

Vetores ou Matrizes unidimensionais

Professor:

Leandro O. Freitas - leandro@politecnico.ufsm.br



Vetores

- Variáveis simples nem sempre são suficientes;
- A linguagem C permite a utilização de uma variável que armazena um **conjunto de dados do mesmo tipo ao mesmo tempo** (variáveis indexadas);
- Uma única variável é definida com um tamanho específico (dimensão), referente ao total de elementos que poderá guardar. Exemplo:

int vet[3]; □ a variável **vet** pode armazenar até 3 valores. Um em cada uma das posições (0, 1, 2).

Posição 0	Posição 1	Posição 2
Valor 1	Valor 2	Valor 3

- Nome de vetores seguem as mesmas regras das outras variáveis.

Características de vetores

- Podem conter diversos valores, mas deve-se definir o número máximo no momento da declaração;
- Todos os valores devem ser do mesmo tipo (estrutura homogênea);
- Possuem apenas um nome que os identifica (nome da variável);
- Cada valor pode ser acessado de forma independente, através do índice;
- Valores podem ser acessados de forma aleatória.



Sintaxe

```
int main() {  
    TIPO VARIÁVEL [DIMENSÃO];  
}
```

- Onde:

- **TIPO**: tipo dos dados que serão armazenados no vetor (int, float, char, etc);
- **VARIÁVEL**: nome do vetor;
- **DIMENSÃO**: tamanho do vetor, ou seja, total de elementos que deverá armazenar. Ex.: [3].

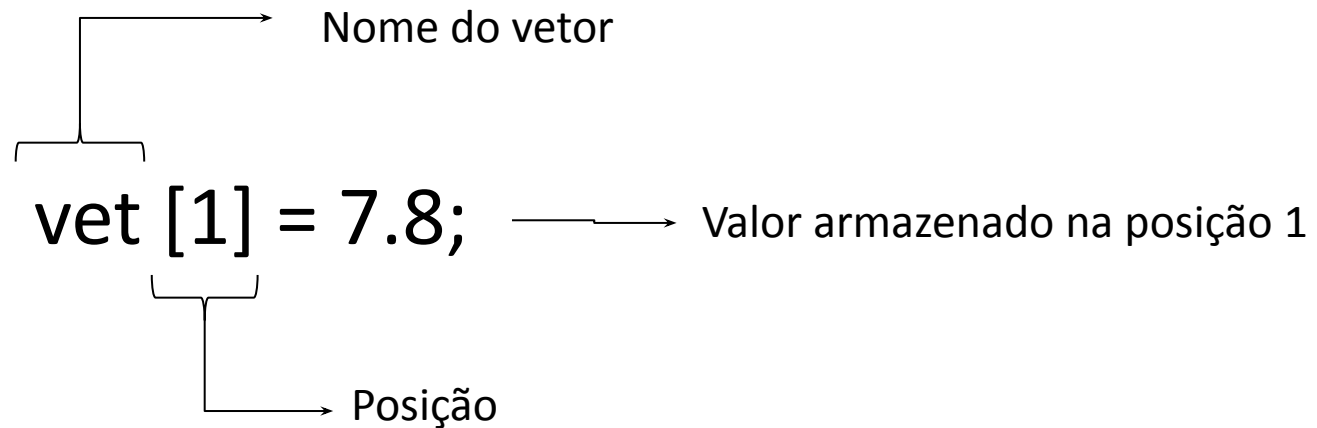
- Exemplo:

```
int main()  
{  
    float notas[3];  
    ...  
}
```

→ Três posições: 0, 1 e 2.

ATENÇÃO!!!

- Não confunda o **índice** com o **valor armazenado** em uma posição do vetor!



