

Função em C

O que são funções?

Em C, **as funções** são blocos de código que realizam uma tarefa específica. Eles ajudam a:

- Reutilizar código
- Organizar melhor o programa
- Dividir problemas grandes em partes menores e mais simples de entender

Quando usar funções?

Funções são indicadas quando:

- Há trechos de código que se repetem
- Queremos dividir o problema em partes menores
- É necessário organizar o código para facilitar a manutenção e o entendimento

Estrutura básica de uma função em C

```
tipo_de_retorno nome_da_funcao(parâmetros) {  
    // bloco de comandos  
    return valor; // se a função não for 'void'  
}
```

Explicando:

- tipo_de_retorno: tipo do dado que a função retorna (int, float, void, etc.).
- nome_da_funcao: nome que você define.
- parâmetros: dados de entrada que a função precisa.
- return: devolver um valor (exceto em funções void).

Exemplo 1: Função que imprime uma mensagem (mais simples possível)

```
#include <stdio.h> // Biblioteca para usar printf  
  
// Define uma função chamada mensagem  
// Não recebe nenhum parâmetro ()  
// Não retorna nada (void)  
void mensagem() {  
    // Imprime uma mensagem na tela  
    printf("Bem-vindo ao estudo de funções em C!\n");  
}  
  
int main() {  
    // Chama a função mensagem  
    mensagem();  
  
    return 0; // Indica que o programa terminou corretamente  
}
```

Exemplo 2: Função que soma dois números

```
#include <stdio.h>

// Define uma função chamada somar
// Recebe dois inteiros como parâmetros: a e b
// Retorna um inteiro (a soma)
int somar(int a, int b) {
    return a + b; // Retorna a soma
}

int main() {
    int x = 10, y = 5; // Declara duas variáveis inteiras
    int resultado;

    // Chama a função somar passando x e y
    resultado = somar(x, y);

    // Imprime o resultado da soma
    printf("Resultado da soma: %d\n", resultado);

    return 0;
}
```

Aqui, a função somar é **reutilizável** com diferentes valores

Exemplo 3 – Função com estrutura de repetição (**calcular o fatorial** de um número)

```
#include <stdio.h>

// Função que calcula o fatorial usando repetição
int fatorial(int n) {
    int resultado = 1; // Inicializa com 1 (neutro da multiplicação)

    for(int i = 1; i <= n; i++) {
        resultado *= i; // Multiplica resultado por i em cada passo
    }

    return resultado; // Retorna o fatorial
}

int main() {
    int numero = 5;

    // Chama a função fatorial com número 5
    printf("Fatorial de %d é %d\n", numero, fatorial(numero));

    return 0;
}
```

Exemplo 4 – Função recursiva (fatorial com recursão)(Recursão é quando a **função chama a si mesma.**)

```
#include <stdio.h>

// Função recursiva para calcular fatorial
int fatorial(int n) {
    // Caso base: se n for 0 ou 1, o fatorial é 1
    if(n == 0 || n == 1) {
        return 1;
    }

    // Chamada recursiva: n * fatorial(n-1)
    return n * fatorial(n - 1);
}

int main() {
    int numero = 6;

    // Chama a função recursiva
    printf("Fatorial de %d é %d\n", numero, fatorial(numero));

    return 0;
}
```

Exemplo 5: Soma e subtração com laço

```
#include <stdio.h>

int soma(int a, int b)
{
    int s = a + b;
    return s;
}

int sub(int a, int b)
{
    int s = a - b;
    return s;
}

int main() {
    int n1, n2, op, res;
    int fim = 0;
```

```
    while(fim==0)
    {
        printf("\nDigite um valor: ");
        scanf("%d",&n1);

        printf("Digite um valor: ");
        scanf("%d",&n2);

        printf("\nEscolha a operação:\n1 -
        Adição\n2 - Subtração\n");
        scanf("%d",&op);

        if(op == 1)
        {
            res = soma(n1,n2);
        }
        if(op == 2)
        {
            res = sub(n1,n2);
        }

        printf("O resultado é: %d",res);

        printf("\n0 - Continuar\n1 - Parar\n");
        scanf("%d",&fim);
    }

    return 0;
}
```

Vetores

- Um **vetor** (ou array) é uma estrutura de dados que armazena **vários elementos do mesmo tipo** (inteiros, floats, char etc.) em **uma única variável**.

