



# Trabalhando com Vetores em C

---

Profa.: Andréia Rodrigues Casare

E-mail: [casareandreia@gmail.com](mailto:casareandreia@gmail.com)



# Algumas dicas do C

---

# Condicional ternário

---

***Sintaxe:***

***condição ? expressão1 : expressão 2;***

**Exemplo:**

**maior = (numA > numB) ? numA : numB;**



# Trabalhando com Strings

---

- Tipo de dado **char**, para armazenar uma string deve ser definido como vetor.
- Exemplos:
  - `char nome[20];`

# Trabalhando com cadeia de caracteres

---

- O **scanf** só lê a string até encontrar o primeiro espaço, se for uma string composta de várias palavras ( frase) só será lido a primeira palavra.
- Nesse caso deve ser usado a função `gets()`.
- `char nome[20];`
- Ex: `gets(nome);`

# Função gets()

---

- A função **gets()** lê uma string do teclado.
- Sintaxe:
  - **gets (nome\_da\_string);**
- Exemplo:

```
int main() {  
    char nome[100];  
    printf ("Digite o seu nome: ");  
    gets(nome);  
    printf("\n Ola %s", nome);  
    return 0;  
}
```



# VETORES

---

São arrays com 1 dimensão

# Arrays Unidimensionais – Vetores

---

- Vetores, também chamados arrays, são uma maneira de armazenar vários dados num mesmo nome de variável através do uso de índices numéricos.
- Por exemplo, se tivemos que criar 20 variáveis do mesmo tipo que querem dizer a mesma coisa, nós não criaríamos -> `int x1, x2, x3, x4, x5, ...` ao invés disso, criaríamos apenas uma variável de vetor para guardar todos os 20 números de uma vez.



# Vetores

---

- Variável composta **unidimensional**
  - Contém espaço para armazenar diversos valores
  - É acessada via um índice
- Trabalhamos até agora com variáveis com somente um valor
  - Exemplo:  $y = 123$
- No caso de vetores, uma mesma variável guarda ao mesmo tempo múltiplos valores
  - Exemplo:  $x_1 = 123, x_2 = 456, \dots$
  - $x = [123, 456, \dots]$

# Declaração de um vetor

---

```
int vet [4];  
float aVet[10];  
char nomes[5][10];
```

# Tamanho do vetor

---

- O importante mesmo é saber que não importa o tamanho do vetor, o **index (número que indica a posição)** sempre começa em 0. Portanto, um vetor de tamanho 20 vai da posição 0 a 19.
- Número de elementos é  $N$ , o índice ou posição do último elemento será  **$N - 1$** .



# Atribuindo valores

---

- `Vet[2] = 10;`
- `aNome[1] = "Andreia";`
- `aVet[5] = 201.50;`

# Exemplo

---

```
#include<stdio.h>
int main()
{
    int vet[5] = {1,2,3,4,5};
    printf("Primeiro valor : %i",vet[0]);
    printf("\n Último valor  : %i",vet[4]);
    printf("\n Somando o segundo e o quarto elementos :
%i",vet[1] + vet[3]);
    vet[2] = 27;
    printf("\n Mudamos o valor do terceiro elemento para:
%i",vet[2]);
    return 0;
}
```

# Mais exemplos

---

```
int main() {  
    int vet[5],i;  
    for (i=0;i<5;i++){  
        printf("Digite o termo %i :",i+1);  
        scanf("%i",&vet[i]);  
    }  
    for (i=0;i<5;i++) {  
        printf("%i    ",vet[i]);  
    }  
    for(i=0;i<5;i++) {  
        vet[i]=vet[i]*2;  
    }  
    printf("\n");  
    printf("Novos valores do vetor\n");  
    for (i=0;i<5;i++) {  
        printf("%i    ",vet[i]);  
    }  
    return 0;  
}
```

# Usando o tipo char – uma palavra

---

```
int main() {
    setlocale(LC_ALL, "Portuguese");
    char nomes[5][100];
    int i;
    for (i=0; i<5; i++){
        printf ("\nDigite um nome: ");
        scanf ("%s", &nomes[i]);
    }
    printf ("\nos nomes digitados foram:\n");
    for (i=0; i<5; i++){
        printf ("%s\n", nomes[i]);
    }
    return 0;
}
```

