

Algoritmos e Estruturas de Dados I

Aula 4 -Estrutura Condicional

Prof. Felipe Lara



Estrutura Condicional

Existem três estruturas básicas em programação:

- Estrutura Sequencial
- **Estrutura Condicional**
- Estrutura Repetitiva

Proposição

Proposição é um enunciado verbal (expressão), ao qual deve ser atribuído, sem ambiguidade, um valor lógico verdadeiro (V) ou falso (F).

- Exemplos:
 - A copa do mundo 2014 foi no Brasil.
 - Todo ser humano é imortal.
 - $2 + 5 < 5$
 - $1 + 1 \geq 2$

Expressões lógicas

Expressões lógicas são expressões compostas de relações que sempre retornam um valor lógico (verdadeiro ou falso).

- Exemplos:
 - $5 + 3 < 10 \Rightarrow V$
 - $1 + 10 <> 11 \Rightarrow F$
 - $5 + 0 = 5 \Rightarrow V$
- Expressões lógicas podem ser unidas usando operadores lógicos.

Operadores lógicos

Atuam sobre expressões lógicas retornando sempre valores lógicos (verdadeiro ou falso).

Operador	Tipo	Resultado
E	Binário	Retorna verdadeiro se ambas expressões forem verdadeiras
OU	Binário	Retorna verdadeiro se pelo menos uma das expressões for verdadeira
NÃO	Unário	Inverte o estado, retorna verdade caso a expressão seja falsa e vice-versa

Operador lógico E (AND)

- Operador **E (AND)**: operador de conjunção
 - Terá valor **V** quando as ambas proposições forem **V**.
 - Basta uma proposição ser **F** para o resultado ser **F**.

- Exemplos:

- $(2 + 5 > 4)$ **E** $(1 + 3 \leq 2)$ \Rightarrow **F**
- $(2 + 5 > 4)$ **E** $(1 + 3 \geq 2)$ \Rightarrow **V**
- $(2 + 5 < 4)$ **E** $(1 + 3 \leq 2)$ \Rightarrow **F**

A	B	A e B
TRUE	TRUE	TRUE
TRUE	FALSE	FALSE
FALSE	TRUE	FALSE
FALSE	FALSE	FALSE

Operador lógico OU (OR)

- Operador **OU (OR)**: operador de disjunção
 - Terá valor **V** quando as uma das proposições forem **V**.
- Exemplos:
 - $(2 + 5 > 4)$ **OU** $(1 + 3 \leq 2)$ \Rightarrow **V**
 - $(2 + 5 > 4)$ **OU** $(1 + 3 \geq 2)$ \Rightarrow **V**
 - $(2 + 5 < 4)$ **OU** $(1 + 3 \leq 2)$ \Rightarrow **F**

A	B	A ou B
TRUE	TRUE	TRUE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE
FALSE	FALSE	FALSE

Operador lógico NÃO (NOT)

- Operador **NÃO (NOT)**: operador de negação
 - Inverte o valor lógico

- Exemplos:

- Não $(2 + 5 > 4)$ \Rightarrow **F**
- Não $(1 + 3 \leq 2)$ \Rightarrow **V**

A	não A
TRUE	FALSE
FALSE	TRUE

Tabela Verdade

A	B	$A \text{ E } B$	$A \text{ OU } B$	$\text{NÃO } (A)$
F	F			
F	V			
V	F			
V	V			

Tabela Verdade

A	B	$A \text{ E } B$	$A \text{ OU } B$	$\text{NÃO } (A)$
F	F	F	F	V
F	V	F	V	V
V	F	F	V	F
V	V	V	V	F

Operadores lógicos em C

Operador	linguagem C	Precedência
NÃO	!	1
E	&&	2
OU		3

- Note que o operador OU (||) é o que possui a menor precedência.

Operadores Relacionais

Operador	Exemplo	Comentário
<code>==</code>	<code>X == Y</code>	O conteúdo de X é igual ao conteúdo de Y
<code>!=</code>	<code>X != Y</code>	O conteúdo de X é diferente do conteúdo de Y
<code><=</code>	<code>X <= Y</code>	O conteúdo de X é menor ou igual ao conteúdo de Y
<code>>=</code>	<code>X >= Y</code>	O conteúdo de X é maior ou igual ao conteúdo de Y
<code><</code>	<code>X < Y</code>	O conteúdo de X é menor que o conteúdo de Y
<code>></code>	<code>X > Y</code>	O conteúdo de X é maior que o conteúdo de Y

