



Algoritmos e Linguagem de Programação

Capitulo 4

Programação com Desvios

Prof. Me. Renato Carioca Duarte



Programação com Desvios - Objetivos

- Apresentar os detalhes sobre o uso de tomadas de decisão em programação no estabelecimento de processamento lógico.
- Apresentar os conceitos de
 - condição, decisão e operadores relacionais,
 - operadores lógicos,
 - decisão simples,
 - decisão composta,
 - decisão seletiva
 - e divisibilidade.



Condição, Decisão e Operadores Relacionais

- Para um programa de computador tomar decisões, é necessário criarmos para ele uma condição.
- Assim, para tomar uma decisão, é necessário existir uma condição.
- **Uma condição pode produzir uma de duas respostas: pode ser uma resposta verdadeira ou uma resposta falsa.**
- A condição, por sua vez, do ponto de vista condicional, é a definição de relação lógica entre variáveis versus variáveis ou variáveis versus constantes definidas com o uso de operadores relacionais abaixo.

Operador Relacional	Descrição
==	Igual a
!=	Diferente de
>	Maior que
<	Menor que
>=	Maior do que ou igual a
<=	Menor do que ou igual a



Condição, Decisão e Operadores Relacionais

- São condições válidas as seguintes estruturas relacionais:

$A == B$, $A != B$, $A > B$, $A < B$, $A >= B$, $A <= B$ ou

$A == 5$, $A != 5$, $A > 5$, $A < 5$, $A >= 5$, $A <= 5$.

- A partir do momento que temos uma condição, podemos em um programa estabelecer ações de desvios condicionais.
- Os desvios condicionais estão associados à utilização de decisões em um programa de computador.
- **Uma decisão a ser tomada, como já comentado, pode ser verdadeira ou falsa.**
- **Se verdadeira, determinada ação pode ser executada;**
- **Se falsa, outra ação pode ou não ser executada.**
- **Portanto, um desvio condicional pode ser simples ou composto.**



Desvio Condicional Simples

- Um desvio condicional será simples quando tivermos uma condição que desvia a execução do programa caso o resultado lógico avaliado de certa condição seja verdadeiro.
- **Se o resultado lógico avaliado for falso, nada acontecerá, e o programa simplesmente segue o seu fluxo de execução.**
- Um desvio condicional simples, nas linguagens de programação C e C++, é realizado por meio da instrução `if`, que possui a seguinte sintaxe:

```
if <(condição)>
{
    <instrução 1 executada quando condição for verdadeira>;
    <instrução 2 executada quando condição for verdadeira>;
    <instrução N executada quando condição for verdadeira>;
}
<instrução executada após condição ser verdadeira e/ou falsa>;
```



Desvio Condicional Simples

- Exemplo: Fazer um algoritmo que receba um número e mostre uma mensagem caso este seja maior que 10.

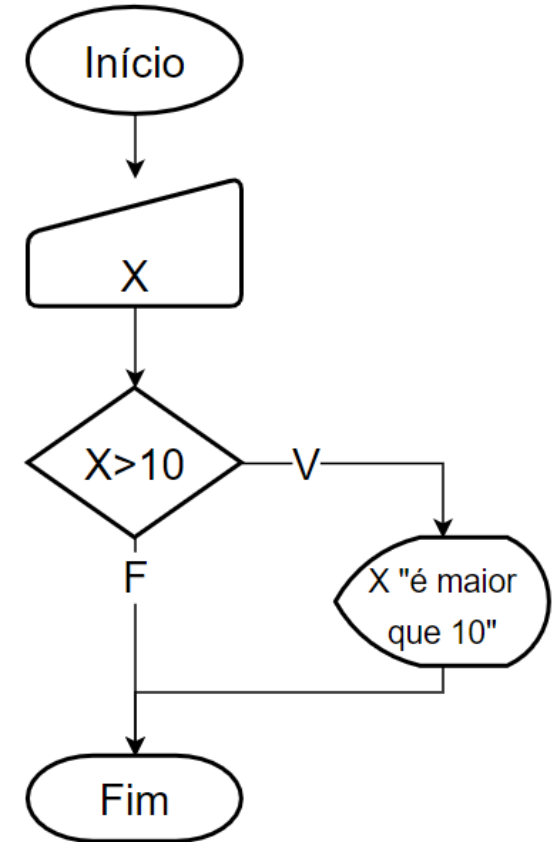




Desvio Condicional Simples

- Exemplo: Fazer um algoritmo que receba um número e mostre uma mensagem caso este seja maior que 10.

```
public static void Main(string[] args)
{
    int x;
    x = int.Parse (Console.ReadLine());
    if (x>10)
    {
        Console.WriteLine ("{0} é maior que 10",x);
    }
}
```





Desvio Condicional Simples

- Exemplo: Fazer um algoritmo que receba um número e mostre uma mensagem caso este seja maior que 10.

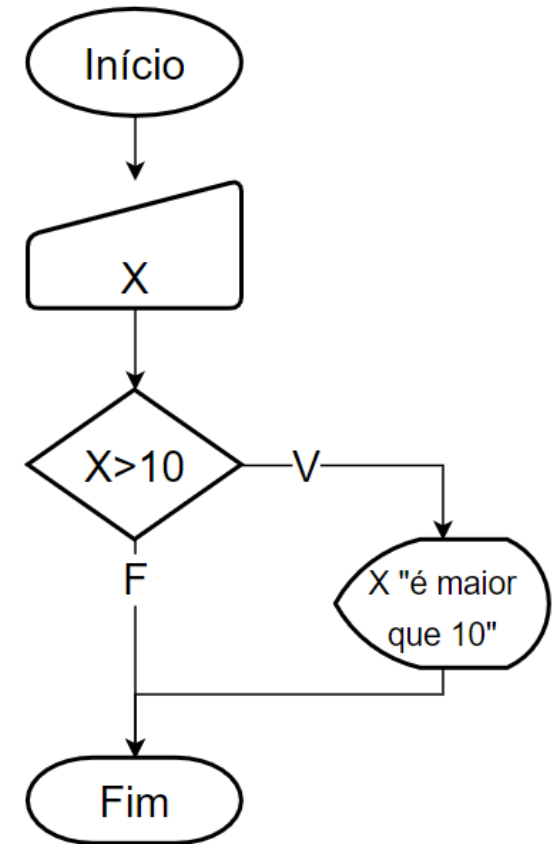
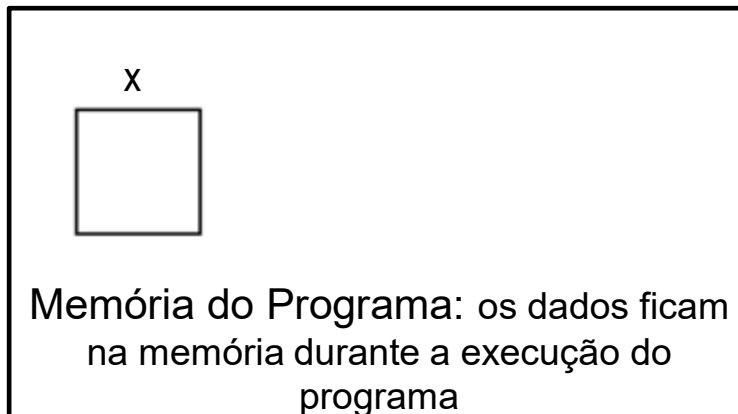
**Vamos fazer um teste de mesa
com a entrada abaixo:**





Desvio Condicional Simples

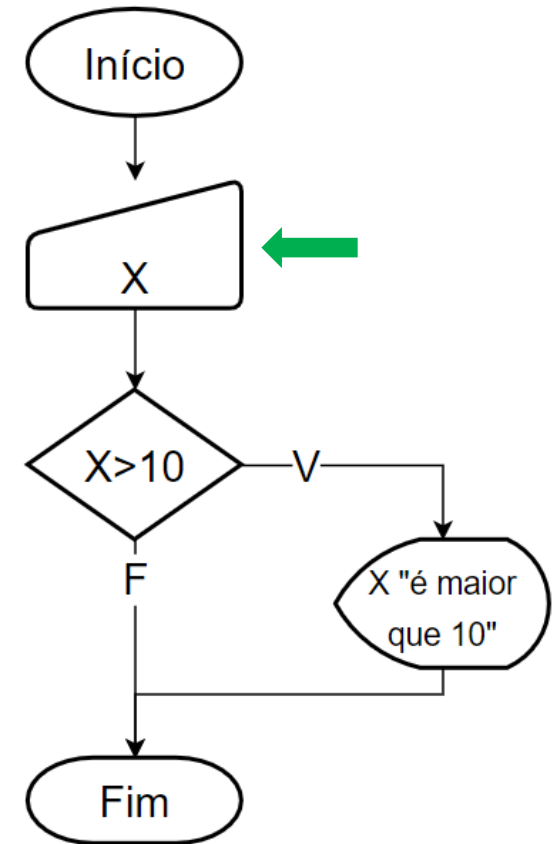
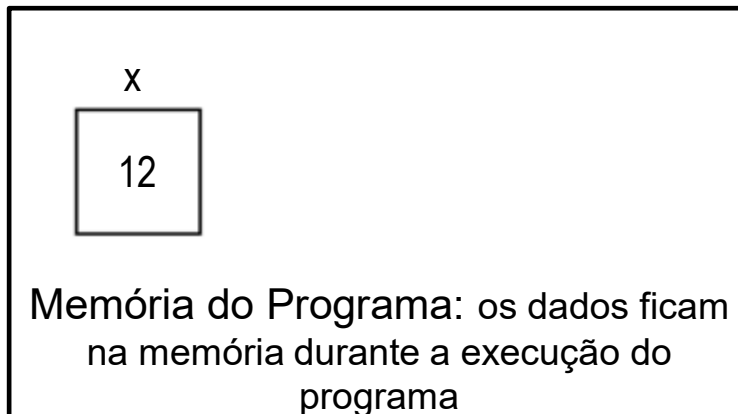
```
public static void Main(string[] args)
{
    → int x;
    x = int.Parse (Console.ReadLine());
    if (x>10)
    {
        Console.WriteLine ("{0} é maior que 10",x);
    }
}
```





Desvio Condicional Simples

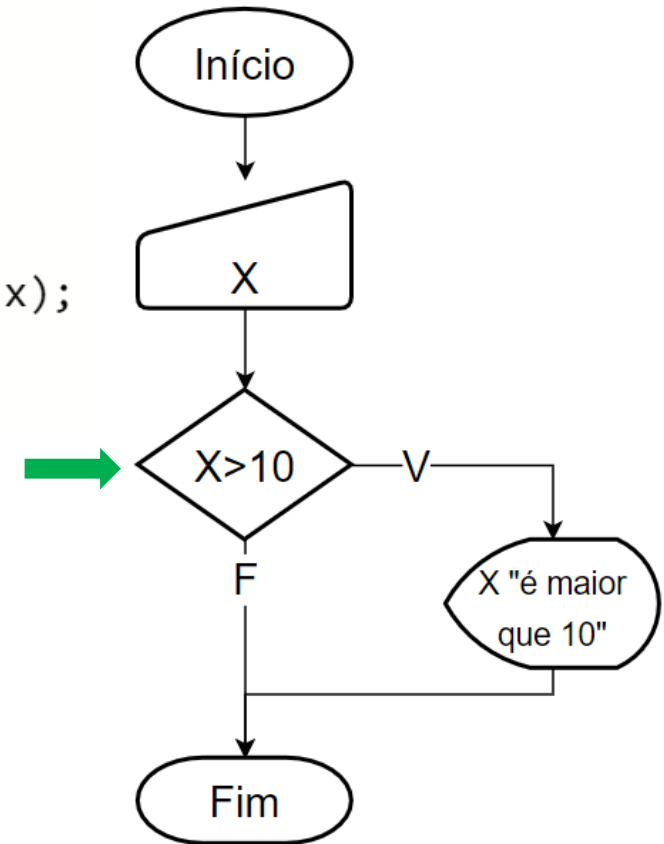
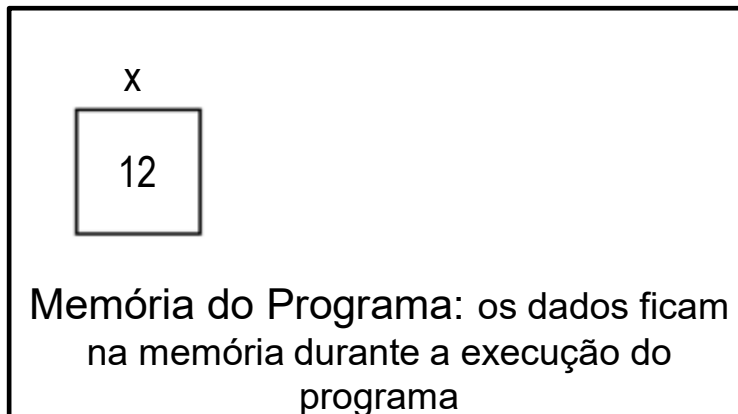
```
public static void Main(string[] args)
{
    int x;
    → x = int.Parse (Console.ReadLine());
    if (x>10)
    {
        Console.WriteLine ("{0} é maior que 10",x);
    }
}
```





