0;
}
```



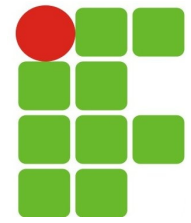
# Exercício de Aprendizagem 1

Desenvolver um programa que efetue a leitura de dez elementos para um vetor A. Construir um vetor B, observando a seguinte lei de formação: se o valor do índice for par, o valor deverá ser multiplicado por 5, sendo ímpar, deverá ser somando com 5. Ao final mostrar o conteúdo do vetor A e B.

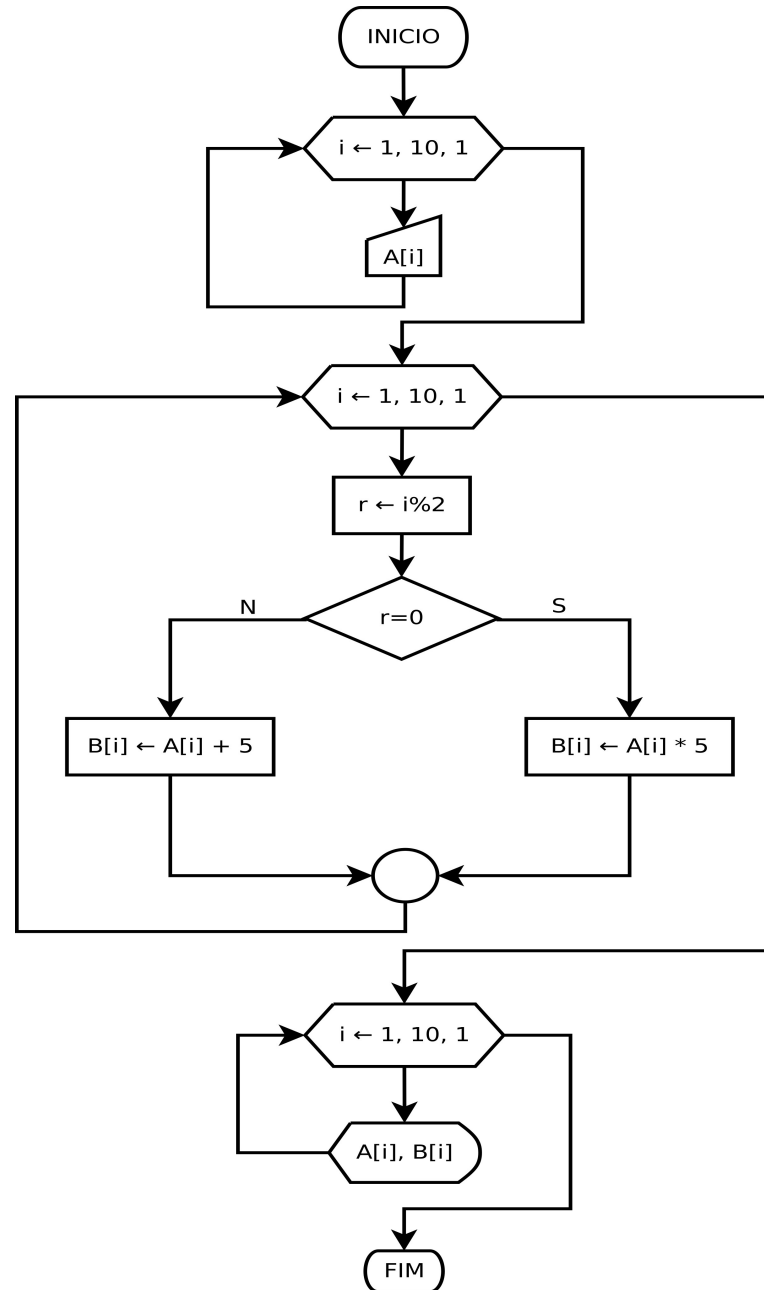


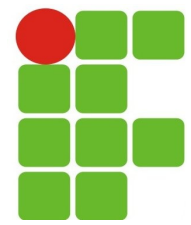
# Algoritmo

- 1) Iniciar o contador de índice, variável  $i$  com 0 em um contador até 9;
- 2) Ler os 10 valores, um a um;
- 3) Verificar se o índice é par se sim multiplica por 5, se não soma 5. Criar o vetor B e atribuir a ela os valores do vetor A devidamente calculados;
- 4) Apresentar os conteúdos dos dois vetores.



# Diagrama

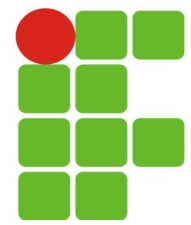




# Exercício de Aprendizagem 2

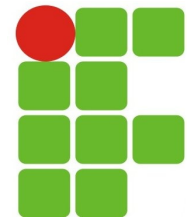