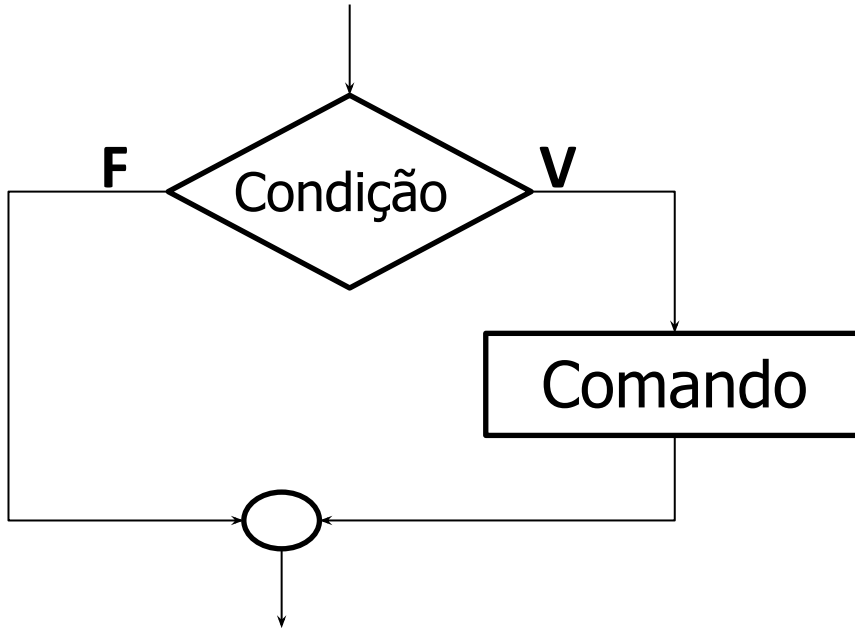
Operadores lógicos em C

- Em C, inteiros são usados para tipos lógicos.
 - Nesse caso, **0 é considerado FALSO** e **1 VERDADEIRO**.
- Podemos usar o tipo **bool** importando "stdbool.h". Nesse caso, o valor será convertido em 0 ou 1.

```
#include <stdbool.h>
#include <stdio.h>
int main(){
    bool x = true;
    bool y = false;
    printf("%d", x); //imprime 1
    printf("%d", y); //imprime 0
    return 0;
}
```

Estrutura Condicional Simples

Fluxograma

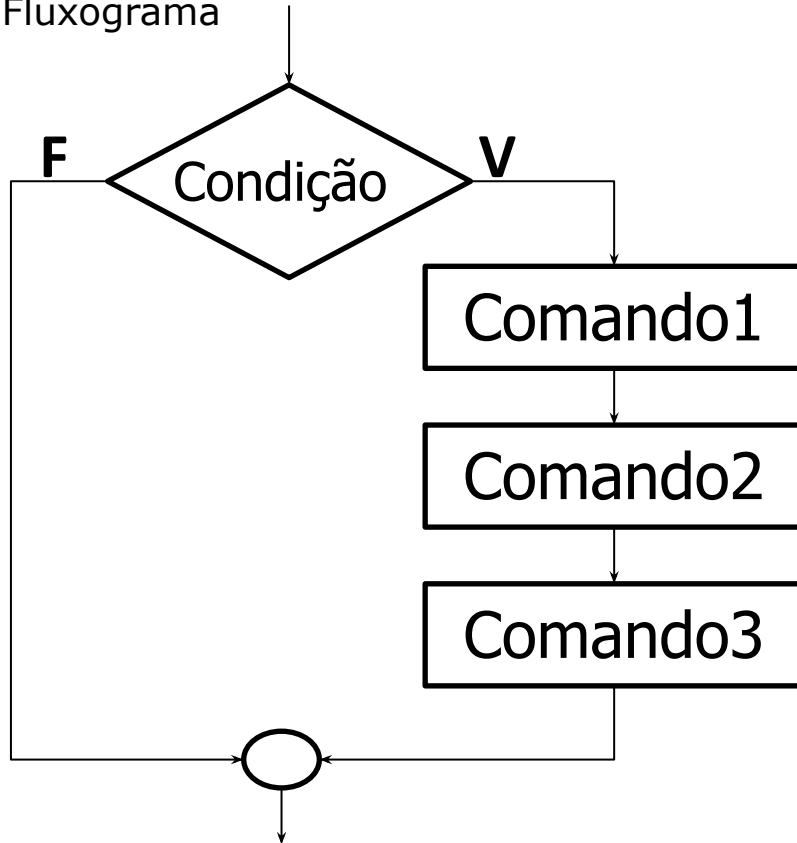


Pseudocódigo

SE *Condição* **ENTÃO**
Comando

Estrutura Condicional Simples

Fluxograma



Pseudocódigo

SE *Condição* **ENTÃO**

INÍCIO

Comando1

Comando2

Comando3

FIM

Estrutura Condicional Simples em C

```
if (Condição)  
    Comando1
```

```
if (Condição) {  
    Comando1  
    Comando2  
    Comando3  
}
```


Exemplo

- **Faça um algoritmo que, após um número ser lido, informa se você ganhou ou perdeu. Você ganha se o número for 5.**

ALGORITMO

DECLARE x NUMÉRICO

LEIA x

SE x = 5 ENTÃO

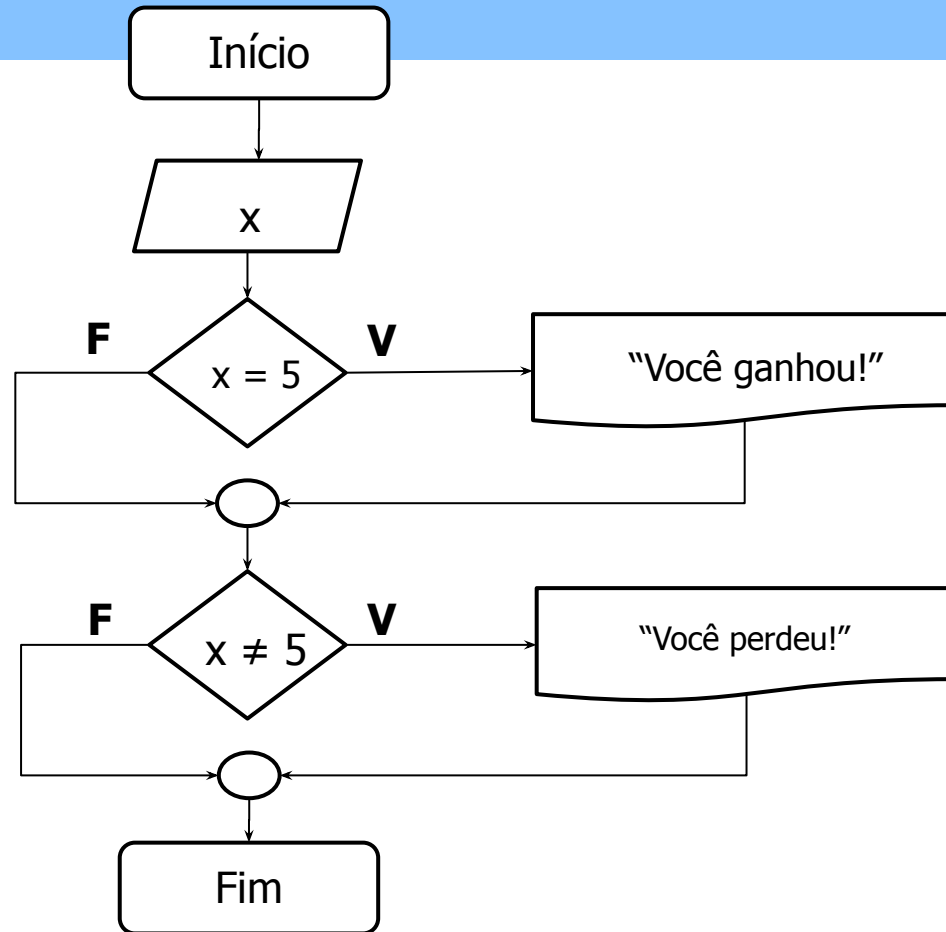
 ESCREVA "Você ganhou!"

