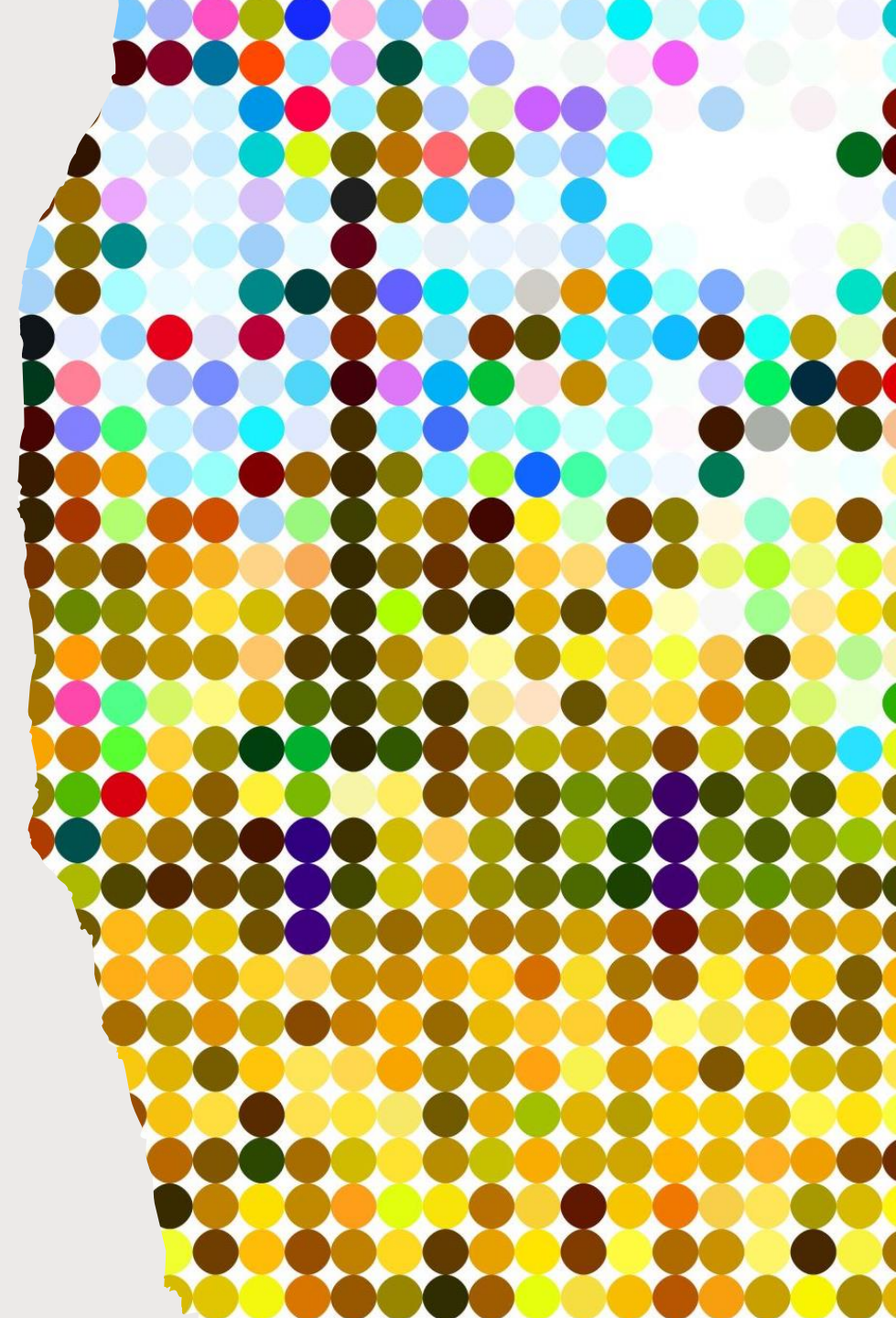


# ESTRUTURAS DE CONTROLE DE FLUXO E CONNECTIVOS LÓGICOS

Algoritmos e Lógica de Programação  
Prof. Dr. Gerson Pastre de Oliveira



# CONNECTIVOS LÓGICOS

- Em linguagens estruturadas, os conectivos lógicos mais comuns são E, OU, NÃO
- A ideia das funções lógicas é organizar testes compostos, nos quais seria possível indicar mais de uma condição – nesse caso, as condições seriam mediadas por um dos conectivos lógicos mencionados
- Por exemplo, se o problema indica como condição que, para fornecer aumento de 10% de salário para um indivíduo, seria necessário testar se o mesmo ganha atualmente mais de 5000 reais e possui idade superior a 30 anos, a lógica do teste importaria indicar testes como  $\text{salário} > 5000$  **E**  $\text{idade} > 30$
- Se, por outro lado, a condição indicasse que qualquer das duas hipóteses seria suficiente, ou seja, ganhar atualmente mais de 5000 reais ou possuir idade superior a 30 anos, a lógica do teste importaria indicar testes como  $\text{salário} > 5000$  **OU**  $\text{idade} > 30$

# CONNECTIVOS LÓGICOS

- Pode-se colocar um número relativamente grande de testes lógicos mediados pelos conectivos E, OU, NÃO
- Soluções com conectivos lógicos podem ser usadas em expressões simples ou aninhadas
- Os símbolos, em linguagem C, são os seguintes:

Conectivo lógico E: **&&**

Conectivo lógico OU: **||**

Conectivo lógico NÃO: **!**

# VALORES LÓGICOS POSSÍVEIS

- O número de testes lógicos de uma expressão determina a quantidade de valores finais possíveis em um conjunto de testes mediados por conectivos lógicos
- De maneira geral, a expressão seria  $q = 2^n$ , onde **q** corresponde à quantidade de resultados possíveis (um de cada vez, dependendo da análise dos testes parciais) e **n** seria a quantidade de testes em uma expressão
- Por exemplo, em uma expressão com 2 testes, temos 4 resultados possíveis; se forem 3 testes, serão 8 resultados possíveis, e assim por diante

# VALORES LÓGICOS POSSÍVEIS

| <i>Conectivo lógico OU</i> |         |           |
|----------------------------|---------|-----------|
| Teste 1                    | Teste 2 | Resultado |
