

## Lista de Exercícios 1

### Algoritmos, Pseudocódigo e Fluxograma

1. Escreva um algoritmo (pseudocódigo) que coloque os números de 1 a 100 na tela na ordem inversa (começando em 100 e terminando em 1).

**variáveis**

**inteiro:** Numero

**início**

```
1   para Numero ← 100 até 1 passo -1 repetir
2       Mostrar Numero
3   fim para
fim
```

2. Faça um algoritmo (pseudocódigo) que apresente na tela a tabela de conversão de graus Celsius para Fahrenheit, de -100 C a 100 C. Use um incremento de 10 C. Observação:  $Fahrenheit = (9 / 5) \times Celsius + 32$ .

**variáveis**

**real:** C, F

**início**

```
1   para C ← -100 até 100 passo 10 repetir
2       F ← (9.0 / 5.0) * C + 32.0
3       Mostrar C, ' Celsius -> ', F, ' Fahrenheit'
4   fim para
fim
```

3. Faça um algoritmo (pseudocódigo) para listar todos os múltiplos positivos do número 7 menores ou iguais a 100.

#### Solução 1

**variáveis**

**inteiro: N**

**início**

```
1   para N ← 7 até 100 passo 7 repetir
2     Mostrar N
3   fim para
fim
```

#### Solução 2

**variáveis**

**inteiro: N**

**início**

```
1   para N ← 1 até 100 repetir
2     se Mod( N, 7 ) = 0 então
3       Mostrar N
4     fim se
5   fim para
fim
```

4. Sendo  $h = 1 + 1/2 + 1/3 + 1/4 + \dots + 1/N$ , construa um algoritmo (pseudocódigo) para calcular o número  $h$ , sendo o número inteiro  $N$  fornecido pelo usuário. Seu algoritmo deve garantir que apenas um valor maior do que zero seja aceito como entrada.

**variáveis**

**real:**  $h$

**inteiro:**  $N, \text{Atual}$

**início**

```
1   repetir
2     Ler  $N$ 
3   enquanto  $N \leq 0$ 

4      $h \leftarrow 1.0$ 
5     para  $\text{Atual} \leftarrow 2$  até  $N$  repetir
6        $h \leftarrow h + 1.0 / \text{Atual}$ 
7     fim para

8   Mostrar  $h$ 
fim
```

5. Elabore um algoritmo (pseudocódigo) que calcule  $N!$  (fatorial de  $N$ ), sendo que o valor de  $N$  é fornecido pelo usuário. Lembre que  $N$  é sempre um valor inteiro e não negativo. Logo, seu algoritmo deve evitar que valores negativos sejam aceitos como entrada. Lembre também que  $N! = 1 \times 2 \times 3 \times \dots \times (N-1) \times N$ , e que  $0! = 1$  por definição.

**variáveis**

**inteiro:** Fatorial, N, Atual

**início**

```
1  repetir
2  Ler N
3  enquanto N < 0

4  Fatorial ← 1
5  para Atual ← 2 até N repetir
6  Fatorial ← Fatorial * Atual
7  fim para

8  Mostrar Fatorial
fim
```

6. Faça um algoritmo (pseudocódigo e fluxograma) que, a partir de um valor informado em centavos, indique a menor quantidade de moedas que representa esse valor. Considere moedas de 1, 5, 10, 25 e 50 centavos, e 1 real.

Exemplo: para o valor 290 centavos, a menor quantidade de moedas é 2 moedas de 1 real, 1 moeda de 50 centavos, 1 moeda de 25 centavos, 1 moeda de 10 centavos e 1 moeda de 5 centavos.