SE x ≠ 5 ENTÃO

 ESCREVA "Você perdeu!"

FIM_ALGORITMO

Exemplo



Exemplo

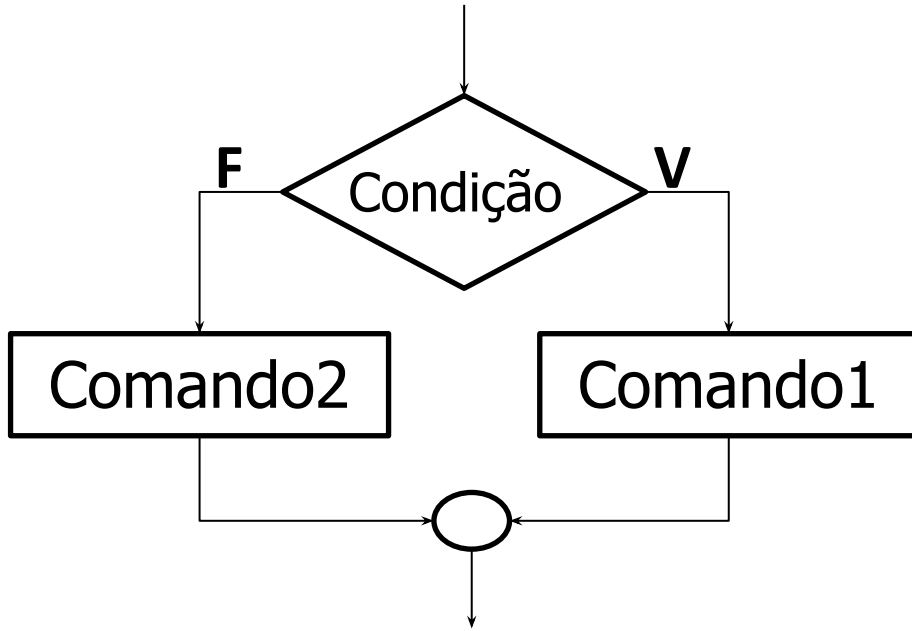
- **Faça um algoritmo que, após um número ser sorteado de 1 e 10, informa se você ganhou ou perdeu. Você ganha se o número for 5.**

```
#include <stdio.h>

int main()
{
    int x;
    printf("Digite o numero sorteado:");
    scanf("%d", &x);
    if(x == 5) {
        printf("Você ganhou!");
    }
    if(x != 5) {
        printf("Você perdeu!");
    }
    return 0;
}
```

