

Projeto OBI

MATRIZ UNIDIMENSIONAL

O que é?

Matriz unidimensional, também tratada como "vetor", armazena dados de forma sequencial, e cada dado é armazenado e recuperado por meio de um número inteiro que representa a sua "posição" nesta "fileira". Por representar uma "reta", pode-se dizer que os dados são armazenados em uma única dimensão.

Vamos a um exemplo de declaração em linguagem C:

```
int vetor [5];
```

int -> tipo de dado do vetor

vetor -> nome da variável

[5] -> quantidade de dados possíveis a serem armazenadas nesse vetor

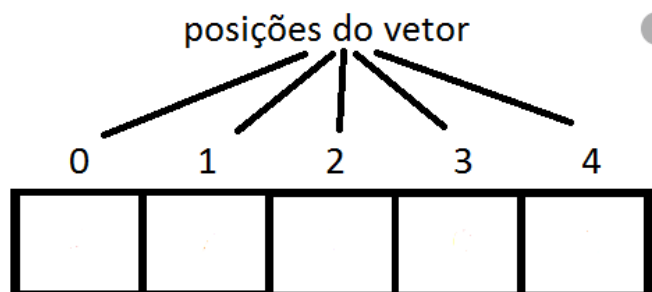
Ilustração:



Na imagem acima, podemos visualizar uma matriz unidimensional com nome vetor, com 5 espaços de armazenamento e cada espaço podendo armazenar um inteiro.

Índice

Para localizar a posição de um item em um vetor usamos um número inteiro denominado índice do vetor. E esse índice começa sempre da posição 0. Você verá melhor com a figura a seguir:



Ou seja:

Vetor [0] é a primeira posição de um vetor, e vetor[1] é a segunda posição desse vetor, e assim por diante. Sabendo isso, agora podemos atribuir valores para esse vetor. Os dados armazenados em um vetor são chamados de itens do vetor.

Projeto OBI

Vamos lá:

Vetor[0] = 4;

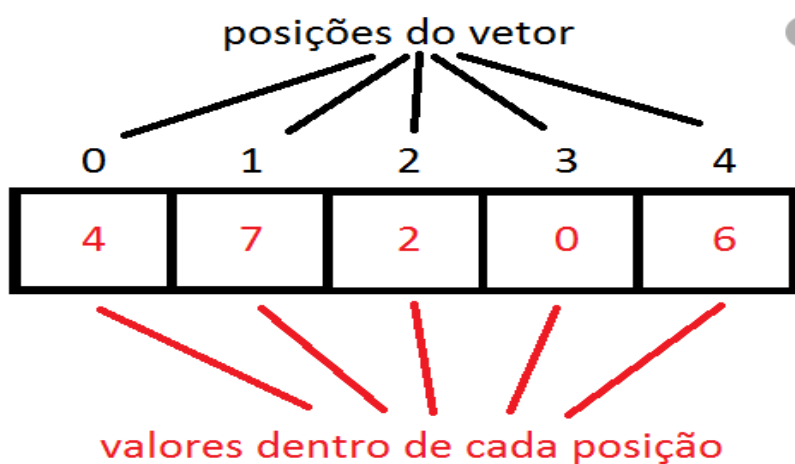
Vetor [1] = 7;

Vetor [2] = 2;

Vetor [3] = 0;

Vetor [4] = 6;

Então nosso vetor fica assim:



Se quiser manipular esses valores, fará da mesma forma. Veja no exemplo a seguir a soma de todos os itens desse vetor:

$$\text{Soma} = \text{vetor}[0] + \text{vetor}[1] + \text{vetor}[2] + \text{vetor}[3] + \text{vetor}[4];$$

Repare que você não está somando $0 + 1 + 2 + 3 + 4$.

Você está somando o conteúdo armazenado nessas posições. No nosso exemplo anterior, você estaria somando $4 + 7 + 2 + 0 + 6$.

Projeto OBI

Vamos ao seguinte exercício:

Faça um programa que solicite 5 números inteiros ao usuário e os armazenem em um vetor. Após isso, exiba:

- 1) O conteúdo do vetor na tela.**
- 2) O conteúdo do vetor na tela na ordem inversa de inserção.**
- 3) O primeiro número e o último.**
- 4) O maior e o menor número.**
- 5) A média e a soma dos números.**

Projeto OBI

1º Passo: solicite 5 números inteiros ao usuário e os armazenem em um vetor

Poderíamos iniciar esse programa com o seguinte código:

```
#include <stdio.h>

main(){
    int numero1, numero2, numero3, numero4, numero5
```

