INSTITUTO FEDERAL
Catarinense
Campus Blumenau

Bacharelado em
ENGENHARIA
ELÉTRICA

Algoritmos - Expressões Lógicas - Operadores Relacionais

Tarefa - Algoritmos - Expressões Lógicas - Operadores Relacionais

1 Expressões Lógicas - Operadores Relacionais

(Texto gerado com auxílio do ChatGPT)

Os operadores relacionais são uma parte importante da lógica de programação em muitas linguagens de programação, incluindo C. Eles são usados para comparar valores e determinar se um valor é maior que, menor que, igual ou diferente de outro valor. Operadores relacionais são comumente usados em declarações condicionais, loops e outras estruturas de controle para controlar o fluxo de um programa.

Existem seis operadores relacionais: menor que ($<$), maior que ($>$), menor ou igual a ($\leq$), maior ou igual a ($\geq$), igual a ($==$) e diferente de ($!=$). Esses operadores são operadores binários, o que significa que eles levam dois operandos (valores) e retornam um valor booleano (verdadeiro ou falso) dependendo do resultado da comparação. A tabela abaixo descreve os operadores relacionais presentes na Linguagem C.

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

Tabela 1: Operadores Relacionais em C

Por exemplo, suponha que temos duas variáveis, x e y, e queremos comparar seus valores usando operadores relacionais. Podemos usar a seguinte sintaxe:

```
1 if (x < y) {
2   // faça algo se x for menor que y
3 }
4
```

```c
if (x >= y) {
  // faça algo se x for maior ou igual a y
}

if (x == y) {
  // faça algo se x for igual a y
}

if (x != y) {
  // faça algo se x for diferente de y
}
```

No primeiro exemplo, o programa executará o código dentro da instrução `if` somente se x for menor que y. No segundo exemplo, o programa executará o código dentro da instrução if somente se x for maior ou igual a y. No terceiro exemplo, o programa executará o código dentro da instrução if somente se x for igual a y. No quarto exemplo, o programa executará o código dentro da instrução if somente se x for diferente de y.

Em resumo, operadores relacionais são uma parte essencial da lógica de programação em C. Eles nos permitem comparar valores e controlar o fluxo de um programa com base no resultado da comparação. Operadores relacionais são usados em declarações condicionais, loops e outras estruturas de controle, e são uma ferramenta importante para qualquer programador C dominar.

2 Expressões Lógicas e Operadores Relacionais

(Texto gerado com auxílio do ChatGPT)

Os operadores relacionais são frequentemente utilizados em conjunto com operadores lógicos para formar expressões lógicas em linguagens de programação. Expressões lógicas são expressões que avaliam para verdadeiro ou falso com base nos valores de seus operandos. Os operadores lógicos mais comumente usados em programação são os operadores `&&` (E lógico), `||` (OU lógico) e `!` (NÃO lógico).

Por outro lado, os operadores relacionais são usados para comparar os valores de dois operandos e determinar se eles são iguais, diferentes, menores, maiores, menores ou iguais ou maiores ou iguais entre si. O resultado de um operador relacional é sempre um valor booleano (true ou false).

Aqui está um exemplo de como usar operadores relacionais e operadores lógicos juntos em C:

```c
#include <stdio.h>

int main() {
    int a = 10, b = 20, c = 30;

    if ((a < b) && (b < c)) {
        printf("a é menor que b e b é menor que c\n");
    } else {
        printf("a não é menor que b ou b não é menor que c\n");
    }

    return 0;
}
```

Neste exemplo, o operador relacional < é usado para comparar os valores das variáveis a, b e c. O operador lógico `&&` é então usado para combinar os resultados dessas operações relacionais em uma única expressão lógica. A expressão lógica resultante será verdadeira apenas se ambas as condições `(a < b)` e `(b < c)` forem verdadeiras.

Se a expressão lógica for verdadeira, o programa imprimirá a mensagem "a é menor que b e b é menor que c". Se a expressão lógica for falsa, o programa imprimirá a mensagem "a não é menor que b ou b não é menor que c".

Ao combinar operadores relacionais e operadores lógicos, você pode criar expressões complexas que podem ser usadas para tomar decisões em seus programas com base em várias condições.

É importante entender o compoartamento desses operadores e como eles interagem entre si para produzir os resultados desejados.

3 Leituras Adicionais

Os livros [Paul Deitel, 2022] e [Brian W. Kernighan, 1988] podem ser utilizados para obter mais informações sobre expressões lógicas.

O livro [Souza, 2010] apresenta um estudo sobre lógica. Ele traz conteúdos básicos como operadores, formas normais, equivalência entre expressões lógicas, demonstrações de argumentos lógicos e teoremas.

4 Exercícios

1. Sejam a, b, c variáveis inteiras e os símbolos $\neg, \wedge, \vee$ representando NÃO, E e OU respectivamente, escreva tabelas verdade (pode ser em papel) para
   (a) $\neg(a$ é igual a 10)
   (b) $(a > 5) \wedge (b < 20)$
   (c) (a é par) $\wedge$ (c é impar)
   (d) (a é par) $\vee$ (c é impar)
   (e) $(c > a) \wedge (c < b)$
   (f) $(c \leq a) \vee (c \geq b)$
   (g) $((c > a) \vee (c > b)) \wedge (\neg((c > a) \wedge (c > b)))$ (Essa expressão é equivalente ao OU EXCLUSIVO (conhecido também como *XOR*. Ela será verdade se e, somente se, $c > a$ ou $c > b$ e não ambos, ou seja, quando apenas uma única expressão de $\{(c > a), (c > b)\}$ for verdadeira.)
2. Escreva um programa em C que deverá conter uma função para cada um dos itens do exercício anterior para avaliar quando a expressão é verdadeira ou falsa (a função deverá ter como parâmetros de entrada variáveis inteiras e devolver como saída um inteiro que representará falso com o valor 0 e verdadeiro com qualquer valor diferente de zero).

   Na função principal (*main*) deverá ser criadas variáveis e atribuidos valores, teste as funções e compare com os valores esperados descritos nas tabelas verdade do exercício anterior.

Referências

[Brian W. Kernighan, 1988] Brian W. Kernighan, D. M. R. (1988). *C Programming Language*. Prentice Hall, 2 edition.

[Paul Deitel, 2022] Paul Deitel, H. D. (2022). *C How to Program*. Pearson, 9 edition.

[Souza, 2010] Souza, J. N. d. (2010). *Lógica para Ciência da Computação*. Bookman Editora.