| V                          | V       | <b>V</b>  |
| V                          | F       | <b>V</b>  |
| F                          | V       | <b>V</b>  |
| F                          | F       | <b>F</b>  |

| <i>Conectivo lógico E</i> |         |           |
|---------------------------|---------|-----------|
| Teste 1                   | Teste 2 | Resultado |
| V                         | V       | <b>V</b>  |
| V                         | F       | <b>F</b>  |
| F                         | V       | <b>F</b>  |
| F                         | F       | <b>F</b>  |

| <i>Conectivo lógico NÃO</i> |           |
|-----------------------------|-----------|
| Teste                       | Resultado |
| V                           | <b>F</b>  |
| V                           | <b>V</b>  |

# VALORES LÓGICOS POSSÍVEIS

| <i>Conectivo lógico OU</i> |         |         |           |
|----------------------------|---------|---------|-----------|
| Teste 1                    | Teste 2 | Teste 3 | Resultado |
| V                          | V       | V       | <b>V</b>  |
| V                          | V       | F       | <b>V</b>  |
| V                          | F       | V       | <b>V</b>  |
| V                          | F       | F       | <b>V</b>  |
| F                          | V       | V       | <b>V</b>  |
| F                          | V       | F       | <b>V</b>  |
| F                          | F       | V       | <b>V</b>  |
| F                          | F       | F       | <b>F</b>  |

| <i>Conectivo lógico E</i> |         |         |           |
|---------------------------|---------|---------|-----------|
| Teste 1                   | Teste 2 | Teste 3 | Resultado |
| V                         | V       | V       | <b>V</b>  |
| V                         | V       | F       | <b>F</b>  |
| V                         | F       | V       | <b>F</b>  |
| V                         | F       | F       | <b>F</b>  |
| F                         | V       | V       | <b>F</b>  |
| F                         | V       | F       | <b>F</b>  |
| F                         | F       | V       | <b>F</b>  |
| F                         | F       | F       | <b>F</b>  |