# Usando o tipo char - frase

---

```
int main() {
    setlocale(LC_ALL, "Portuguese");
    char nomes[5][100];
    int i;
    for (i=0; i<5; i++){
        printf ("\nDigite um nome: ");
        gets(nomes[i]);
    }
    printf ("\nOs nomes digitados foram:\n");
    for (i=0; i<5; i++){
        printf ("%s\n", nomes[i]);
    }
    return 0;
}
```



# Outro exemplo

```
int main() {
    setlocale(LC_ALL, "Portuguese");
    int num_1[5], num_2[5], novo_vet[5], i;
    for (i=0; i<5; i++){
        printf("\n Digite o número :");
        scanf("%i", &num_1[i]);
    }
    for (i=0; i<5; i++){
        printf("\n Digite o número :");
        scanf("%i", &num_2[i]);
    }
    for (i=0; i<5; i++) {
        novo_vet[i] = num_1[i] + num_2[i];
    }
    printf("\n Novos valores do vetor\n ");
    for (i=0; i<5; i++) {
        printf("%i  ", novo_vet[i]);
    }
    return 0;
}
```

# Exercício 1

---

Faça um programa que preencha um vetor de 10 números inteiros, calcule e mostre:

- O maior número
- Crie um novo vetor multiplicando o maior número por todos os números do vetor
- Imprima na tela o maior número do vetor e o novo vetor.

## Exercício 2

---

Faça um programa que receba a idade de 10 pessoas. calcule e mostre:

- A quantidade de pessoas com idade superior a 50 anos
- A média das idades
- A quantidade de pessoas com idade menor que a média das pessoas que responderam essa pesquisa.
- A percentagem de pessoas com idade superior a 30 anos.

## Exercício 3

---

Faça um programa que receba a idade, a altura e o peso de 10 pessoas, calcule e mostre:

- A quantidade de pessoas com idade superior a 50 anos
- A média das alturas das pessoas com idade entre 10 e 20 anos
- A percentagem de pessoas com peso inferior a 40 quilos entre todas as pessoas analisadas

## Exercício 4

---

Faça um programa que leia 15 nomes de pessoas e após lidos seja possível pesquisar no vetor se um determinado nome foi digitado. O programa deverá deixar que sejam pesquisados nomes até que seja informado "FIM" no nome.



## Exercício 5

---

Faça um programa que preencha um vetor de 10 números do tipo float, calcule e mostre:

- a média dos números do vetor
- a maior número
- o menor número

## Exercício 6

---

Faça um programa que preencha um vetor de 10 números, calcule e mostre:

- a quantidade de números primos existentes no vetor;
- A soma de todos os números maiores que 10 e múltiplos de 5;
- A quantidade de números que sejam maiores ou iguais a 50 e menores ou iguais a 100.

## Exercício 7

---

Faça um programa que preencha um vetor de 10 números.

Depois de preenchido o vetor 1, monte um novo vetor 2 preenchendo com zero as posições do vetor 1 que tenha valores menores que 10 e preencha com 1 (um) as posições que tenha valores iguais a 10 ou maiores que 10.

Exiba na tela o vetor 1 e vetor 2.



## Exercício 8

---

- Elaborar um programa que leia 10 números e armazene-os no vetor A. Construir um vetor B do mesmo tipo, em que cada elemento seja a metade de cada elemento do vetor A. Apresentar os elementos do vetor A em ordem decrescente e os elementos do vetor B em ordem crescente.

```

1.  #include<stdio.h>
2.  int main()
3.  {
4.      int vet[4],aux ,i, k;
5.      for (i=0;i<4;i++){
6.          printf("Digite o termo %i",i+1;
7.          scanf("%i",&vet[i]);
8.
9.      for (i=0;i<4;i++) {
10.         printf("%i  ",vet[i]);
11.     }
12.     printf("\n");
13.     for (i=0;i<3;i++) {
14.         for (k=i;k<4;k++) {
15.             if(vet[i]>vet[k]) {
16.                 aux=vet[i];
17.                 vet[i]=vet[k];
18.                 vet[k]=aux;
19.             }
20.         }
21.     }
22.     printf("Ordem crescente\n");
23.     for (i=0;i<4;i++) {
24.         printf("%i  ",vet[i]);
25.     }
26.     return 0;
27. }

```