Porém, e se precisássemos armazenar 100 números? Por isso se faz necessário o uso do vetor:

```
#include <stdio.h>

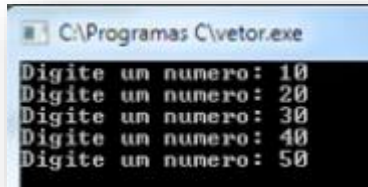
main(){
    int numero[5];
```

Agora vamos solicitar ao usuário 5 números e armazená-los nesse vetor. Usamos a variável *i* para passar pelo vetor. Iniciando ela em 0 (1ª posição do vetor), e terminando em 4 (5ª posição do vetor.) Conforme o código a seguir:

```
//Carregamento
for(i=0; i < 5; i++){
    printf("Digite o %d numero: ", i+1);
    scanf("%d",&numero[i]);
}
```

Executamos e temos o seguinte programa:

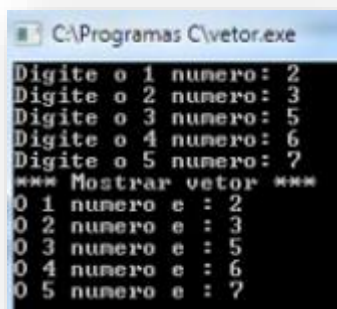
Projeto OBI



2º Passo: Vamos imprimir na tela todos os itens desse vetor. Seguindo a mesma lógica do carregamento, usamos um for com a variável i passando pelas 5 posições do vetor. Para cada posição, imprimimos o número contido naquela posição.

```
//Mostrar
printf("*** Mostrar vetor ***\n");
for (i=0; i<5; i++){
    printf("O %d numero e : %d \n", i+1, numero[i]);
}
```

Executamos e temos o seguinte programa:

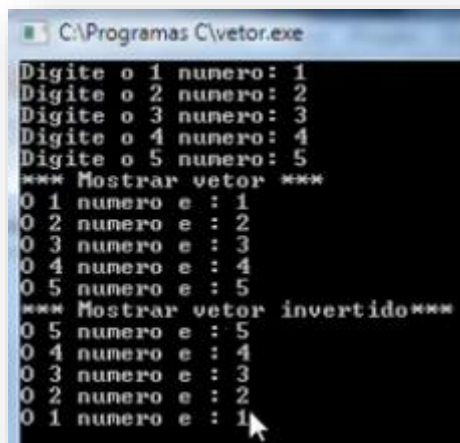


3º Passo: Mostrar vetor invertido. A lógica é a mesma, porém iniciamos o for com i = 4 (5ª posição do vetor) e vamos decrescendo até i = 0 (1ª posição do vetor).

```
//Mostrar invertido
printf("*** Mostrar vetor invertido***\n");
for (i=4; i>=0; i--){
    printf("O %d numero e : %d \n", i+1, numero[i]);
}
```

Projeto OBI

Executamos e temos o seguinte programa:



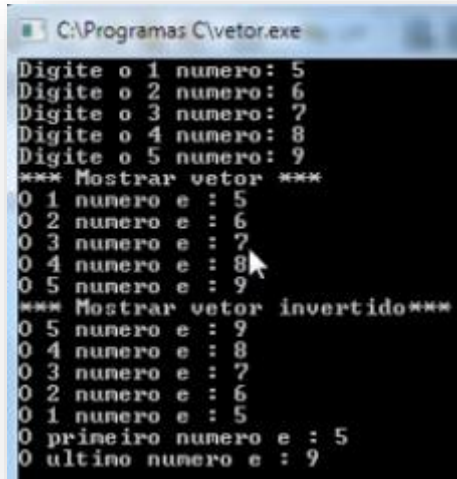
```
C:\Programas\C\vetor.exe
Digite o 1 numero: 1
Digite o 2 numero: 2
Digite o 3 numero: 3
Digite o 4 numero: 4
Digite o 5 numero: 5
*** Mostrar vetor ***
0 1 numero e : 1
0 2 numero e : 2
0 3 numero e : 3
0 4 numero e : 4
0 5 numero e : 5
*** Mostrar vetor invertido***
0 5 numero e : 5
0 4 numero e : 4
0 3 numero e : 3
0 2 numero e : 2
0 1 numero e : 1
```

4º Passo: Mostrar o primeiro e o último número. Apenas imprimir os valores solicitados, Já sabemos que `numero[0]` é a primeira posição do vetor, enquanto `numero[4]` é a última.

```
//primeiro e ultimo
printf("O primeiro numero e : %d \n", numero[0]);
printf("O ultimo numero e : %d \n", numero[4]);
```

Executamos e temos o seguinte código:

