

## **Algumas funcionalidades da biblioteca <string.h>**

[https://www.w3schools.com/c/c\\_strings\\_functions.php](https://www.w3schools.com/c/c_strings_functions.php)

Leitura de palavras

Adicionar a biblioteca: <string.h>

Definir a variável tipo char: char palavra[50];

A função fgets() lê caracteres da posição atual do fluxo até e incluindo o primeiro caractere de nova linha (\n): fgets(palavra,sizeof(palavra),stdin);

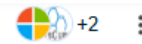
Para contar a quantidade de caracteres: strlen(palavra)

```
*****/  
#include <stdio.h>  
#include <string.h>  
  
char palavra[50];  
int main()  
{  
    printf("Leitura de palavra\n");  
    printf("Digite uma palavra");  
    fgets(palavra,sizeof(palavra),stdin);  
    printf("palavra %s",palavra);  
    printf("Sua palavra tem: %d letras \n",strlen(palavra));  
    return 0;  
}
```



## Estrutura condicional IF-ELSE

Visão geral criada por IA



Ouvir

Em C, a estrutura `if/else` permite que o programa execute diferentes blocos de código com base em uma condição. A condição é avaliada, e se for verdadeira, o bloco de código dentro do `if` é executado. Caso contrário, se existir um `else`, o bloco de código dentro do `else` é executado. [🔗](#)

Estrutura básica:

```
C

if (condição) {
    // Código a ser executado se a condição for verdadeira
} else {
    // Código a ser executado se a condição for falsa
}
```

Operadores relacionais: são símbolos utilizados para comparar um valor com outro.

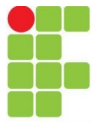
Símbolo	Nome do Operador	Exemplo	Significado
>	Maior que	$x > y$	$x$ é maior que $y$ ?
>=	Maior ou igual	$x \geq y$	$x$ é maior ou igual a $y$ ?
<	Menor que	$x < y$	$x$ é menor que $y$ ?
<=	Menor ou igual	$x \leq y$	$x$ é menor ou igual a $y$ ?
==	Igualdade	$x == y$	$x$ é igual a $y$ ?
!=	Diferente de	$x != y$	$x$ é diferente de $y$ ?

a)  $5 == 5$  ?  
b)  $5 < 5$  ?  
c)  $5 > 5$  ?  
d)  $5 \geq 5$  ?  
e)  $5 != 5$  ?  
f)  $5 != 6$  ?

**Desafio:** faça um algoritmo que leia dois valores inteiros e mostre se são iguais ou diferentes entre si.

`else if:`

Para verificar múltiplas condições, utiliza-se a estrutura `else if`. Essa estrutura permite adicionar mais verificações condicionais após o `if` original. Se a condição do `if` for falsa, o programa avalia a condição do `else if`. Se essa condição for verdadeira, o bloco de código correspondente é executado. Um `else` final pode ser adicionado para lidar com o caso em que nenhuma das condições anteriores é verdadeira, [diz o WsCube Tech.](#) [🔗](#)



C

```
#include <stdio.h>

int main() {
    int temperatura = 25;

    if (temperatura > 30) {
        printf("Está quente.\n");
    } else if (temperatura > 20) {
        printf("Está agradável.\n");
    } else {
        printf("Está frio.\n");
    }

    return 0;
}
```

**Desafio:** faça um algoritmo que leia duas notas inteiras, calcule a média e mostre o resultado conforme a seguinte regra:  
Se média < 5 “Reprovado”; Se média >= 5 E média < 6 “Recuperação”; Se média >= 6 “Aprovado”;

### Operadores Lógicos:

- **&& ou and (E):** Retorna verdadeiro apenas se ambos os operandos forem verdadeiros.
- **|| ou or (OU):** Retorna verdadeiro se pelo menos um dos operandos for verdadeiro.
- **! ou not (NÃO):** Inverte o valor booleano do operando. Se for verdadeiro, retorna falso, e vice-versa.