

# Operadores Lógicos

## Módulo 4 Aula 1

Linguagem C, o Curso Definitivo WR Kits

Autor: Dr. Eng. Wagner Rambo

# Operadores Lógicos

- ▶ Quando precisamos fazer alguma análise lógica de expressões, recorremos aos operadores lógicos. Com eles, é possível de se combinar operandos e também expressões de operadores relacionais.
- ▶ Os operadores lógicos são necessários em contextos em que apenas uma única condição não é o bastante para a tomada de decisão.
- ▶ Exemplo:
- ▶ Ouviremos o som se a frequência for maior que 20Hz e menor que 20kHz.

# C tem 3 operadores lógicos

- ▶ `&&`      AND lógico
  - ▶ `||`      OR lógico
  - ▶ `!`      NOT negação
- 
- ▶ Seguem a mesma tabela verdade das lógicas AND (E), OR (OU) e NOT (negação).
  - ▶ Para o resultado de uma lógica AND ser verdadeiro, TODAS as condições devem ser verdadeiras.
  - ▶ Para o resultado de uma lógica OR ser verdadeiro, UMA ou MAIS condições precisam ser verdadeiras.
  - ▶ Para o resultado de uma lógica NOT ser verdadeiro, a condição avaliada deverá ser falsa.

# Exemplo lógica E

- ▶ O programa abaixo identifica a faixa de valores entre 10 e 20.

```
main()
{
    int val;

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

    if(val>=10 && val<=20)
        printf("Faixa de valores entre 10 e 20.\n");

    else
        printf("Faixa de valores menor que 10 ou maior que 20.\n");
}
```

# Exemplo lógica OU

- ▶ O programa abaixo aceita a opção em maiúscula ou minúscula.

```
main()
{
    char opt;

    printf("Digite s para sim e n para nao: ");
    scanf("%c",&opt);

    if(opt == 's' || opt == 'S')
        printf("Voce escolheu sim.\n");

    if(opt == 'n' || opt == 'N')
        printf("Voce escolheu nao.\n");

}
```

# Exemplo lógica NOT

- ▶ Este programa verifica se o valor digitado foi zero ou não.

```
main()
{
    char teste;

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

    if(!teste)
        printf("O valor foi ZERO.\n");
    else
        printf("Valor diferente de ZERO.\n");
}
```

# Lógica OU Exclusivo (XOR)

- ▶ A Linguagem C não apresenta um operador lógico para XOR. OBS.: Não confundir com operadores bit a bit, que veremos mais adiante. Porém, o algoritmo a seguir combina operadores lógicos e relacionais para se obter a lógica OU Exclusivo, onde, para o resultado ser verdadeiro, o número de condições verdadeiras precisa ser ímpar.

```
main(){
    int opA, opB, xor;

    printf("Entre com opA: ");
    scanf("%d",&opA);
    printf("Entre com opB: ");
    scanf("%d",&opB);
    xor = (opA || opB) && !(opA && opB);

    printf("%d XOR %d = %d\n",opA,opB,xor);
}
```

# Precedência dos operadores

- ▶ Os operadores relacionais e lógicos em C têm a seguinte precedência:

- ▶
  - Maior
  - !
  - > >= < <=
  - == !=
  - &&
  - ||
  - Menor



# Combinando operadores

- ▶ Pode-se combinar diversos operadores em uma mesma expressão.
- ▶ Exemplo:

```
15>11 && 8<4 || 7>=2
```

- ▶ Essa expressão resultará em verdadeiro ou falso?

# Exercício proposto

- ▶ Escreva um programa em C que solicite a idade do indivíduo e se ele tem carro (s ou S) ou não tem (n ou N).
- ▶ Os indivíduos sem carro e com idade entre 18 e 50 anos receberão 500 Reais de auxílio transporte. Nos demais casos, o auxílio será de 200 Reais.
- ▶ Indivíduo entre 18 e 50 anos sem carro recebe 500 Reais.
- ▶ Indivíduo com qualquer idade com carro recebe 200 Reais.
- ▶ Indivíduo com menos de 18 e mais de 50 anos recebem 200 Reais.

# Bibliografia

- ▶ DAMAS, Luís; Linguagem C, décima edição.
- ▶ Disponível em: <https://amzn.to/3nGdlbN>