

Revisão

Prof. John Lenon C. Gardenghi

3 Elementos básicos de algoritmos

3.2 Operadores de comparação e lógicos

Os operadores de comparação são

- Igual: == (não confundir com o =)
- Diferente: !=
- Maior: >
- Maior ou igual: >=
- Menor: <
- Menor ou igual: <=

São operadores **binários**, isto é, operam sempre sobre dois operadores.

Uma comparação sempre retorna **verdadeiro** ou **falso**. Em linguagem C, zero representa falso, e qualquer coisa diferente de zero é verdadeiro.

Exemplo 1. *O código a seguir funciona? Se sim, qual a sua saída?*

```
#include <stdio.h>
int main() {
    int a = 45, b = 30, c = 30, d;
    d = ( b == c != a );
    printf( "%d\n", d );
    return 0;
}
/* Saída = 1 */
```

Quando queremos combinar comparações, podemos usar **operadores lógicos**. São eles:

- Não: !
- E: &&
- Ou: ||

A combinação de comparações nos dá uma **condição**.

Exemplo 2. Suponha que *diaSemana* varie de 1 a 7, onde 1 é domingo e 7 é sábado, e seja *hora* a hora do dia entre 0 e 23.

Se *diaSemana* = 2 e *hora* = 10, quais os valores a seguir?

```
ocupado1 = ( diaSemana == 2 || diaSemana == 5 ) && ( hora >= 14 && hora < 16 );
ocupado2 = diaSemana == 2 || ( diaSemana == 5 && ( hora >= 14 && hora < 16 ));
ocupado3 = ( diasemana == 2 || ( diaSemana == 5 && hora >=14 )) && hora < 16;
ocupado4 = (( diaSemana == 2 || diaSemana == 5 ) && hora >= 14 ) && hora < 16;
```

```
/* ocupado1 = 0 */
/* ocupado2 = 1 */
/* ocupado3 = 1 */
/* ocupado4 = 0 */
```

Desafio: Elabore uma expressão que indique que você está ocupado nas 2as até às 16h, nas 5as o dia inteiro e nos outros dias após às 14h.

Juntando todos os operadores, temos a seguinte lista de prioridades:

Prioridade	Operador	Ordem
1	() Parênteses	esquerda-direita
2	* Multiplicação / Divisão % Resto	esquerda-direita
3	+ Soma - Subtração	esquerda-direita
4	< Menor <= Menor ou igual > Maior >= Maior ou igual	esquerda-direita
5	== Igual != Diferente	esquerda-direita
6	&& E	esquerda-direita
7	Ou	esquerda-direita
8	= Igual	direita-esquerda

3.3 Estruturas de desvio condicional

Sintaxe:

```
if ( condicao ) {  
    // código se condição verdadeira  
} else {  
    // código se condição for falsa  
}
```

Exemplo 3. *Faça um programa que leia um número inteiro n . Se ele estiver entre 1 e 100, imprima “OK”.*

```
#include <stdio.h>  
int main() {  
    int n;  
    scanf( "%d", &n );  
    if ( n >= 1 && n <= 100 ) {  
        printf( "OK\n" );  
    }  
    return 0;  
}
```

Discussão:

- Uso das chaves dispensável no if.
- Posso fazer assim?

```
if ( 1 <= n <= 100 )
```

E assim?

```
if ( 1 <= n && n <= 100 )
```

Exemplo 4. *Faça um algoritmo que leia dois inteiros a e b e determine se são múltiplos ou não.*

```
#include <stdio.h>  
int main() {  
    int a, b;  
    scanf( "%d %d", &a, &b );  
    if ( ( a >= b && a%b == 0 ) || ( b > a && b%a == 0 ) ) {  
        printf( "São múltiplos\n" );  
    } else {  
        printf( "Não são múltiplos\n" );  
    }  
    return 0;  
}
```

Exemplo 5. Ler três números inteiros a , b e c e imprimir o maior deles.

```
#include <stdio.h>
int main() {
    int a, b, c;
    scanf( "%d %d %d", &a, &b, &c );
    printf( "Maior: " );
    if ( a >= b && a >= c ) {
        printf( "%d\n", a );
    } else {
        if ( b >= a && b >= c ) {
            printf( "%d\n", b );
        } else {
            if ( c >= a && c >= b ) {
                printf( "%d\n", c );
            }
        }
    }
    return 0;
}
```

Discussão

- Podemos ocultar a condição $c \geq a \ \&\& \ c \geq b$? (Sim.)
- Podemos mudar a condição $b \geq a \ \&\& \ b \geq c$ apenas para $b \geq c$? (Sim, pois se o primeiro if não vale, então o maior será b ou c.)

3.4 Estruturas de repetição

Estruturas de repetição são blocos de código que devem ser repetidos até que alguma condição seja satisfeita. Em computação, chamamos isso de **laço** (ou *loop*, do inglês). Cada rodada de um laço é chamada de **iteração**.

Todo laço deve ter três elementos básicos bem definidos:

1. inicialização,
2. passo e
3. parada.

Em linguagem C, contamos com algumas estruturas de repetição. As principais são

1.

```
while( condicao ) {
    // código a ser executado enquanto a condição for verdadeira
}
```
2.

```
for( inicializacao; condicao; passo ) {
    // código a ser executado enquanto a condição for verdadeira
}
```

Exemplo 6. *O laço a seguir está bem definido?*

```
#include <stdio.h>
int main() {
    int n;
    n = 9;
    while ( n != 0 ) {
        printf( "n = %d", n );
        n = n - 2;
    }
    return 0;
}
```