

# Algoritmos e Programação

Aula 02

Desenvolvimento de Algoritmos

Representação de Algoritmos

Fluxogramas

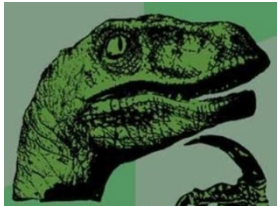


**Universidade Federal do Pampa**

# Desenvolvimento de Algoritmos

# Desenvolvimento de Algoritmos

- E se utilizarmos um **Algoritmo** para fazer **Algoritmos**?



# Desenvolvimento de Algoritmos

- E se utilizarmos um **Algoritmo para fazer Algoritmos?**
  - 4 passos básicos
  - Pode ser adaptado para problemas gerais

Algoritmo: **FAZER\_ALGORITMO**

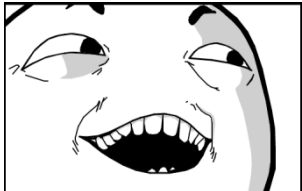
Passo 1: **Entender o Problema**

Passo 2: **Elaborar o Algoritmo**

Passo 3: **Implementar o Algoritmo**

Passo 4: **Verificar os Resultados**

*Fim\_Algoritmo*



# Desenvolvimento: Passo 1

## ENTENDER O PROBLEMA

- Definir o problema
- Qual o **OBJETIVO** fundamental?
- O que é preciso **FAZER**? O que é preciso encontrar?
- Quais são os dados de **ENTRADA**?
- Quais são os dados de **SAÍDA**?
- Dicas
  - Examinar um caso mais simples do problema (entendimento)
  - Listar os casos especiais/exceções (do problema e do usuário)
- Observação
  - **Entender o problema** é uma parte **crucial**. Influi diretamente no resultado! **Se você começar mal, terminará como?** Será que vai terminar?

# Desenvolvimento: Passo 2

## ELABORAR O ALGORITMO

- IDENTIFICAR as ETAPAS na resolução do problema
- DETALHAR CADA UMA delas até encontrar uma sequência de operações básicas (abordagem topdown)
- Dicas
  - Fazer tabelas/diagramas/equações para formalizar as ideias
  - Não esquecer casos especiais (tratamento diferente)
- Observações
  - A maior parte do tempo é ‘gasta’ nesta etapa. Quanto melhor for o resultado desta etapa, menos tempo será perdido nos próximos passos!
  - Projeto ruim = Implementação Prejudicada!

# Desenvolvimento: Passo 3

## IMPLEMENTAR O ALGORITMO

- **ESCREVER** o algoritmo na linguagem escolhida
  - A linguagem deve ser adequada ao problema
- **MODULARIZAR** o programa
  - Utilização de funções
- Favorecer a **LEGIBILIDADE** do código escrito
- Dica
  - **USAR IDENTIFICAÇÃO!!** (jamais esqueça disto)
  - Usar comentários
- Observação
  - Use **SEMPRE: PASSOS SIMPLES**

# Desenvolvimento: Passo 4

## VERIFICAR OS RESULTADOS

- Interpretar a solução
  - Verificar sua adequação ao problema original
- Testar o algoritmo para diversos casos (exaustivamente, inclua os casos especiais/exceções)
- Dica
  - NÃO ESQUEÇA DOS CASOS ESPECIAIS/EXCEÇÕES!
- Observação
  - Verificação é a parte mais delicada
    - Após esta etapa o software pode parar nas mãos do cliente...





# Desenvolvimento de Algoritmos

- São apenas **4** passos
  - Mas não é simples...

Algoritmo: **FAZER\_ALGORITMO**

Passo 1: **Entender o Problema**

Passo 2: **Elaborar o Algoritmo**

Passo 3: **Implementar o Algoritmo**

Passo 4: **Verificar os Resultados**

*Fim\_Algoritmo*

- **PRESTE MUITA ATENÇÃO**
  - Estes passos podem ser adequados para qualquer área
  - **Nesta disciplina**
    - **Problemas simples**
  - **No mundo real**
    - **Problemas com dimensões MUITO maiores!**
    - Na computação
      - Equipes distintas para cada etapa.

# Representação de Algoritmos usando Fluxogramas

# aula passada...

## Representações de Algoritmos

- **NARRATIVA**

- Representação puramente textual
- Usa linguagem natural (no nosso caso, pt-br)
- Problema da ambiguidade

AULA PASSADA

- **FLUXOGRAMA**

- Representação gráfica
- Mais legível para humanos

HOJE

- **PSEUDOCÓDIGO**

- Representação textual
- Mais próxima do computador
- Linguagem de programação em pt-br

FUTURO

- **LINGUAGEM DE PROGRAMAÇÃO**

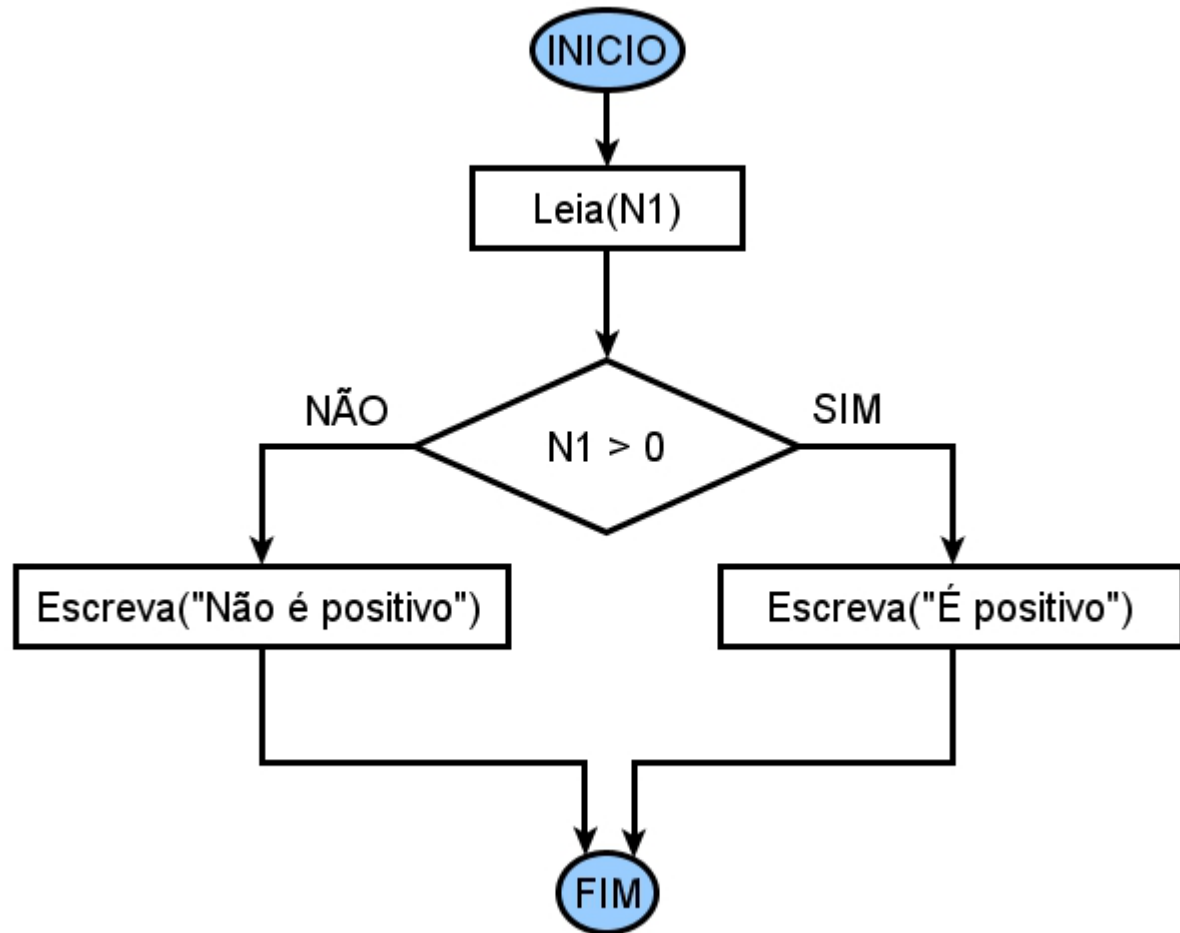
- Adequada ao computador
- Adequada a programadores

FUTURO

# Fluxogramas

- Representam algoritmos de forma **gráfica**

Veremos em detalhes →



# Fluxogramas

- Representam algoritmos de forma **gráfica**
- Quais as vantagens?
  - Facilita a **compreensão**
  - Útil para representação de algoritmos em alto nível de abstração
- Quais as desvantagens?
  - É necessário **aprender** a simbologia dos fluxogramas
  - **Difícil** de **corrigir**
  - Pode ser **muito extenso** para algoritmos mais complexos

# Fluxogramas: Símbolos usados



**Ínício** ou **Fim** do Fluxograma

Indica onde **começa** e onde **termina**)



Operação de **Entrada de Dados**

(Usada para **leitura** de dados)

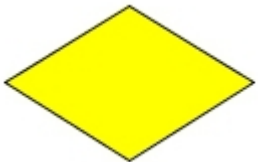


Não será usado.  
Usaremos Atribuição



Operação de **Atribuição**

(usada para **atribuir** um **valor** ou realizar uma **operação**)



Operação de **Decisão**

(usada para fazer **testes**)



Operação de **Saída de Dados**

(usada para **escrita** de dados)



Não será usado.  
Usaremos Atribuição

# Fluxogramas: Operações

Leia(X)

Recebe um valor do usuário e armazena na variável X.

Escreva(X)

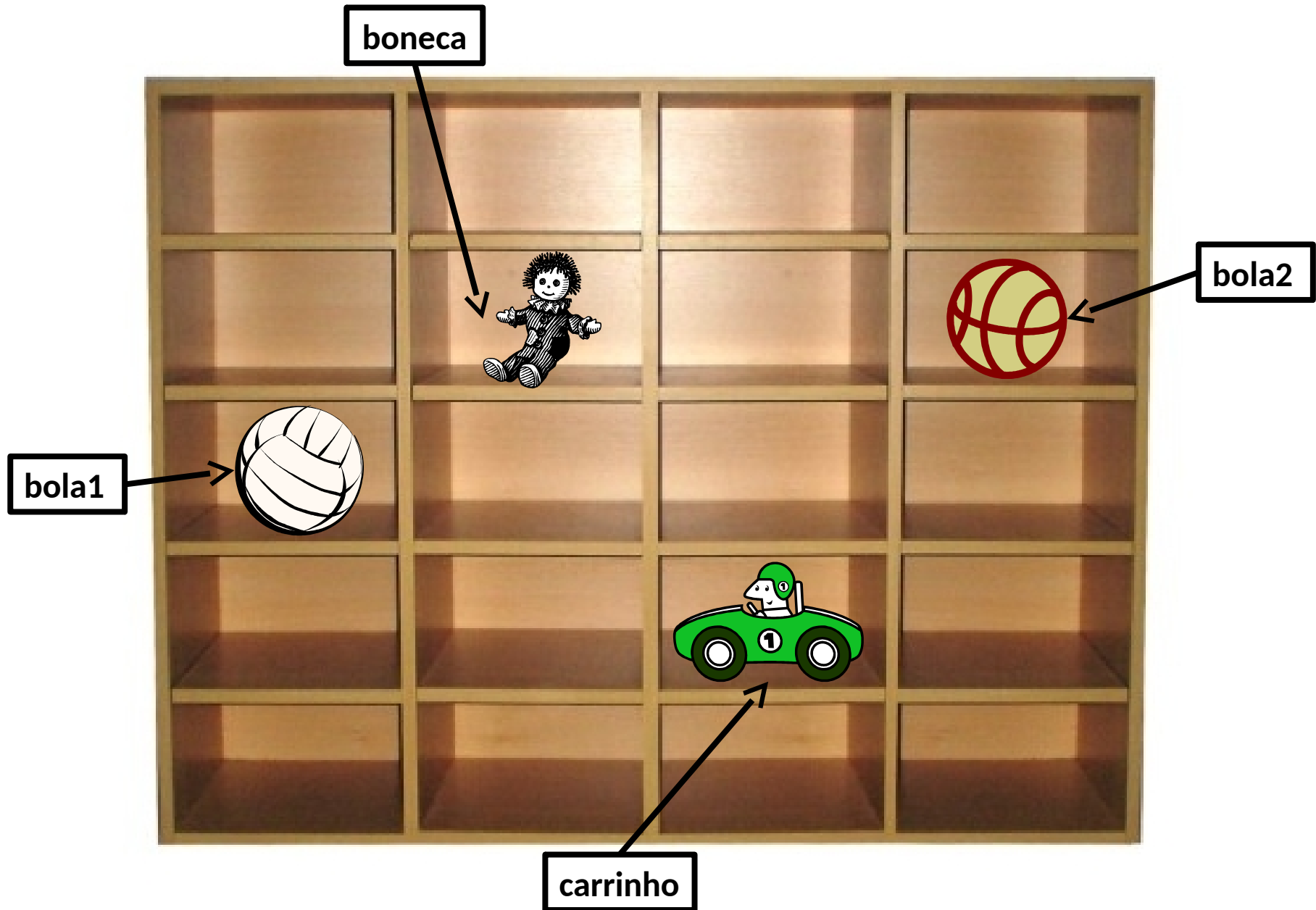
Imprime o valor armazenado na variável X.

$X = \text{Expressão}$

Armazena o resultado da Expressão na variável X.