Estrutura Condicional Composta

Fluxograma



Pseudocódigo

SE *Condição* **ENTÃO**

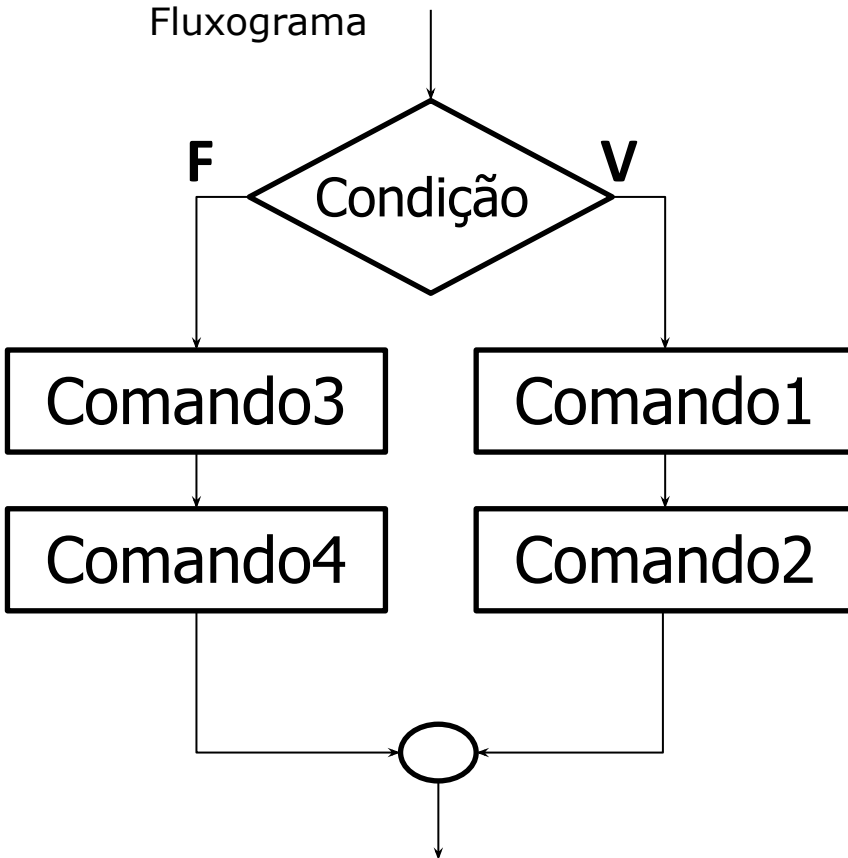
Comando1

SENÃO

Comando2

Estrutura Condicional Composta

Fluxograma



Pseudocódigo

SE *Condição* **ENTÃO**

INÍCIO

Comando1

Comando2

FIM

SENÃO

INÍCIO

Comando3

Comando4

FIM

Estrutura Condicional Composta em C

```
if (Condição)
```

```
    Comando1
```

```
else
```

```
    Comando2
```

```
if (Condição) {
```

```
    Comando1
```

```
    Comando2
```

```
} else {
```

```
    Comando3
```

```
    Comando4
```

```
}
```

Exemplo

- **Faça um algoritmo que, após um número ser lido, informa se você ganhou ou perdeu. Você ganha se o número for 5.**

ALGORITMO

DECLARE x NUMÉRICO

LEIA x

SE x = 5 ENTÃO

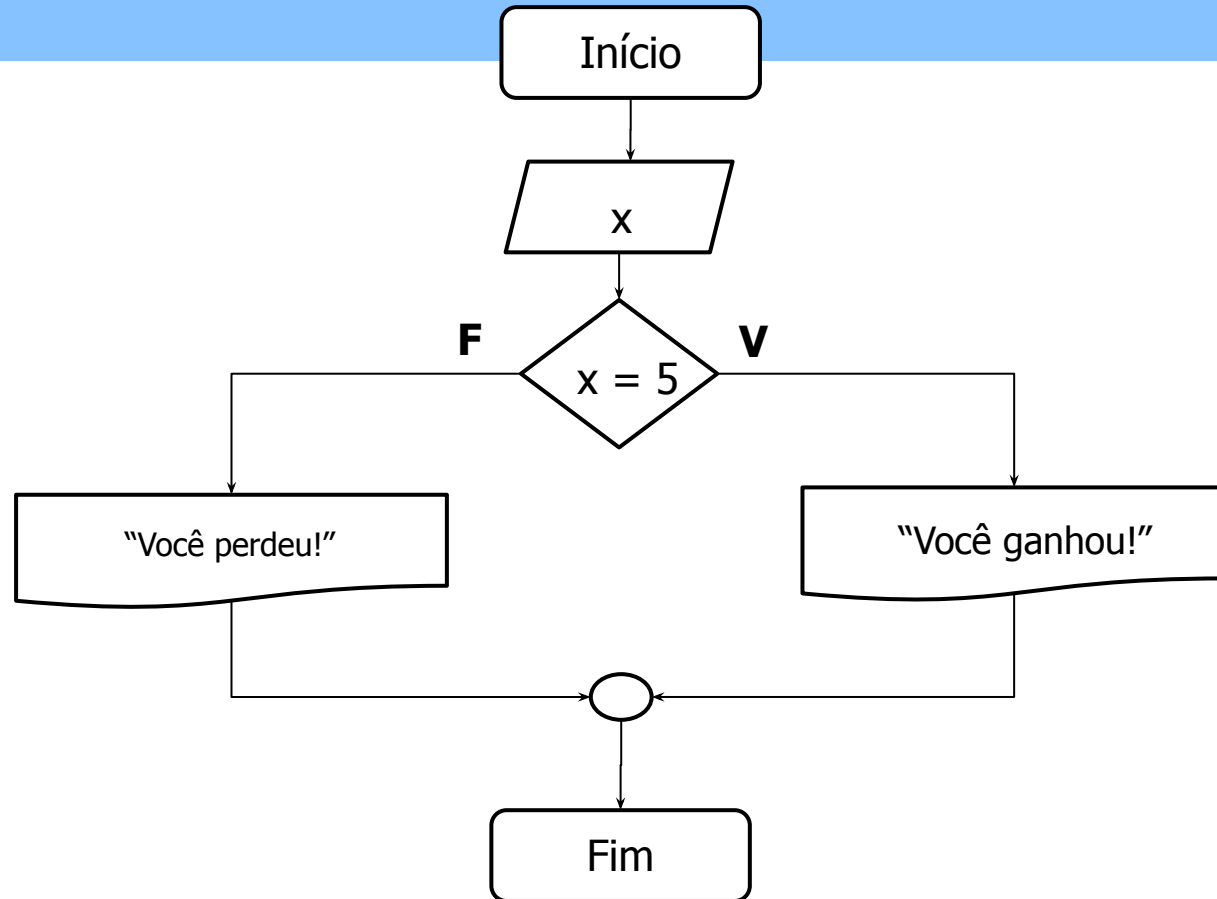
 ESCREVA "Você ganhou!"

SENÃO

 ESCREVA "Você perdeu!"