Desvio Condicional Simples

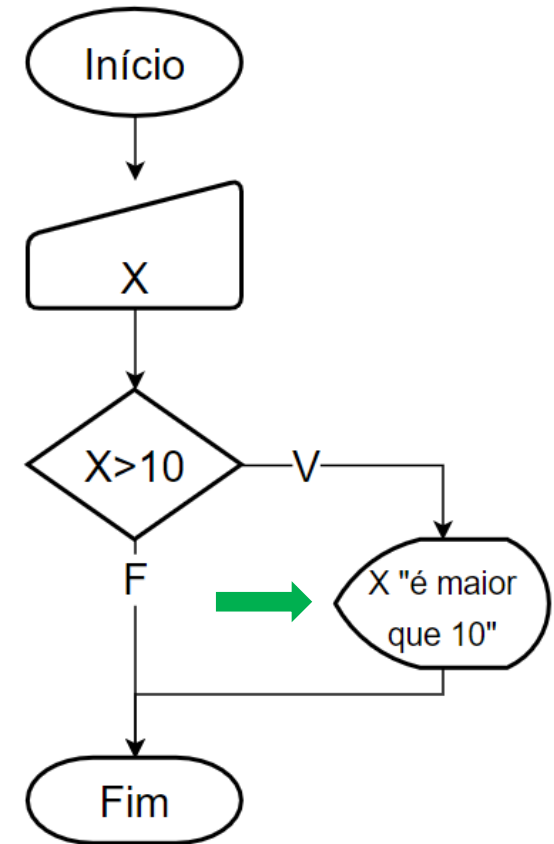
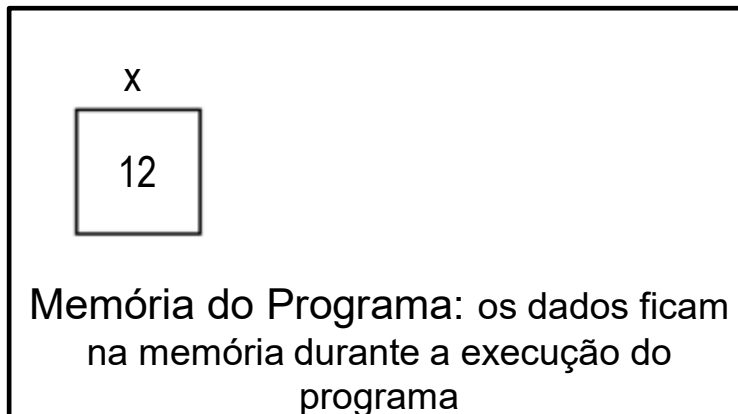
```
public static void Main(string[] args)
{
    int x;
    x = int.Parse (Console.ReadLine());
    → if (x>10) verdadeiro
    {
        Console.WriteLine ("{0} é maior que 10",x);
    }
}
```





Desvio Condicional Simples

```
public static void Main(string[] args)
{
    int x;
    x = int.Parse (Console.ReadLine());
    if (x>10)
    {
        Console.WriteLine ("{0} é maior que 10",x);
    }
}
```

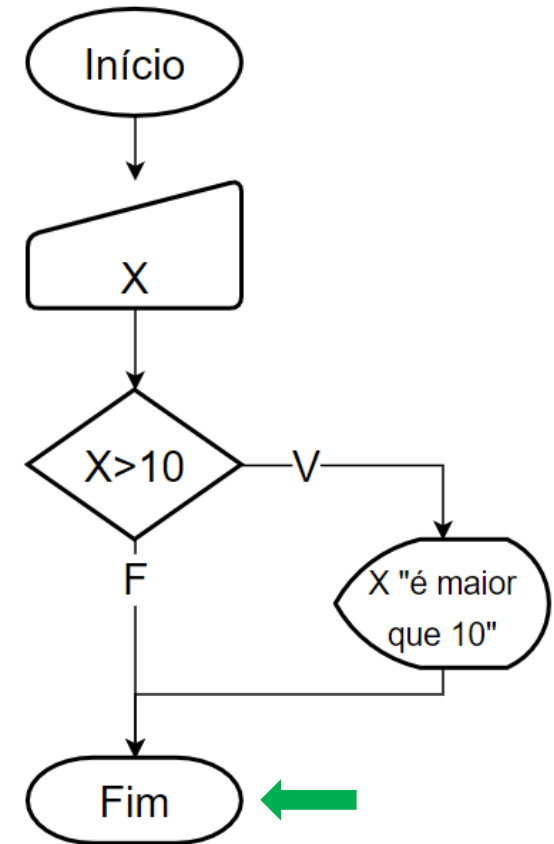
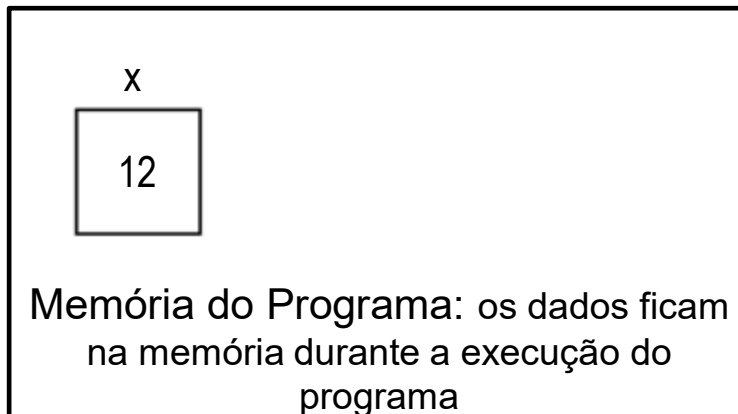




Desvio Condicional Simples

```
public static void Main(string[] args)
{
    int x;
    x = int.Parse (Console.ReadLine());
    if (x>10)
    {
        Console.WriteLine ("{0} é maior que 10",x);
    }
    }

```





Desvio Condicional Simples

- Exemplo: Fazer um algoritmo que receba um número e mostre uma mensagem caso este seja maior que 10.

**Vamos fazer um teste de mesa
com a entrada abaixo:**

3

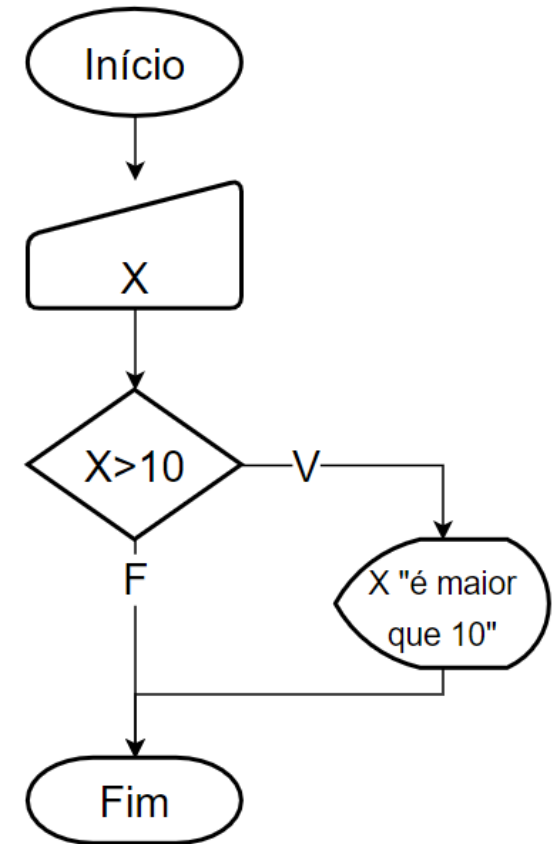
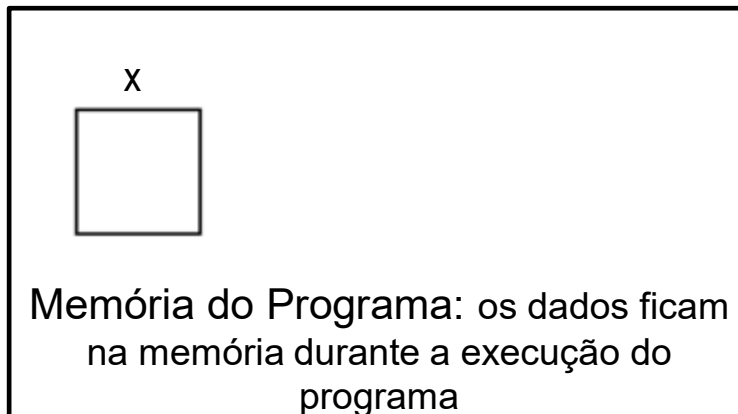


não mostra nada



Desvio Condicional Simples

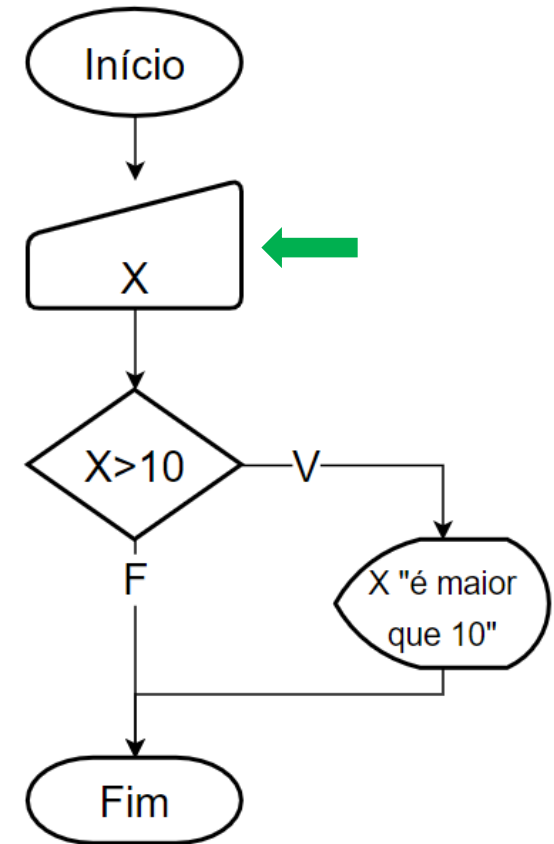
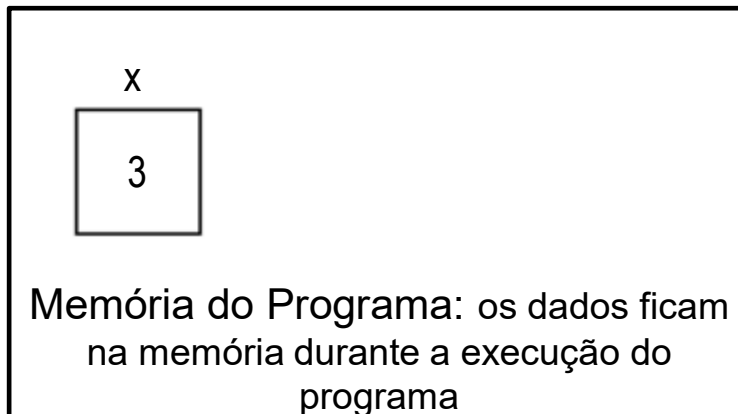
```
public static void Main(string[] args)
{
    → int x;
    x = int.Parse (Console.ReadLine());
    if (x>10)
    {
        Console.WriteLine ("{0} é maior que 10",x);
    }
}
```





