

Algoritmos com Seleção

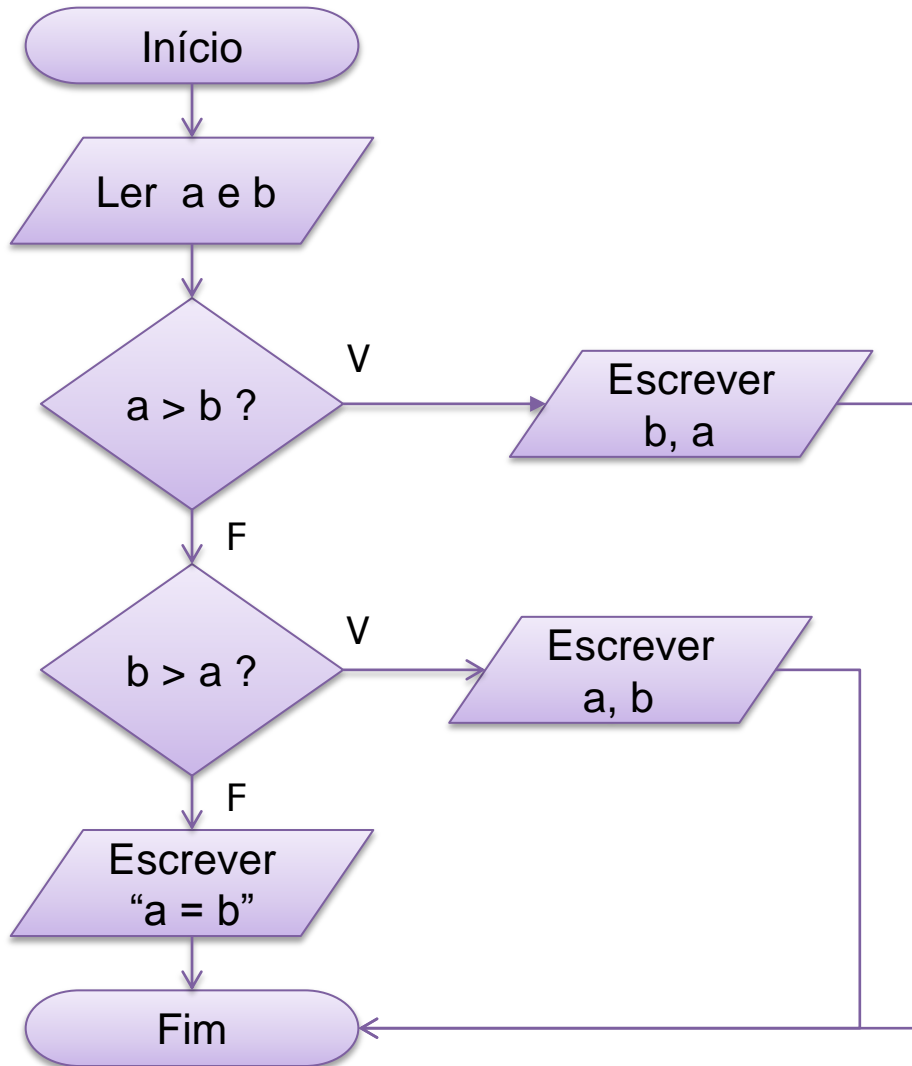
Continuação

**Seleções aninhadas, tabelas verdade e
precedência de operadores**

Considere o enunciado

- Ler dois valores e mostrá-los na saída em ordem crescente ou informar caso os valores forem iguais
 - **Entradas:** dois valores em ordem qualquer
 - **Saída:** os dois valores lidos apresentados em ordem crescente
 - **Processamento:** preparação da apresentação dos dois valores

Algoritmo – Ordem Crescente



```
1. Principal ()
2. Início
3.   Ler a, b;
4.   Se (a > b)
5.     Então
6.       Escrever b, a
7.     Senão
8.       Se (b > a)
9.         Então
10.          Escrever a, b;
11.        Senão
12.          Escrever "a eh igual a b"
13.      Fim
14.  Fim
15. Fim
```