FIM_ALGORITMO

Exemplo



Exemplo

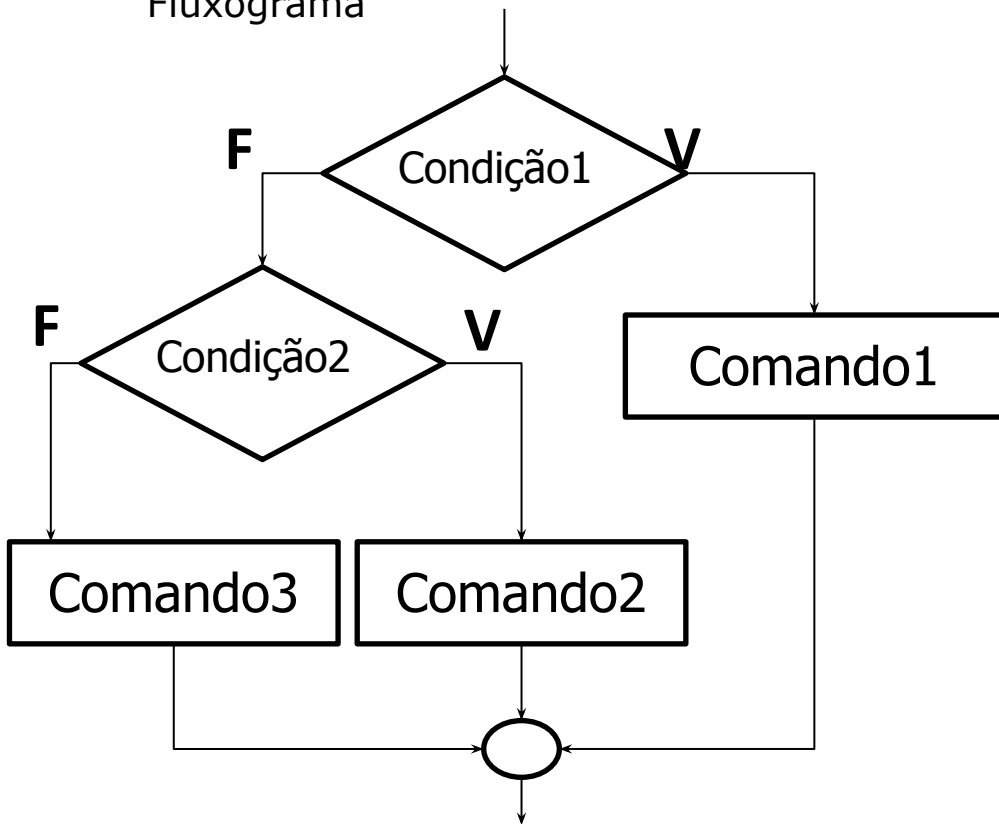
- **Faça um algoritmo que, após um número ser lido, informa se você ganhou ou perdeu. Você ganha se o número for 5.**

```
#include <stdio.h>

int main()
{
    int x;
    printf("Digite o numero sorteado:");
    scanf("%d", &x);
    if(x == 5) {
        printf("Voce ganhou!");
    } else {
        printf("Voce perdeu!");
    }
    return 0;
}
```

Estrutura Condicional Aninhada

Fluxograma



Pseudocódigo

```
SE Condição1 ENTÃO  
    Comando1  
SENÃO  
    SE Condição2 ENTÃO  
        Comando2  
    SENÃO  
        Comando3
```

Estrutura Condicional Aninhada

SE *Condição1* **ENTÃO**

Comando1

SENÃO

SE *Condição2* **ENTÃO**

Comando2

SENÃO

Comando3

SE *Condição1* **ENTÃO**

Comando1

SENÃO **SE** *Condição2*
ENTÃO

Comando2

SENÃO

Comando3

Estrutura Condicional Aninhada em C

```
if (Condição1) {  
    Comando1  
} else {  
    if (Condição2) {  
        Comando2  
    } else {  
        Comando3  
    }  
}
```

```
if (Condição1) {  
    Comando1  
} else if (Condição2) {  
    Comando2  
} else {  
    Comando3  
}
```

Exemplo

- **Faça um algoritmo que, após um número ser lido, informa se você ganhou ou perdeu. Você ganha se o número for 5 ou 7.**

ALGORITMO

DECLARE x NUMÉRICO

LEIA x

SE x = 5 ENTÃO

 ESCREVA "Você ganhou!"

SENÃO

 SE x = 7 ENTÃO

 ESCREVA "Você ganhou!"

 SENÃO

 ESCREVA "Você perdeu!"

FIM_ALGORITMO

Exemplo

- **Faça um algoritmo que, após um número ser lido, informa se você ganhou ou perdeu. Você ganha se o número for 5 ou 7.**

ALGORITMO

DECLARE x NUMÉRICO

LEIA x

SE x = 5 ENTÃO

 ESCREVA "Você ganhou!"

SENÃO SE x = 7 ENTÃO

 ESCREVA "Você ganhou!"

SENÃO

 ESCREVA "Você perdeu!"