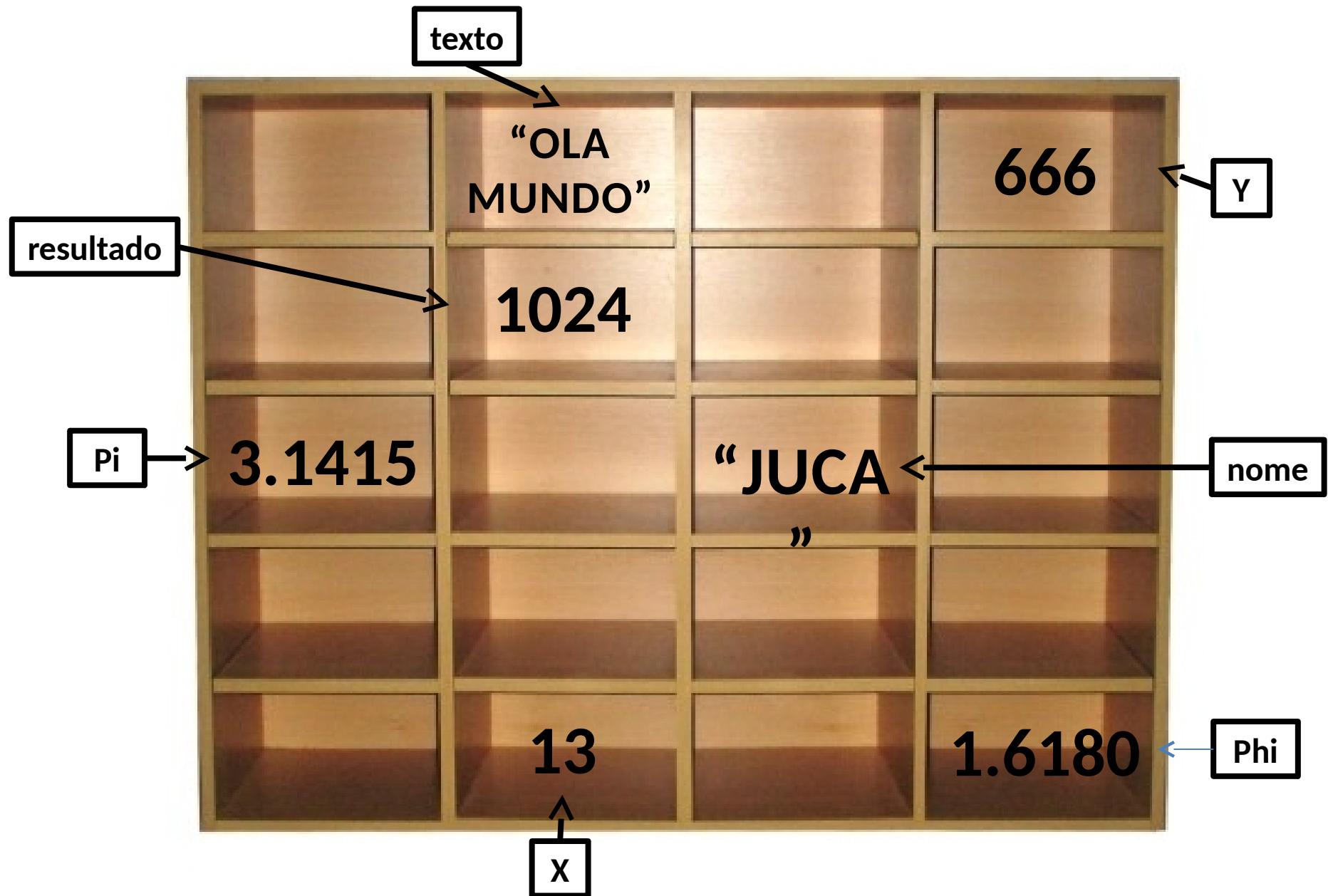
**Variáveis** são espaços de memória onde podemos armazenar números, textos etc.

# Fluxogramas: Variáveis

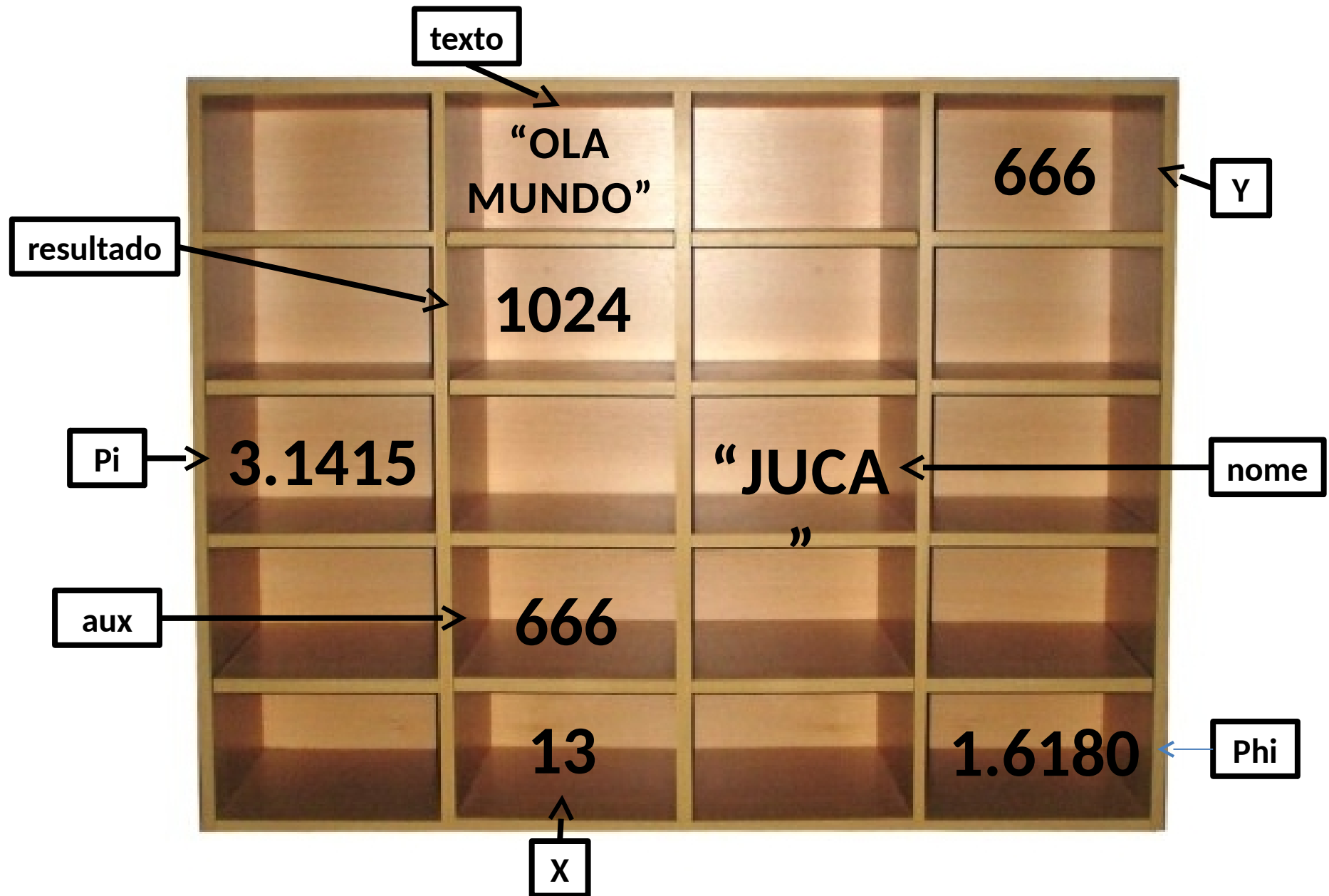




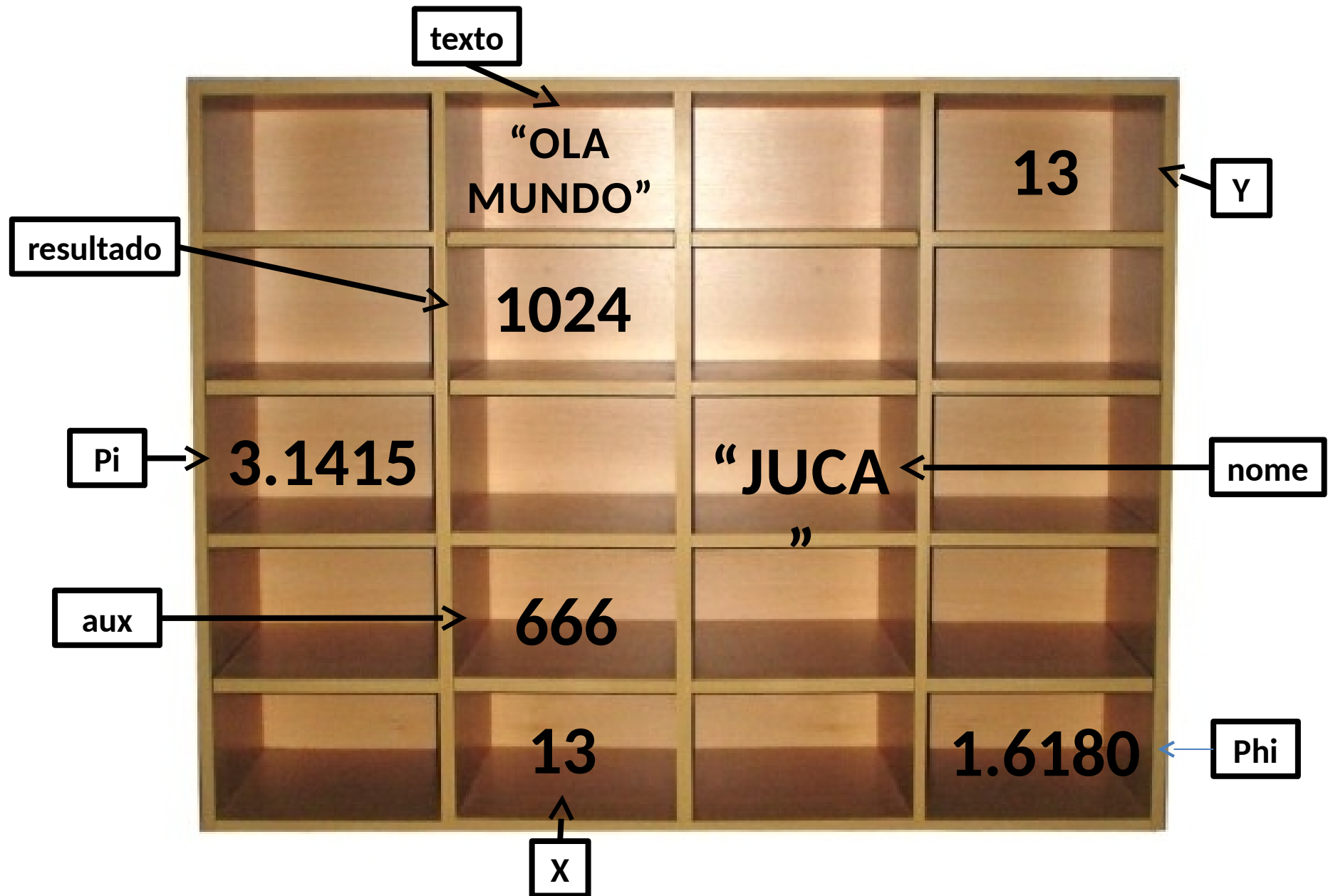
# Fluxogramas: Variáveis



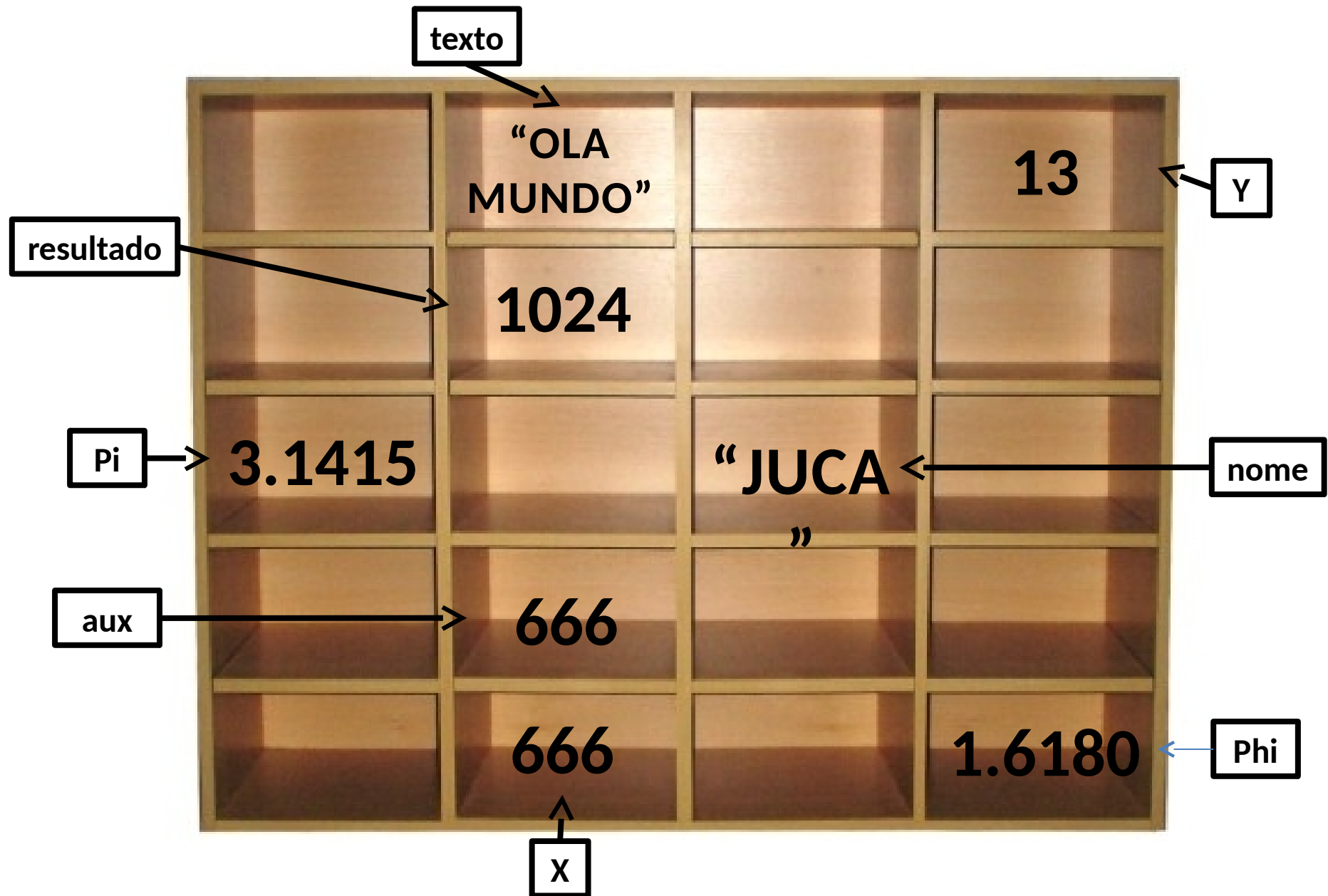
# Fluxogramas: Variáveis



# Fluxogramas: Variáveis



# Fluxogramas: Variáveis

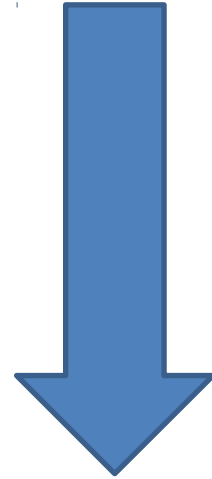


# Fluxogramas

Mas, e as setas??