Desenvolver um programa que efetue a leitura de dez elementos de um vetor A. No final, apresente o total da soma de todos os elementos que sejam ímpares.



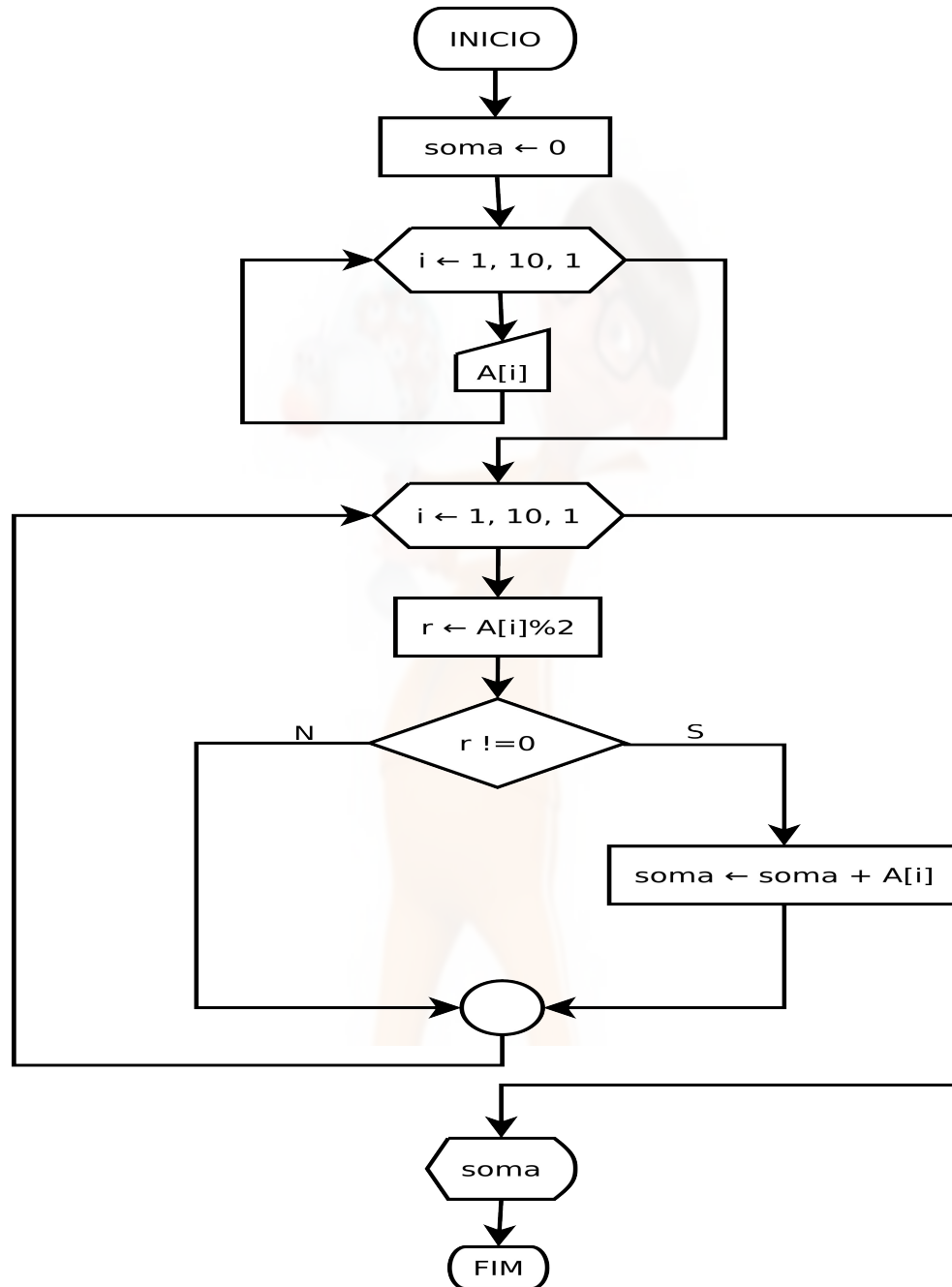


# Algoritmo

- 1) Iniciar o contador de índice, variável  $i$  como 0 em um contador até 9;
- 2) Ler os 10 valores, um a um;
- 3) Verificar se o elemento é ímpar; se sim efetuar a soma dos elementos;
- 4) Apresentar o total somado de todos os elementos ímpares do vetor A.



# Diagrama







# Código

```
#include <iostream>

using namespace std;

int main( ) {

    int i, r, soma, vetorA [10];

    soma = 0;

    for (i = 0; i < 10; i++) {

        cout << "Digite valor "<<(i+1)<<" : ";
        cin >> vetorA[i];
    }

    for (i = 0; i < 10; i++) {

        r = vetorA[i]%2;

        if (r != 0) {

            soma = soma + vetorA[i];
        }
    }
}
```

```
    cout << "Soma : "<<soma;

    return 0;
}
```