Cada elemento é identificado por um **índice**, começando do **zero**.

```
int numeros[5] = {10, 20, 30, 40, 50};
```

Índice	Valor
numeros[0]	10
numeros[1]	20
numeros[2]	30
numeros[3]	40
numeros[4]	50

Quando usar vetores?

Use vetores quando:

- Precisa **armazenar e processar vários dados** do mesmo tipo (ex: notas, temperaturas, idades).
- Deseja **acessar os dados rapidamente** por índice.
- Quer **usar laços de repetição (for/while)** para percorrer os dados.

Exemplo 1 – Vetor simples com acesso manual

```
#include <stdio.h>

int main() {
    // Cria e inicializa um vetor com 5 valores
    int numeros[5] = {10, 20, 30, 40, 50};

    // Imprime cada valor manualmente
    printf("Posição 0: %d\n", numeros[0]);
    printf("Posição 1: %d\n", numeros[1]);
    printf("Posição 2: %d\n", numeros[2]);
    printf("Posição 3: %d\n", numeros[3]);
    printf("Posição 4: %d\n", numeros[4]);

    return 0;
}
```

acesso individual dos elementos. Mas não é prático se o vetor for grande.

Exemplo 2 - Percorrer vetor com FOR

```
#include <stdio.h>

int main() {
    int numeros[5] = {2, 4, 6, 8, 10};

    // Percorre o vetor de 0 até 4
    for(int i = 0; i < 5; i++) {
        printf("numeros[%d] = %d\n", i, numeros[i]);
    }

    return 0;
}
```

Por que $i < 5$?

Porque os índices vão de **0 a 4**, totalizando 5 elementos.

Exemplo 3 – Lendo valores do teclado e armazenando no vetor

```
#include <stdio.h>

int main() {
    int valores[3]; // Vetor de 3 posições

    // Lê os 3 valores do usuário
    for(int i = 0; i < 3; i++) {
        printf("Digite o valor da posição %d: ", i);
        scanf("%d", &valores[i]); // Armazena no índice correspondente
    }

    // Exibe os valores lidos
    for(int i = 0; i < 3; i++) {
        printf("valores[%d] = %d\n", i, valores[i]);
    }

    return 0;
}
```

Use vetores sempre que quiser **armazenar múltiplas entradas**.

Exemplo 4 – Calcular a média das notas usando vetor

```
#include <stdio.h>

int main() {
    float notas[4];      // Vetor para 4 notas
    float soma = 0;      // Variável para acumular as notas
    float media;

    // Lê as 4 notas
    for(int i = 0; i < 4; i++) {
        printf("Digite a nota %d: ", i + 1);
        scanf("%f", &notas[i]); // Armazena no vetor
        soma += notas[i];        // Soma a nota na variável soma
    }

    media = soma / 4; // Calcula a média

    printf("Média das notas: %.2f\n", media);

    return 0;
}
```

usamos float pois notas podem ter casas decimais.

É um caso clássico de uso de vetores para **processamento estatístico**.

Exemplo 5 Criando uma função que recebe um vetor como parâmetro

```
#include <stdio.h>

// Função que imprime os elementos de um vetor
void mostrarVetor(int v[], int tamanho) {
    for(int i = 0; i < tamanho; i++) {
        printf("v[%d] = %d\n", i, v[i]);
    }
}

int main() {
    int dados[5] = {1, 3, 5, 7, 9};

    // Chamada da função e envio do vetor
    mostrarVetor(dados, 5);

    return 0;
}
```

Em C, ao passar um vetor para uma função, você **não precisa** indicar o tamanho no colchete, pois o vetor é passado por referência.