Desvio Condicional Simples

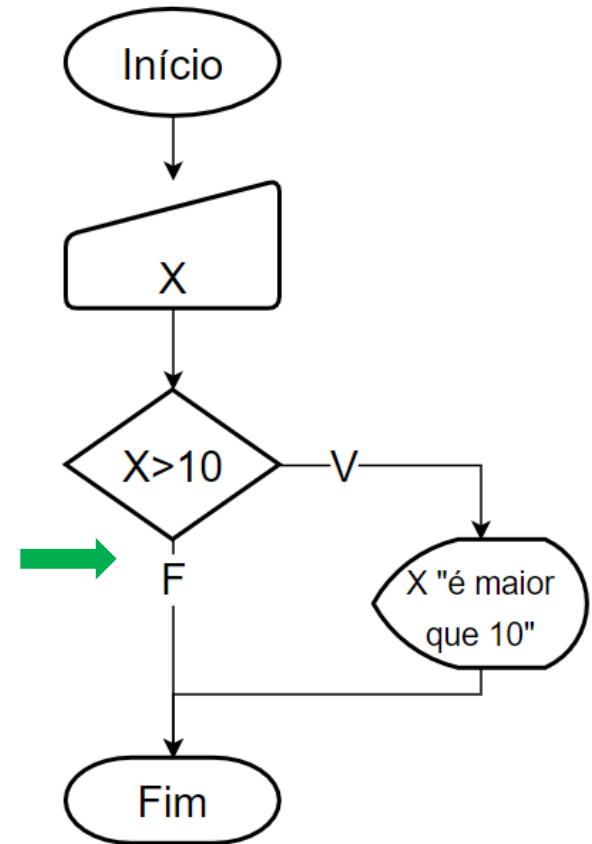
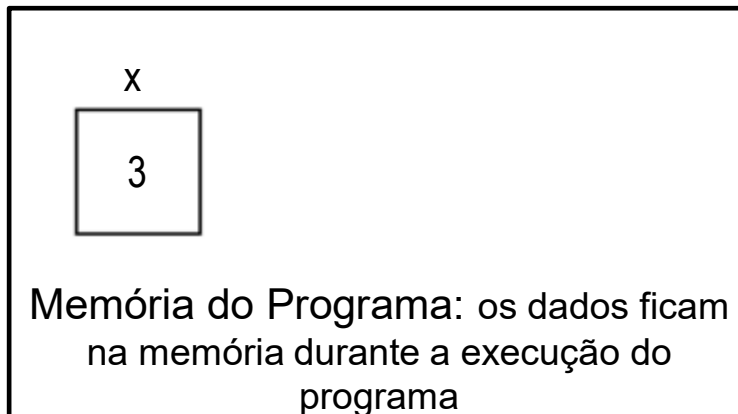
```
public static void Main(string[] args)
{
    int x;
    → x = int.Parse (Console.ReadLine());
    if (x>10)
    {
        Console.WriteLine ("{0} é maior que 10",x);
    }
}
```





Desvio Condicional Simples

```
public static void Main(string[] args)
{
    int x;
    x = int.Parse (Console.ReadLine());
    → if (x>10) falso
    {
        Console.WriteLine ("{0} é maior que 10",x);
    }
}
```

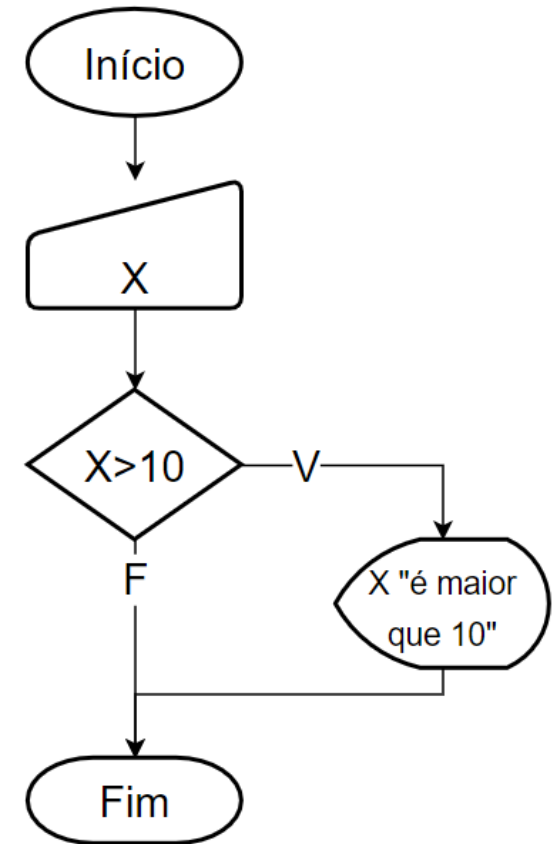
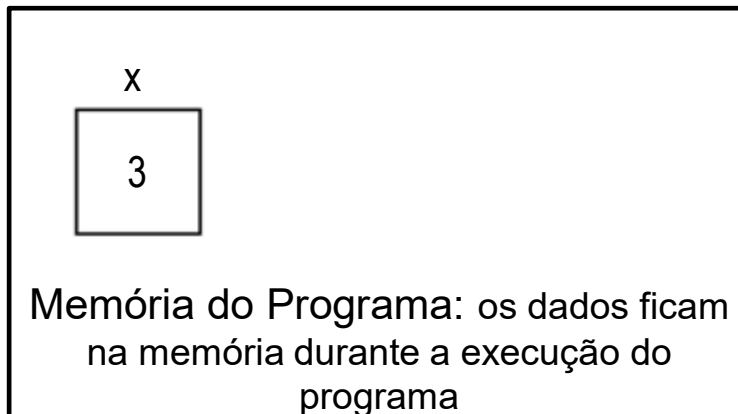




Desvio Condicional Simples

```
public static void Main(string[] args)
{
    int x;
    x = int.Parse (Console.ReadLine());
    if (x>10)
    {
        Console.WriteLine ("{0} é maior que 10",x);
    }
    }

```





Desvio Condicional Composto

- Numa **instrução If... else**, se a condição for verdadeira, será executada a instrução posicionada entre as instruções if e else.
- Sendo a condição falsa, será executada a instrução posicionada logo após a else, como é indicado na seguinte sintaxe:

```
if <(condição)>
{
    <instrução 1 executada quando condição for verdadeira>;
    <instrução 2 executada quando condição for verdadeira>;
    <instrução N executada quando condição for verdadeira>;
}
else
{
    <instrução 1 executada quando condição for falsa>;
    <instrução 2 executada quando condição for falsa>;
    <instrução N executada quando condição for falsa>;
}
<instrução executada após condição ser verdadeira ou falsa>;
```



Desvio Condicional Composto

- Exemplo: Crie um algoritmo que leia dois valores inteiros distintos e informe qual é o maior.

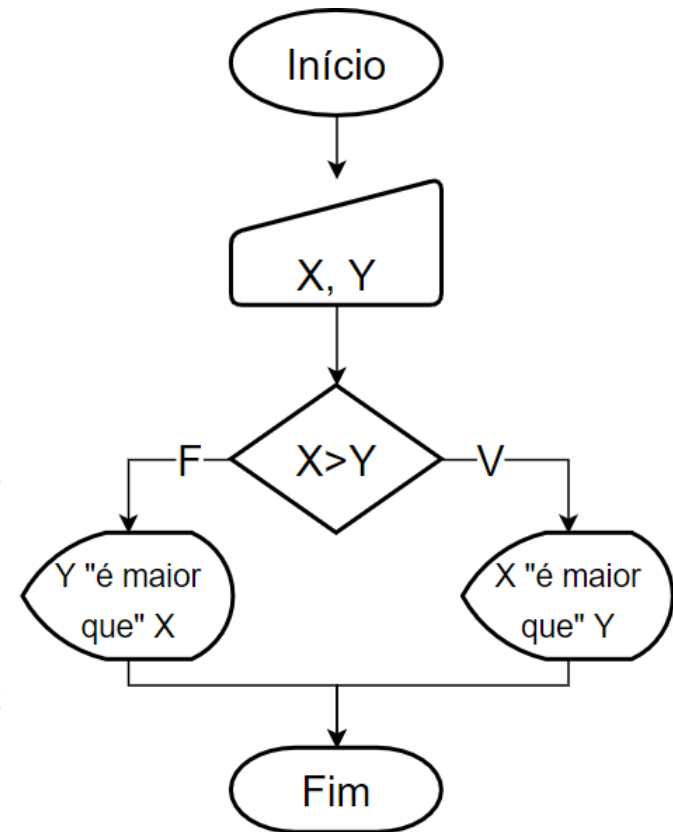




Desvio Condicional Composto

- Exemplo: Crie um algoritmo que leia dois valores inteiros distintos e informe qual é o maior.

```
public static void Main (string[] args)
{
    int x, y;
    x = int.Parse (Console.ReadLine ());
    y = int.Parse (Console.ReadLine ());
    if (x>y)
    {
        Console.WriteLine ("{0} é maior que {1}",x,y);
    }
    else
    {
        Console.WriteLine ("{0} é maior que {1}",y,x);
    }
}
```





Desvio Condicional Composto

- Exemplo: Crie um algoritmo que leia dois valores inteiros distintos e informe qual é o maior.

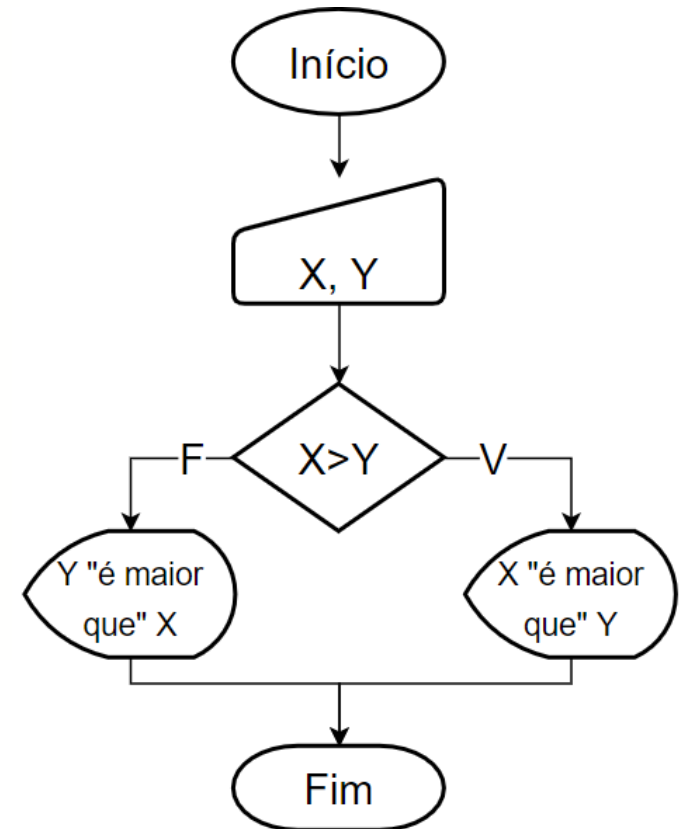
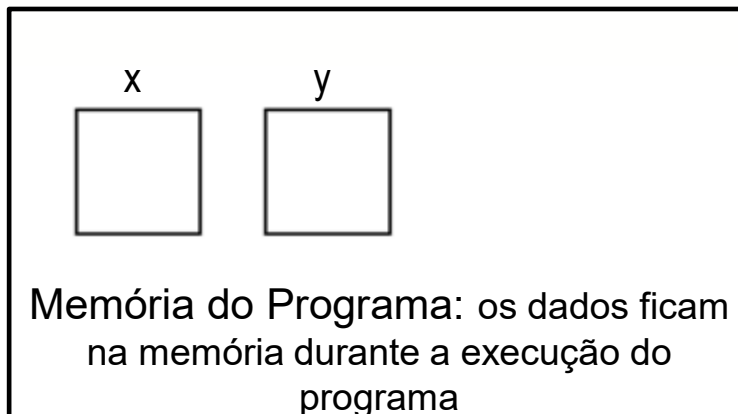
**Vamos fazer um teste de mesa
com a entrada abaixo:**