EXEMPLOS

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
    int idade; float sal;
```

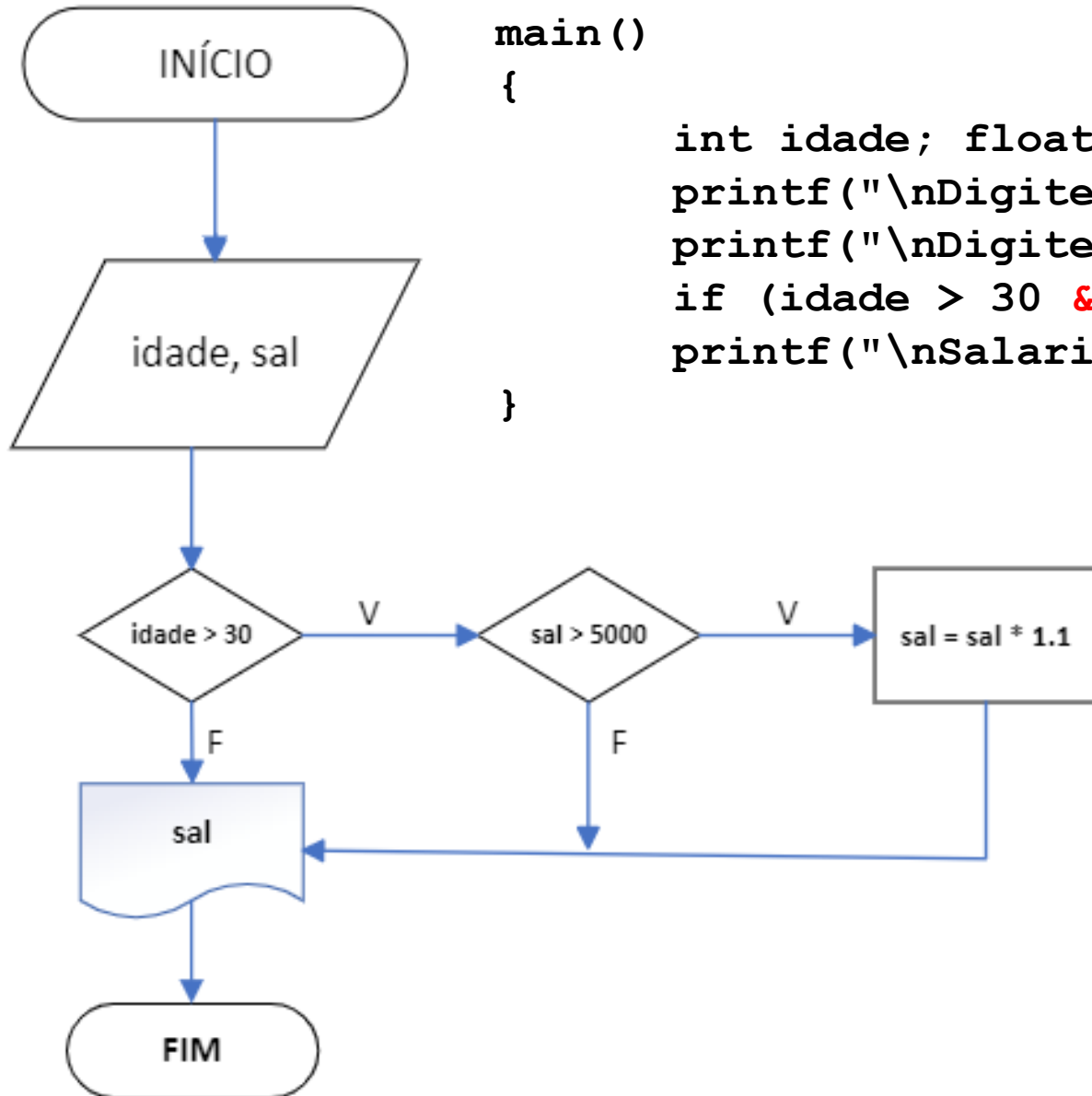
```
    printf("\nDigite a idade    = "); scanf("%d",&idade);
```