Projeto OBI



```
CAProgramas Cvetor.exe
Digite o 1 numero: 5
Digite o 2 numero: 6
Digite o 3 numero: 7
Digite o 4 numero: 8
Digite o 5 numero: 9
*** Mostrar vetor ***
0 1 numero e : 5
0 2 numero e : 6
0 3 numero e : 7
0 4 numero e : 8
0 5 numero e : 9
*** Mostrar vetor invertido***
0 5 numero e : 9
0 4 numero e : 8
0 3 numero e : 7
0 2 numero e : 6
0 1 numero e : 5
0 primeiro numero e : 5
0 ultimo numero e : 9
```

5º passo: Mostrar o maior número. Para isso, criamos uma variável chamada “maior”, e iniciamos em 0. A lógica contida dentro da estrutura condicional if é simples: se o conteúdo do vetor na posição indicada por i for maior que o valor da variável “maior”, a variável recebe esse valor. Ao passar pelas 5 posições, o maior valor dentre os 5 estará contido na variável maior.

```
int numero[5];
int i, maior=0;
```

```
if (numero[i] > maior){
    maior = numero[i];
}
```

Colocamos essa lógica dentro daquele primeiro for de carregamento dos valores.

```
//Carregamento
for(i=0; i < 5; i++){
    printf("Digite o %d numero: ", i+1);
    scanf("%d",&numero[i]);
    if (numero[i] > maior){
        maior = numero[i];
    }
}
```

Projeto OBI

E inserimos uma impressão, para mostrar ao usuário qual o maior valor.

```
//primeiro e ultimo  
printf("O primeiro numero e : %d \n", numero[0]);  
printf("O ultimo numero e : %d \n", numero[4]);  
printf("O maior numero e : %d \n", maior); }
```

Executamos e temos o seguinte código:

```
CA\Programas\Cvetor.exe  
Digite o 1 numero: 2  
Digite o 2 numero: 5  
Digite o 3 numero: 3  
Digite o 4 numero: 1  
Digite o 5 numero: 4  
*** Mostrar vetor ***  
0 1 numero e : 2  
0 2 numero e : 5  
0 3 numero e : 3  
0 4 numero e : 1  
0 5 numero e : 4  
*** Mostrar vetor invertido***  
0 5 numero e : 4  
0 4 numero e : 1  
0 3 numero e : 3  
0 2 numero e : 5  
0 1 numero e : 2  
0 primeiro numero e : 2  
0 ultimo numero e : 4  
0 maior numero e : 5
```

6º passo: Mostrar o menor número. A lógica do menor número é a mesma. Criamos uma variável menor iniciando com 10000, só por exemplo. Para cada valor inserido, é feito um if para saber se esse valor é menor que o valor da variável menor. Se sim, a variável menor recebe esse valor, e o looping continua até passarmos por todo o vetor.

```
int numero[5];  
int i, maior=0, menor=10000;
```

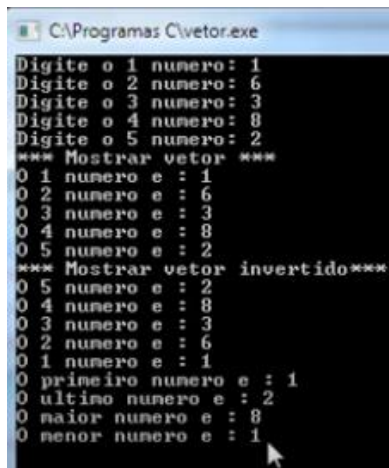
```
//Carregamento  
for(i=0; i < 5; i++){  
    printf("Digite o %d numero: ", i+1);  
    scanf("%d", &numero[i]);  
    if (numero[i] > maior){  
        maior = numero[i];  
    }  
    if (numero[i] < menor){  
        menor = numero[i];  
    }  
}
```


Projeto OBI

Inserimos uma impressão para mostrar ao usuário o menor valor.

```
//primeiro e ultimo  
printf("O primeiro numero e : %d \n", numero[0]);  
printf("O ultimo numero e : %d \n", numero[4]);  
printf("O maior numero e : %d \n", maior);  
printf("O menor numero e : %d \n", menor);
```

Executamos e temos o seguinte programa:



```
C:\Programas\Civetor.exe  
Digite o 1 numero: 1  
Digite o 2 numero: 6  
Digite o 3 numero: 3  
Digite o 4 numero: 8  
Digite o 5 numero: 2  
*** Mostrar vetor ***  
0 1 numero e : 1  
0 2 numero e : 6  
0 3 numero e : 3  
0 4 numero e : 8  
0 5 numero e : 2  
*** Mostrar vetor invertido***  
0 5 numero e : 2  
0 4 numero e : 8  
0 3 numero e : 3  
0 2 numero e : 6  
0 1 numero e : 1  
0 primeiro numero e : 1  
0 ultimo numero e : 2  
0 maior numero e : 8  
0 menor numero e : 1
```