Desvio Condicional Composto

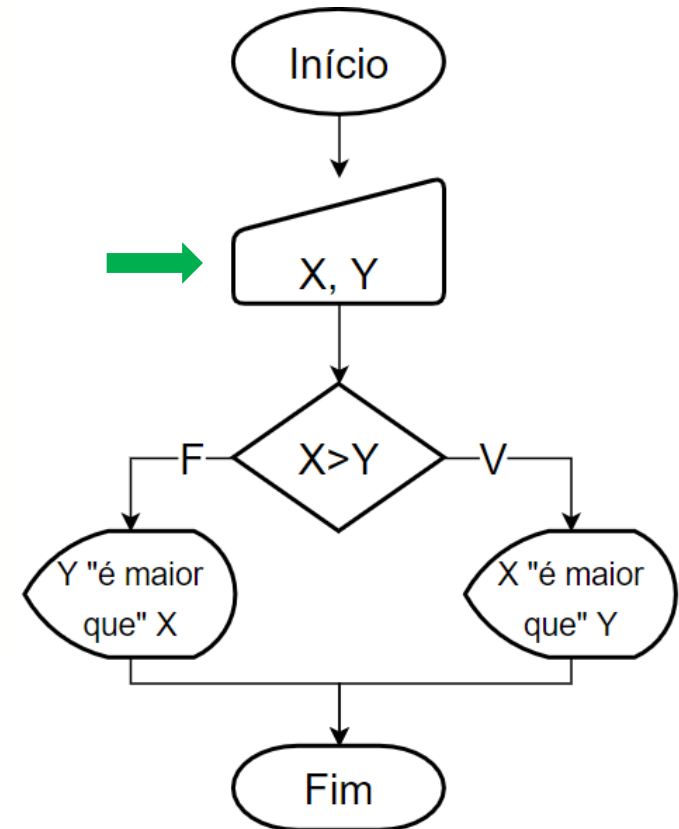
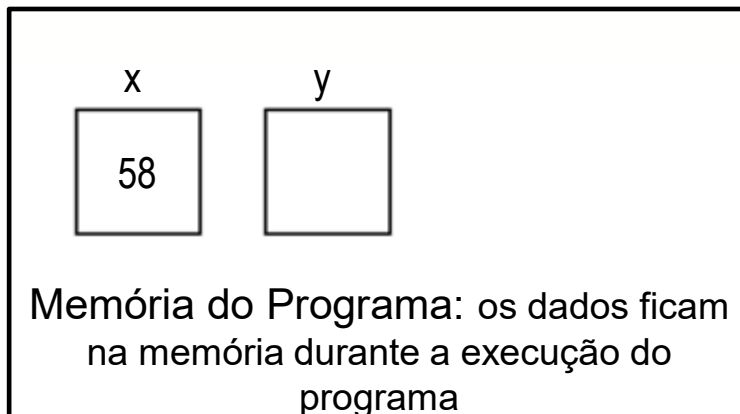
```
public static void Main (string[] args)
{
    int x, y;
    x = int.Parse (Console.ReadLine ());
    y = int.Parse (Console.ReadLine ());
    if (x>y)
    {
        Console.WriteLine ("{0} é maior que {1}",x,y);
    }
    else
    {
        Console.WriteLine ("{0} é maior que {1}",y,x);
    }
}
```





Desvio Condicional Composto

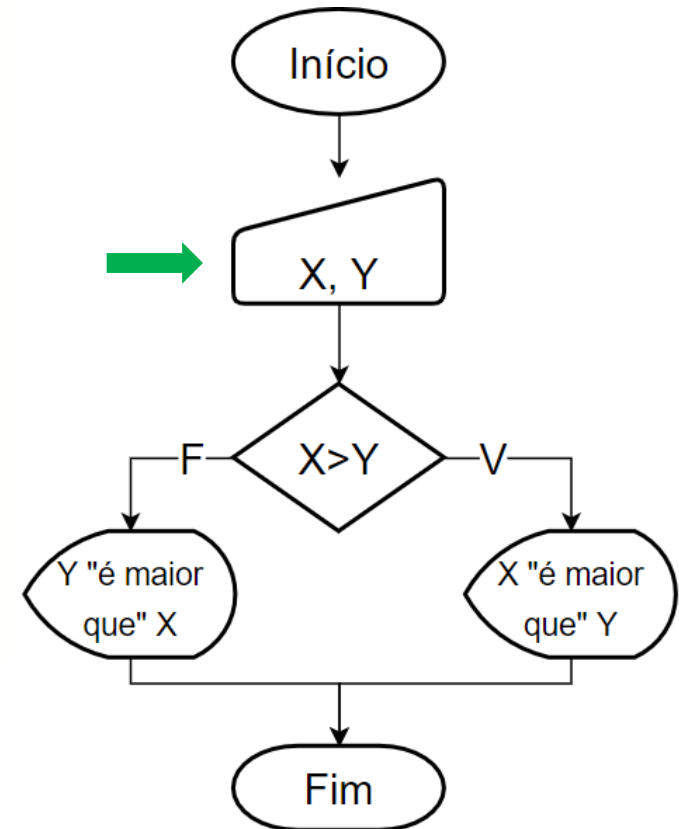
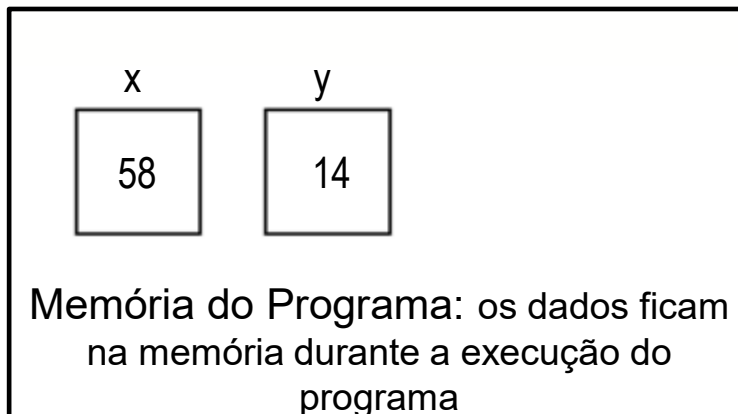
```
public static void Main (string[] args)
{
    int x, y;
    x = int.Parse (Console.ReadLine ());
    y = int.Parse (Console.ReadLine ());
    if (x>y)
    {
        Console.WriteLine ("{0} é maior que {1}",x,y);
    }
    else
    {
        Console.WriteLine ("{0} é maior que {1}",y,x);
    }
}
```





Desvio Condicional Composto

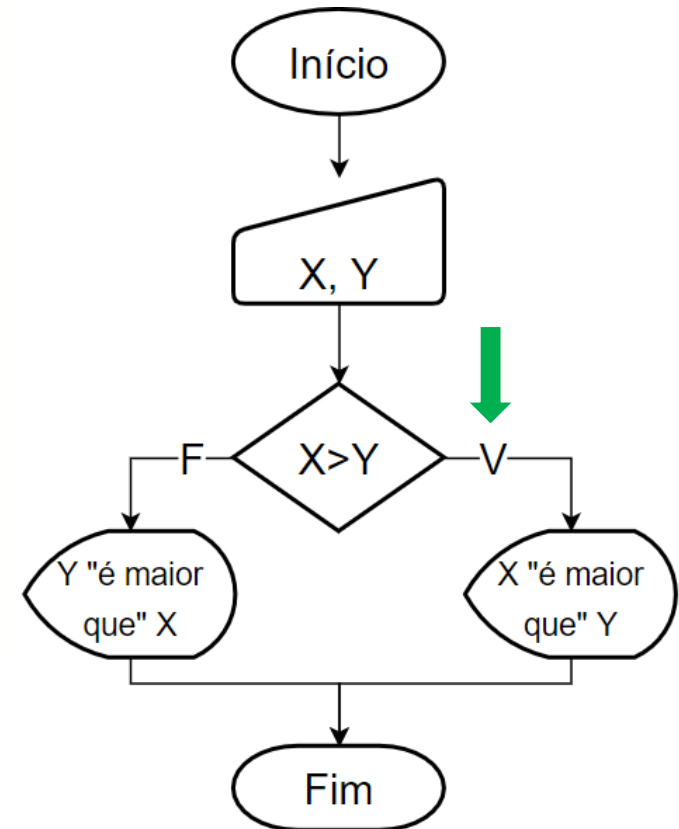
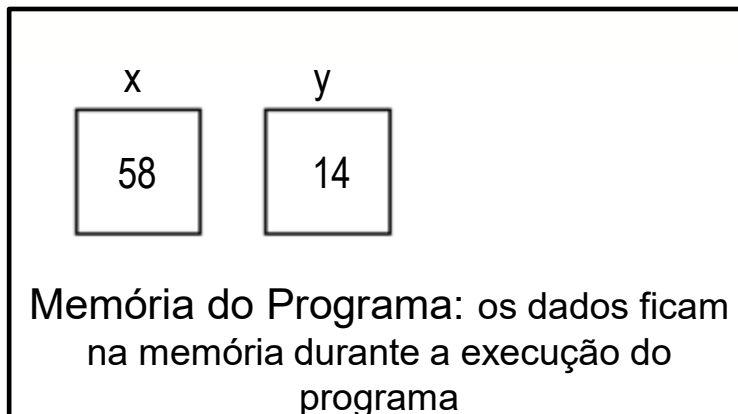
```
public static void Main (string[] args)
{
    int x, y;
    x = int.Parse (Console.ReadLine ());
    y = int.Parse (Console.ReadLine ());
    if (x>y)
    {
        Console.WriteLine ("{0} é maior que {1}",x,y);
    }
    else
    {
        Console.WriteLine ("{0} é maior que {1}",y,x);
    }
}
```





Desvio Condicional Composto

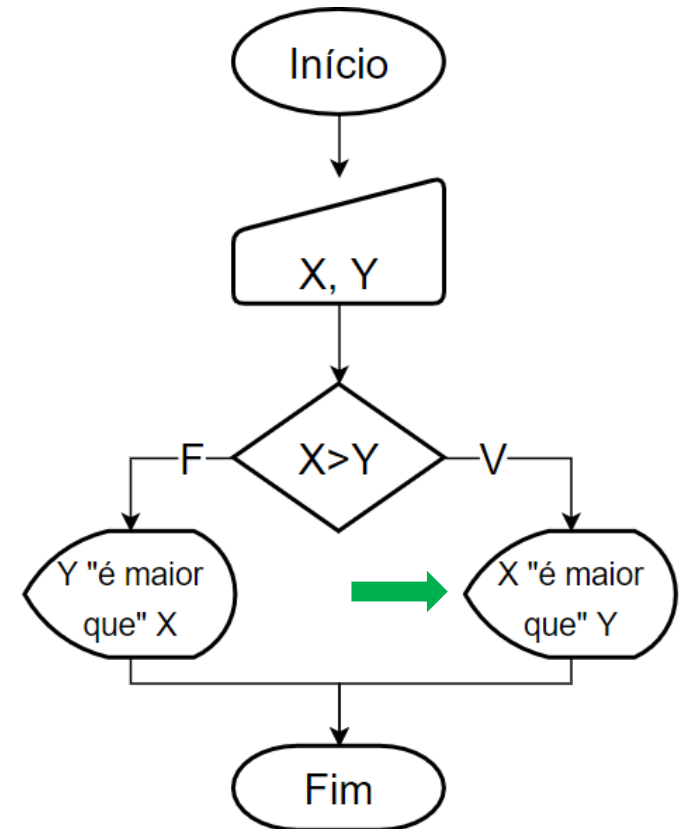
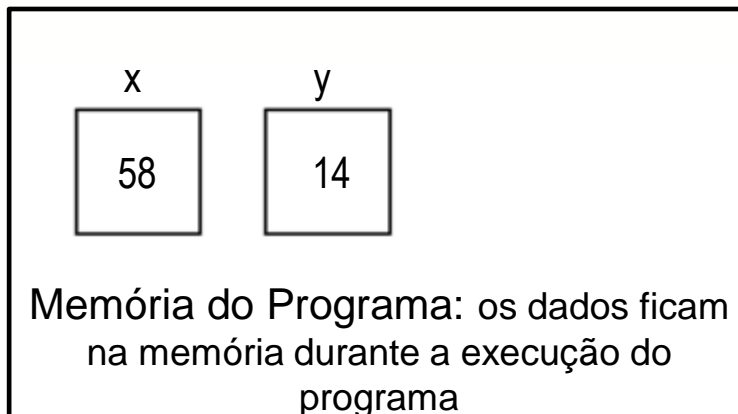
```
public static void Main (string[] args)
{
    int x, y;
    x = int.Parse (Console.ReadLine ());
    y = int.Parse (Console.ReadLine ());
    if (x>y) verdadeiro
    {
        Console.WriteLine ("{0} é maior que {1}",x,y);
    }
    else
    {
        Console.WriteLine ("{0} é maior que {1}",y,x);
    }
}
```