**variáveis**

**inteiro:** Valor, NMoedas

**início**

```
1  repetir
2  Ler Valor
3  enquanto Valor < 0
4  se Valor <> 0 então
5      NMoedas ← Valor / 100
6      se NMoedas > 0 então
7          Mostrar NMoedas, ' moeda(s) de 1 Real'
8          Valor ← Valor - (NMoedas * 100)
9      fim se
10     NMoedas ← Valor / 50
11     se NMoedas > 0 então
12         Mostrar NMoedas, ' moeda(s) de 50 centavos'
13         Valor ← Valor - (NMoedas * 50)
14     fim se
15     NMoedas ← Valor / 25
16     se NMoedas > 0 então
17         Mostrar NMoedas, ' moeda(s) de 25 centavos'
18         Valor ← Valor - (NMoedas * 25)
19     fim se
20     NMoedas ← Valor / 10
```

Observação

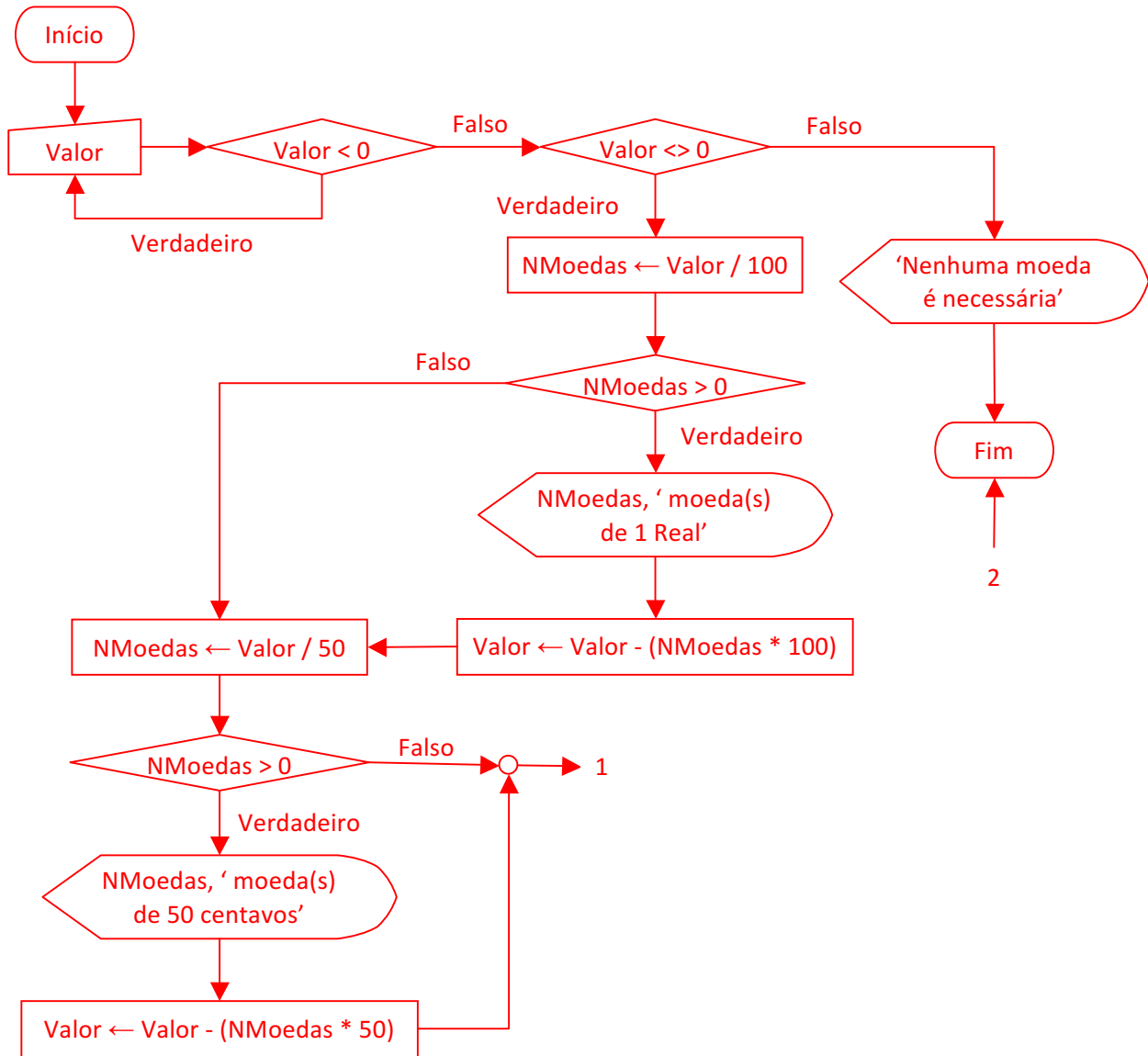
Lembre que em programas de computador a divisão envolvendo dois números inteiros retorna a parte inteira do resultado da divisão. Isso também vale para algoritmos.

