



Programação de Computadores

Comandos de Controle do Programa

Neste tópico abordaremos os comandos de decisão e de repetição da linguagem C.

Prof. Ciro Cirne Trindade / Prof. Thiago Ferauche



Expressões Lógicas

- Verdadeiro e Falso em C
 - Representa verdadeiro e falso por valores inteiros, 1 e 0, respectivamente
 - Qualquer coisa diferente de 0 é verdadeira
 - C99
 - `_Bool`
 - `stdbool.h`
 - `bool`
 - `true`
 - `false`

Expressões Lógicas

- Operadores relacionais

Símbolo	Operação
>	maior
>=	maior ou igual
<	menor
<=	menor ou igual
==	igualdade
!=	diferente

Expressões Lógicas

- Operadores lógicos

Símbolo	Operador
&&	E lógico
	OU lógico
!	NEGAÇÃO lógica

- Se $exp1$ e $exp2$ são duas expressões simples:
 - $exp1 \ \&\& \ exp2$ é verdadeira se as duas, $exp1$ e $exp2$, forem verdadeiras
 - $exp1 \ || \ exp2$ é verdadeira se uma das duas, $exp1$ ou $exp2$, for verdadeira ou ambas forem verdadeiras
 - $!exp1$ é verdadeira se $exp1$ for falsa e vice-versa

Expressões Lógicas

- Prioridade dos operadores lógicos

Operador	Prioridade
!	1º
&&	2º
	3º

Os operadores unários têm prioridade sobre os binários

- Os operadores aritméticos têm prioridade sobre os relacionais que por sua vez têm prioridade sobre os lógicos

a + b > c - d * 2 || f >= g + 1 && !h

2º 6º 4º 3º 9º 7º 5º 8º 1º

Comandos de Seleção

- **if**

- A forma geral do comando **if** é:

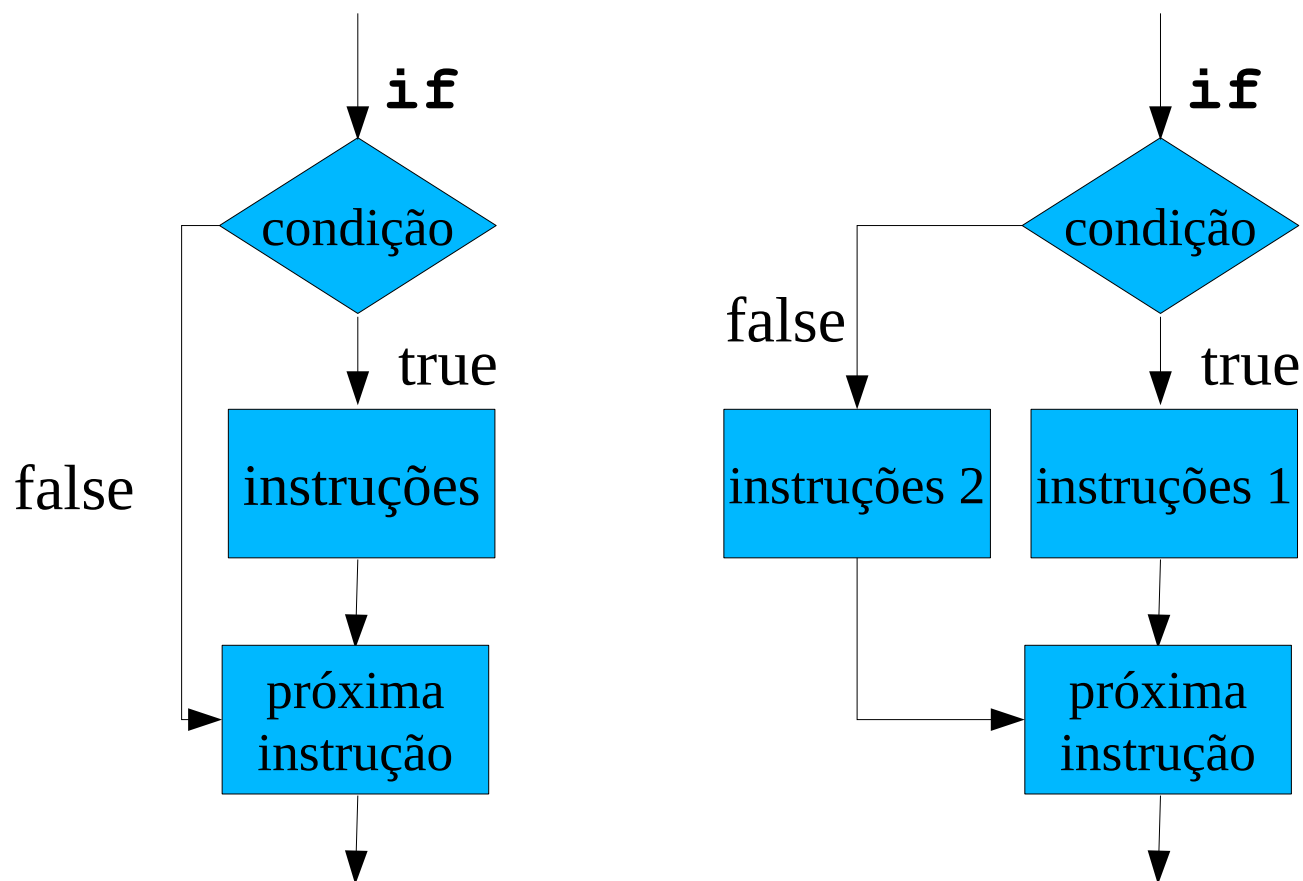
```
if (expressão) {  
    sequência de comandos 1;  
} else {  
    sequência de comandos 2;  
}
```