Desvio Condicional Composto

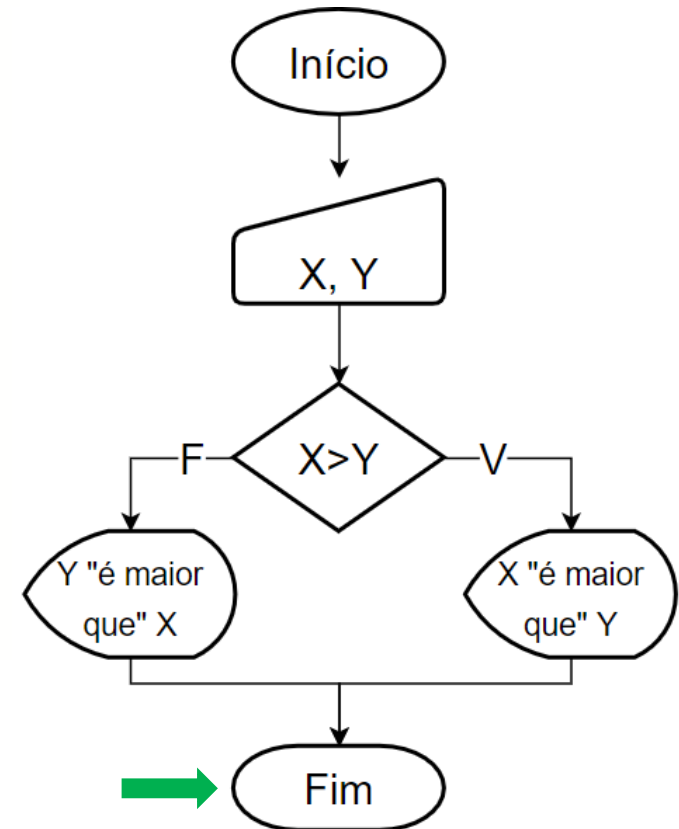
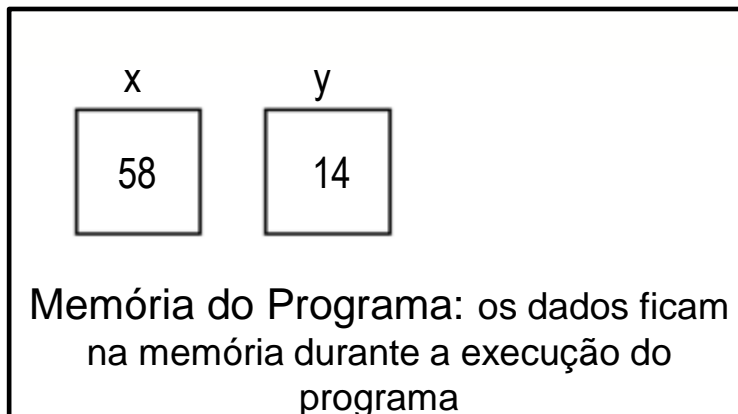
```
public static void Main (string[] args)
{
    int x, y;
    x = int.Parse (Console.ReadLine ());
    y = int.Parse (Console.ReadLine ());
    if (x>y)
    {
        Console.WriteLine ("{0} é maior que {1}",x,y);
    }
    else
    {
        Console.WriteLine ("{0} é maior que {1}",y,x);
    }
}
```





Desvio Condicional Composto

```
public static void Main (string[] args)
{
    int x, y;
    x = int.Parse (Console.ReadLine ());
    y = int.Parse (Console.ReadLine ());
    if (x>y)
    {
        Console.WriteLine ("{0} é maior que {1}",x,y);
    }
    else
    {
        Console.WriteLine ("{0} é maior que {1}",y,x);
    }
}
```





Desvio Condicional Composto

- Exemplo: Crie um algoritmo que leia dois valores inteiros distintos e informe qual é o maior.

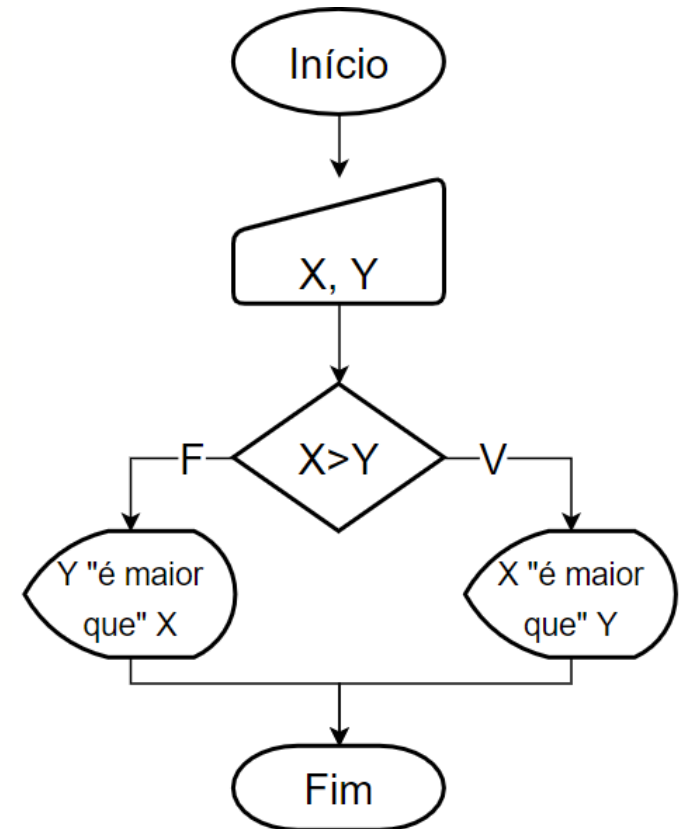
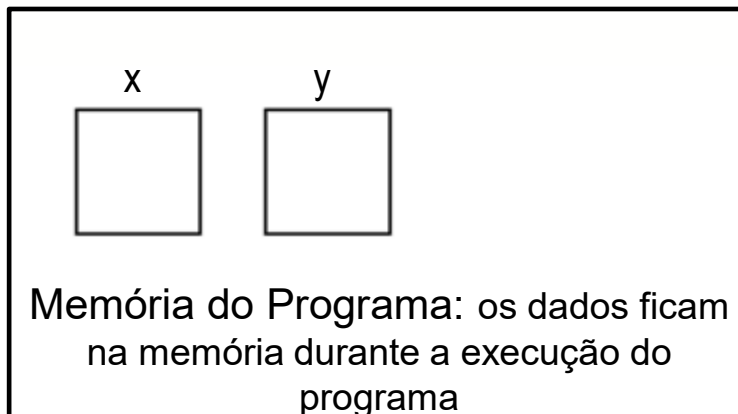
**Vamos fazer um teste de mesa
com a entrada abaixo:**





Desvio Condicional Composto

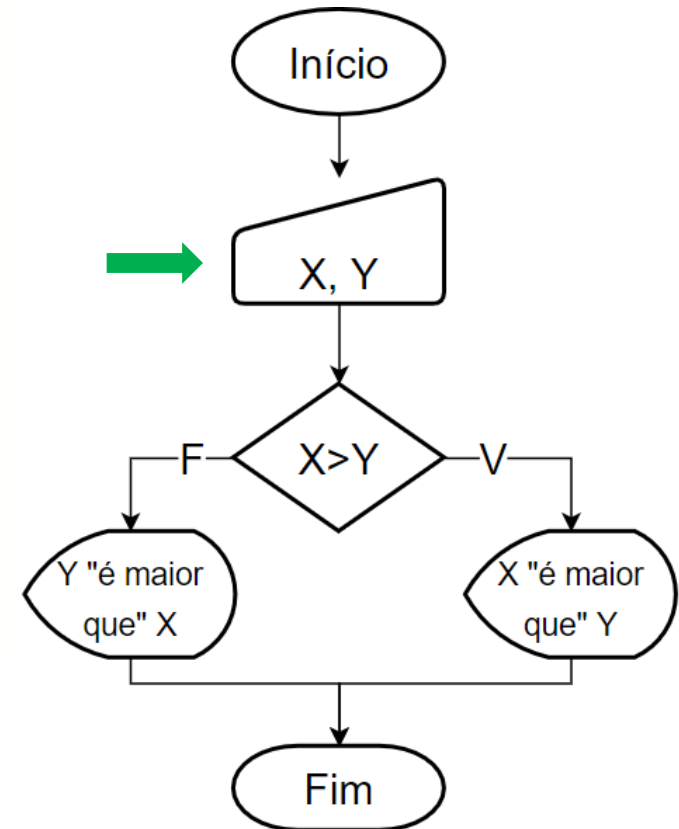
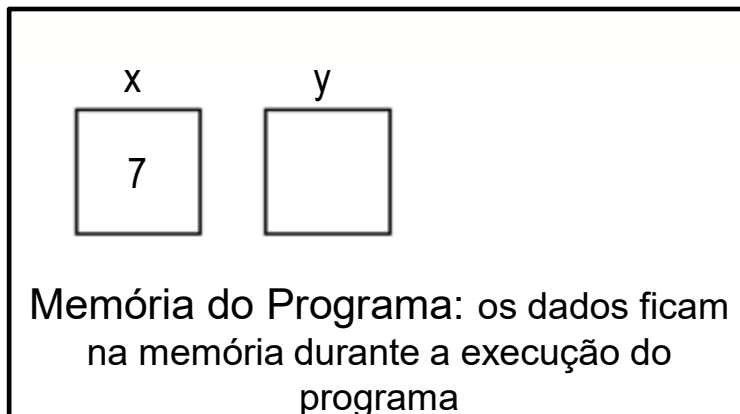
```
public static void Main (string[] args)
{
    int x, y;
    x = int.Parse (Console.ReadLine ());
    y = int.Parse (Console.ReadLine ());
    if (x>y)
    {
        Console.WriteLine ("{0} é maior que {1}",x,y);
    }
    else
    {
        Console.WriteLine ("{0} é maior que {1}",y,x);
    }
}
```





Desvio Condicional Composto

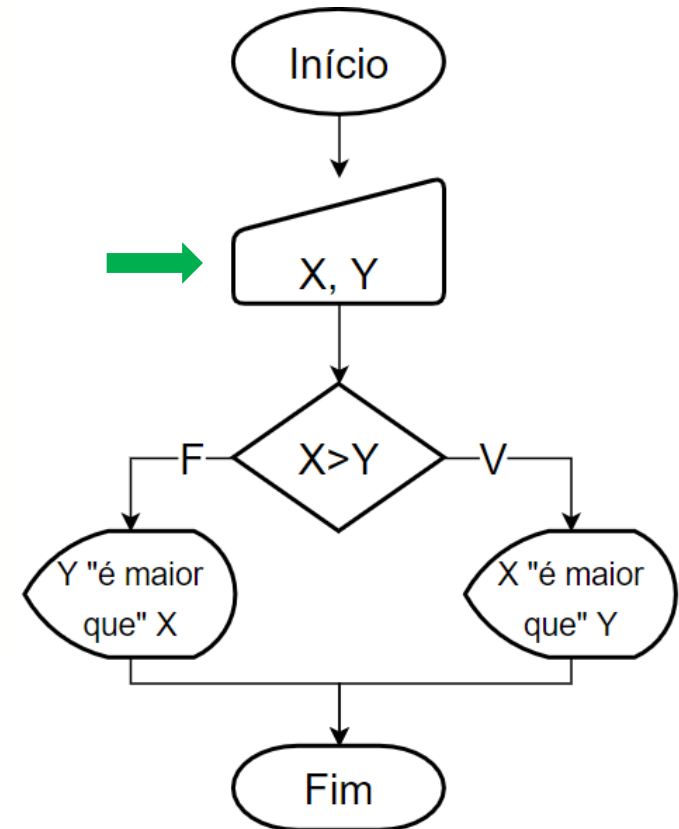
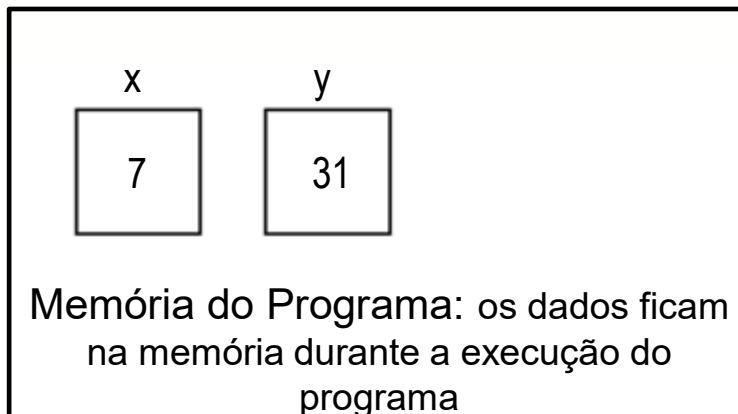
```
public static void Main (string[] args)
{
    int x, y;
    x = int.Parse (Console.ReadLine ());
    y = int.Parse (Console.ReadLine ());
    if (x>y)
    {
        Console.WriteLine ("{0} é maior que {1}",x,y);
    }
    else
    {
        Console.WriteLine ("{0} é maior que {1}",y,x);
    }
}
```





