

# Algoritmos - AULA 4

---

## Sumário

1	Introdução ao if.....	1
2	Operadores relacionais .....	2
3	Estrutura básica do if .....	2
4	Exemplo completo .....	3
5	Uso da String .....	4
5.1	A biblioteca string.....	4
5.2	Outra forma de ler um string pelo teclado.....	5
6	Exercícios .....	6

---

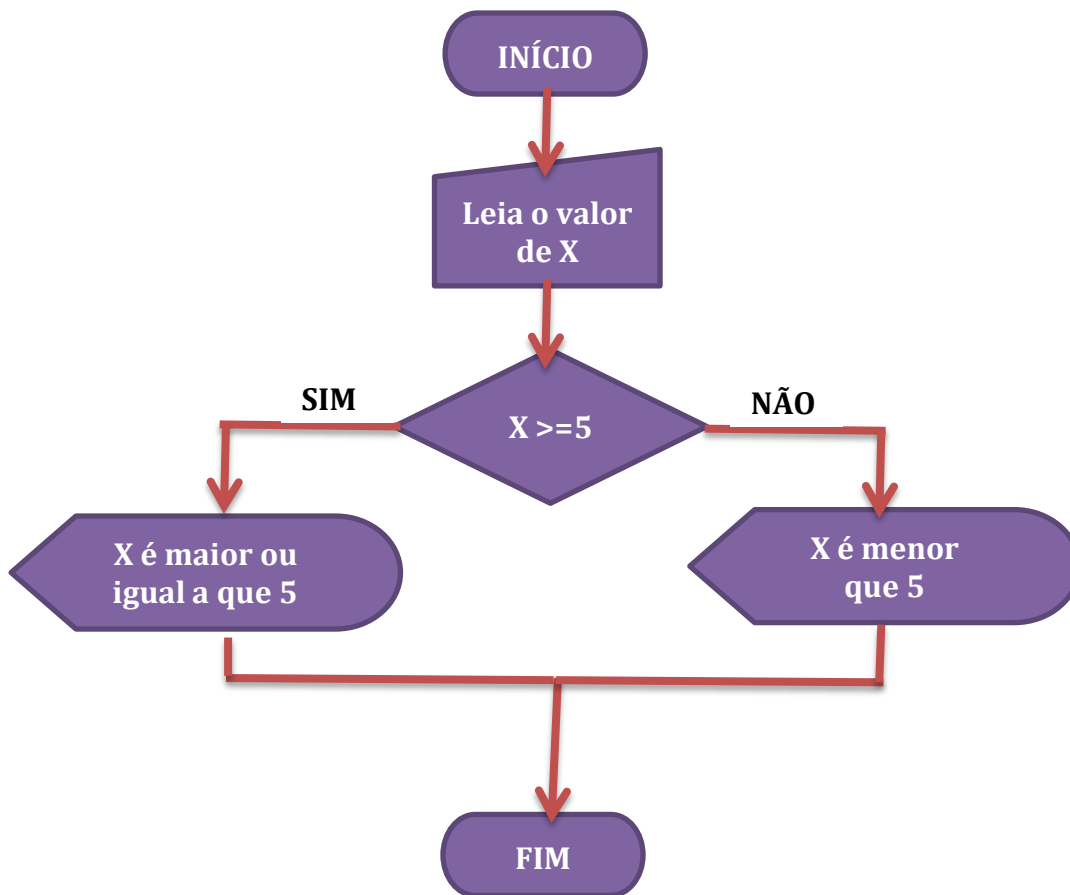
## 1 Introdução ao if

Imagine a seguinte situação: você entra no site do Detran para consultar as multas já processadas, débitos de taxas de licenciamento, IPVA e seguro obrigatório de um determinado veículo. Para isso você precisa entrar com a identificação da placa e o número do renavam do referido veículo. Esse consulta por mostrar a situação do carro junto ao Detran ou indicar um erro. Nessas situações você poderia utilizar um dos comandos de controle de fluxo são aqueles que permitem ao programador alterar a sequência de execução do programa.