**FLUXO**grama

Diagrama de Fluxo



Para que servem as setas?

Informar a **direção**

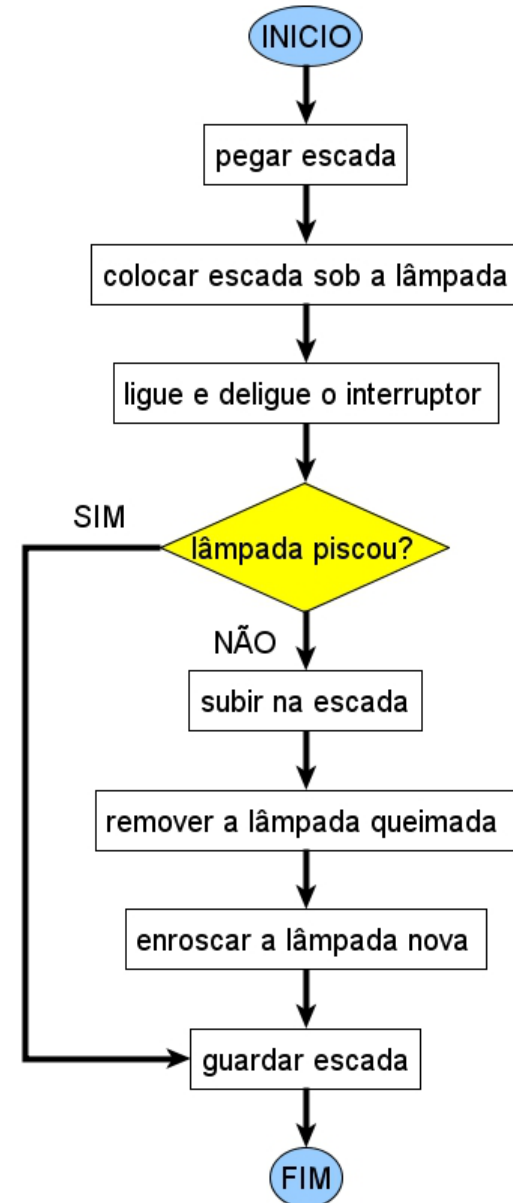
Determinar o **fluxo de execução do algoritmo**

Ligam o **INÍCIO** ao **FIM** do fluxograma

## Exemplos de Fluxogramas

# Fluxograma: Exemplo 1

- Algoritmo para trocar a lâmpada queimada!
- Losango representa uma decisão!
- Fluxo de execução:
  - Se afirmação **VERDADEIRA** segue seta com **SIM**
  - Se afirmação **FALSA** segue seta com **NÃO**



# Exemplo 1: Caso 1

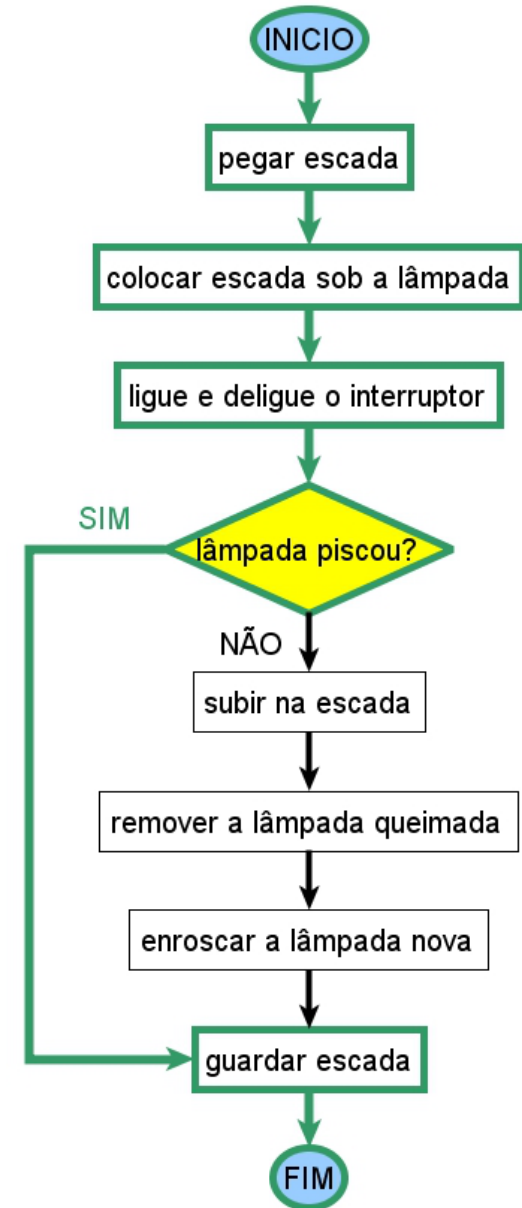


Se afirmação

**VERDADEIRA**

Se **lâmpada piscou**

segue seta com **SIM**

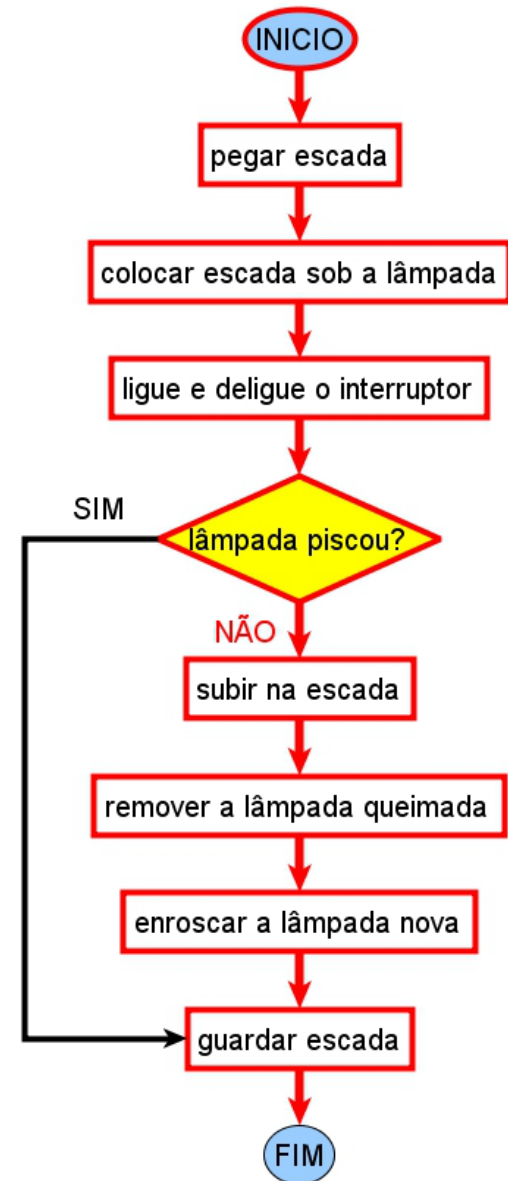




# Exemplo 1: Caso 2



Se afirmação **FALSA**  
Se **lâmpada não piscou**  
segue seta com **SIM**



# Fluxograma: Exemplo 2

Algoritmo que lê dois números N1 e N2, e imprime (escreve) o maior deles.

**N1 = 1º número**

**N2 = 2º número**

# Fluxograma: Exemplo 2

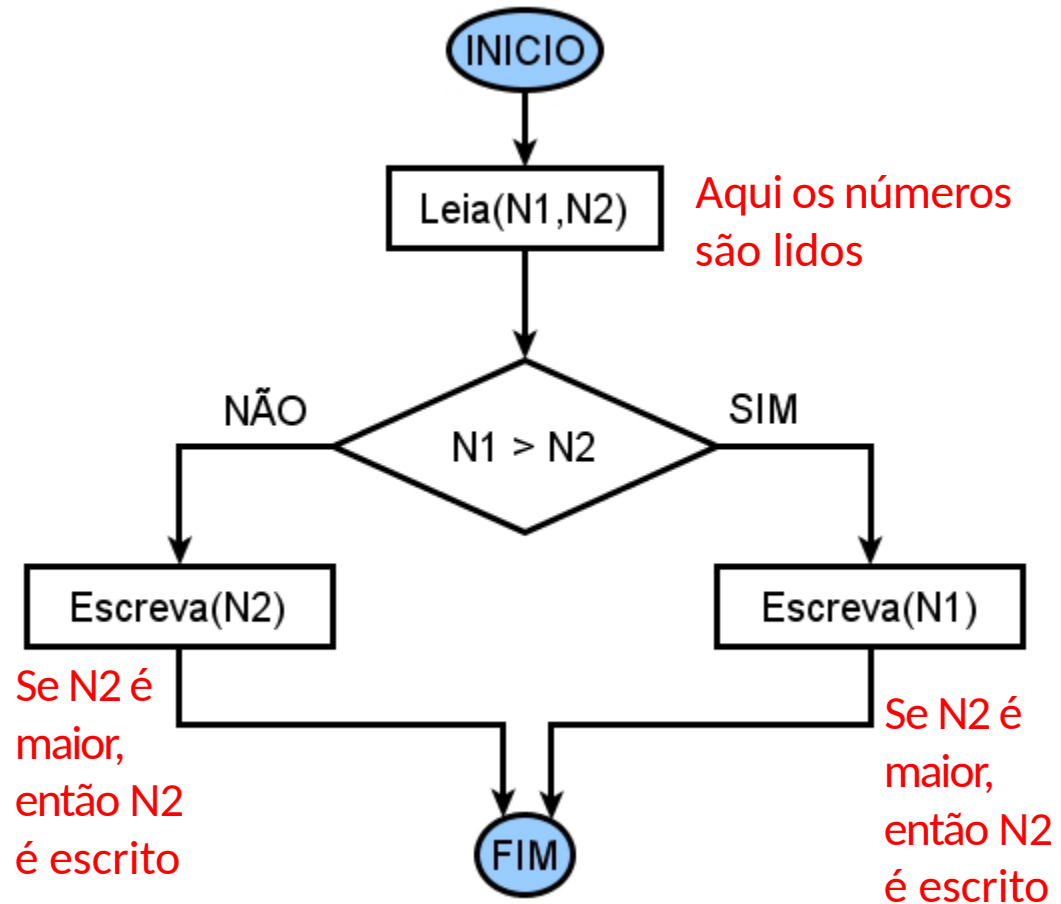
Algoritmo que lê dois números N1 e N2, e imprime (escreve) o maior deles.

**N1 = 1º número**

**N2 = 2º número**

## OBS:

Se os números forem iguais, então a variável N2 será escrita.



## Exemplo 2: Caso 1

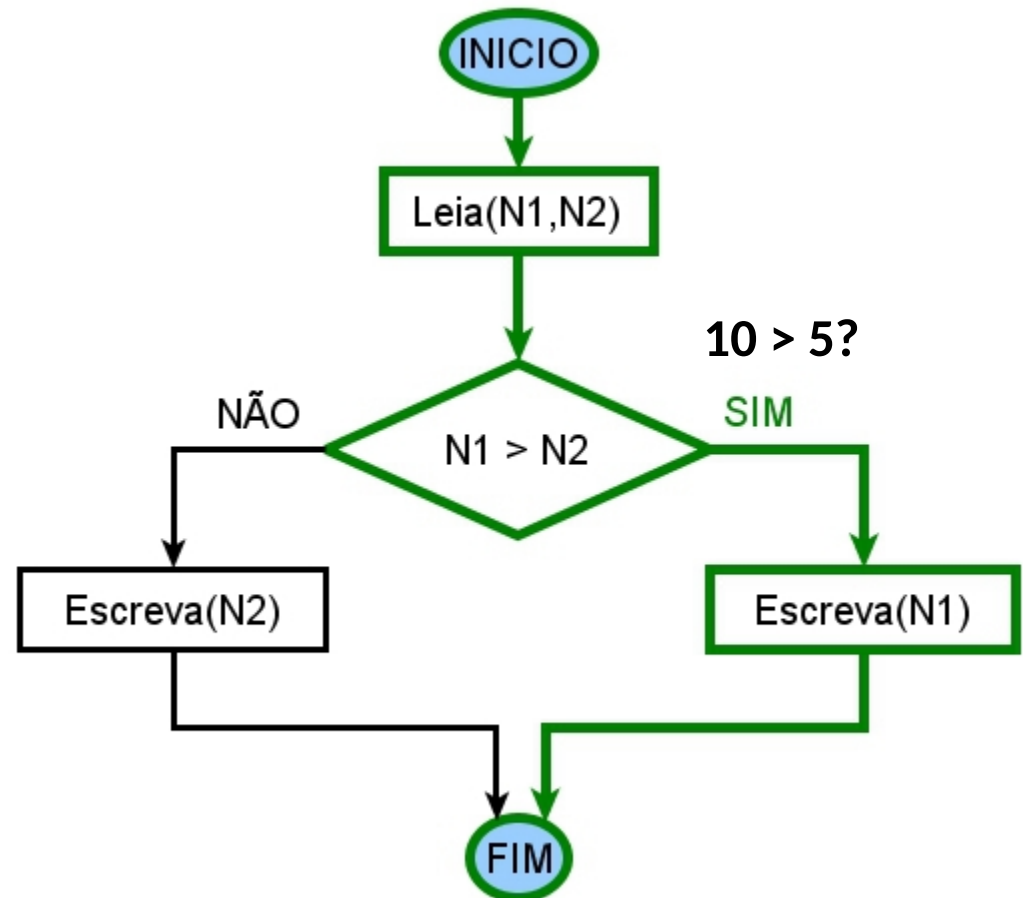
Se ( $N1 > N2$ )

$N1 = 10$

$N2 = 5$

Saída:

10



## Exemplo 2: Caso 2

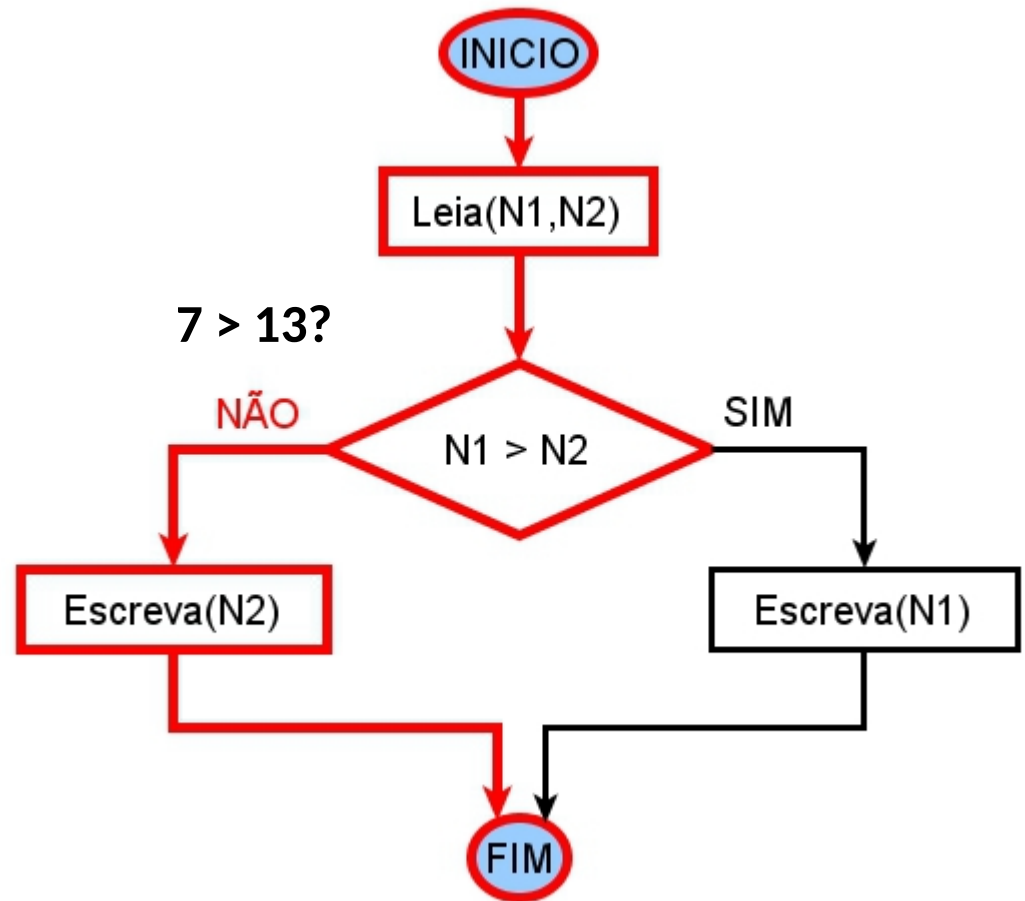
Se ( $N1 > N2$ )

$N1 = 7$

$N2 = 13$

Saída:

13



# Fluxograma: Exemplo 3

**Algoritmo que conta  
até 5, e escreve o valor  
sendo contado.**

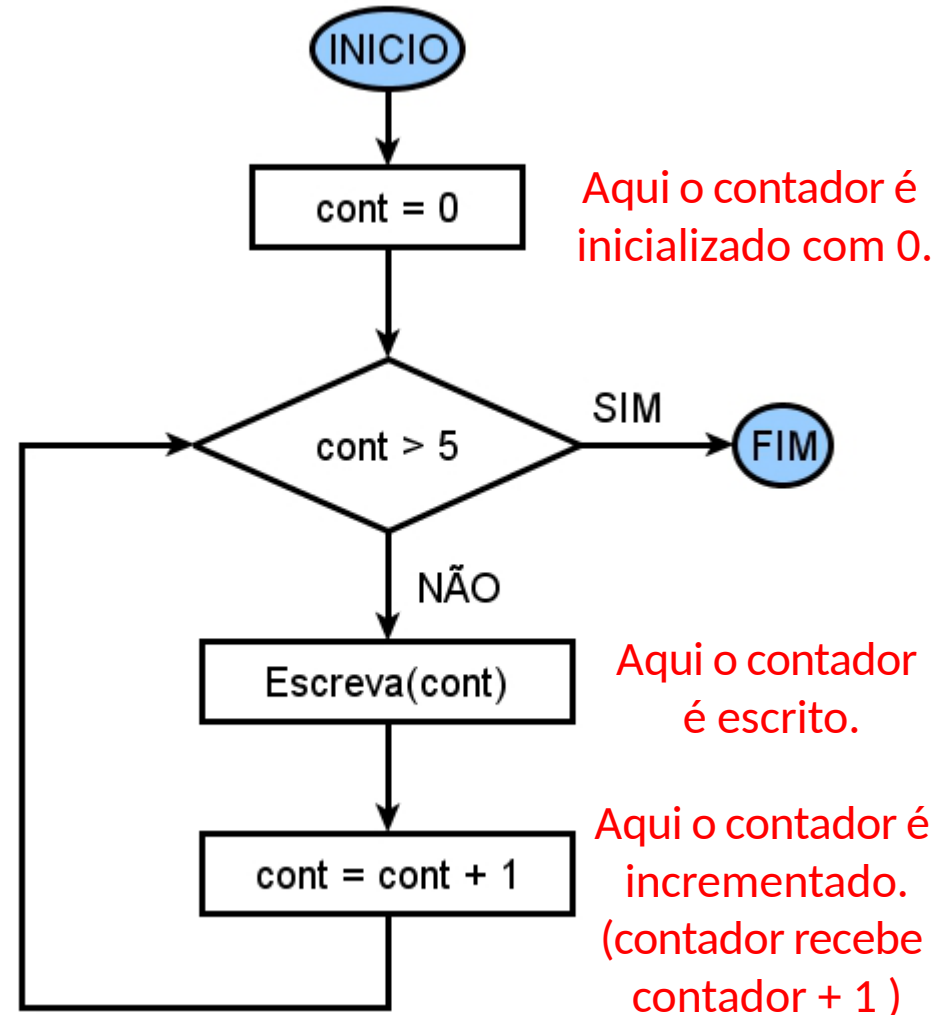
# Fluxograma: Exemplo 3

Algoritmo que conta  
até 5, e escreve o valor  
sendo contado.

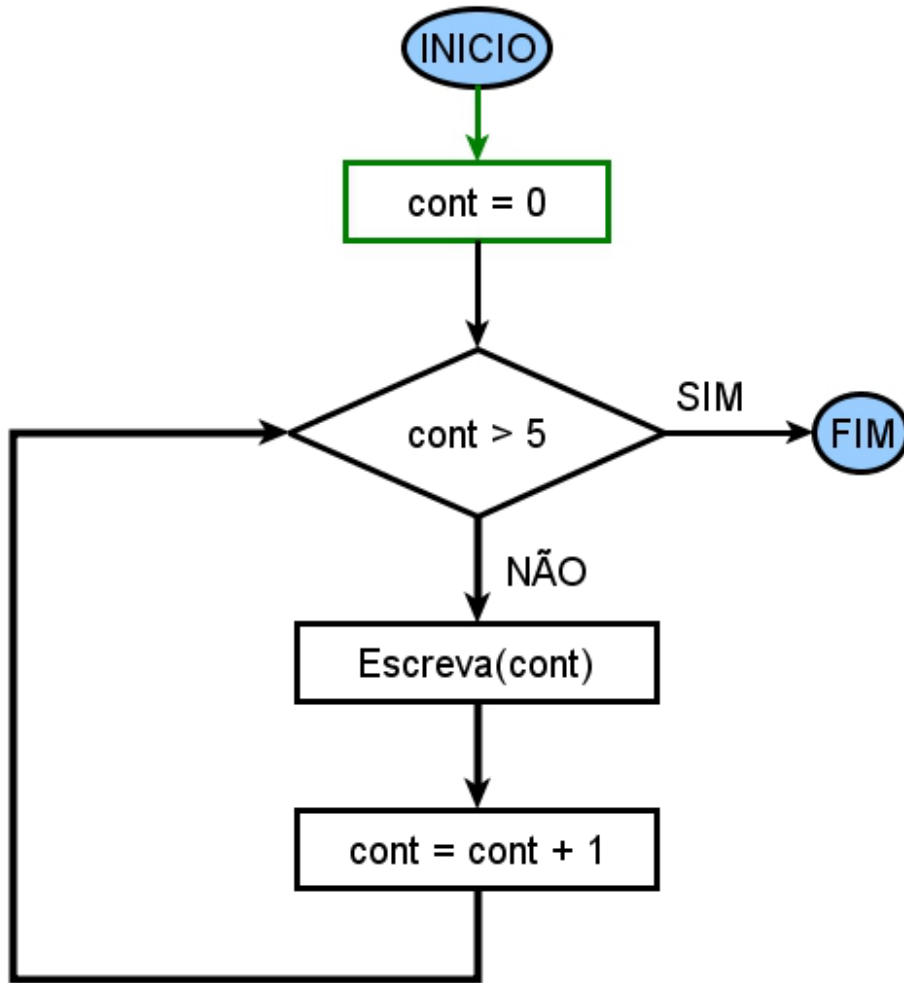
cont = contador auxiliar

**FLUXO**grama

Siga o fluxo das setas



# Exemplo 3



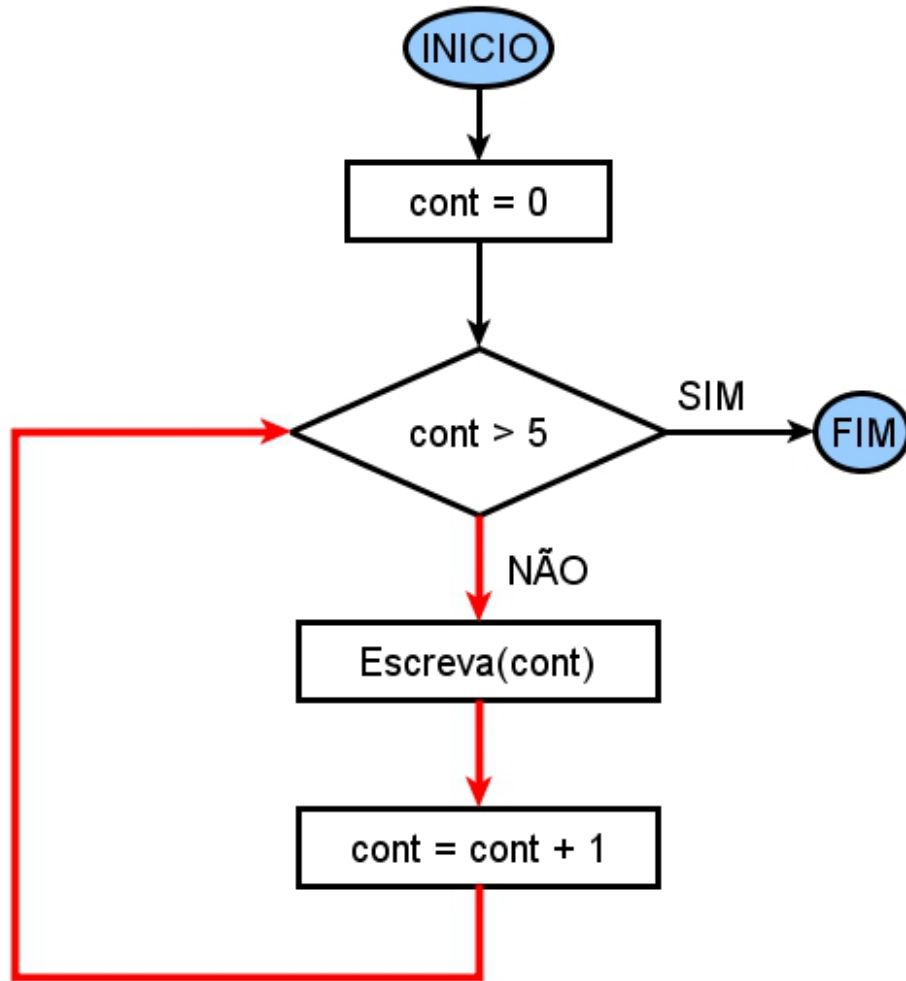
TESTE	SAÍDA	CONT
		0

INÍCI  
O

Saída:



# Exemplo 3

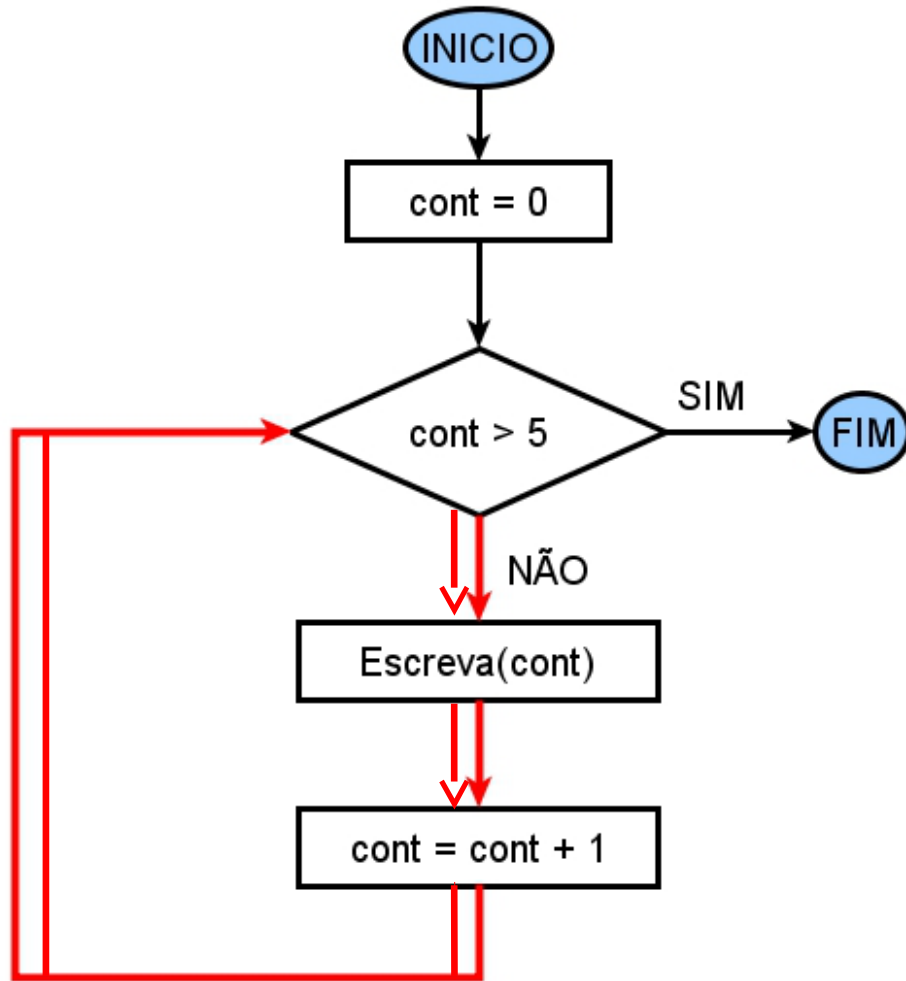


TESTE	SAÍDA	CONT
		0
F	0	1

INÍCI  
O

Saída: 0

# Exemplo 3

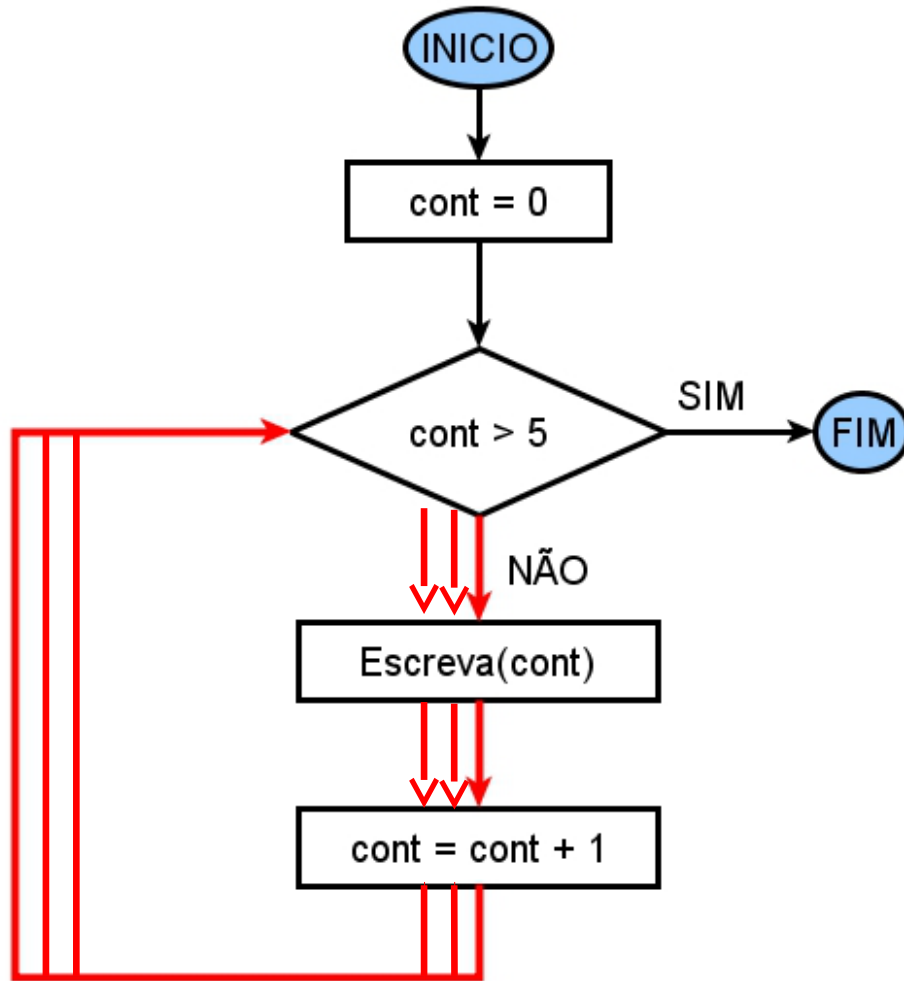


TESTE	SAÍDA	CONT
		0
F	0	1
F	1	2

INÍCI  
O

Saída: 0 1

# Exemplo 3

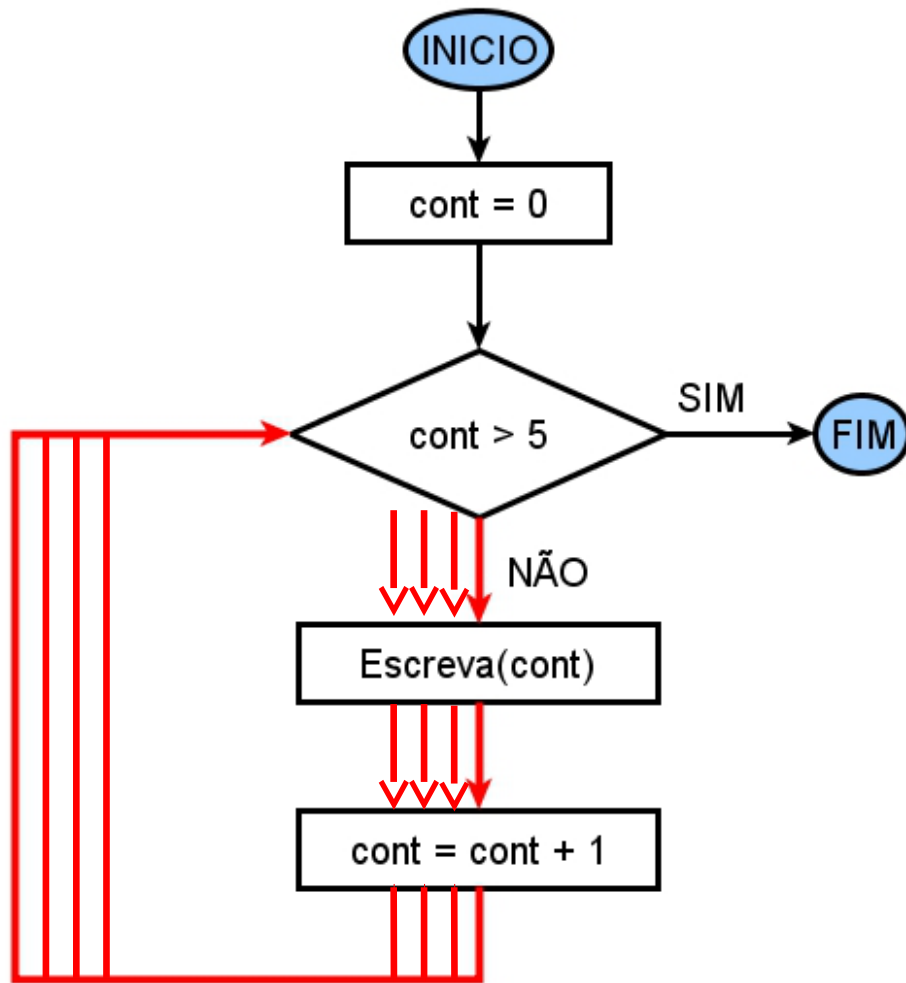


TESTE	SAÍDA	CONT
		0
F	0	1
F	1	2
F	2	3

INÍCI  
O

Saída: 0 1 2

# Exemplo 3

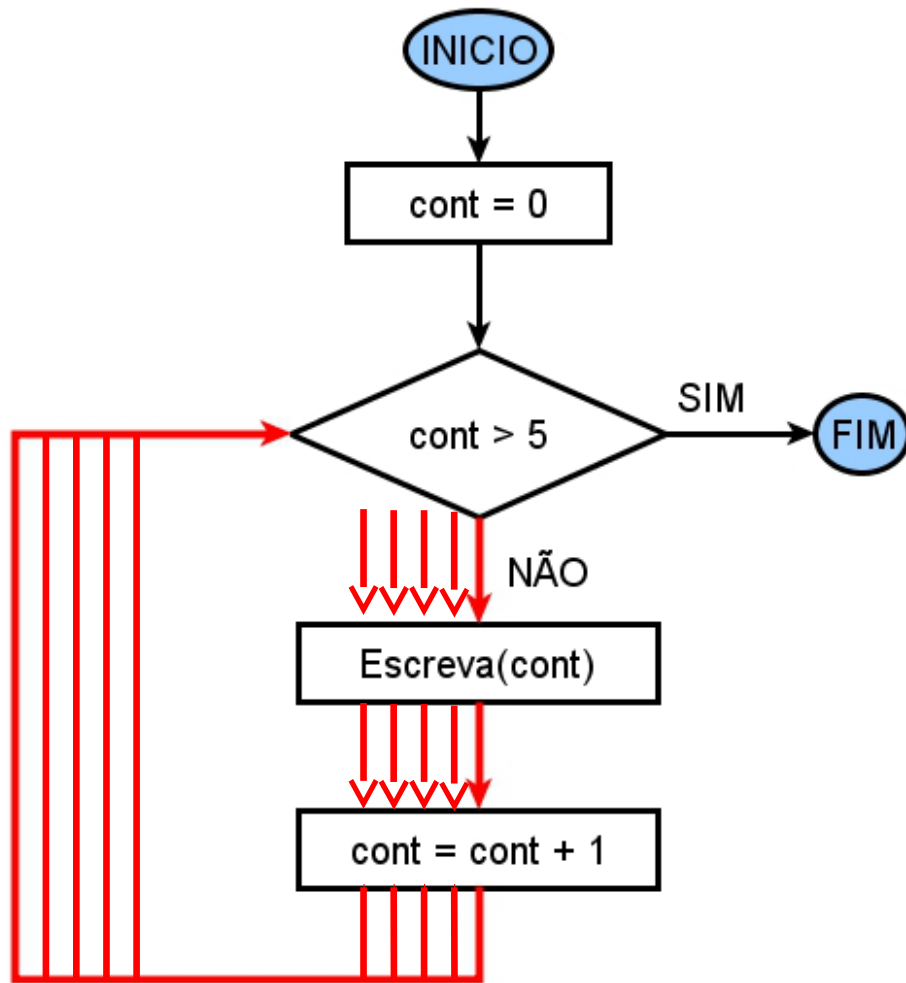


TESTE	SAÍDA	CONT
		0
F	0	1
F	1	2
F	2	3
F	3	4

INÍCI  
O

Saída: 0 1 2 3

# Exemplo 3

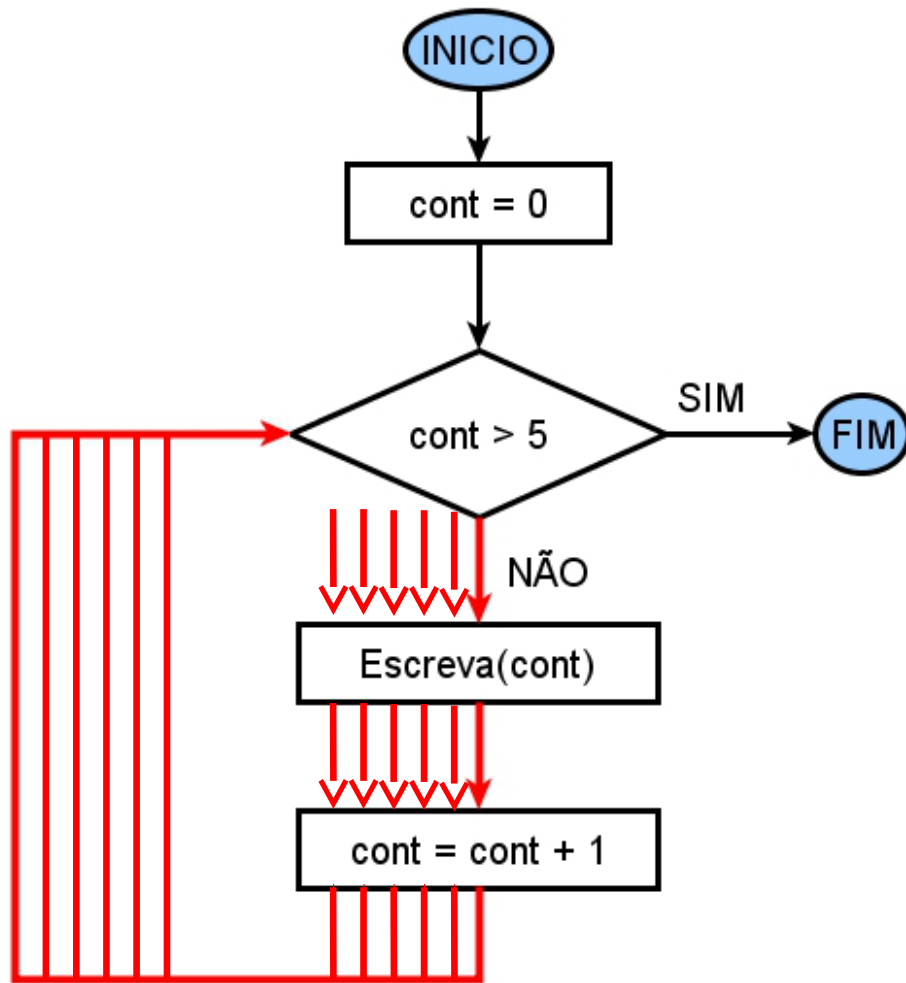


TESTE	SAÍDA	CONT
		0
F	0	1
F	1	2
F	2	3
F	3	4
F	4	5

INÍCI  
O

Saída: 0 1 2 3 4

# Exemplo 3

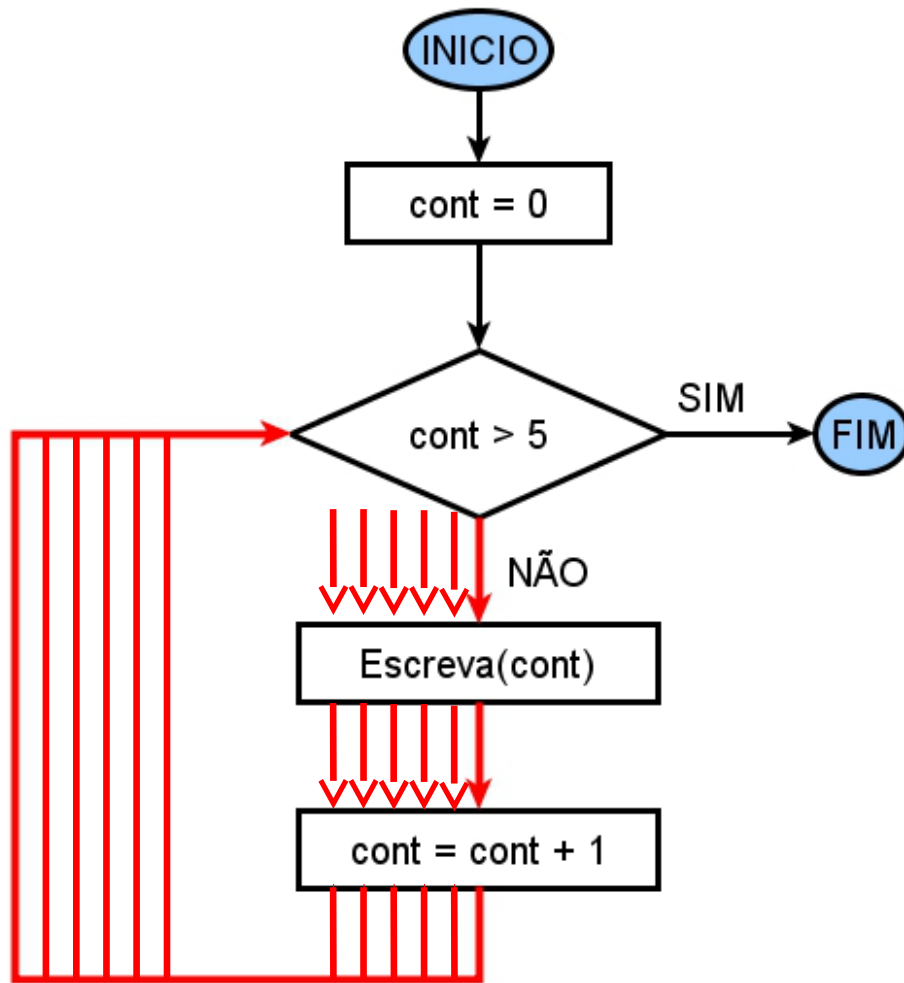


TESTE	SAÍDA	CONT
		0
F	0	1
F	1	2
F	2	3
F	3	4
F	4	5