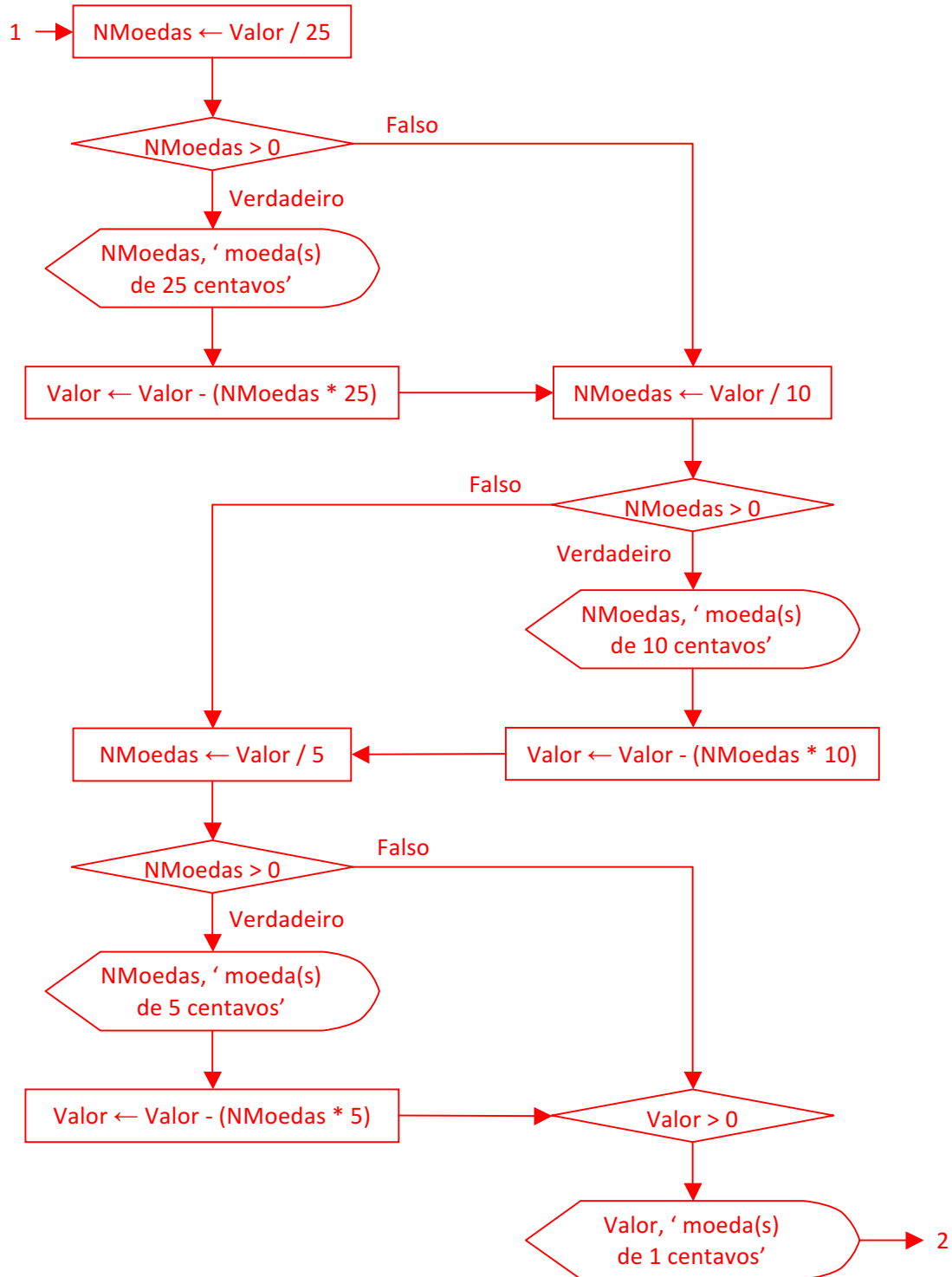
```
21     se NMoedas > 0 então
22         Mostrar NMoedas, ' moeda(s) de 10 centavos'
23         Valor ← Valor - (NMoedas * 10)
24     fim se

25     NMoedas ← Valor / 5
26     se NMoedas > 0 então
27         Mostrar NMoedas, ' moeda(s) de 5 centavos'
28         Valor ← Valor - (NMoedas * 5)
29     fim se

30     se Valor > 0 então
31         Mostrar Valor, ' moeda(s) de 1 centavo'
32     fim se

33     se não
34         Mostrar 'Nenhuma moeda é necessária'
35     fim se
fim
```