O comando if representa uma tomada de decisão do tipo "SE isto ENTÃO aquilo". A condição do comando if é uma expressão que será avaliada. Se o resultado for qualquer coisa diferente de zero (verdadeira) a declaração as instruções dentro do if serão executadas. Se o resultado for zero (falsa) as instruções dentro do if NÃO serão executadas.

Por exemplo:

1. Leia um valor e guarde em X
2. Se X for maior ou igual a 5
  - a. Mostre uma mensagem dizendo que é maior ou igual a 5
3. Senão
  - a. Mostre uma mensagem dizendo que é menor que 5



## 2 Operadores relacionais

Para utilizar o if você precisa também utilizar um dos operadores que realizam comparações entre variáveis. São eles:

Operador	Ação
>	Maior do que
>=	Maior ou igual a
<	Menor do que
<=	Menor ou igual a
==	Igual a
!=	Diferente de

## 3 Estrutura básica do if

Se você quiser que alguma instrução seja executada apenas se um determinado teste for verdadeiro, poderá utilizar a seguinte sintaxe:

```

if (teste)
{
    instruções
}
  
```

Onde:

- if quer dizer SE em inglês;
- teste são as comparações realizadas com os operadores relacionais;
- { } delimitam as instruções que serão executadas se o teste for verdadeiro.

Caso você queira que algumas instruções sejam executadas apenas quando um determinado teste for verdadeiro, e outras apenas quando determinado teste for falso, poderá utilizar a seguinte sintaxe:

```
if (teste)
{
    instruções
}
else
{
    instruções
}
```

Onde:

- if quer dizer SE em inglês;
- teste são as comparações realizadas com os operadores relacionais;
- As primeiras { } delimitam as instruções do if que serão executadas se o teste for verdadeiro;
- else quer dizer SENÃO em inglês;
- As últimas { } delimitam as instruções do else que serão executadas se o teste for falso.

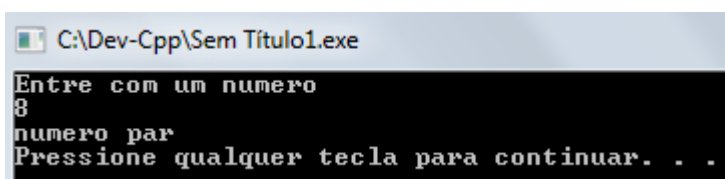
### Observação

Se tiver apenas uma única instrução dentro do if ou do else, as chaves do referido if ou else serão opcionais.

## 4 Exemplo completo

Ler um número e determinar se o mesmo é par ou ímpar. Um número é par se for divisível exato por 2, ou seja, se o referido número for dividido por 2 e for par o resto será zero.

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int numero, resto;
    printf("Entre com um numero\n");
    scanf("%d",&numero);
    resto = numero % 2;
    if(resto == 0)
        printf("numero par\n");
    else
        printf("numero impar\n");
    system("pause");
    return 0;
}
```



## 5 Uso da String

Imagine a seguinte situação: você precisa armazenar o seu nome ou o seu endereço. Nesse caso, para armazenar caracteres em uma variável a mesma deveria ser uma string.

Uma string é uma série de caracteres, e um caractere é o mesmo que um byte. Em programação, cada caractere armazenado na memória de uma string é representado por um valor numérico.

Para representar uma string em C você deve fazer o seguinte:

**char nomeDaVariável[tamanho];**

Onde:

- char indica que se trata de armazenamento de caracteres;
- nomeDaVariável é o nome identificador que o desenvolvedor definiu;
- [tamanho] define o número de máximo de caracteres a serem armazenados na string.

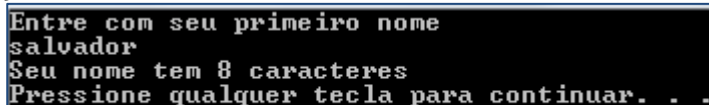
### 5.1 A biblioteca string

A biblioteca padrão string da linguagem C contém várias funções de manipulação de strings. Para usar essas funções, o seu programa deve incluir o arquivo string.h.

- **strlen(string)**: recebe uma string e devolve o seu comprimento, ou seja, devolve o número de caracteres armazenados.

**Exemplo:**

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main( )
{
    int tamanho;
    char nome[20];
    printf("Entre com seu primeiro nome\n");
    scanf("%s",&nome);
    tamanho = strlen(nome);
    printf("Seu nome tem %d caracteres\n",tamanho);
    system("pause");
    return 0;
}
```