F	5	6

INÍCI  
O

Saída: 0 1 2 3 4 5

# Exemplo 3



TESTE	SAÍDA	CONT
		0
F	0	1
F	1	2
F	2	3
F	3	4
F	4	5
F	5	6
V		6

INÍCI  
O

FIM

Saída: 0 1 2 3 4 5

# Fluxograma: Exemplo 4

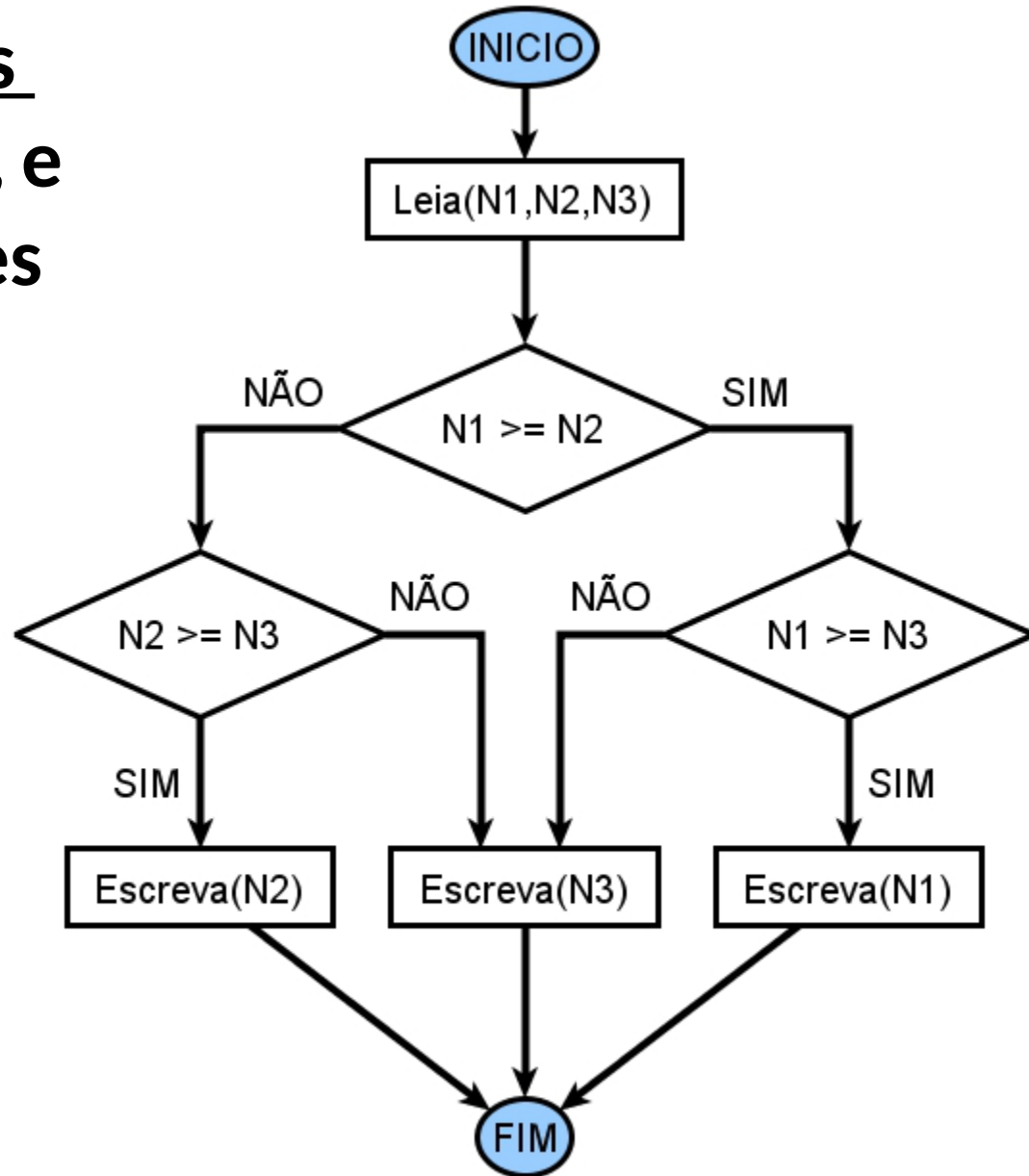
**Algoritmo que lê três  
números N1, N2 e N3, e  
escreve o maior deles**



# Fluxograma: Exemplo 4

Algoritmo que lê três números N1, N2 e N3, e escreve o maior deles

**N1 = 1º número**  
**N2 = 2º número**  
**N3 = 3º número**



# Exemplo 4: Caso 1

SE ( $N1 > N3 > N2$ )

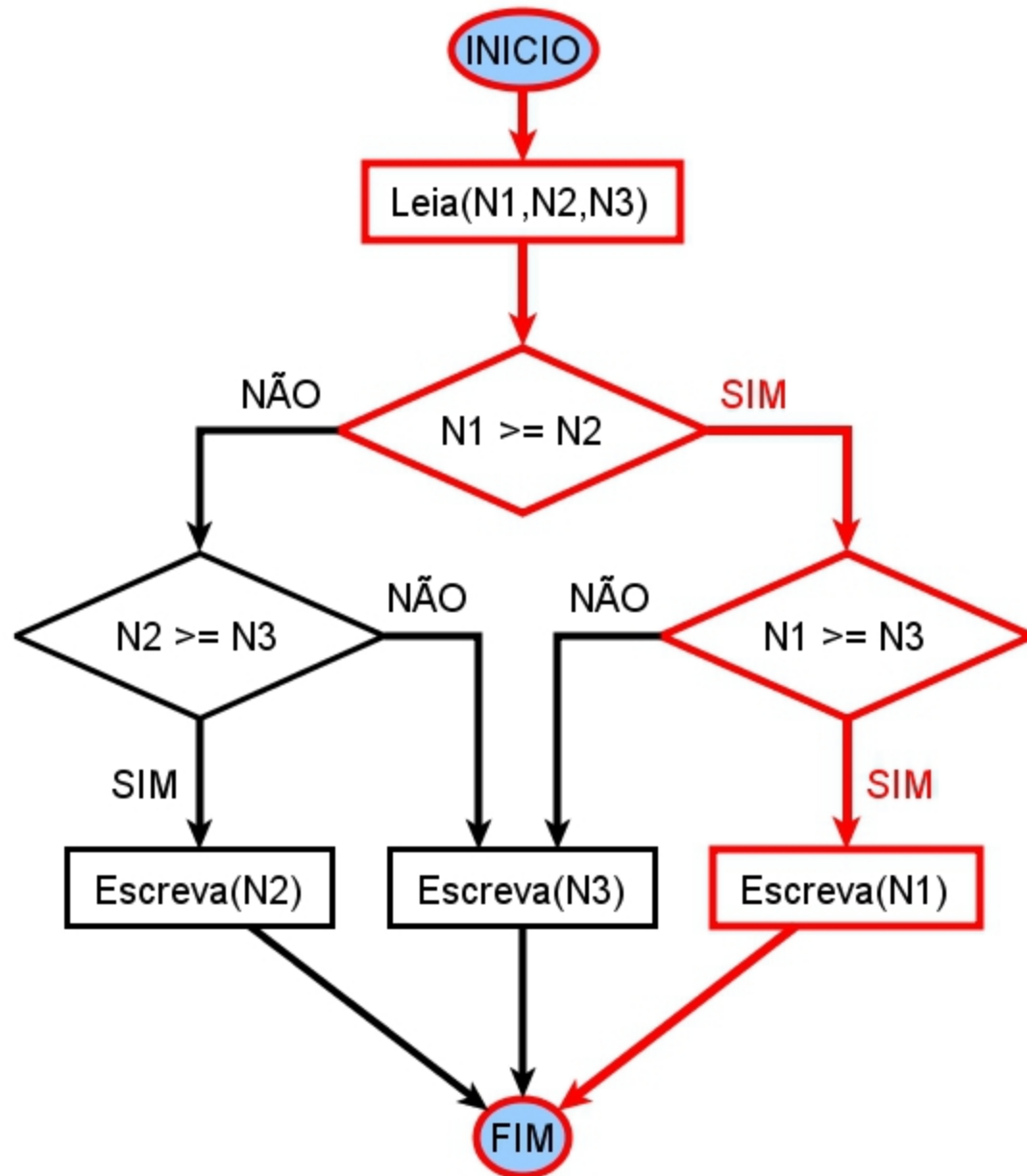
$N1 = 7$

$N2 = 3$

$N3 = 4$

Saída:

**7**



# Exemplo 4: Caso 2

SE ( $N3 > N1 > N2$ )

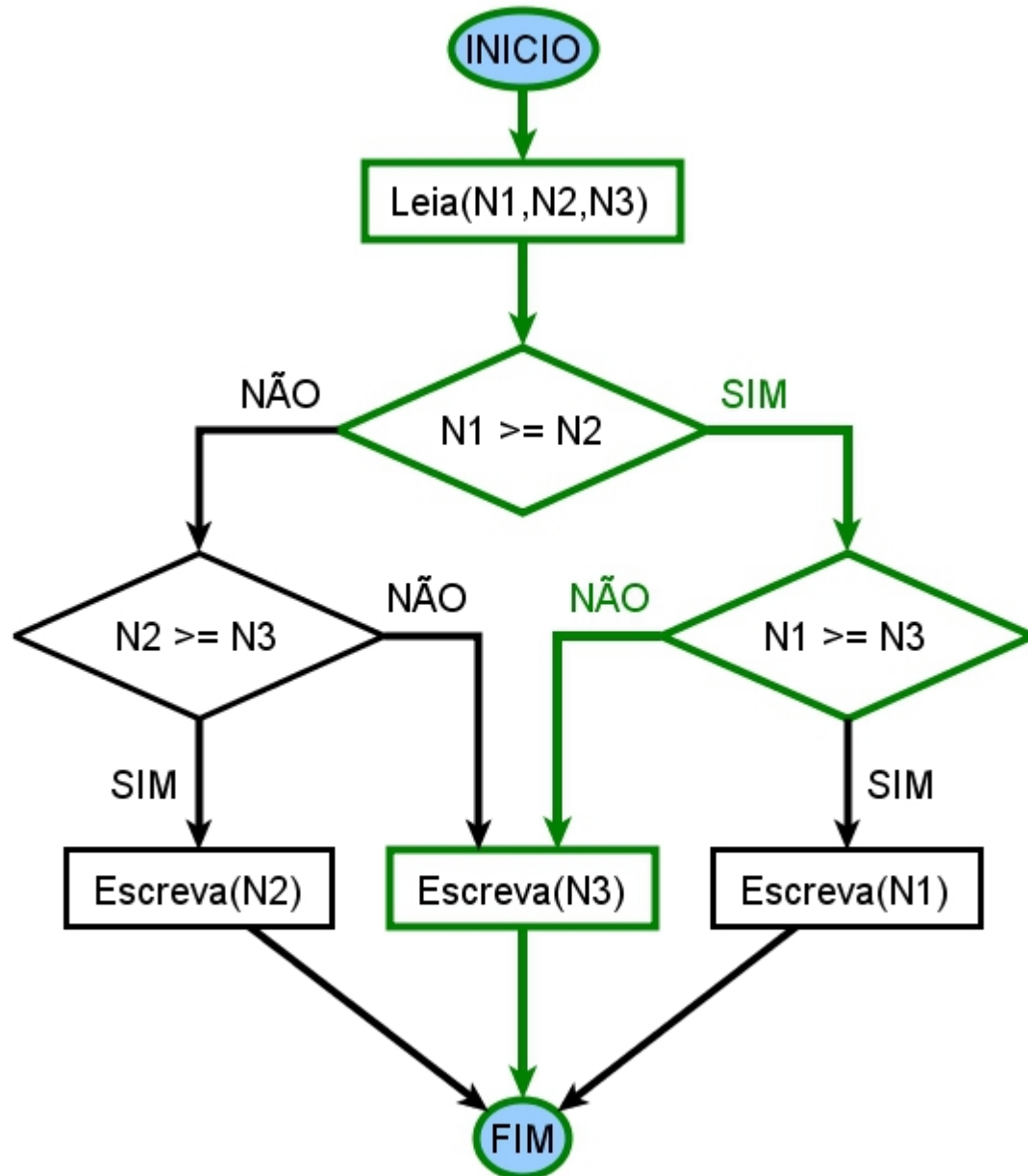
$N1 = 4$

$N2 = 3$

$N3 = 7$

Saída:

7



# Exemplo 4: Caso 3

**SE ( $N2 > N3 > N1$ )**

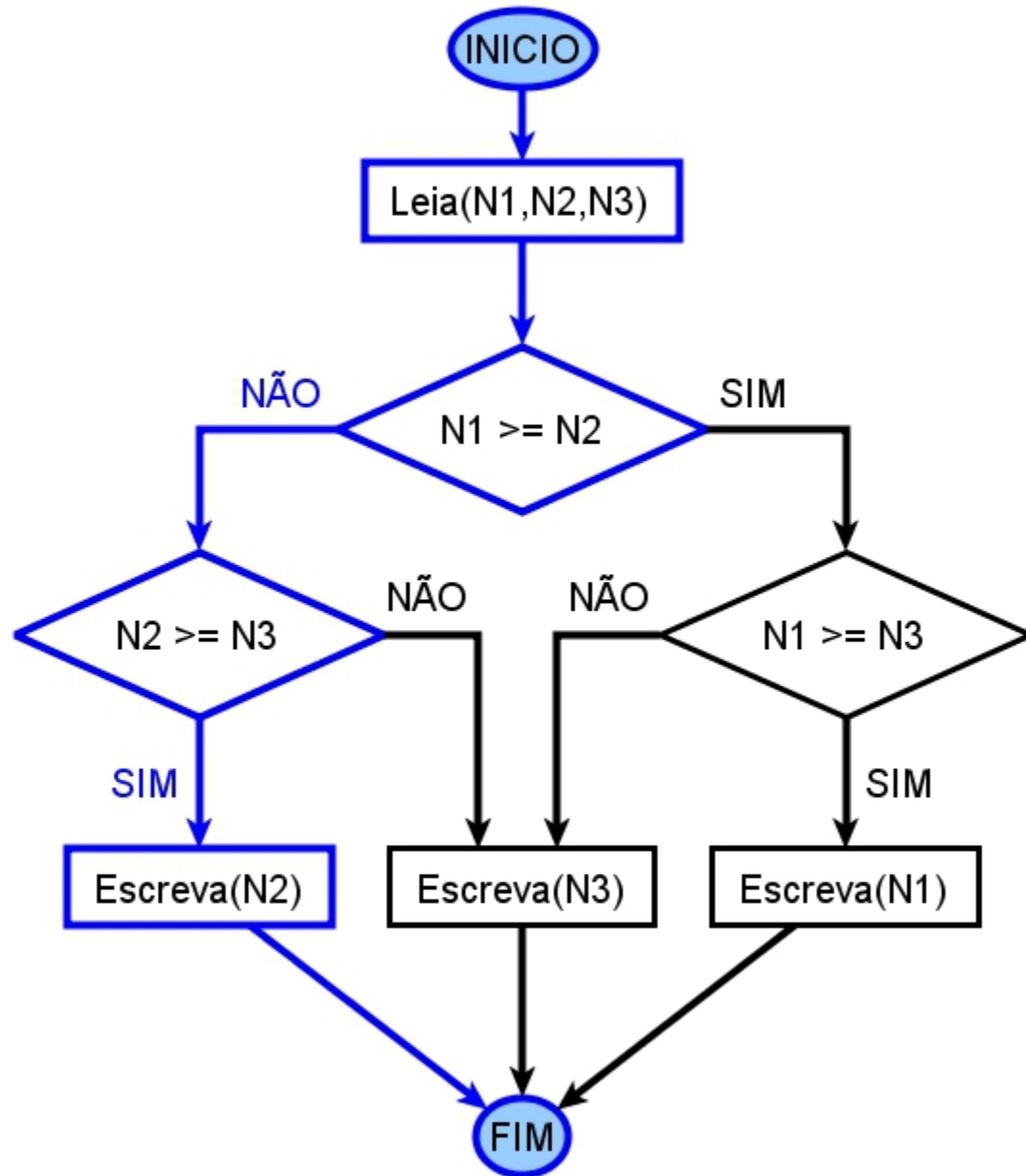
**$N1 = 3$**

**$N2 = 7$**

**$N3 = 4$**

**Saída:**

**7**



# Exemplo 4: Caso 4

SE ( $N3 > N2 > N1$ )

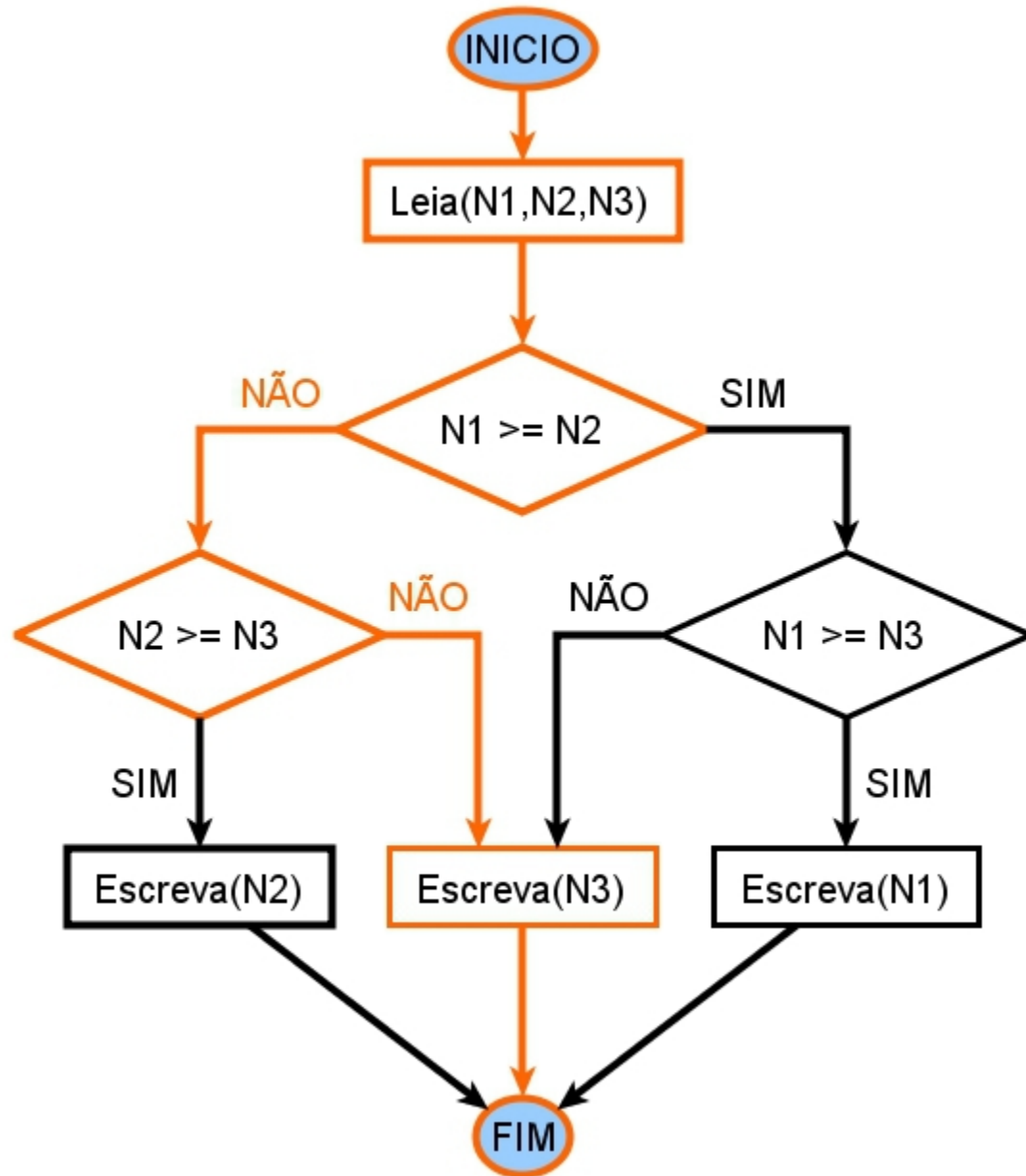
$N1 = 3$

$N2 = 4$

$N3 = 7$

Saída:

7



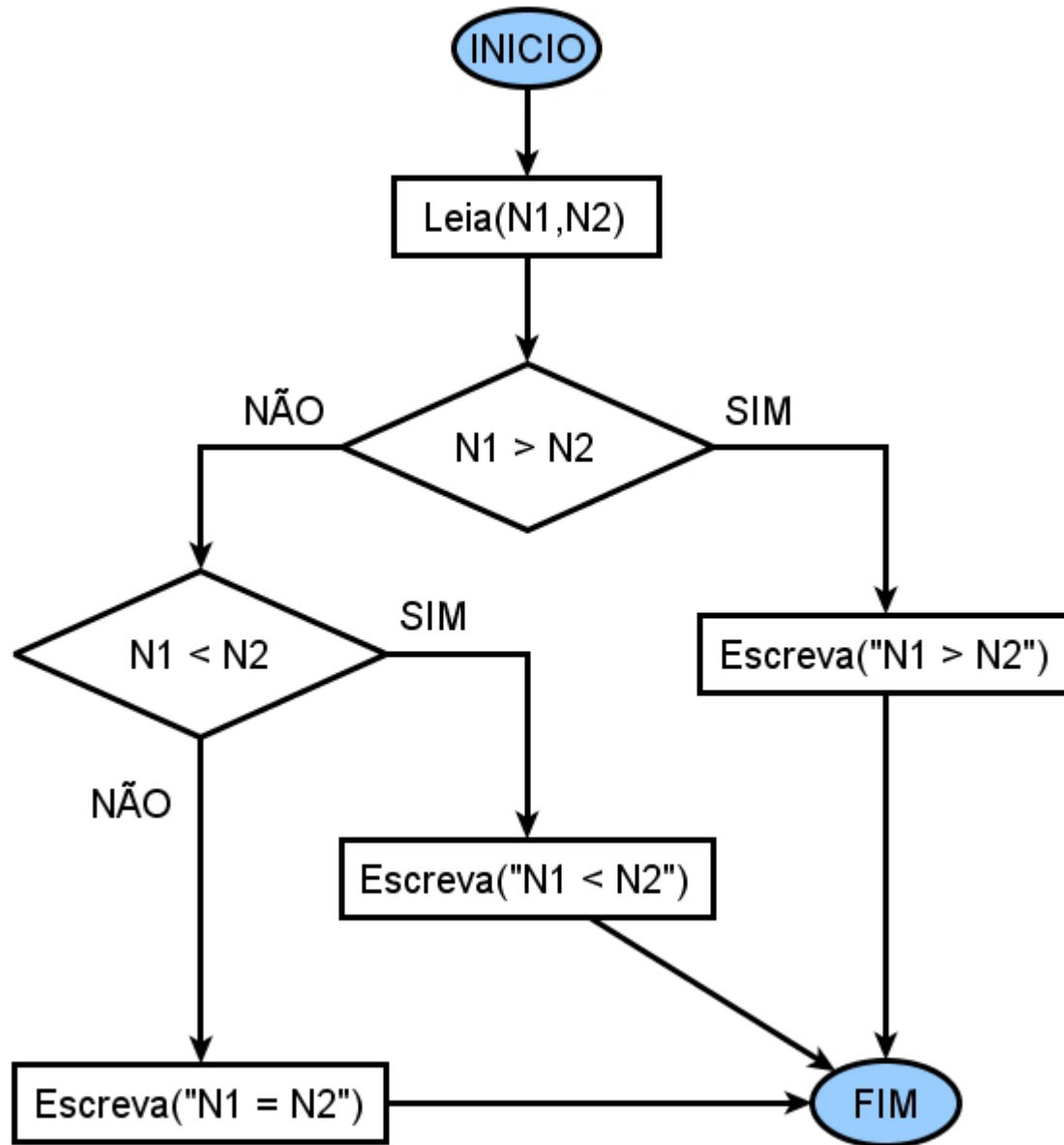
## Exercícios de Fluxograma

# Fluxograma: Exercício A

- Desenhe um fluxograma para o algoritmo que compara dois números ( $N1$  e  $N2$ ), e escreve/imprime a relação entre eles.

Use os textos " $N1 > N2$ ", " $N1 < N2$ ", ou " $N1 = N2$ " para imprimir as saídas.

# Exercício A



Vamos ver  
caso a caso...