7. João tem 1,50 metros e cresce 2 centímetros por ano, enquanto Maria tem 1,10 metros e cresce 3 centímetros por ano. Construa um algoritmo (pseudocódigo e fluxograma) que calcule e imprima quantos anos serão necessários para que Maria seja maior que João.

Observação: a solução abaixo não utiliza manipulação algébrica, que, na prática, é mais simples, porém não treina o uso de estruturas de repetição.

**variáveis**

**inteiro:** N

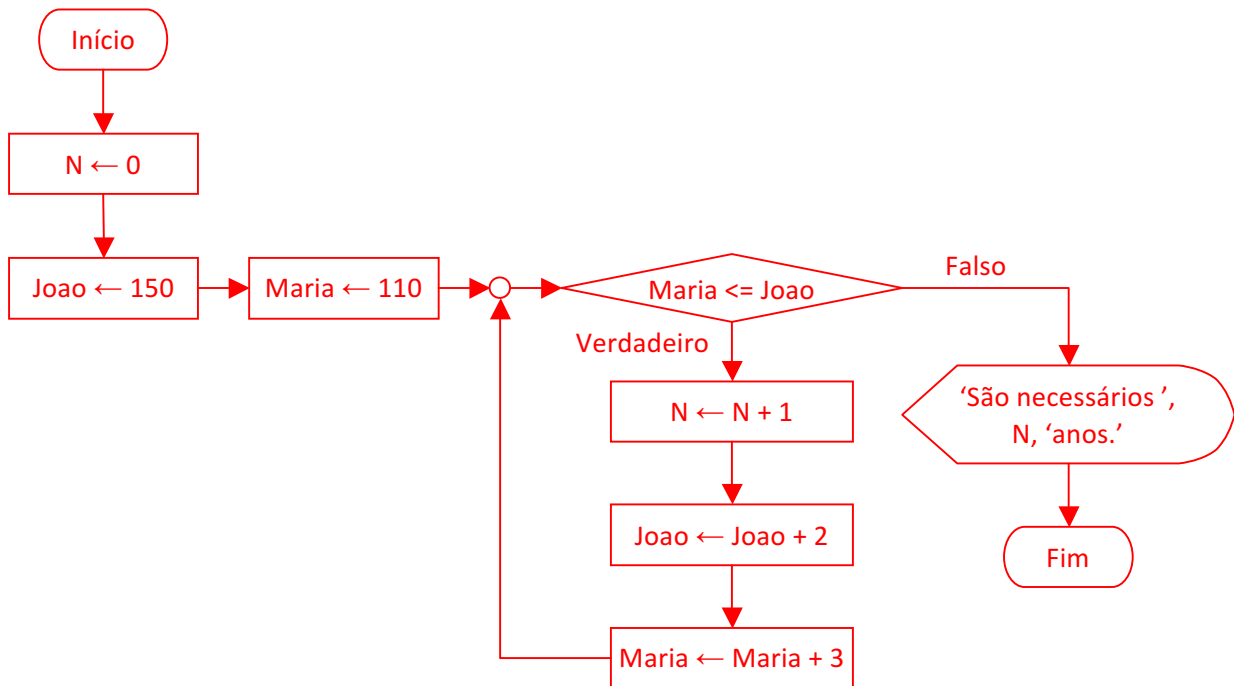
**real:** Joao, Maria

**início**

```
1     N ← 0
2     Joao ← 150
3     Maria ← 110

4     enquanto Maria <= Joao repetir
5         N ← N + 1
6         Joao ← Joao + 2
7         Maria ← Maria + 3
8     fim enquanto

9     Mostrar 'São necessários ', N, ' anos.'
fim
```



8. Faça um algoritmo (pseudocódigo e fluxograma) que pergunte ao usuário quantos números deseja somar. Em seguida, leia a quantidade informada de números e apresente o valor da soma, quantos números são maiores que 7 e quantos números são maiores que 9.