```
    printf("\nDigite o salario = "); scanf("%f",&sal);
```

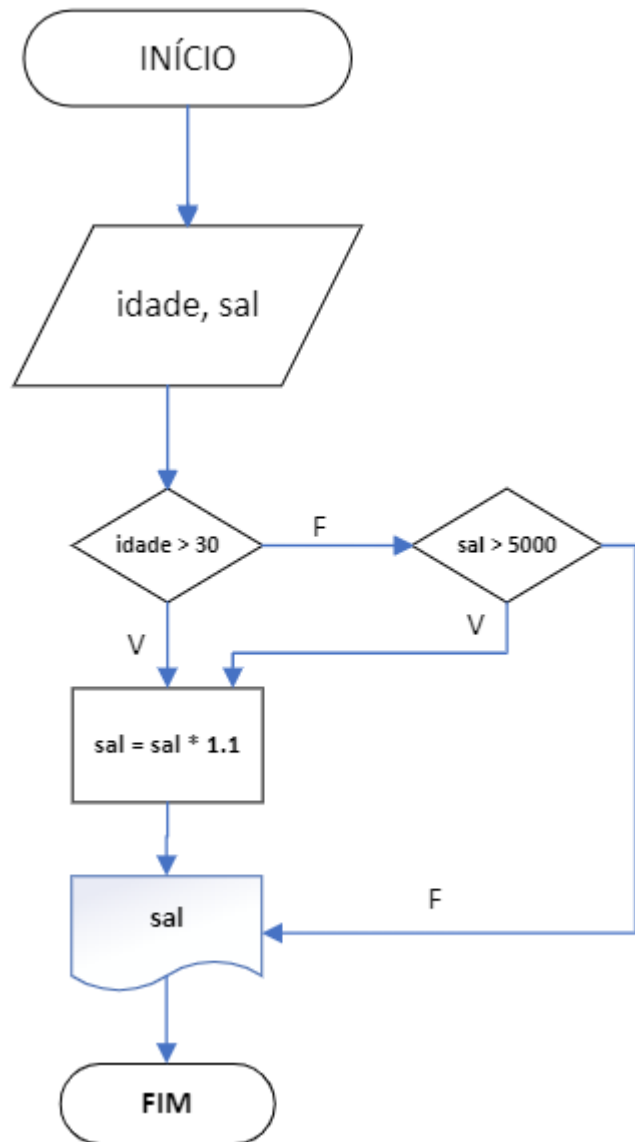
```
    if (idade > 30 && sal > 5000) sal = sal * 1.1;
```

```
    printf("\nSalario final = %.2f",sal);
```

```
}
```







```
#include <stdio.h>
```

```
{
```

```
    int idade; float sal;
```

```
    printf("\nDigite a idade    = "); scanf("%d",&idade);
```

```
    printf("\nDigite o salario = "); scanf("%f",&sal);
```

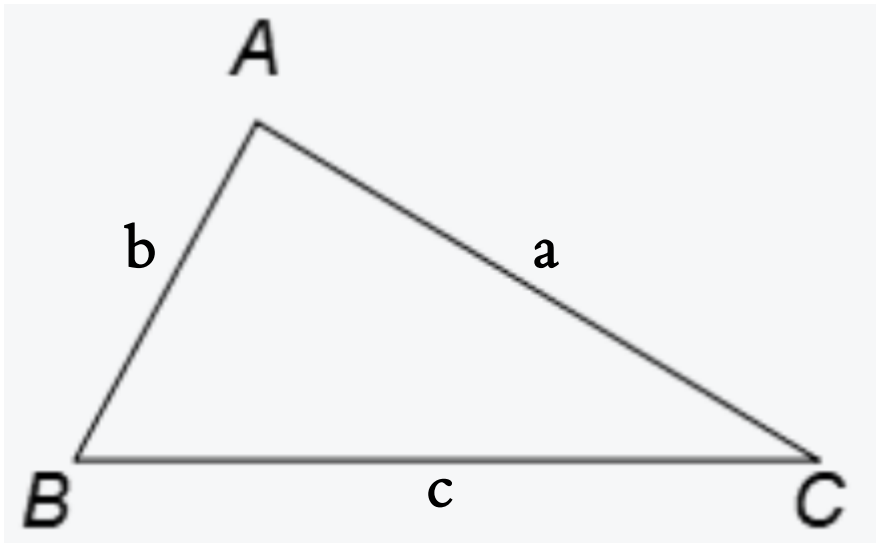
```
    if (idade > 30 || sal > 5000) sal = sal * 1.1;
```