Exemplo 6: Função que calcula a soma dos elementos do vetor

```
#include <stdio.h>

// Função que retorna a soma de todos os elementos
int somarElementos(int v[], int tamanho) {
    int soma = 0;

    for(int i = 0; i < tamanho; i++) {
        soma += v[i];
    }

    return soma;
}

int main() {
    int numeros[] = {10, 20, 30, 40, 50};

    int resultado = somarElementos(numeros, 5);

    printf("Soma: %d\n", resultado);

    return 0;
}
```

Exemplo 7 Encontrando o maior e o menor valor

```
#include <stdio.h>

void encontrarMaiorMenor(int v[], int tamanho) {
    int maior = v[0];
    int menor = v[0];

    for(int i = 1; i < tamanho; i++) {
        if(v[i] > maior) maior = v[i];
        if(v[i] < menor) menor = v[i];
    }

    printf("Maior: %d\n", maior);
    printf("Menor: %d\n", menor);
}

int main() {
    int valores[] = {12, 5, 78, 3, 42};

    encontrarMaiorMenor(valores, 5);

    return 0;
}
```

Bom para análises estatísticas. Você pode adaptar isso para encontrar **posição**, **média**, ou **valores repetidos**.

Exercícios: Básico

- 1. Escreva uma função chamada `mensagem()` que apenas imprime "Olá, mundo das funções!" e chame essa função a partir da `main()`.**
- 2. Crie uma função `dobro(int x)` que recebe um número inteiro como parâmetro e retorna o seu dobro. Use a função na `main()` e mostre o resultado.**
- 3. Faça uma função `soma(int a, int b)` que retorna a soma de dois números inteiros. Solicite os dois números na `main()` e exiba o resultado da função.**
- 4. Declare um vetor de 5 inteiros, atribua valores manualmente e use um `for` para imprimir cada valor na tela.**
- 5. Leia 5 valores inteiros do teclado e armazene-os em um vetor. Depois, mostre os valores lidos.**

Exercícios: Intermediário

- 6. Crie uma função `mostrarVetor(int v[], int tamanho)` que recebe um vetor de inteiros e imprime todos os seus elementos. Teste com um vetor de 6 posições.**
- 7. Faça uma função `somaVetor(int v[], int tamanho)` que retorna a soma de todos os elementos do vetor. Mostre o resultado na `main()`.**
- 8. Crie uma função `mediaNotas(float notas[], int qtd)` que calcula a média de notas e retorna o valor. Leia 4 notas do usuário, chame a função e mostre a média com duas casas decimais.**
- 9. Escreva uma função `maiorMenor(int v[], int tamanho)` que imprime o maior e o menor valor contido no vetor.**
- 10. Faça uma função `parOuImpar(int numero)` que retorna 1 se for par e 0 se for ímpar. Leia 5 números do usuário e informe se são pares ou ímpares, utilizando a função.**

Nivel 3 - Avancado

- 11. Crie uma função buscaElemento(int v[], int tamanho, int valor) que retorna o índice do valor no vetor ou -1 se não for encontrado. Teste com diferentes valores.**
- 12. Faça uma função contarOcorrencias(int v[], int tamanho, int valor) que retorna quantas vezes um valor aparece no vetor.**
- 13. Crie uma função ordenar(int v[], int tamanho) que ordena o vetor em ordem crescente. Mostre o vetor antes e depois da ordenação.**
- 14. Escreva uma função mediaMaiorQue(float v[], int tamanho, float media) que imprime apenas os elementos maiores que a média.**

Ultimo ex.

- Monte um programa com **menu de opções**:

```
1 - Inserir valores no vetor  
2 - Mostrar vetor  
3 - Somar elementos  
4 - Calcular média  
5 - Sair
```

Use uma função diferente para cada opção do menu.