

Introdução a Algoritmos

Raoni F. S. Teixeira

Aula 4 - Expressões Lógicas, Comandos Condicionais e Legibilidade de código

1 Expressões Lógicas

1.1 Definição

Expressões lógicas são aquelas que realizam uma operação lógica (como E, OU e NÃO) e retornam um valor booleano — verdadeiro ou falso. Funcionam de modo semelhante às expressões relacionais, mas operam sobre resultados lógicos.

1.2 Operadores Lógicos em C

Operador	Significado	Exemplo
&&	E lógico (AND)	<code>a > 0 && b < 10</code>
	OU lógico (OR)	<code>a > 0 b < 10</code>
!	NÃO lógico (NOT)	<code>!(a > 0)</code>

1.3 Tabelas Verdade

Operador E (&&) Retorna verdadeiro apenas se ambas as expressões forem verdadeiras.

Op1	Op2	Resultado
V	V	V
V	F	F
F	V	F
F	F	F

```
a == 0 && b == 0
```

Operador OU (||) Retorna verdadeiro quando pelo menos uma das expressões for verdadeira.

Op1	Op2	Resultado
V	V	V
V	F	V
F	V	V
F	F	F

```
a == 0 || b == 0
```

Operador NÃO (!) Inverte o valor lógico da expressão.

Op1	Resultado
V	F
F	V

```
!(a == 0)
```

1.4 Simplificações Úteis

As seguintes equivalências ajudam a escrever expressões mais simples:

$$\begin{aligned}!(a == b) &\Leftrightarrow a != b \\!(a != b) &\Leftrightarrow a == b \\!(a > b) &\Leftrightarrow a <= b \\!(a < b) &\Leftrightarrow a >= b \\!(a >= b) &\Leftrightarrow a < b \\!(a <= b) &\Leftrightarrow a > b\end{aligned}$$

2 Combinando Comandos Condicionais

Com expressões lógicas, é possível simplificar múltiplos ifs aninhados.

```
if (cond1) {  
    if (cond2) {  
        if (!cond3) {  
            comando;  
        }  
    }  
}
```

```
if (cond1 && cond2 && !cond3) {  
    comando;  
}
```

3 Mais Comandos Condicionais

3.1 Revisão: if-else-if

Quando há múltiplas condições baseadas em um mesmo valor, o código pode ficar extenso. Exemplo tradicional:

```
int main(void) {  
    int a;  
    scanf("%d", &a);  
  
    if (a == 10129)  
        printf("Maria_Candida_Moreira_Telles\n");  
    else if (a == 33860)  
        printf("Larissa_Garcia_Alfonsi\n");  
    else if (a == 33967)  
        printf("Leonardo_Kozlowiski_Kenupp\n");  
    else  
        printf("Nenhum_aluno_com_RA_informado!\n");  
}
```

3.2 O comando switch

O comando switch simplifica esse tipo de estrutura quando todas as condições testam o valor de uma única variável.

```
switch (variavel) {
    case valor1:
        comandos;
        break;
    case valor2:
        comandos;
        break;
    default:
        comandos; // opcional
}
```

```
switch (a) {
    case 10129:
        printf("Maria_Candida_Moreira_Telles\n");
        break;
    case 33860:
        printf("Larissa_Garcia_Alfonsi\n");
        break;
    case 33967:
        printf("Leonardo_Kozlowiski_Kenupp\n");
        break;
    default:
        printf("O_aluno_não_está_matriculado\n");
}
```

3.3 Comportamento do switch

A execução começa no primeiro case cujo valor coincide com a variável, e continua até encontrar um break ou o final do bloco. O comando default é executado se nenhuma das condições for verdadeira (semelhante ao último else em um if-else-if).

4 Legibilidade de Código

4.1 Definição e Importância

Legibilidade é a qualidade do código que permite seu fácil entendimento por outros programadores. Um código é escrito uma vez, mas lido e mantido muitas vezes.

Um código legível:

- facilita a manutenção e reuso;
- reduz erros;
- é mais facilmente revisado por outras pessoas.

4.2 Características de um Código Legível

- Comentários claros e objetivos;
- Boa formatação (indentação, espaçamento e nomes consistentes);
- Documentação adequada e padronizada.

4.3 Exemplos de Boas e Más Práticas**Comentários Mal:**

```
#include <stdio.h>
#include <math.h>
int main() {
    // declara tr s vari veis
    float x, y, z;
    // l um valor do teclado
    scanf("%f", &x);
    // l outro valor
    scanf("%f", &y);
    // calcula o x ao quadrado mais y ao quadrado
    z = x*x + y*y;
    // tira a raiz de z
    z = sqrt(z);
    // mostra o valor de z
    printf("%f\n", z);
}
```

Bem:

```
#include <stdio.h>
#include <math.h>
int main() {
    // catetos e hipotenusa
    float x, y, z;

    // l valores dos catetos
    scanf("%f", &x);
```

```
scanf("%f", &y);

// aplica Pit goras
z = sqrt(x*x + y*y);

// devolve a hipotenusa
printf("%f\n", z);
}
```

Indentação Mal:

```
if(n==1){printf("Unidade");}
else if (n>=0) {
if (n%3)
printf("Deixa resto");
else printf("N o deixa");
}else{printf("Negativo");}}
```

Bem:

```
if (n == 1) {
    printf("Unidade");
} else if (n >= 0) {
    if (n % 3)
        printf("Deixa resto");
    else
        printf("N o deixa");
} else {
    printf("Negativo");
}
```

Nomes de Variáveis Mal:

```
float n1, n2, n3, n4, aluno;
int exercices;
```

Bem:

```
float nota1, nota2;
float peso1, peso2;
float media;
int exercicios;
```

Documentação Mal:

```
#include <stdio.h>
int main() {
    float r, v;
```

```
scanf("%f", &r);  
v = (4.0*3.1425*r*r*r)/3.0;  
printf("%f", v);  
}
```

Bem:

```
#include <stdio.h>  
/* Descricao: Le o raio de uma esfera e imprime seu volume.  
   ENTRADA: raio da esfera.  
   Saida: volume da esfera.  
   Pre-condicao: raio > 0.  
   AUTOR: Joaozinho */  
int main() {  
    float r, v;  
    scanf("%f", &r);  
    v = (4.0 * 3.1415 * r * r * r) / 3.0;  
    printf("%f", v);  
}
```

5 Exercícios

1. Refaça os exercícios da aula anterior utilizando operadores lógicos para simplificar o código:
 - (a) Programa que imprime "SIM" se o número for par e maior que 10, ou ímpar e menor que 50. Caso contrário, imprime "NAO".
 - (b) Programa que ordena três números inteiros.
2. Escreva um programa que leia duas datas (ano, mês, dia) e determine qual ocorreu primeiro.
3. Troque seu código com um colega e avalie sua legibilidade (comentários, indentação, nomes, etc.).