

Introdução a Algoritmos

Raoni F. S. Teixeira

Aula 3 - Expressões Relacionais e Comandos Relacionais.

1 Valores Booleanos

1.1 Definição

Um **valor booleano** (nome derivado de George Boole, matemático inglês) é uma constante que representa o **valor de verdade** de uma sentença — podendo ser **verdadeira** ou **falsa**.

Exemplos:

- “O céu é azul.” \Rightarrow verdadeiro
- “A lua é feita de queijo.” \Rightarrow falso

1.2 Booleanos em C

Em C, valores booleanos são representados por números inteiros:

- 0 — falso
- 1 ou qualquer valor diferente de zero — verdadeiro

2 Expressões Relacionais

2.1 Definição

Expressões relacionais são aquelas que comparam duas expressões e retornam um valor booleano (verdadeiro ou falso).

```
a = 10;
b = 5;
if (a > b)
    printf("a é maior que b\n");
```

2.2 Operadores Relacionais

Os principais operadores relacionais em C são:

Operador	Significado	Exemplo
==	Igualdade	a == b
!=	Diferente	a != b
>	Maior que	a > b
<	Menor que	a < b
>=	Maior ou igual a	a >= b
<=	Menor ou igual a	a <= b

2.3 Exemplos de Expressões Relacionais

```
a == b    // verdadeiro se a e b t m o mesmo valor
a != b    // verdadeiro se a e b s o diferentes
```

```
a > b     // verdadeiro se a maior que b
a < b     // verdadeiro se a menor que b
a >= b    // verdadeiro se a maior ou igual a b
a <= b    // verdadeiro se a menor ou igual a b
```

3 Comandos Condicionais

3.1 Definição

Um comando condicional permite que um trecho do programa seja executado apenas se uma determinada condição for verdadeira.

3.2 O comando `if`

A estrutura básica do comando `if` é:

```
if (condição)
    comando;
```

ou, para múltiplos comandos:

```
if (condição) {
    comando1;
    comando2;
}
```

Os comandos dentro das chaves são executados somente se a condição for verdadeira.

3.3 Blocos de Comandos

Um **bloco de comandos** é um conjunto de instruções delimitadas por chaves `{` e `}`.

```
int main(void) {    // início do bloco
    int a;
    a = 1;
}                    // fim do bloco
```

3.4 Exemplo de Uso do `if`

```
int main(void) {
    int a;
    scanf("%d", &a);
    if (a % 2) {
        printf("O valor é ímpar.\n");
    }
}
```

4 Comando if/else

O comando if/else permite definir ações alternativas para os casos verdadeiro e falso.

```
if (condição) {  
    // comandos se a condição verdadeira  
} else {  
    // comandos se a condição falsa  
}
```

```
if (a > 0)  
    printf("Número positivo\n");  
else  
    printf("Número negativo ou zero\n");
```

4.1 Ambiguidade no uso do else

O else é sempre associado ao if mais próximo que ainda não possui um else.

```
if (cond1)  
    if (cond2)  
        comando1;  
    else  
        comando2; // associado ao if(cond2)
```

Para evitar confusão, use sempre chaves:

```
if (cond1) {  
    if (cond2)  
        comando1;  
} else {  
    comando2;  
}
```

5 Estruturas if-else-if

Quando há múltiplas alternativas, podemos usar uma sequência encadeada de if-else-if.

5.1 Exemplo 1 — Múltiplos testes independentes

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

Neste caso, todos os testes são avaliados, mesmo após um deles ser verdadeiro.

5.2 Exemplo 2 — Encadeamento if-else-if

```
int main(void) {
    int a;
    scanf("%d", &a);
    if (a == 10129)
        printf("Maria_C ndida_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");
}
```

Na construção if-else-if:

- Assim que uma condição é verdadeira, o bloco correspondente é executado;
- As demais condições não são testadas;
- O último else (sem if) serve como opção padrão.

6 Exercícios

1. Escreva um programa que lê um número inteiro e imprime:

- “SIM” se o número for par e maior que 10, ou se for ímpar e menor que 50;
 - “NAO” caso contrário.
2. Escreva um programa que leia três números inteiros e os imprima em ordem crescente.