**variáveis**

**real:** Atual, Soma

**inteiro:** i, Qtd, QtdMaior7, QtdMaior9

**início**

```
1      Soma ← 0
2      QtdMaior7 ← 0
3      QtdMaior9 ← 0

4      Ler Qtd

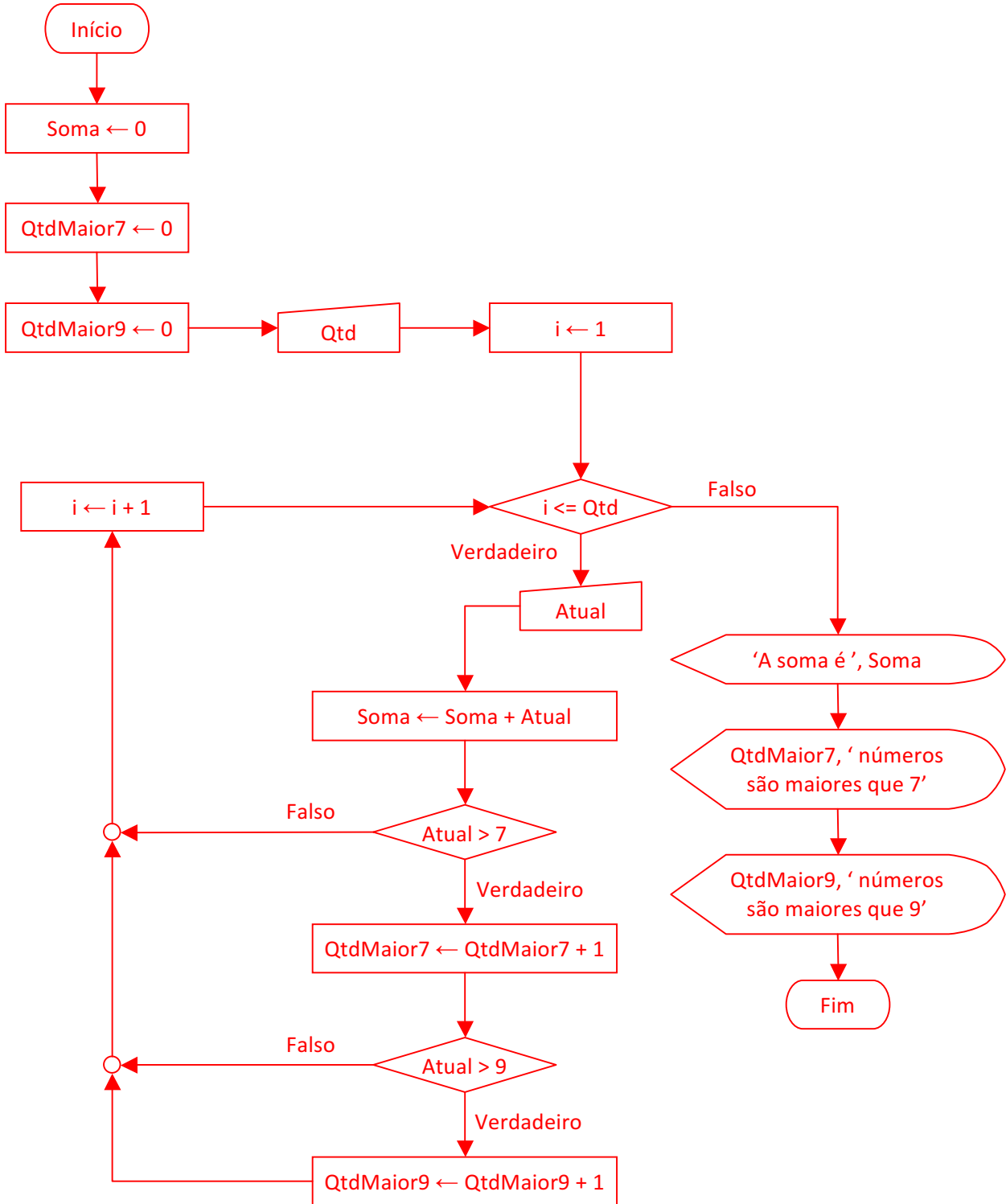
5      i ← 1
6      enquanto i <= Qtd repetir
7          Ler Atual

8          Soma ← Soma + Atual

9          se Atual > 7 então
10             QtdMaior7 ← QtdMaior7 + 1
11             se Atual > 9 então
12                 QtdMaior9 ← QtdMaior9 + 1
13             fim se
14         fim se

15         i ← i + 1
16     fim enquanto

17     Mostrar 'A soma é ', Soma
18     Mostrar QtdMaior7, ' números são maiores que 7'
19     Mostrar QtdMaior9, ' números são maiores que 9'
fim
```



9. Faça um algoritmo (pseudocódigo e fluxograma) que lê o nome de um produto, o preço e a quantidade comprada. Escreva o nome do produto comprado e o valor total a ser pago, considerando que são oferecidos descontos pelo número de unidades compradas, segundo a tabela abaixo:
- a) Até 10 unidades: valor total
  - b) De 11 a 20 unidades: 10% de desconto
  - c) De 21 a 50 unidades: 20% de desconto
  - d) Acima de 50 unidades: 25% de desconto