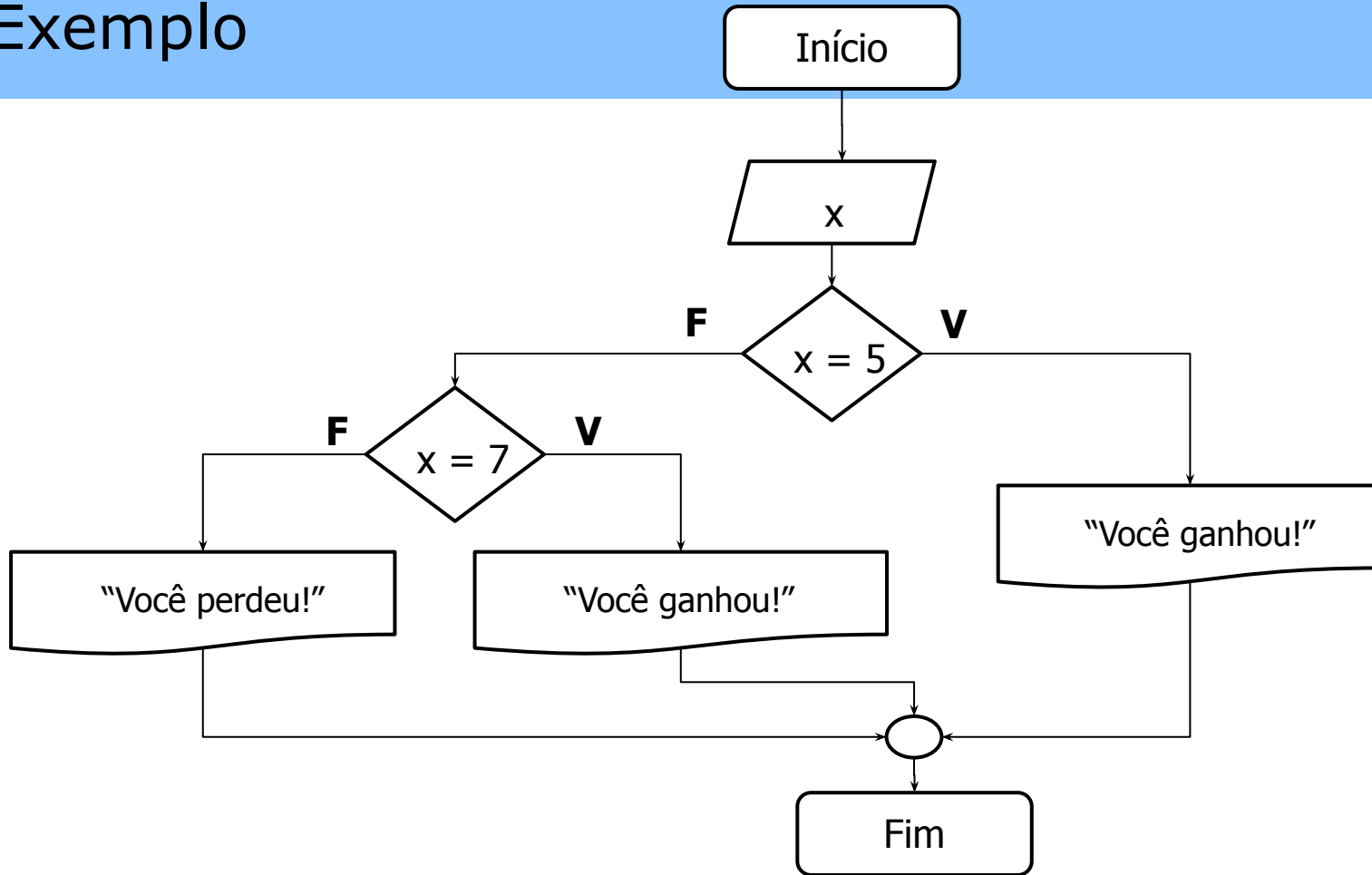
FIM_ALGORITMO

Exemplo

- **Faça um algoritmo que, após um número ser lido, informa se você ganhou ou perdeu. Você ganha se o número for 5 ou 7.**

```
int main()
{
    int x;
    printf("Digite o numero sorteado:");
    scanf("%d", &x);
    if(x == 5) {
        printf("Voce ganhou!");
    }
    else if(x == 7) {
        printf("Voce ganhou!");
    }
    else {
        printf("Voce perdeu!");
    }
    return 0;
}
```

Exemplo



Exemplo

- **Faça um algoritmo que, após um número ser lido, informa se você ganhou ou perdeu. Você ganha se o número for 5 ou 7.**

ALGORITMO

DECLARE x NUMÉRICO

LEIA x

SE x = 5 OU x = 7 ENTÃO

 ESCREVA "Você ganhou!"

SENÃO

 ESCREVA "Você perdeu!"