```
    printf("\nSalario final = %.2f",sal);
```

```
}
```

## EXEMPLO

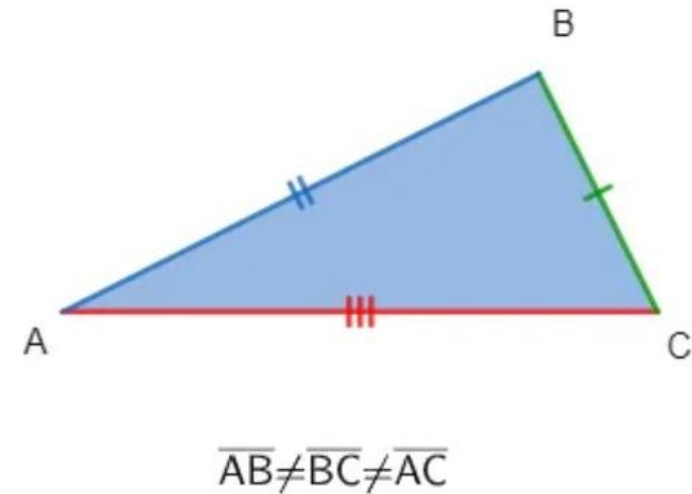
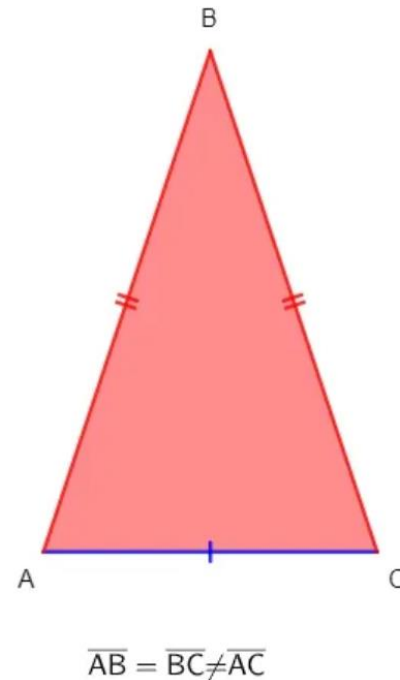
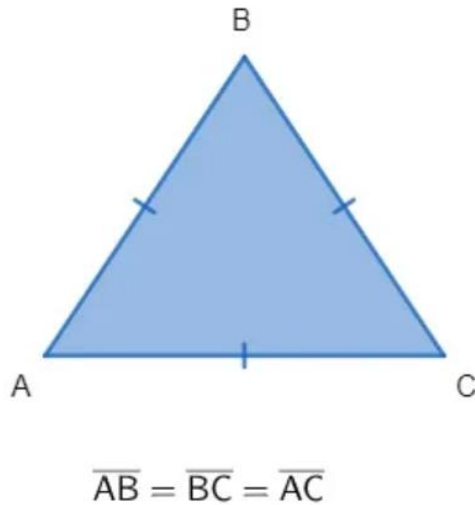
- Dadas 3 medidas, compostas por números reais, indique se as mesmas poderiam formar os lados de um triângulo; caso, possam, classifique o triângulo quanto aos lados

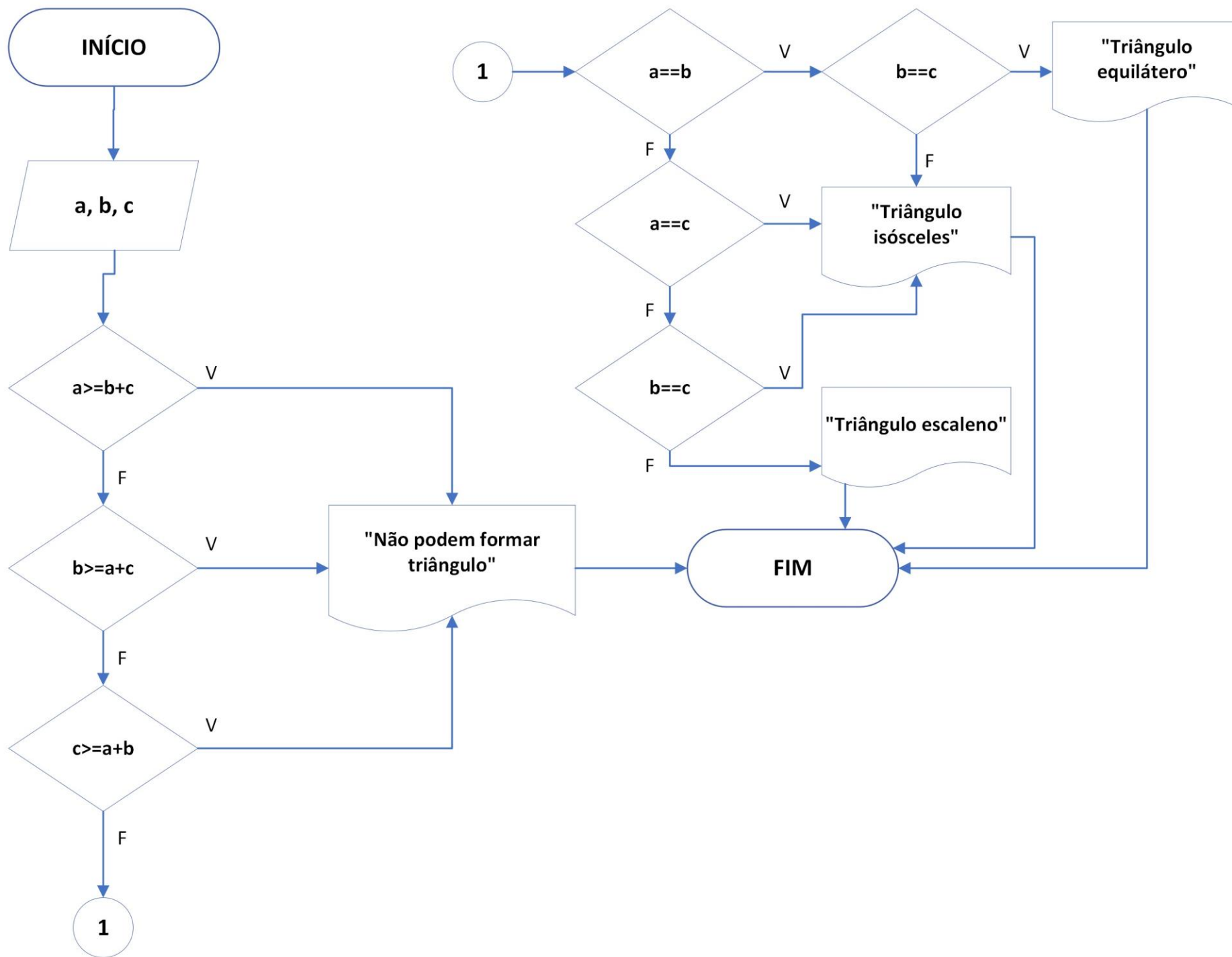


$$\begin{cases} AB < AC + BC \\ AC < AB + BC \\ BC < AB + AC \end{cases}$$

## EXEMPLO

- Dadas 3 medidas, compostas por números reais, indique se as mesmas poderiam formar os lados de um triângulo; caso, possam, classifique o triângulo quanto aos lados





```
#include <stdio.h>
```

```
main()
```

```
{
```

```
    float a, b, c;
```

```
    printf("\na = "); scanf("%f",&a);
```

```
    printf("\nb = "); scanf("%f",&b);
```

```
    printf("\nc = "); scanf("%f",&c);
```

```
    if (a >= b+c || b >= a+c || c >= a+b) printf("\nLados nao formam triangulo");
```

```
    else if (a==b && b==c) printf("\nTriangulo equilatero");
```

```
        else if (a==b || b==c || a==c) printf("\nTriangulo isosceles");
```

```
            else printf("\nTriangulo escaleno");
```

```
}
```