# Introdução à Programação Programação com desvios

#### Nelson Carvalho Sandes

Centro de Ciências Tecnológicas - CCT Universidade Federal do Cariri

2019





## **Tópicos**

- 1 Operadores relacionais
- 2 Instrução if
- 3 Instrução if...else
- 4 Estruturas aninhadas
- 5 Operadores lógicos





# **Tópicos**

- 1 Operadores relacionais
- 2 Instrução if
- 3 Instrução if...else
- 4 Estruturas aninhadas
- 5 Operadores lógicos





# Operadores relacionais

- É possível expressar sentenças que sejam falsas ou verdadeiras em C.
- Para isso, utiliza-se operadores relacionais.

Operadores relacionais						
Símbolo	Significado	Exemplo	Resultado			
>	maior que	5 > 3	1 (V)			
<	menor que	4 < 3	0 (F)			
>=	maior ou igual que	10 >= 10	1 (V)			
<=	menor ou igual que	7 <= 10	1 (V)			
== igual a		7 == 7	1 (V)			
! = diferente de (não igual)		7!=7	0 (F)			







# **Tópicos**

- 1 Operadores relacionais
- 2 Instrução if
- 3 Instrução if...else
- 4 Estruturas aninhadas
- 5 Operadores lógicos





#### Instrução if

- As condições servem para indicar quando uma parte do programa deve ser executada ou não.
- Em C, a estrutura de decisão é o if.

```
if (condição) {
    Instrução 1, caso condição seja verdadeira.
    Instrução 2, caso condição seja verdadeira.
    .
    .
    .
    .
}
```





Instrução if: Exemplo 1

Este algoritmo, no final de sua execução, indica qual número digitado pelo usuário é maior.

```
int a, b;
printf("Digite dois valores \n");
scanf("%d %d", &a, &b);
if (a > b) {
    printf("O primeiro número é maior \n");
}
if (b > a) {
    printf("O segundo número é maior \n");
}
```





Instrução if: Exemplo 2

■ Exemplo que diz se carro é novo ou velho:

```
int idade_carro;
printf("Digite a idade do seu carro \n");
scanf("%d", &idade_carro);
if (idade_carro <= 3) {
    printf("Seu carro é novo \n");
}
if (idade_carro > 3) {
    printf("Seu carro é velho \n");
}
```





Instrução if: Exercício

■ Escreva um programa que pergunte a velocidade do carro de um usuário. Caso ultrapasse 80km/h, exiba uma mensagem dizendo que o usuário foi multado. Nesse caso, exiba o valor da multa cobrando R\$5,00 por km acima de 80km/h





# **Tópicos**

- 1 Operadores relacionais
- 2 Instrução if
- 3 Instrução if...else
- 4 Estruturas aninhadas
- 5 Operadores lógicos





Instrução if...else

- Quando existem situações em que a segunda condição é o inverso da primeira (exemplo do carro), podemos usar a cláusula else para especificar o que fazer caso a condição do if for falsa. Sem precisar usar a instrução if novamente.
- Estrutura if...else

```
if (condição) {
     Executa código caso a condição seja verdadeira.
} else {
     Executa código caso a condição seja falsa.
}
```





Instrução if...else: Exemplo do carro

■ Exemplo do algoritmo do carro utilizando a estrutura if...else

```
int idade_carro;
printf("Digite a idade do seu carro \n");
scanf("%d", &idade_carro);
if (idade_carro <= 3) {
    printf("Seu carro é novo \n");
} else {
    printf("Seu carro é velho \n");
}</pre>
```





Instrução if...else: Exemplo média final

 Algoritmo que recebe a média ponderada mp e a nota final avf como entrada. Em seguida, informa aprovação ou reprovação.

```
float mp, avf, mf;
printf("Digite a média ponderada \n");
scanf("%f', &mp);
printf("Digite a nota da avaliação final \n");
scanf("%f', &avf);
mf = (mp + avf)/2
if (mf >= 5) {
    printf("Aprovado \n");
} else {
    printf("Reprovado \n");
}
```





Instrução if...else: Exercícios

Escreva um programa que pergunte o salário do funcionário e calcule o valor do aumento. Para salários maiores que R\$1.250, calcule um aumento de 10%. Para salários com valores menores ou iguais, use uma taxa de 15%.





# **Tópicos**

- 1 Operadores relacionais
- 2 Instrução if
- 3 Instrução if...else
- 4 Estruturas aninhadas
- 5 Operadores lógicos





- 1 Nem sempre os programas serão simples.
- 2 Muitas vezes, precisaremos aninhar vários **if** para obter o comportamento desejado do programa.
- 3 Aninhar, nesse caso, é utilizar um **if** dentro de outro.





- Suponha que temos diferentes categorias de produto.
- Dependendo da categoria comprada, o preço do produto será diferente.
- Como fazer um algoritmo, com o conteúdo visto em sala, que leia a categoria de um produto e determine o preço dele?