Posição 0	Posição 1	Posição 2	Posição 3, 4...	Posição N
1º valor	7.8	2º valor	...	N-ésimo valor

Declaração e atribuição de valores



Sistemas para Internet
UFSM

- Manualmente:

```
int main()
{
    float notas [3];
    notas[0] = 4.0;
    notas[1] = 3.4;
    notas[2] = 10.0;
    <conjunto de instruções>;
}
```

- OU -

```
int main()
{
    float notas [3] = {5.5, 4.4, 3.1};
    <conjunto de instruções>;
}
```

- Leitura do teclado:

```
int main()
{
    float notas[3];
    int i;

    for (i=0; i<3; i++)
    {
        scanf("%f", &notas[i]);
    }

    return 0;
}
```

Impressão na tela

- Manualmente:

```
int main()
{
    float notas[3];
    int i;

    notas[0] = 4.0;
    notas[1] = 3.4;
    notas[2] = 10.0;

    printf("%.2f", notas[0]);
    printf("%.2f", notas[1]);
    printf("%.2f", notas[2]);
    return 0;
}
```

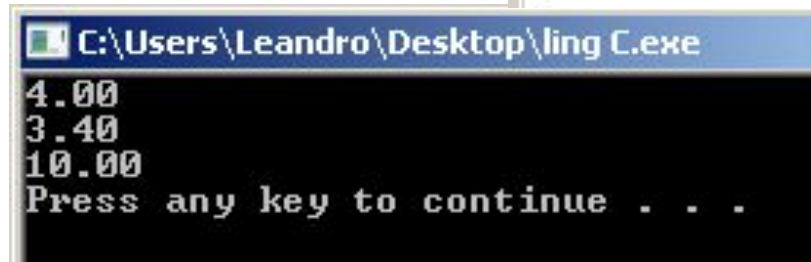
- Com laço de repetição:

```
int main()
{
    float notas[3];
    int i;

    notas[0] = 4.0;
    notas[1] = 3.4;
    notas[2] = 10.0;

    for (i=0; i<3; i++)
    {
        printf("%.2f \n", notas[i]);
    }

    return 0;
}
```



```
C:\Users\Leandro\Desktop\ling C.exe
4.00
3.40
10.00
Press any key to continue . . .
```



Processamento de elementos

- Fazer a leitura 5 elementos de um vetor. Depois, realizar a soma do primeiro com último. O resultado deve ser apresentado ao usuário:

```
#include<stdio.h>

int main()
{
    int v[5];
    int i, soma;

    for (i=0;i<5;i++)
    {
        scanf("%d", &v[i]);
    }
    soma = v[0] + v[4];
    printf("%d", soma);
    return 0;
}
```




Passagem de parâmetros

1. Na chamada da função, deve-se passar apenas o nome do vetor entre parênteses, como argumento.
2. No cabeçalho da função deve-se **declarar um vetor** com seu tipo e dimensão (mesmos usados na função chamadora).

```
1 #include<stdio.h>
2
3 void exemploVetor (int vet[3]) → Passo 2
4 {
5     //conjunto de instruções
6 }
7
8 int main() {
9 {
10     int vetor[3] = {10, 20, 30};
11     exemploVetor(vetor); → Passo 1
12 }
```

Passagem de parâmetros

- Ao contrário do que acontece com variáveis simples, as alterações de valores do vetor feitas dentro de uma função são refletidas fora dela. Exemplo:

```
3- void exemplo(int vet[3]){  
4     vet[0] = 10;  
5     vet[1] = 20;  
6     vet[2] = 30;  
7 }  
8  
9- int main(){  
10     int vet[3] = {50, 60, 70};  
11     exemplo(vet);  
12     printf("%d \n", vet[0]);  
13     printf("%d \n", vet[1]);  
14     printf("%d \n", vet[2]);  
15     return 0;  
16 }
```

1

2

Saída:
10
20
30

Exercícios

- Ler 5 elementos de um vetor e apresentá-los ao usuário.
- Desenvolver um algoritmo que leia 5 elementos de um vetor A e no final apresente o total da soma de todos os elementos ímpares.
- Desenvolver um programa que leia 5 elementos de um vetor A. Construir um vetor B de mesmo tipo e copiar os valores do vetor A para o B, mantendo-os nas posições originais.