Desvio Condicional Composto

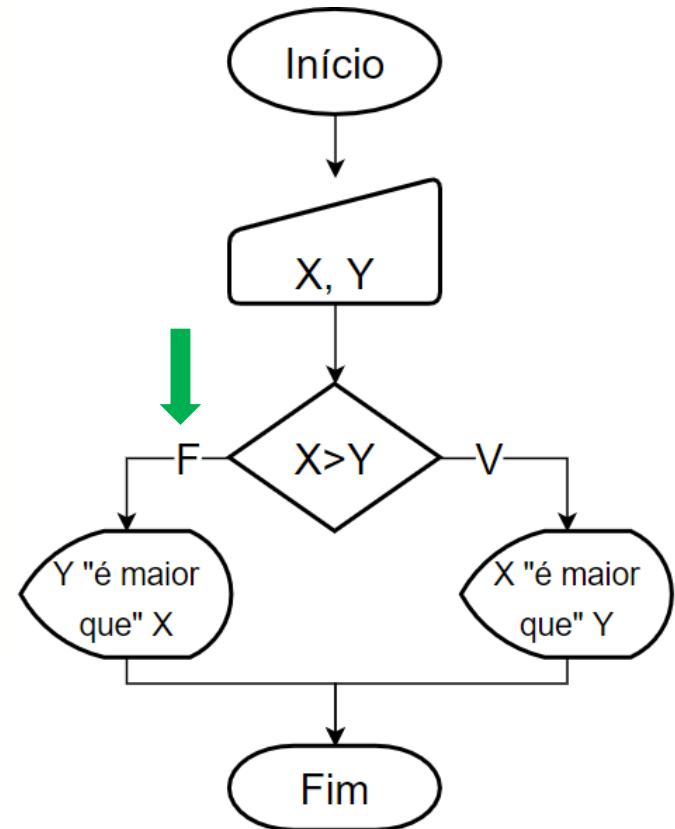
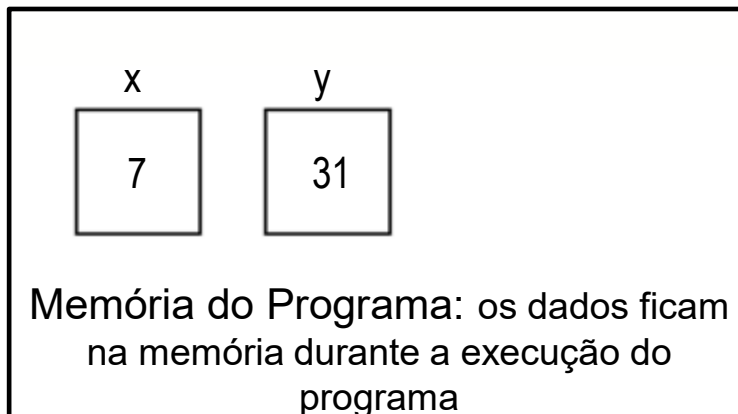
```
public static void Main (string[] args)
{
    int x, y;
    x = int.Parse (Console.ReadLine ());
    y = int.Parse (Console.ReadLine ());
    if (x>y)
    {
        Console.WriteLine ("{0} é maior que {1}",x,y);
    }
    else
    {
        Console.WriteLine ("{0} é maior que {1}",y,x);
    }
}
```





Desvio Condicional Composto

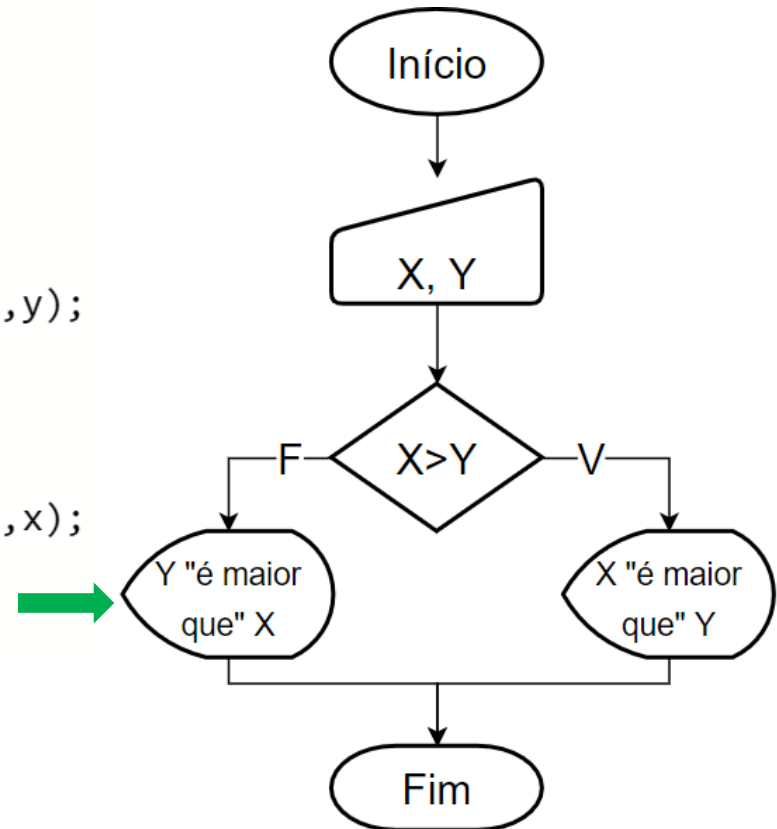
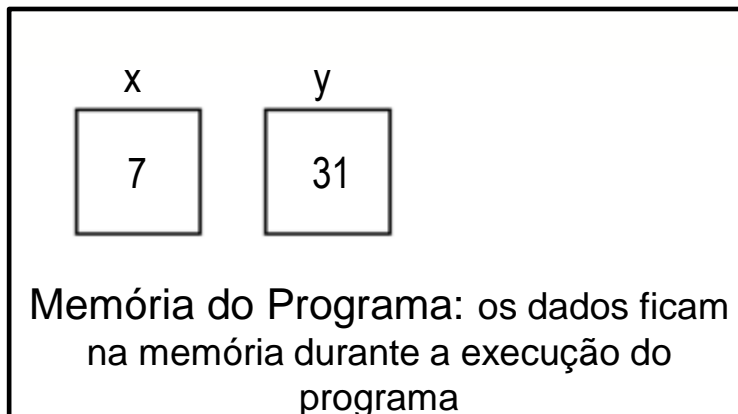
```
public static void Main (string[] args)
{
    int x, y;
    x = int.Parse (Console.ReadLine ());
    y = int.Parse (Console.ReadLine ());
    if (x>y) falso
    {
        Console.WriteLine ("{0} é maior que {1}",x,y);
    }
    else
    {
        Console.WriteLine ("{0} é maior que {1}",y,x);
    }
}
```





Desvio Condicional Composto

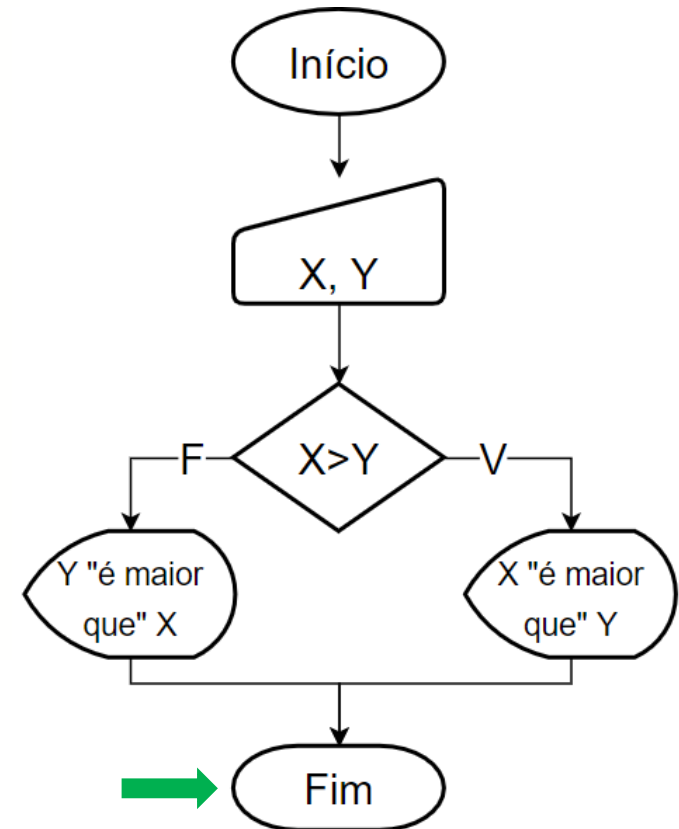
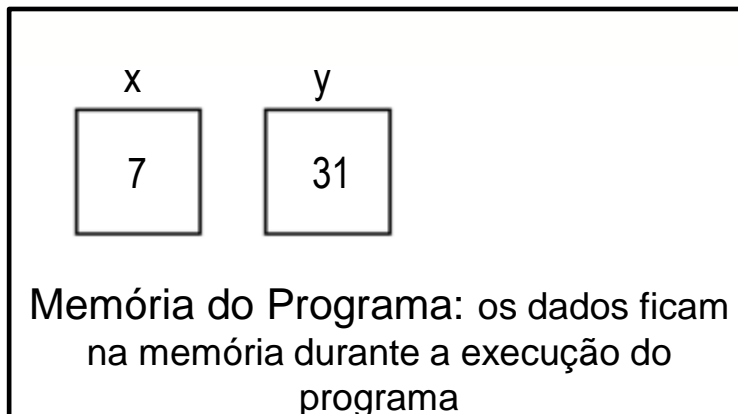
```
public static void Main (string[] args)
{
    int x, y;
    x = int.Parse (Console.ReadLine ());
    y = int.Parse (Console.ReadLine ());
    if (x>y)
    {
        Console.WriteLine ("{0} é maior que {1}",x,y);
    }
    else
    {
        Console.WriteLine ("{0} é maior que {1}",y,x);
    }
}
```





Desvio Condicional Composto

```
public static void Main (string[] args)
{
    int x, y;
    x = int.Parse (Console.ReadLine ());
    y = int.Parse (Console.ReadLine ());
    if (x>y)
    {
        Console.WriteLine ("{0} é maior que {1}",x,y);
    }
    else
    {
        Console.WriteLine ("{0} é maior que {1}",y,x);
    }
}
```





Desvio Condicional Seletivo

- Ao trabalharmos com grande quantidade de desvios condicionais, poderemos ter um programa de difícil interpretação, e para auxiliar e facilitar esse tipo de necessidade temos a **estrutura de controle com múltipla escolha**, denominada switch, cuja sintaxe é:

```
switch <variável>
{
    case <opção 1>: <operação 1>;      break;
    case <opção 2>: <operação 2>;      break;
    case <opção N>: <operação N>;      break;
    default      : <operação default>; break;
}
```

<variável> : nome da variável a ser controlada na decisão;

<opção> : conteúdo da variável a ser verificado;

<operação>: execução de alguma ação específica.



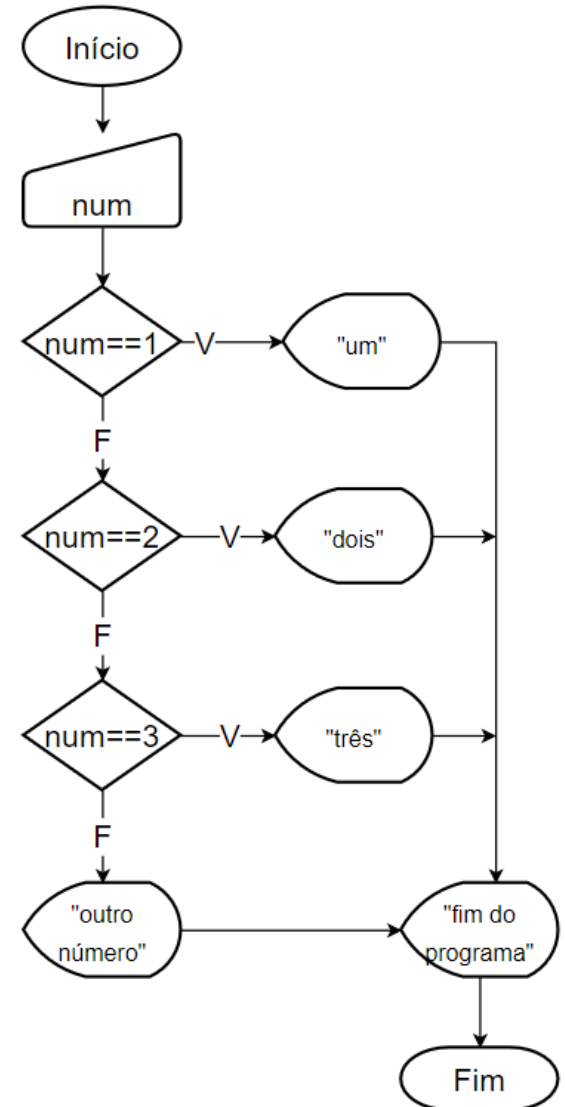
Desvio Condicional Seletivo

- A **instrução default da estrutura de seleção switch e qualquer operação a ela associada são opcionais**, não caracterizando obrigatoriedade de uso.
- **A instrução break tem a finalidade de desviar o processamento para fora do comando switch**. Isso é necessário porque após o processamento de cada case ocorre retorno do fluxo de ação do comando para a primeira instrução após sua chamada, que nesse caso será break.
- A variável usada numa estrutura switch para controlar a ação desse tipo de desvio na linguagem C# pode operar com **variáveis do tipo inteiro ou cadeia**.