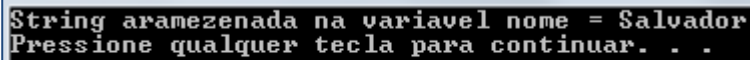
```
Entre com seu primeiro nome
salvador
Seu nome tem 8 caracteres
Pressione qualquer tecla para continuar. . .
```

- **strcpy(string1,string2)**: recebe duas strings e copia o conteúdo do segundo string para o endereço de memória da primeira string. O conteúdo original da primeira string é perdido.

**Exemplo:**

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
```

```
int main( )
{
    char nome[20];
    strcpy(nome,"Salvador");
    printf("String aramezenada na variavel nome = %s\n",nome);
    system("pause");
    return 0;
}
```

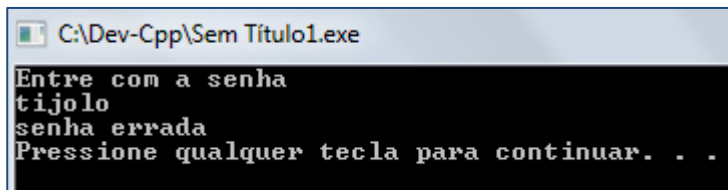


```
String aramezenada na variavel nome = Salvador
Pressione qualquer tecla para continuar. . .
```

- **strcmp(string1,string2):** recebe duas strings, compara as duas e retorna um valor.
  - devolve um número negativo se a primeira string for menor que a segunda;
  - devolve 0 se as duas strings são iguais;
  - devolve um número positivo se a primeira string for maior que a segunda.

#### Exemplo:

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main( )
{
    char senha[13];
    int valor;
    printf("Entre com a senha\n");
    scanf("%s",&senha);
    valor = strcmp(senha,"alasca");
    if(valor==0)
        printf("senha correta\n");
    else
        printf("senha errada\n");
    system("pause");
    return 0;
}
```



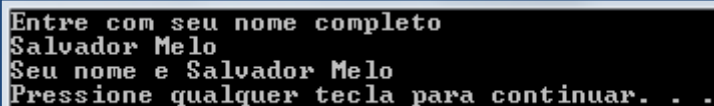
```
C:\Dev-Cpp\Sem Título1.exe
Entre com a senha
tijolo
senha errada
Pressione qualquer tecla para continuar. . .
```

## 5.2 Outra forma de ler um string pelo teclado

Quando você utiliza a função `scanf( )` para ler uma string, você consegue armazenar na variável indicada apenas até que a referida função encontre um espaço em branco, quando então para de armazenar.

Se quiser armazenar caracteres com espaços em branco, como no caso do seu nome completo ou do seu endereço, deverá utilizar a função `gets( )`, da seguinte forma:

```
#include<stdio.h>
#include<stdlib.h>
int main( )
{
    char nomeCompleto[230];
    printf("Entre com seu nome completo\n");
    gets(nomeCompleto);
    printf("Seu nome e %s\n",nomeCompleto);
    system("pause");
    return 0;
}
```



```
Entre com seu nome completo
Salvador Melo
Seu nome e Salvador Melo
Pressione qualquer tecla para continuar. . .
```

## 6 Exercícios

- 1 Teste e analise os exemplos
- 2 Leia a idade de uma pessoa e mostra se a mesma é maior de idade ou não.
- 3 Leia dois números e informa se os mesmos são iguais ou diferentes.
- 4 Leia dois números diferentes e informe qual deles é o maior.
- 5 Leia o nome e quatro notas de um aluno durante o semestre, calcula a média aritmética dessas notas, e informa se o aluno foi aprovado ou reprovado. A média para aprovação é 6.0 (seis).
- 6 Leia o nome de uma cidade e informe se o referido nome tem mais de 10 caracteres ou não.
- 7 Leia um CPF, no formato de string, e informe se o mesmo tem 11 dígitos ou não.
- 8 Leia um CPF, no formato de string, e informe se é o seu CPF ou não.