Categoria	Produto	
1	R\$10,00	
2	R\$18,00	
3	R\$23,00	





```
int categoria; float preco;
scanf("%d", &categoria);
if (categoria == 1) {
     preco = 10:
} else {
     if (categoria == 2) {
          preco = 18:
     } else {
          if (categoria == 3) {
               preco = 23;
          } else {
               printf("Categoria inexistente \n");
               preco = 0;
printf("O preço do produto é: %f \n", preco);
```





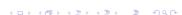
- Solução anterior não é elegante.
- Coloca vários if...else dentro de outros.
- Uma solução mais legível, é utilizar a instrução if logo após a else.





```
int categoria; float preco;
scanf("%d", &categoria);
if (categoria == 1) {
     preco = 10;
} else if (categoria == 2) {
     preco = 18;
} else if (categoria == 3) {
     preco = 23:
} else {
     printf("Categoria inexistente \n");
    preco = 0:
printf("O preço do produto é: %f \n", preco);
```





Mais exemplos de estruturas aninhadas

 Escreva um programa que recebe um inteiro como entrada e produza os resultados "positivo", "negativo" ou "nulo" como saída.

```
int n;
scanf("%d", n);
if (n > 0) {
    printf("Positivo \n");
} else if (n < 0) {
    printf("Negativo \n");
} else {
    printf("Nulo \n");
}</pre>
```





#### Instrução switch

- A instrução if, logo após de uma instrução else, facilita muito a legibilidade do código.
- Apesar disso, quando se tem um número muito elevado de categorias, a instrução switch é mais adequada.
- Vamos supor o mesmo exemplo com 5 categorias. Como resolve-lo usando a instrução switch?

Categoria	Produto	
1	R\$10,00	
2	R\$18,00	
3	R\$23,00	
4	R\$26,00	
5	R\$31,00	

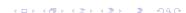




#### Instrução switch

```
int categoria; float preco;
preco = 0;
scanf("%d", &categoria);
switch (categoria) {
     case 1: preco = 10; break;
     case 2: preco = 18; break;
     case 3: preco = 23; break:
     case 4: preco = 26; break;
     case 5: preco = 31; break;
     default: printf("Categoria inexistente \n"); break;
printf("O preço do produto é: %f \n", preco);
```





#### Exercícios

Escreva um programa que leia dois números e faça uma operação matemática básica entre eles. O usuário deverá selecionar o tipo de operação realizada: 1 - Soma, 2 -Subtração, 3 - Multiplicação e 4 - Divisão.





#### Exercícios

Escreva um programa que calcule o preço a pagar pelo fornecimento de energia elétrica. Pergunte a quantidade de kWh consumida e o tipo de instação: R para residências, I para indústrias e C para comércios. Calcule o preço a pagar de acordo com a tabela a seguir:

Preço por tipo de consumo				
Tipo	Faixa	Preço		
Residencial	Até 500	R\$0,40		
Residencial	Acima de 500	R\$0,65		
Comercial	Até 1000	R\$0,55		
Comerciai	Acima de 1000	R\$0,60		
Industrial	Até 5000	R\$0,55		
maustriai	Acima de 5000	R\$0,60		





# **Tópicos**

- 1 Operadores relacionais
- 2 Instrução if
- 3 Instrução if...else
- 4 Estruturas aninhadas
- 5 Operadores lógicos





- Assim como na lógica formal, boa parte das linguagens de programação utilizam operadores de negação, conjunção e disjunção.
- Em C, se utiliza da expressão "!" para negação, "&&" para conjunções e "||" para disjunções.

Operador	Operação	Explicação	Exemplo	Res
!x	Negação	Resultado contrário à x	!(5 < 3)	V
x && y	Conjunção	V se x e y forem V	(1 == 1) && (3 > 2)	V
x    y	Disjunção	V se x ou y for V	(2 == 1)    (2 > 4)	F





#### Conjunções

 Assim como na lógica formal, o operador e (&&) retorna verdadeiro apenas quando todas as condições usadas forem simultaneamente verdadeiras.

```
int numero;
scanf("%d", &numero);
if ((numero >= 20) && (numero <= 90)) {
    printf("O valor está entre 20 e 90 \n");
} else {
    printf("O valor não está entre 20 e 90 \n");
}</pre>
```





#### Conjunções

■ Programa que pergunte ao usuário as horas [0-23] e mostra na tela "Bom dia", "Boa tarde" ou "Boa noite".

```
int horas;
printf("Que horas são? [0-23] \n");
scanf("%d", &horas);
if ((horas > 3) && (horas < 12)) {
    printf("Bom dia \n");
} else if (horas >= 12) && (horas < 18) {
    printf("Boa tarde \n");
} else:
    printf("Boa noite \n");
}</pre>
```





Exercício

 Escreva um programa que leia três números e que imprima o maior e o menor.





#### Disjunções

 Assim como na lógica formal, o operador ou (||) retorna verdadeiro se pelo menos uma das condições mencionadas for verdadeira.





#### Negação

 Assim como na lógica formal, o operador not inverte o valor lógico de uma condição.

```
int valor; 

printf("Digite um número inteiro \n"); 

scanf("%d", &valor); 

if !(valor >= 10) { 

    printf("Valor menor do que 10 \n"); 

} else { 

    printf("Valor maior ou igual a 10 \n") 

}
```





#### Exercício

Escreva um programa que leia três valores para os lados A, B e C de um triângulo. Esse programa deve verificar se os lados fornecidos formam realmente um triângulo. Se essa condição for verdadeira, deve ser indicado o tipo de triângulo formado: isósceles, escaleno ou equilátero.

#### Dicas:

- 1 Para uma figura de três lados ser um triângulo, um lado sempre deve ser menor do que a soma dos outros dois. Ou seja: (A < B + C) e (B < A + C) e (C < A + B)
- 2 Um triângulo equilátero possui os três lados iguais.
- 3 Um triângulo isósceles possui 2 lados iguais e um diferente.
- 4 Um triângulo é escaleno quando todos os seus lados são diferentes.