O **else** é
opcional

- Que pode ser entendido da seguinte forma:
 - se **expressão** for verdadeira, então a **sequência de comandos 1** é executada
 - caso contrário (**expressão** for falsa), a **sequência de comandos 2** é executada

Comandos de seleção

- **if versus if-else**



Exemplo do uso do `if`

```
/* programa que dá 20% de desconto se o número de
   ingressos comprados for maior que 5 */
#include <stdio.h>
int main() {
    float preco, custo;
    int num_ingressos;
    printf("Preço do ingresso: R$ ");
    scanf("%f", &preco);
    printf("Número de ingressos: ");
    scanf("%d", &num_ingressos);
    custo = num_ingressos * preco;
    if (num_ingressos > 5) {
        custo = custo - custo * 0.2;
    }
    printf("Custo de %d ingressos: R$ %.2f\n",
           num_ingressos, custo);
    return 0;
}
```




Exemplo do uso do if-else

```
/* programa que determina o maior entre 2 números
   Inteiros diferentes */
#include <stdio.h>
int main() {
    int a, b, maior;
    printf("Digite 2 numeros inteiros diferentes: ");
    scanf("%d %d", &a, &b);
    if (a > b) {
        maior = a;
    }
    else {
        maior = b;
    }
    printf("Maior = %d\n", maior);
    return 0;
}
```



Exemplo do uso do if-else encaixados

```
/* programa que determina o maior entre 2 números
   Inteiros sem a premisa de que são diferentes */
#include <stdio.h>
int main() {
    int a, b, maior;
    printf("Digite dois numeros inteiros:");
    scanf("%d %d", &a, &b);
    if (a == b) {
        printf("Os numeros sao iguais\n");
    }
    else {
        if (a > b) {
            maior = a;
        }
        else {
            maior = b;
        }
        printf("Maior = %d\n", maior);
    }
    return 0;
}
```

Exercícios

1) Escreva um programa que dado o salário de um funcionário, imprimir o desconto do INSS segundo a tabela a seguir:

Salário	Desconto do INSS
Até R\$ 1.045,00	7,5%
De R\$ 1.045,01 a R\$ 2.089,60	9,0%
De R\$ 2.089,61 a R\$ 3.134,40	12,0%
De R\$ 3.134,41 a R\$ 6.101,66	14,0%
Atenção: O teto de contribuição do INSS é de R\$ 6.101,66	



Exercícios

2) Um banco concederá um crédito especial aos seus clientes de acordo com o saldo médio do último ano. Faça um programa em C que receba o saldo médio de um cliente e calcule o valor do crédito, de acordo com a tabela a seguir. Mostre o valor do crédito.

Saldo Médio	Percentual
Acima de R\$ 2.000,00	200% do saldo médio
A partir de R\$ 2.000,00 até R\$ 1.000,00 (exclusive)	150% do saldo médio
A partir de R\$ 1.000,00 até R\$ 500,00 (exclusive)	100% do saldo médio
Até R\$ 500,00 (inclusive)	75% do saldo médio

Comandos de Seleção

■ **switch**

- Usado no lugar de um bloco de **if-else-if...**
- Testa múltiplas opções
- Forma geral:

Testa a igualdade entre as constantes e a expressão

```
switch (expressão) {  
    case constante1: sequência de comandos;  
        break;  
    case constante2: sequência de comandos;  
        break;  
    ...  
    default: sequência de comandos;  
}
```

opcional

Termina a
execução do
switch

Comandos de Seleção

- Operador condicional ternário (?:)
 - O ?: é chamado operador condicional ternário porque requer três operandos
 - Forma geral:
$$\text{Exp1} \ ? \ \text{Exp2} \ : \ \text{Exp3};$$
 - A **Exp1** é avaliada primeiro
 - Se seu valor for diferente de zero (verdadeiro), a **Exp2** é avaliada e seu resultado será o valor da expressão
 - Se **Exp1** for zero (falso), a **Exp3** é avaliada e será o valor da expressão