Desvio Condicional Seletivo

```
public static void Main (string[] args)
{
    int num;
    num = int.Parse (Console.ReadLine());
    switch (num)
    {
        case 1:
            Console.WriteLine ("um");
            break;
        case 2:
            Console.WriteLine ("dois");
            break;
        case 3:
            Console.WriteLine ("três");
            break;
        default:
            Console.WriteLine ("outro numero");
            break;
    }
    Console.WriteLine ("fim do programa");
}
```





Desvio Condicional Composto

**Vamos fazer um teste de mesa
com a entrada abaixo:**

2

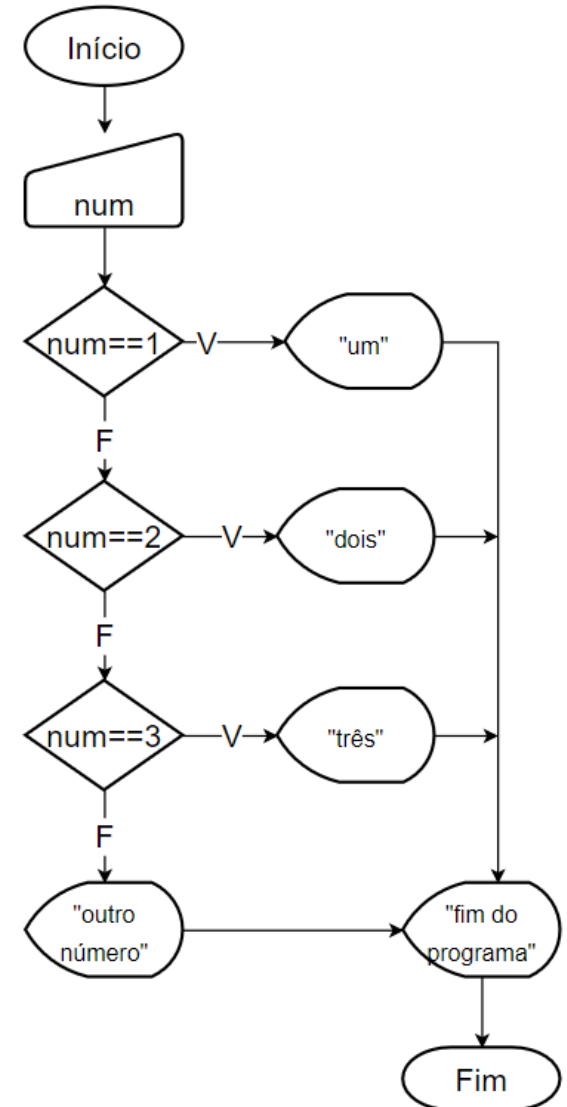


dois



Desvio Condicional Seletivo

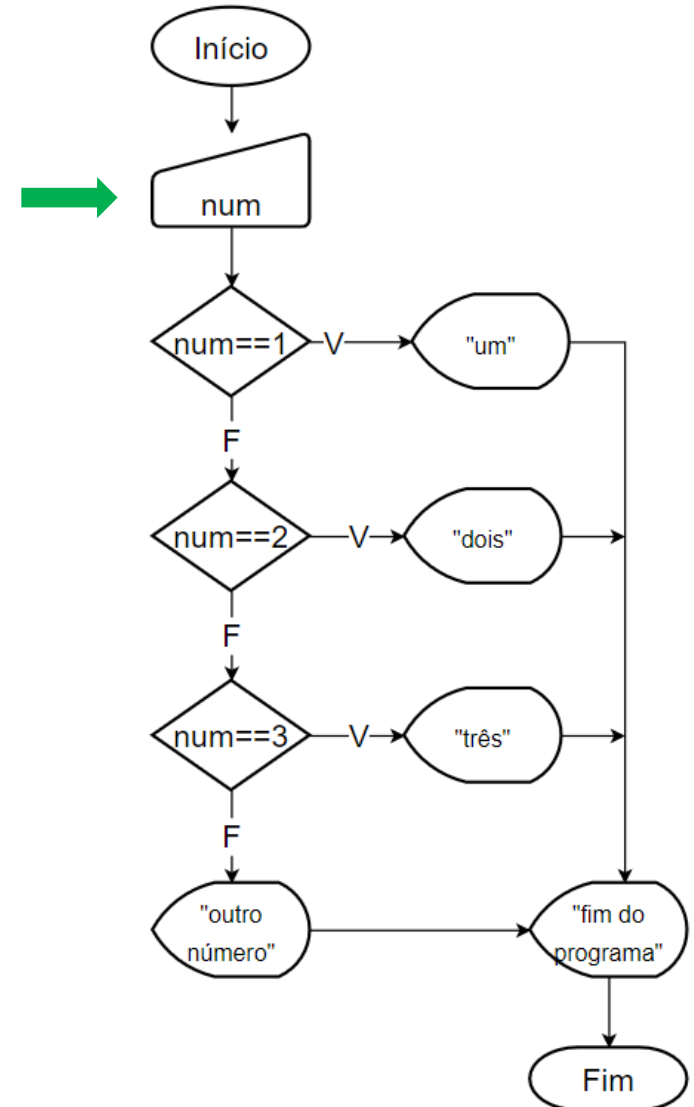
```
public static void Main (string[] args)
{
    int num;
    num = int.Parse (Console.ReadLine());
    switch (num)
    {
        case 1:
            Console.WriteLine ("um");
            break;
        case 2:
            Console.WriteLine ("dois");
            break;
        case 3:
            Console.WriteLine ("três");
            break;
        default:
            Console.WriteLine ("outro numero");
            break;
    }
    Console.WriteLine ("fim do programa");
}
```





Desvio Condicional Seletivo

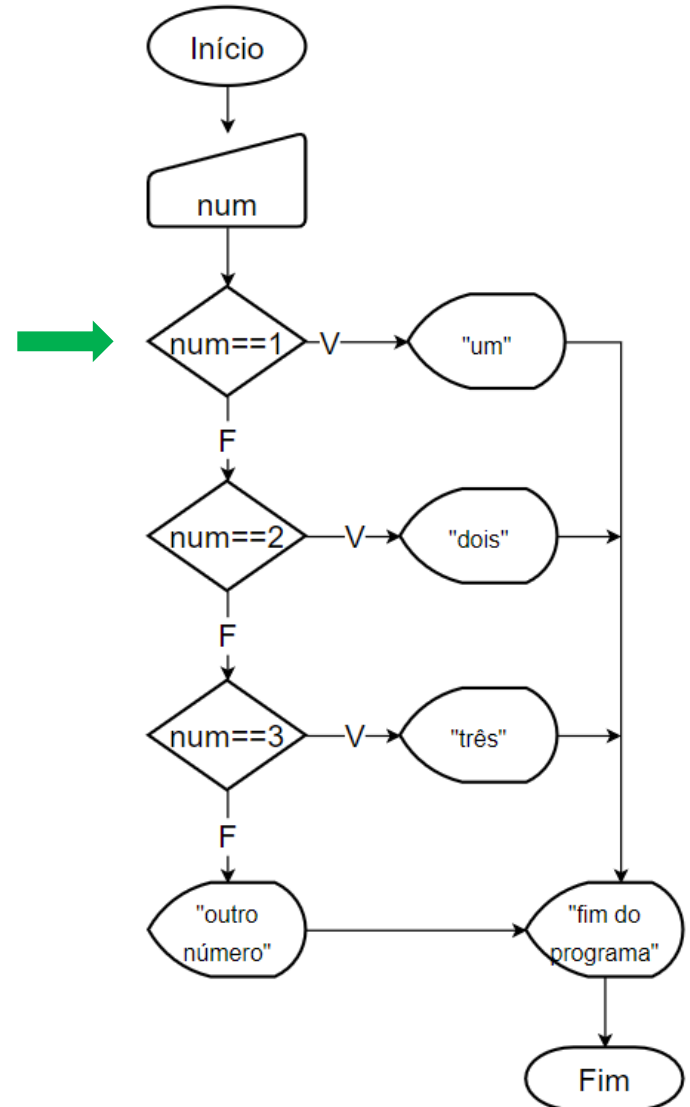
```
public static void Main (string[] args)
{
    int num;
    num = int.Parse (Console.ReadLine());
    switch (num)
    {
        case 1:
            Console.WriteLine ("um");
            break;
        case 2:
            Console.WriteLine ("dois");
            break;
        case 3:
            Console.WriteLine ("três");
            break;
        default:
            Console.WriteLine ("outro numero");
            break;
    }
    Console.WriteLine ("fim do programa");
}
```





Desvio Condicional Seletivo

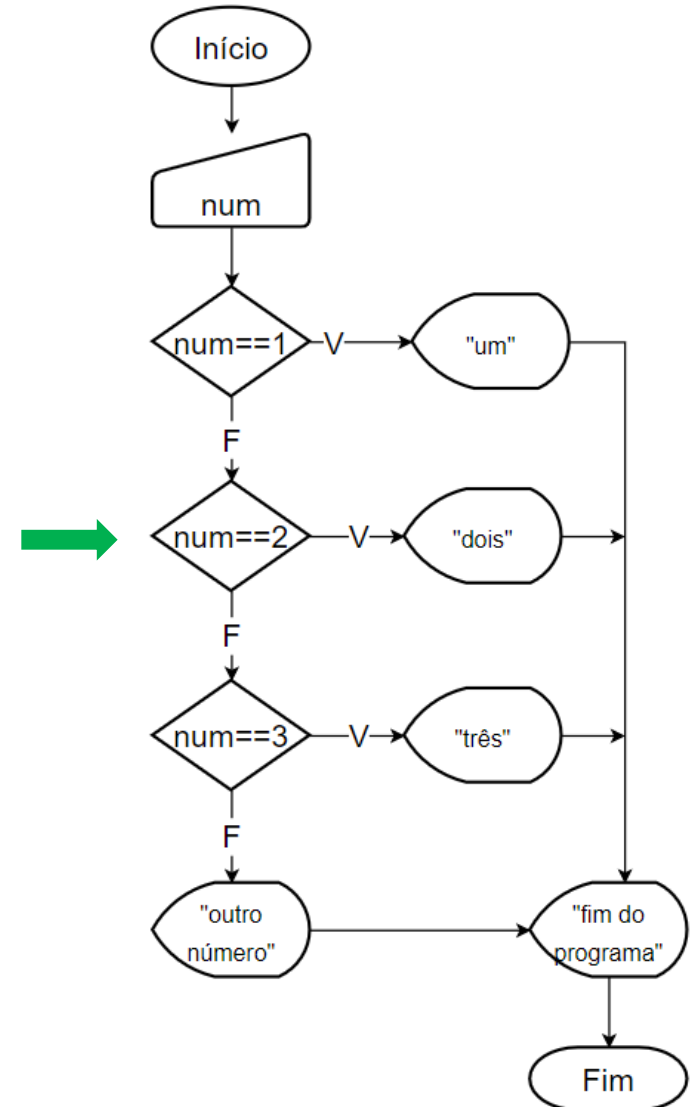
```
public static void Main (string[] args)
{
    int num;
    num = int.Parse (Console.ReadLine());
    switch (num)
    {
        case 1:
            Console.WriteLine ("um");
            break;
        case 2:
            Console.WriteLine ("dois");
            break;
        case 3:
            Console.WriteLine ("três");
            break;
        default:
            Console.WriteLine ("outro numero");
            break;
    }
    Console.WriteLine ("fim do programa");
}
```