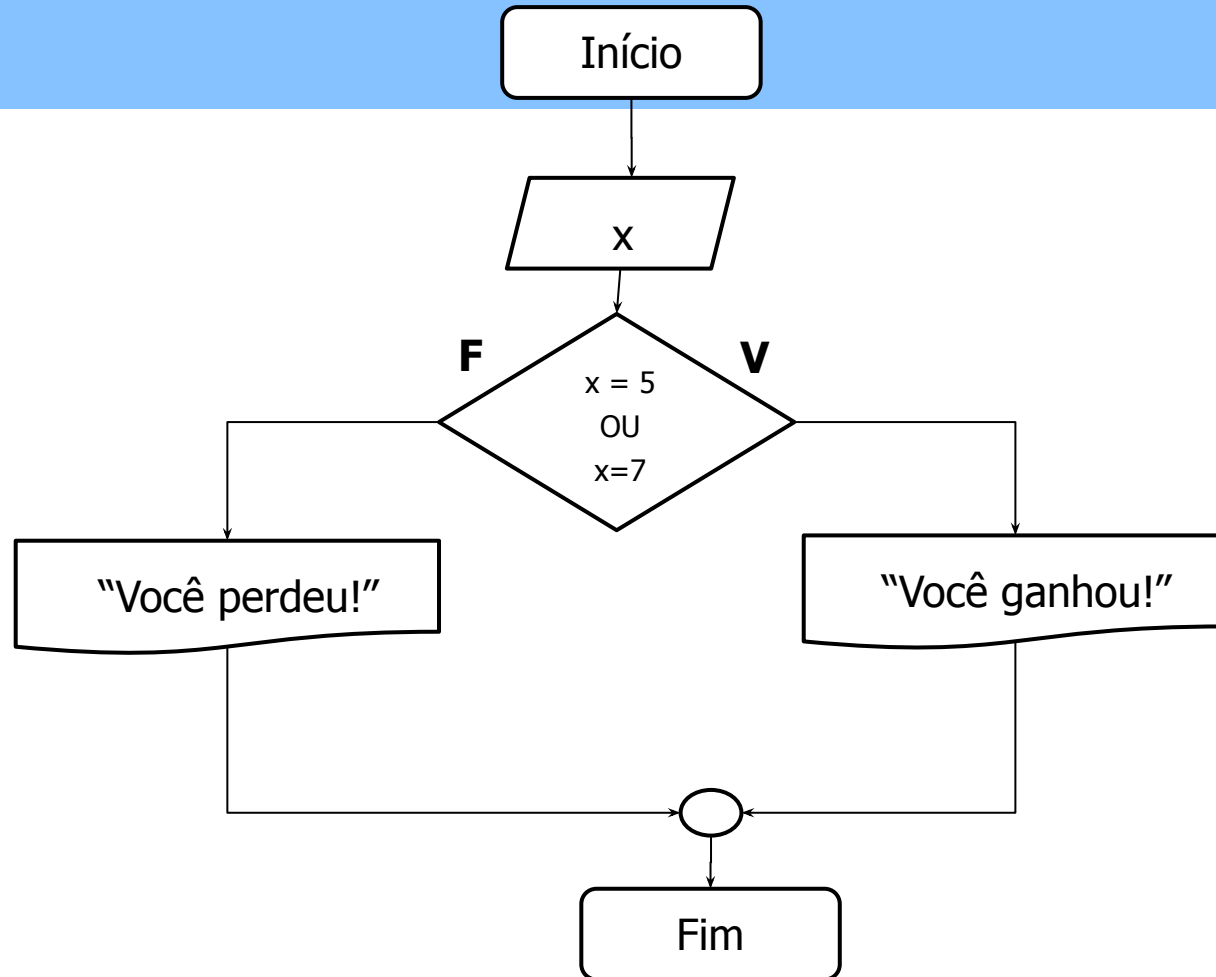
FIM_ALGORITMO

Exemplo

- **Faça um algoritmo que, após um número ser lido, informa se você ganhou ou perdeu. Você ganha se o número for 5 ou 7.**

```
int main()
{
    int x;
    printf("Digite o numero sorteado:");
    scanf("%d", &x);
    if(x == 5 || x == 7) {
        printf("Voce ganhou!");
    }
    else {
        printf("Voce perdeu!");
    }
    return 0;
}
```

Exemplo



Exemplo

- **Faça um algoritmo que, após um número ser sorteado de 1 e 10, informa se você ganhou ou perdeu. Você ganha se o número for 5 ou 7.**

```
#include <stdio.h>

int main()
{
    int x;
    printf("Digite o numero sorteado:");
    scanf("%d", &x);
    if(x != 5 && x != 7) {
        printf("Voce perdeu!");
    } else {
        printf("Voce ganhou!");
    }
    return 0;
}
```

Exemplo

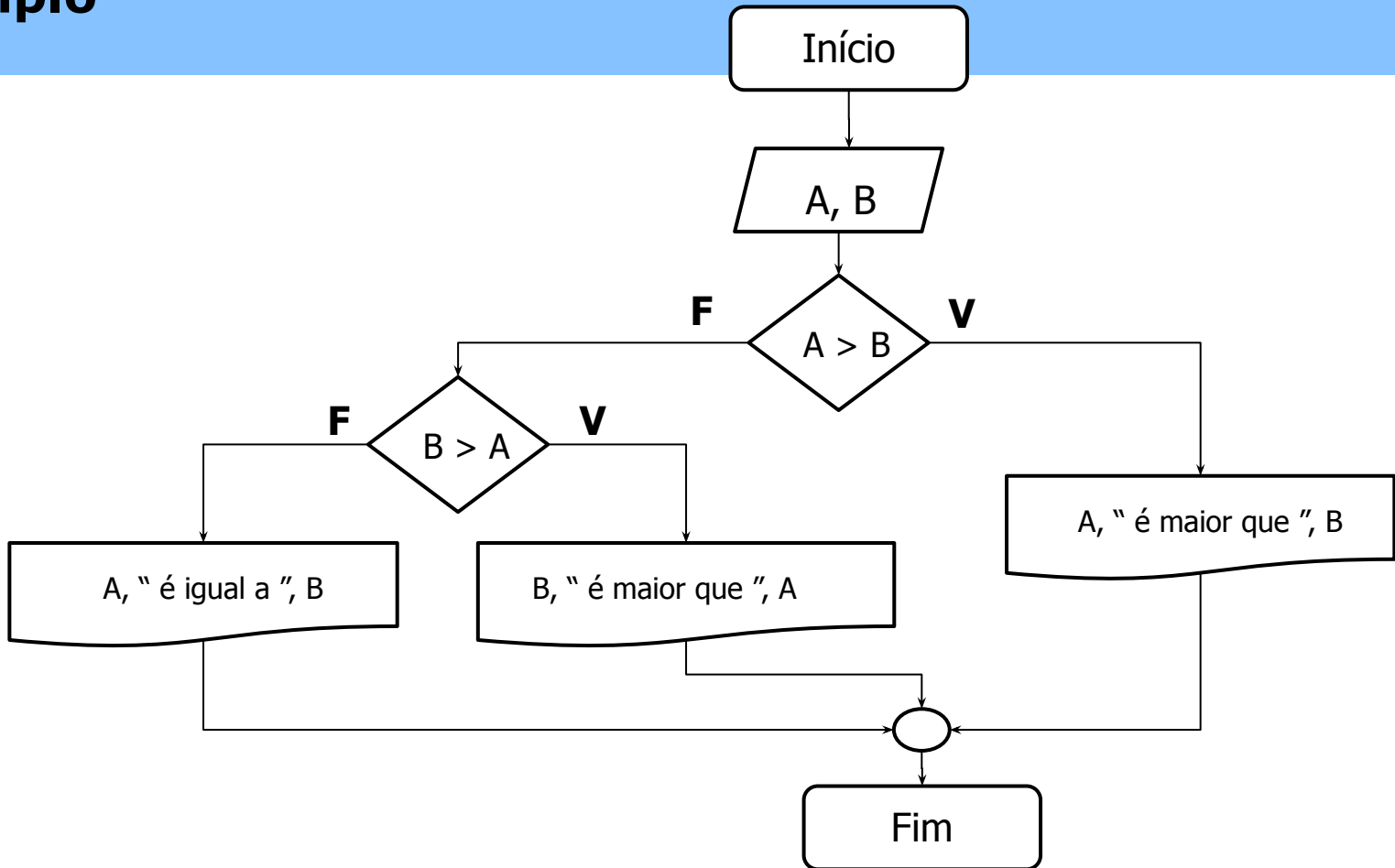
Faça um algoritmo que receba dois números inteiros e imprima o maior deles.