Comandos de Repetição

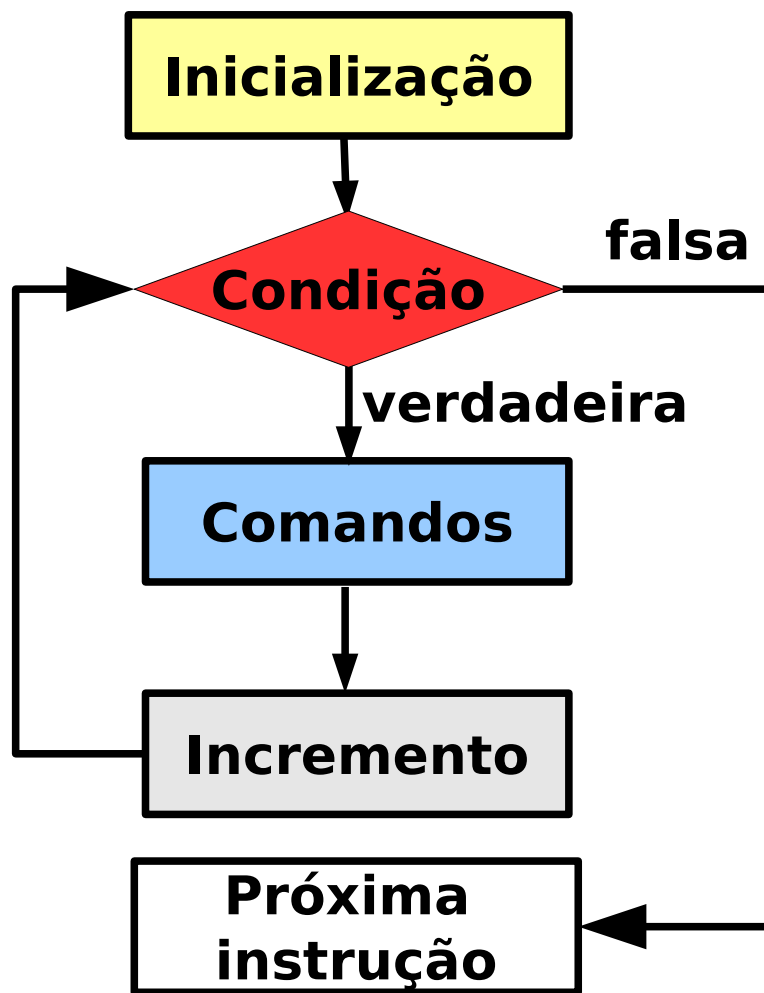
- O Laço **for**

- Muito flexível e poderoso
- Forma geral

```
for(inicialização; condição; incremento) {  
    sequência de comandos;  
}
```

- **inicialização**: geralmente uma atribuição usada para definir a variável de controle do laço
- **condição**: expressão que determina quando o laço terminará
- **incremento**: define como a variável de controle do laço será modificada cada vez que o laço é repetido

Fluxo de execução do for





Operadores de incremento (++) e decremento (--)

- Operadores unários
 - ++ soma 1 a seu operando
 - -- subtrai 1 de seu operando
- 2 modos: pré-fixado e pós-fixado
 - Pós-fixado: o operador segue a variável
 - O valor da variável é utilizado na expressão e só depois o valor da variável é atualizado
 - Pré-fixado: operador precede a variável
 - O valor da variável é atualizado e depois seu valor é usado na expressão

Operadores aritméticos de Atribuição ($+=$, $-=$, $*=$, $/=$, $\%=$)

- Cada um desses operadores é usado com uma variável a sua esquerda e uma expressão à direita
 - $x \text{ op} = \text{exp};$
- Equivale a
 - $x = x \text{ op } (\text{exp}) ;$



Comandos de Repetição

- O Laço **while**

- Forma geral

```
while (condição) {  
    sequência de comandos;  
}
```

- O comando ou sequência de comandos são repetidos enquanto a condição for verdadeira
 - A condição é avaliada no início, antes da sua execução, podendo até não ser executado nenhuma vez



Comandos de Repetição

- O Laço **do-while**

- Forma geral

```
do {  
    sequência de comandos;  
} while (condição);
```

- A sequência de comandos é repetida enquanto a condição é verdadeira
 - A condição é estada no final do laço
 - Isso significa que o laço `do-while` é executado pelo menos uma vez



Comandos de Repetição

- Comando `break`
- Quando usado dentro de um laço, força a saída do laço
- Exemplo:

```
#include <stdio.h>
int main() {
    char ch;
    for(;;) { // laço infinito
        printf("Digite um caractere (. p/ sair): ");
        ch = getchar();
        getchar();
        if (ch == '.') break;
        printf("Codigo ASCII de %c: %d\n", ch, ch);
    }
    return 0;
}
```

Comandos de Repetição

- Comando `continue`
 - É usado dentro de um laço para pular para a próxima iteração
 - Exemplo:

```
#include <stdio.h>
int main() {
    int i;
    for (i = 0; i < 10; i++) {
        printf("\n%d antes do continue", i);
        if (i > 5) continue;
        printf("\tapos o continue");
    } // fim do for
    return 0;
}
```



Referências

- SCHILDT, Herbert. *C Completo e Total*. 3. ed., Makron Books, 1997.
- DEITEL, H.M.; DEITEL, P.J.. *Como Programar em C*. LTC, 2. ed., 1999.
- MIZRAHI, V.V.. *Treinamento em Linguagem C*. Módulo 1, Makron Books, 1990.
- PRATA, Stephen. *C Primer Plus*. 6. ed., Addison Wesley, 2014.