Desvio Condicional Seletivo

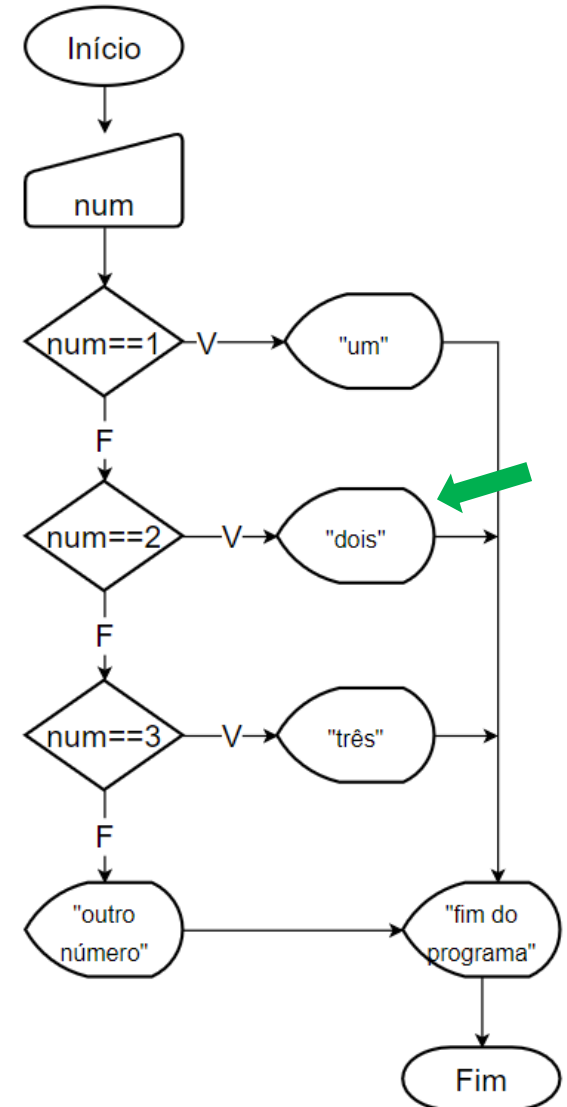
```
public static void Main (string[] args)
{
    int num;
    num = int.Parse (Console.ReadLine());
    switch (num)
    {
        case 1:
            Console.WriteLine ("um");
            break;
        case 2:
            Console.WriteLine ("dois");
            break;
        case 3:
            Console.WriteLine ("três");
            break;
        default:
            Console.WriteLine ("outro numero");
            break;
    }
    Console.WriteLine ("fim do programa");
}
```





Desvio Condicional Seletivo

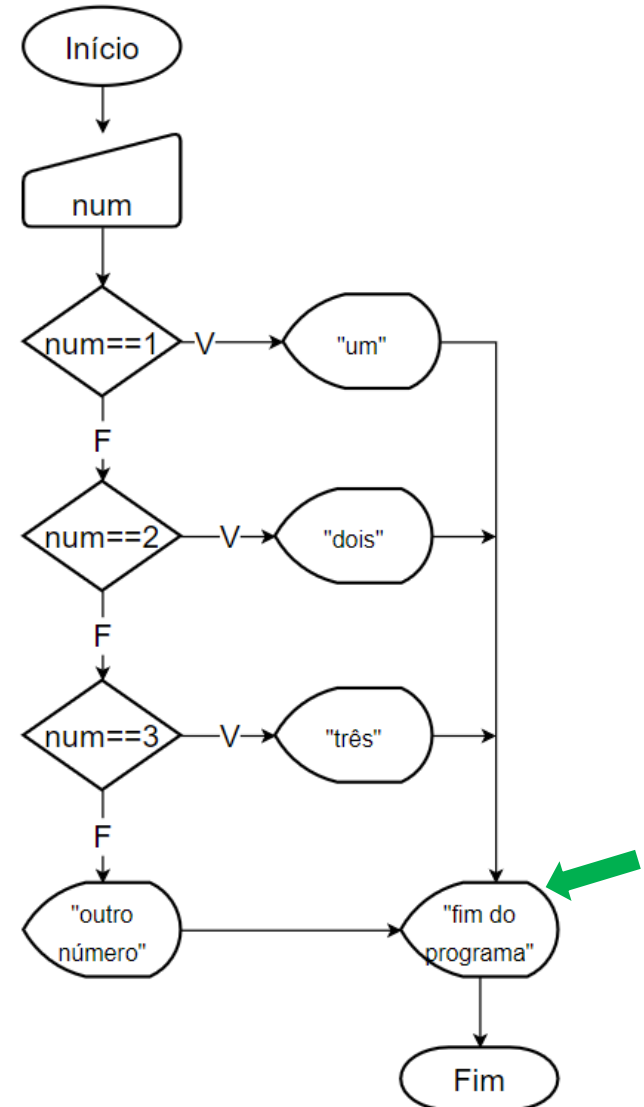
```
public static void Main (string[] args)
{
    int num;
    num = int.Parse (Console.ReadLine());
    switch (num)
    {
        case 1:
            Console.WriteLine ("um");
            break;
        case 2:
            Console.WriteLine ("dois");
            break;
        case 3:
            Console.WriteLine ("três");
            break;
        default:
            Console.WriteLine ("outro numero");
            break;
    }
    Console.WriteLine ("fim do programa");
}
```





Desvio Condicional Seletivo

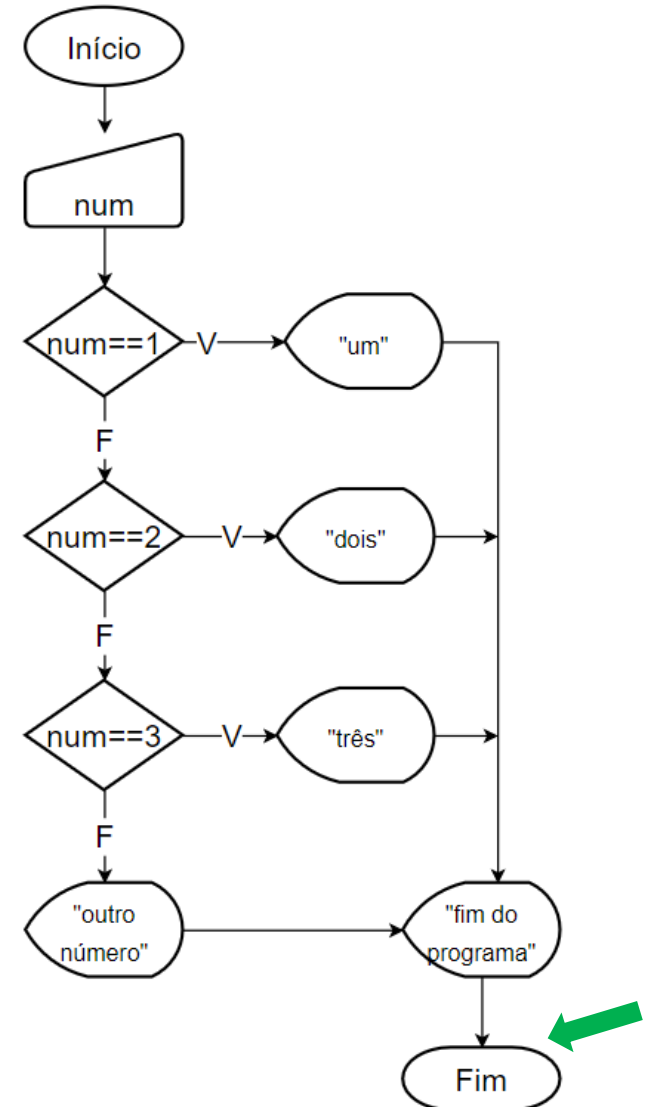
```
public static void Main (string[] args)
{
    int num;
    num = int.Parse (Console.ReadLine());
    switch (num)
    {
        case 1:
            Console.WriteLine ("um");
            break;
        case 2:
            Console.WriteLine ("dois");
            break;
        case 3:
            Console.WriteLine ("três");
            break;
        default:
            Console.WriteLine ("outro numero");
            break;
    }
    Console.WriteLine ("fim do programa");
}
```





Desvio Condicional Seletivo

```
public static void Main (string[] args)
{
    int num;
    num = int.Parse (Console.ReadLine());
    switch (num)
    {
        case 1:
            Console.WriteLine ("um");
            break;
        case 2:
            Console.WriteLine ("dois");
            break;
        case 3:
            Console.WriteLine ("três");
            break;
        default:
            Console.WriteLine ("outro numero");
            break;
    }
    Console.WriteLine ("fim do programa");
}
```





Operadores Lógicos

- Os operadores lógicos && e || permitem mais de uma condição para a tomada de uma única decisão.
- Já o operador logico ! tem por finalidade a negação do estado lógico de uma condição.

Operador Lógico	Descrição
&&	AND = E
	OR = Ou
!	NOT = Não



Operadores Lógicos

- Tabela Verdade do E

A	B	A & B
V	V	V
V	F	F
F	V	F
F	F	F

- Tabela Verdade do OU

A	B	A B
V	V	V
V	F	V
F	V	V
F	F	F

- Tabela Verdade do NOT

A	!A
V	F
F	V



Divisibilidade

- Quando o resto de uma divisão de números naturais é igual a zero, tem-se divisibilidade, ou seja, resultado de divisão exata.
- A linguagem C# possui como operador aritmético para auxiliar o cálculo de divisibilidade o símbolo % (porcentagem).
- Usamos esse operador aritmético quando necessitamos calcular o valor do resto de uma divisão de valores inteiros
- Exemplo:

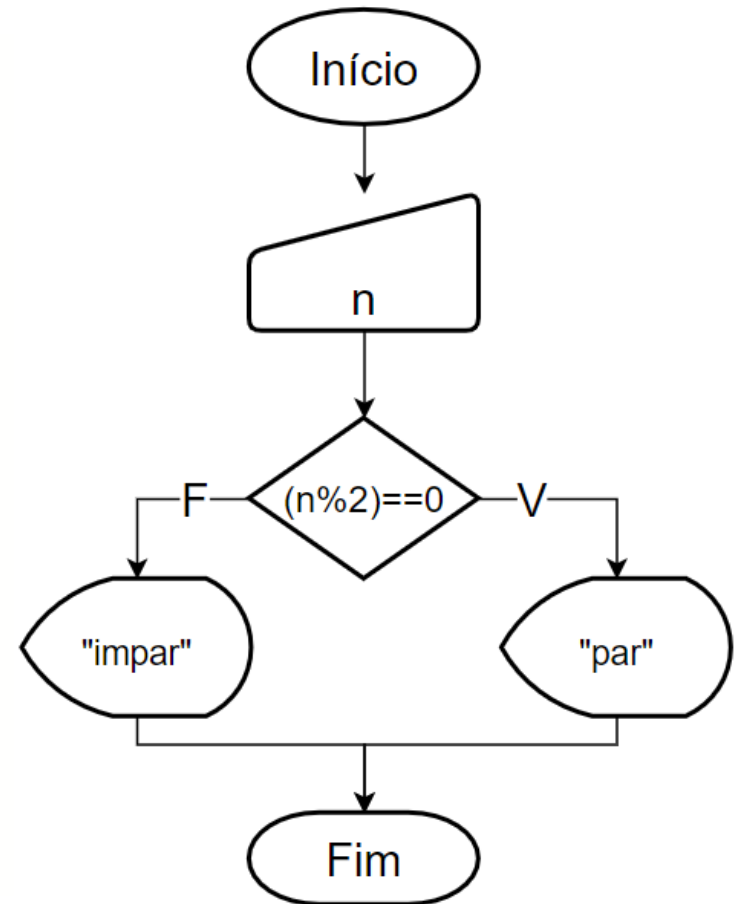
```
X = 4%2;    // x é zero  
Y = 5%3;    // y é 2  
z = 6%3;    // z é zero
```



Exemplo

- Programa escreve na tela se o numero digitado pelo usuário é par ou impar.

```
public static void Main (string[] args)
{
    int n;
    n = int.Parse (Console.ReadLine ());
    if (n%2 == 0)
    {
        Console.WriteLine ("par");
    }
    else
    {
        Console.WriteLine ("impar");
    }
}
```





Exercícios

1. Elabore o fluxograma e a implementação de um programa que peça dois números ao usuário e exiba “igual” caso sejam iguais ou mostre qual o maior.
2. Elabore o fluxograma e a implementação de um programa que receba três inteiros e mostre qual deles é o maior e qual o menor.
3. Elabore o fluxograma e a implementação de um programa que recebe um inteiro e exiba na tela se ele é múltiplo de 7 ou não.
4. Para doar sangue é necessário ter entre 18 e 67 anos. Elabore o fluxograma e a implementação em C de um programa que pergunte a idade de uma pessoa e diga se ela pode doar sangue ou não. Use alguns dos operadores lógicos OU (||) e E (&&).