Programa – Ordem Crescente

```
1  #include <stdio.h>
2
3  int main ()
4  {
5      int a, b;
6      printf ("\nO primeiro nro.: ");
7      scanf ("%d", &a);
8      printf ("\nO segundo nro.: ");
9      scanf ("%d", &b);
10     if (a > b)
11         printf ("\nEm ordem crescente: %d e %d ", b, a);
12     else
13     {
14         if (b > a)
15             printf ("\nEm ordem crescente: %d e %d ", a, b);
16         else
17             printf ("\n%d e %d são iguais", a, b);
18     }
19     return 0;
20 }
```

if mais externo

if aninhado

Comandos de Seleção Encadeados ou Aninhados

1. Se (condição1)

2. Então

3. Se (condição2)

4. Então

5. comando1;

6. Senão

7. comando2;

8. comando3;

9. Fim

10. Senão

11. Se (condição3)

12. Então

13. comando4;

14. Fim

15. Fim

*Um comando de seleção aninhado dentro do **então***

*Outro aninhado dentro do **senão***

Agora indentação passa a ser fundamental!

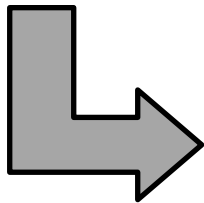
Comandos de Seleção

Encadeados ou Aninhados

- Podem aparecer quantos comandos aninhados forem necessários

Exemplo em C:

```
if (a > 10)
    if (x != 7)
        if (s == 1)
            printf ("Tudo ok!");
```



Nesse caso é análogo a:

```
if (a > 10 && x != 7 && s == 1)
    printf ("Tudo ok!");
```

Comandos de Seleção Encadeados ou Aninhados

Qual **else** pertence a qual **if**?

```
if (condição 1)
  if (condição 2)
    if (condição 3)
      comando;
    else comando;
  else comando;
```

E nesse caso?

```
if (condição 1)
  if (condição 2)
  {
    if (condição 3)
      comando;
  }
  else comando;
else comando;
```

ATENÇÃO: ao aninhar comandos de seleção, usar chaves, se necessário, para que os **elses** correspondam aos **ifs** corretos.

Proposta de Estudo

- Fazer um algoritmo em pseudo-código para somar ou multiplicar dois valores inteiros, dependendo de um terceiro valor fornecido pelo usuário
- Informar também caso a operação digitada seja inválida
 - **Entradas:**
 - Um valor para a operação (1 para soma ou 2 multiplicação)
 - Mais dois valores inteiros
 - **Saída:**
 - Soma ou multiplicação dos dois valores inteiros lidos ou mensagem de erro
 - **Processamento:**
 - Realização do cálculo solicitado

Algoritmo – Soma ou multiplica dois valores inteiros lidos

```
1.  Principal ()
2.  Início
3.      Ler codigo, valor1, valor2;
4.      Se (codigo = 1)
5.          Então
6.              resultado = valor1 + valor2;
7.              Imprimir resultado;
8.      Senão Se (codigo = 2)
9.          Então
10.             resultado = valor1 * valor2;
11.             Imprimir resultado;
12.         Senão
13.             Escrever "Código fornecido inválido";
14.         Fim
15.     Fim
16. Fim
```

Tabelas Verdade

Tabela verdade do operador **e** lógico (&&)

Seja **a = 10** e **b = 5**

Tabela verdade

Operando 1 Operando 2

(a == 10 && b == 5)

(a > 8 && b > 10)

(a != 10 && b == 5)

(a < 10 && b > 5)

Verdade Operador && (e)		
Operando 1	Operando 2	Resultado
V	V	V
V	F	F
F	V	F
F	F	F

Tabelas Verdade

Tabela verdade do operador **ou** lógico (**||**)

Seja **a = 10** e **b = 5**

Tabela verdade

Operando 1 Operando 2

(**a == 10** **||** **b == 5**)

(**a > 8** **||** **b > 10**)

(**a != 10** **||** **b == 5**)

(**a < 10** **||** **b > 5**)

Verdade Operador && (e)		
Operando 1	Operando 2	Resultado
V	V	V
V	F	V
F	V	V
F	F	F

Tabelas Verdade

Tabela verdade do operador **negação (!)**

Seja a = 10

Operando

`(! (a == 10))`

A negação de uma verdade é uma falsidade


`(! (a != 10))`

A negação de uma falsidade é uma verdade

Tabela verdade

Operador ! (not)	
Operando	Resultado
V	F
F	V

Precedência de Operadores Lógicos em C

Operador	Prioridade
< <= > >=	Maior
== !=	
&&	
? :	Menor

Parênteses

permitem definir a precedência de operadores na resolução das expressões lógicas!