7º passo: Mostrar a soma dos itens do vetor. Criamos uma variável soma, iniciando em 0. Incluímos ao final do for de carregamento o seguinte comando:

```
soma = soma+numero[i];
```

Isso faz com que, para cada valor recebido para numero[i] (de 0 até 4), será somado ao valor da variável soma.

```
int numero[5];  
int i, maior=0, menor=10000, soma=0;
```

Projeto OBI

```
//Carregamento
for(i=0; i < 5; i++){
    printf("Digite o %d numero: ", i+1);
    scanf("%d",&numero[i]);
    if (numero[i] > maior){
        maior = numero[i];
    }
    if (numero[i] < menor){
        menor = numero[i];
    }
    soma = soma+numero[i];
}
```

E incluímos uma impressão com a soma desses valores.

```
//primeiro e ultimo
printf("O primeiro numero e : %d \n", numero[0]);
printf("O ultimo numero e : %d \n", numero[4]);
printf("O maior numero e : %d \n", maior);
printf("O menor numero e : %d \n", menor);
printf("A soma e : %d \n", soma);
int soma
```

Executamos e temos o seguinte programa:

Projeto OBI

```
C:\Programas C\vetor.exe
Digite o 1 numero: 1
Digite o 2 numero: 2
Digite o 3 numero: 3
Digite o 4 numero: 4
Digite o 5 numero: 5
*** Mostrar vetor ***
0 1 numero e : 1
0 2 numero e : 2
0 3 numero e : 3
0 4 numero e : 4
0 5 numero e : 5
*** Mostrar vetor invertido***
0 5 numero e : 5
0 4 numero e : 4
0 3 numero e : 3
0 2 numero e : 2
0 1 numero e : 1
0 primeiro numero e : 1
0 ultimo numero e : 5
0 maior numero e : 5
0 menor numero e : 1
A soma e : 15
```

8º Passo: calcular a média dos itens do vetor. Incluímos a variável media. Repare que a variável media é do tipo float, pois iremos atribuir o resultado de uma divisão à ela, então precisamos de números reais, e não inteiros.

```
int numero[5];
int i, maior=0, menor=10000, soma=0;
float media;
```

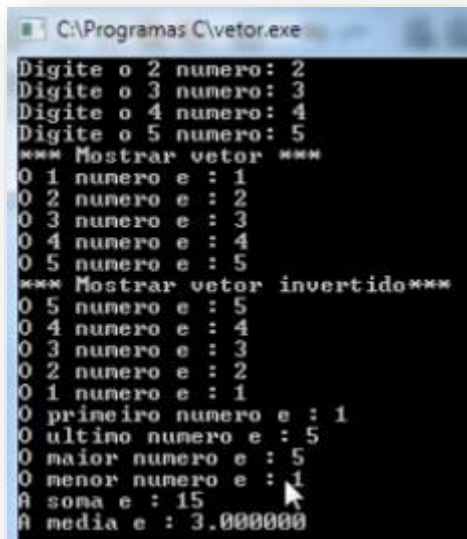
Basicamente é pegar o valor da variável soma, e dividir pelo total de itens do vetor. Como já sabemos, é cinco. Fazer a impressão da média dos valores.

```
//primeiro e ultimo
printf("O primeiro numero e : %d \n", numero[0]);
printf("O ultimo numero e : %d \n", numero[4]);
printf("O maior numero e : %d \n", maior);
printf("O menor numero e : %d \n", menor);
printf("A soma e : %d \n", soma);
media = soma/5;
printf("A media e : %f \n", media);
```

Projeto OBI

No printf, é possível colocar %.2f para limitar o número de casas após a vírgula.

Executamos e temos o seguinte programa finalizado:



```
C:\Programas\Civetor.exe
Digite o 2 numero: 2
Digite o 3 numero: 3
Digite o 4 numero: 4
Digite o 5 numero: 5
*** Mostrar vetor ***
0 1 numero e : 1
0 2 numero e : 2
0 3 numero e : 3
0 4 numero e : 4
0 5 numero e : 5
*** Mostrar vetor invertido***
0 5 numero e : 5
0 4 numero e : 4
0 3 numero e : 3
0 2 numero e : 2
0 1 numero e : 1
0 primeiro numero e : 1
0 ultimo numero e : 5
0 maior numero e : 5
0 menor numero e : 1
A soma e : 15
A media e : 3.000000
```