# Exercício A: Caso 1

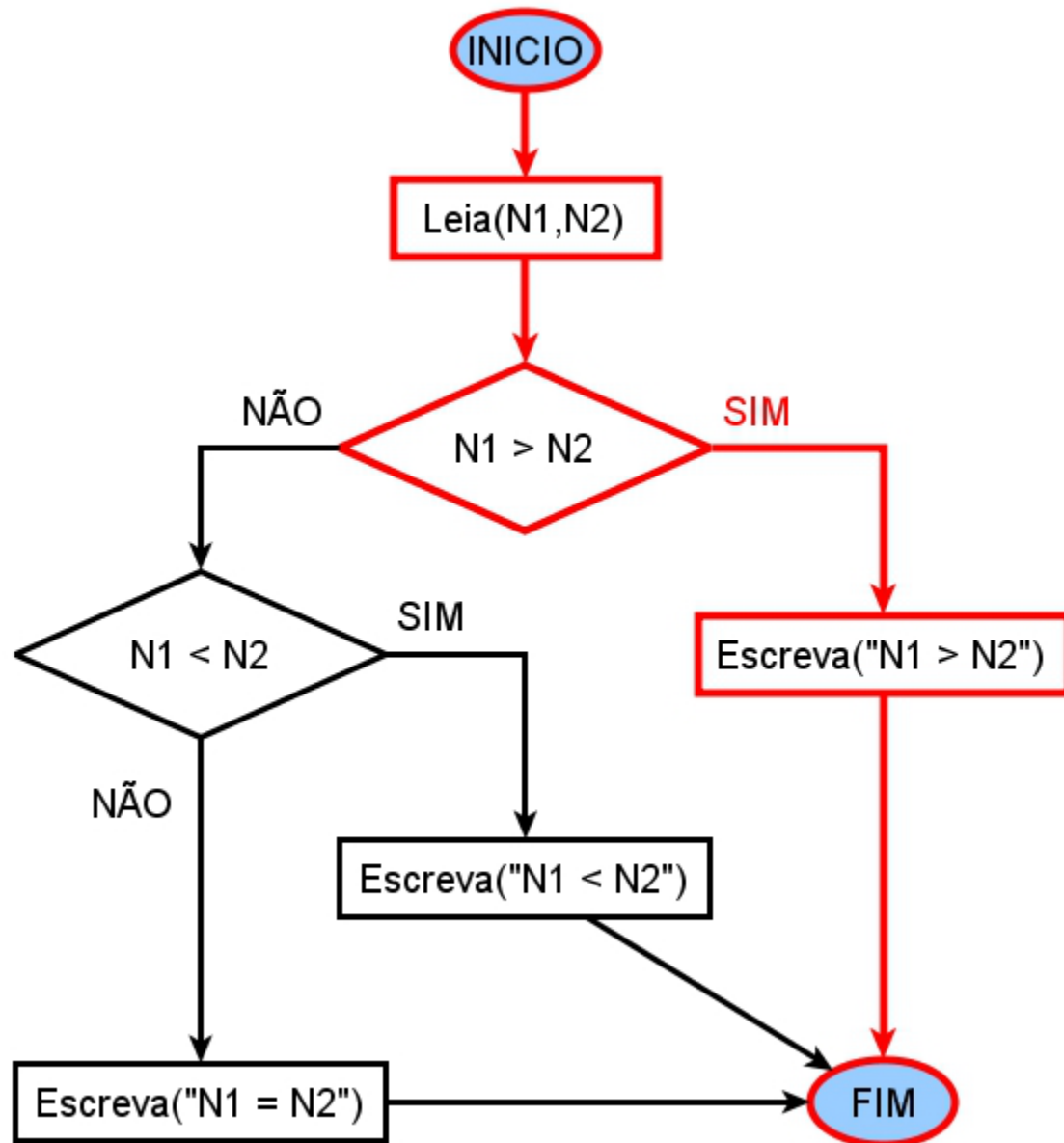
SE ( $N1 > N2$ )

$N1 = 8$

$N2 = 4$

Saída:

$N1 > N2$



# Exercício A: Caso 2

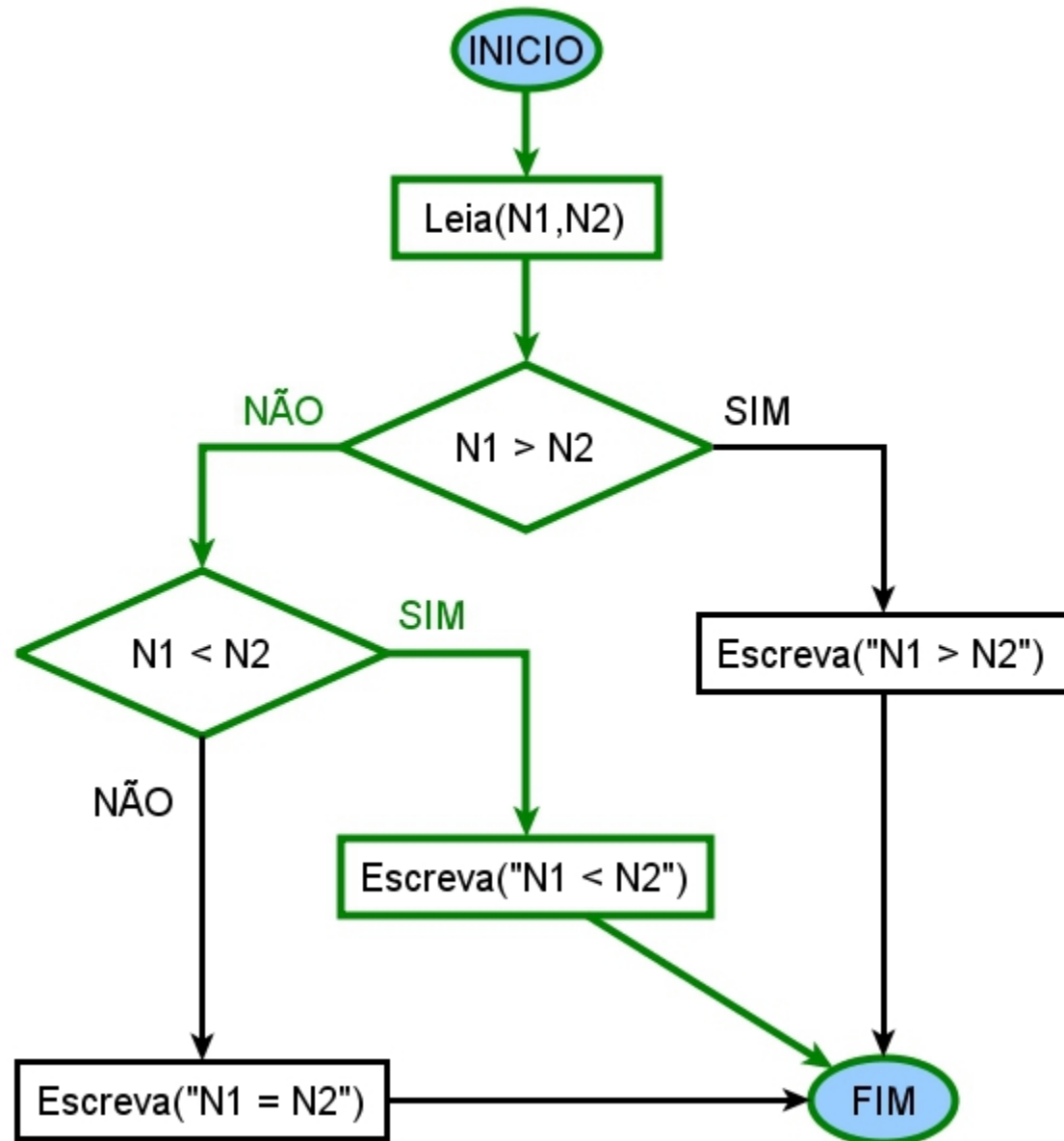
SE ( $N2 > N1$ )

$N1 = 4$

$N2 = 8$

Saída:

$N1 < N2$



# Exercício A: Caso 3

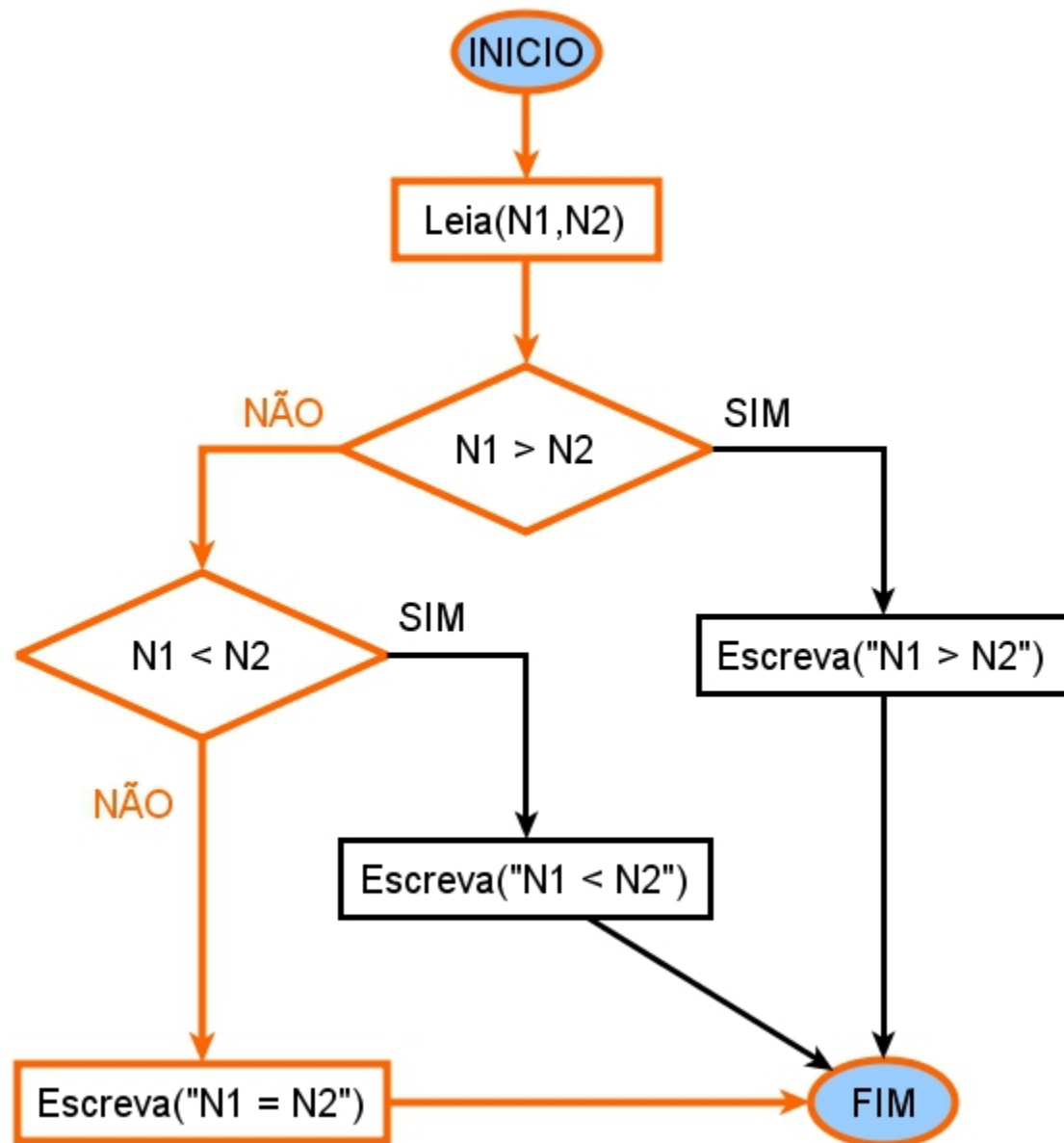
SE ( $N1 = N2$ )

$N1 = 4$

$N2 = 4$

Saída:

$N1 = N2$

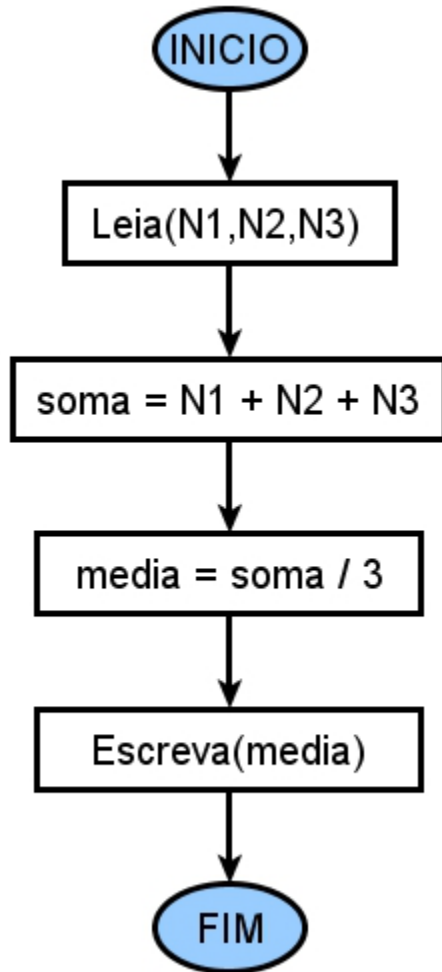


# Fluxograma: Exercício B

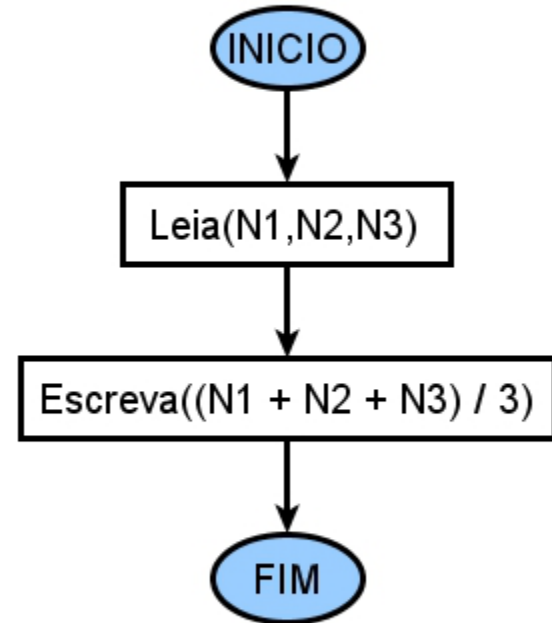
- Desenhe um fluxograma para o algoritmo que lê três números (N1, N2 e N3), calcula a média entre eles e escreve o resultado.

# Exercício B

Recebe 3 números, calcula e imprime a média.

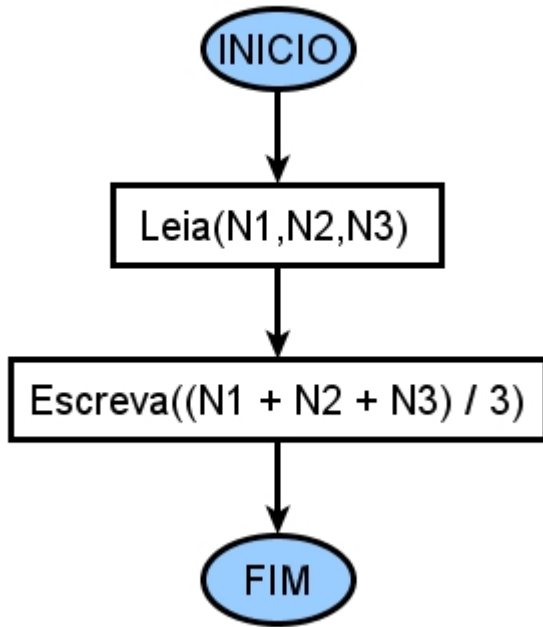


O  
U



# Exercício B: Exemplos

## Exemplo 1:



Se:

N1 = 2

N2 = 4

N3 = 60

Saída:

22

## Exemplo 2:

Se:

N1 = 6

N2 = 7

N3 = 8

Saída:

7