Exemplo

Faça um algoritmo que receba dois números inteiros e imprima o maior deles.

```
int main()
{
    int A, B;
    puts("Digite o primeiro numero:");
    scanf("%d", &A);
    puts("Digite o segundo numero:");
    scanf("%d", &B);
    if(A > B) {
        printf("%d é maior que %d", A, B);
    }
    else if(B > A) {
        printf("%d é maior que %d", B, A);
    }
    else {
        printf("%d é igual a %d", A, B);
    }
    return 0;
}
```

Exemplo



Exemplo

Faça um programa que receba um número inteiro e verifique se é par ou ímpar.

Exemplo

Faça um programa que receba um número inteiro e verifique se é par ou ímpar.

```
#include <stdio.h>

int main()
{
    int num;
    puts("Digite o numero:");
    scanf("%d", &num);
    if(num % 2 == 0) {
        printf("%d e par", num);
    }

    else {
        printf("%d e impar", num);
    }
    return 0;
}
```

Estrutura Condicional Switch-Case

- Ao invés de usar múltiplos if-else, temos a alternativa do switch-case.
- A cláusula switch seleciona um dos múltiplos blocos a ser executado.
- A cláusula switch é avaliada uma única vez
- O valor da expressão é comparado com os valores de cada case
- Se a comparação resultar verdadeiro, o bloco é executado
- A cláusula break interrompe a execução do switch naquele momento
- A cláusula default é opcional e especifica código a ser executado caso nenhuma comparação tenha correspondido
- Switch só pode ser usado com expressões representadas por inteiros

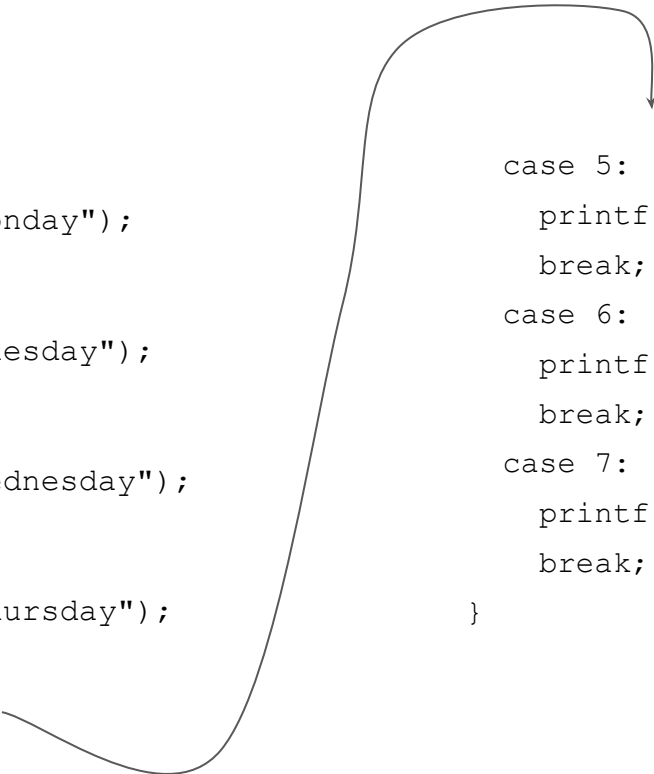
```
switch(expressão) {  
    case x:  
        // code block  
        break;  
    case y:  
        // code block  
        break;  
    default:  
        // code block  
}
```

Estrutura Condicional Switch-Case

Qual o valor impresso a seguir?

```
int day = 3;

switch (day) {
    case 1:
        printf("Monday");
        break;
    case 2:
        printf("Tuesday");
        break;
    case 3:
        printf("Wednesday");
        break;
    case 4:
        printf("Thursday");
        break;
    case 5:
        printf("Friday");
        break;
    case 6:
        printf("Saturday");
        break;
    case 7:
        printf("Sunday");
        break;
}
```



Estrutura Condicional Switch-Case

Qual o valor impresso a seguir?

```
int day = 3;

switch (day) {
    case 1:
        printf("Monday");
        break;
    case 2:
        printf("Tuesday");
        break;
    case 3:
        printf("Wednesday");
        break;
    case 4:
        printf("Thursday");
        break;
    case 5:
        printf("Friday");
        break;
    case 6:
        printf("Saturday");
        break;
    case 7:
        printf("Sunday");
        break;
}
```

Wednesday

Operador Ternário Condicional (?:)

- É uma alternativa mais compacta ao if-else.

```
variavel = condição ? codigoCasoVerdadeiro: codigoCasoFalso;
```

- No código acima, "variavel" recebe "codigoCasoVerdadeiro" se "condição" for verdadeira.
- Caso contrário, "variavel" recebe "codigoCasoFalso".

Operador Ternário Condicional (?:)

- Exemplo:

```
int x, b = 10;  
if (b < 20)  
    x = 100;  
else  
    x = 200;
```

- É equivalente a:

```
int x, b = 10;  
x = b < 20 ? 100 : 200;
```

Operador Ternário Condicional (?:)

- Exemplo:

```
int time = 20;
if (time < 18) {
    printf("Good day.");
} else {
    printf("Good evening.");
}
```

- É equivalente a:

```
int time = 20;
(time < 18) ? printf("Good day.") : printf("Good evening.");
```

Dúvidas?