**variáveis**

**caractere:** Nome

**real:** Preço, Desc, Total

**inteiro:** Qtd

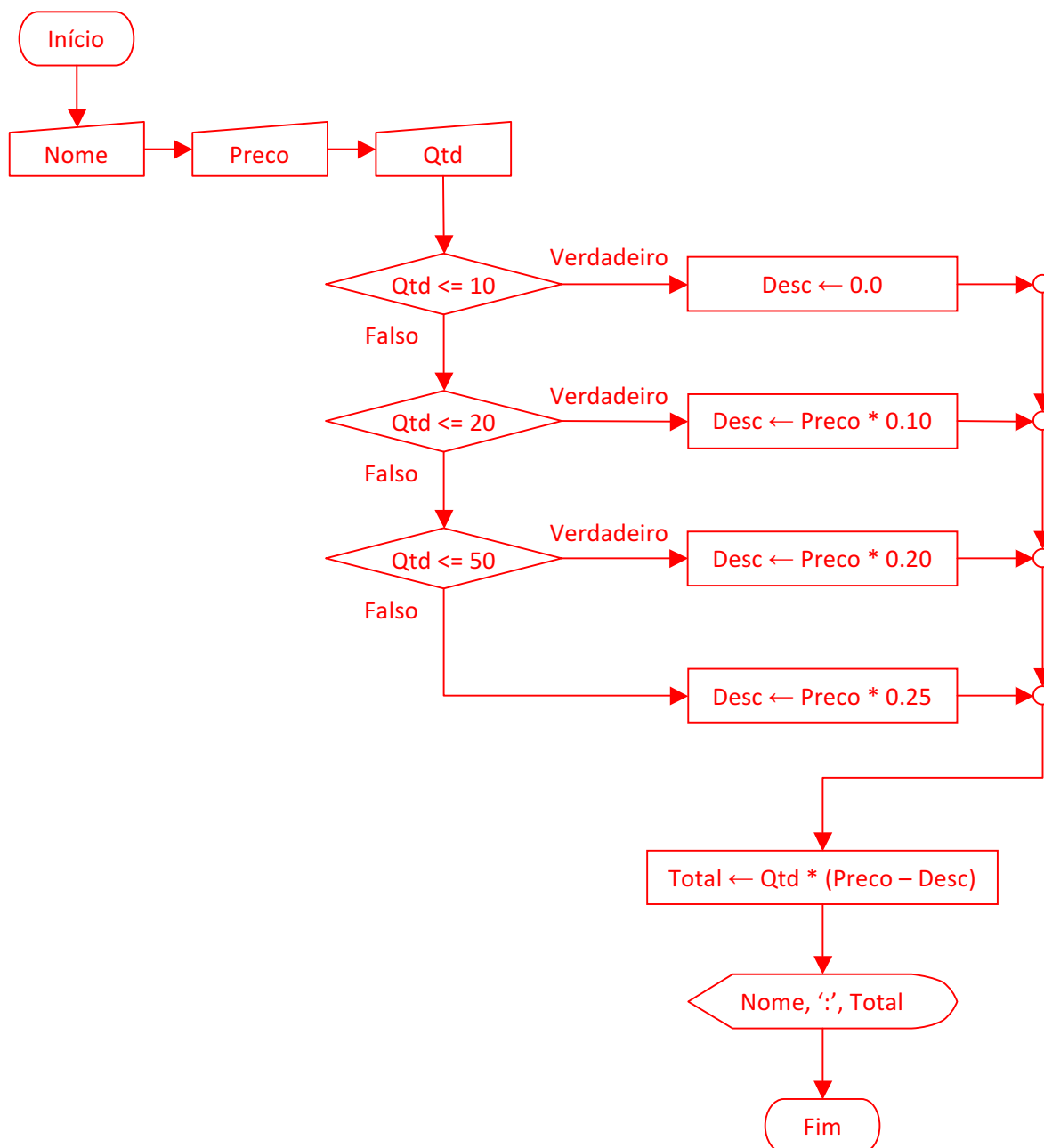
**início**

```
1      Ler Nome
2      Ler Preço
3      Ler Qtd

4      se Qtd <= 10 então
5          Desc ← 0.0
6      se não
7          se Qtd <= 20 então
8              Desc ← Preço * 0.10
9          se não
10             se Qtd <= 50 então
11                 Desc ← Preço * 0.20
12             se não
13                 Desc ← Preço * 0.25
14             fim se
15         fim se
16     fim se

17     Total ← Qtd * (Preço - Desc)

18     Mostrar Nome, ': ', Total
fim
```



10. Construa um algoritmo (pseudocódigo e fluxograma) para determinar e mostrar o número de dígitos de um número inteiro informado.

Solução 1

**variáveis**

**inteiro:** Num, Qtd

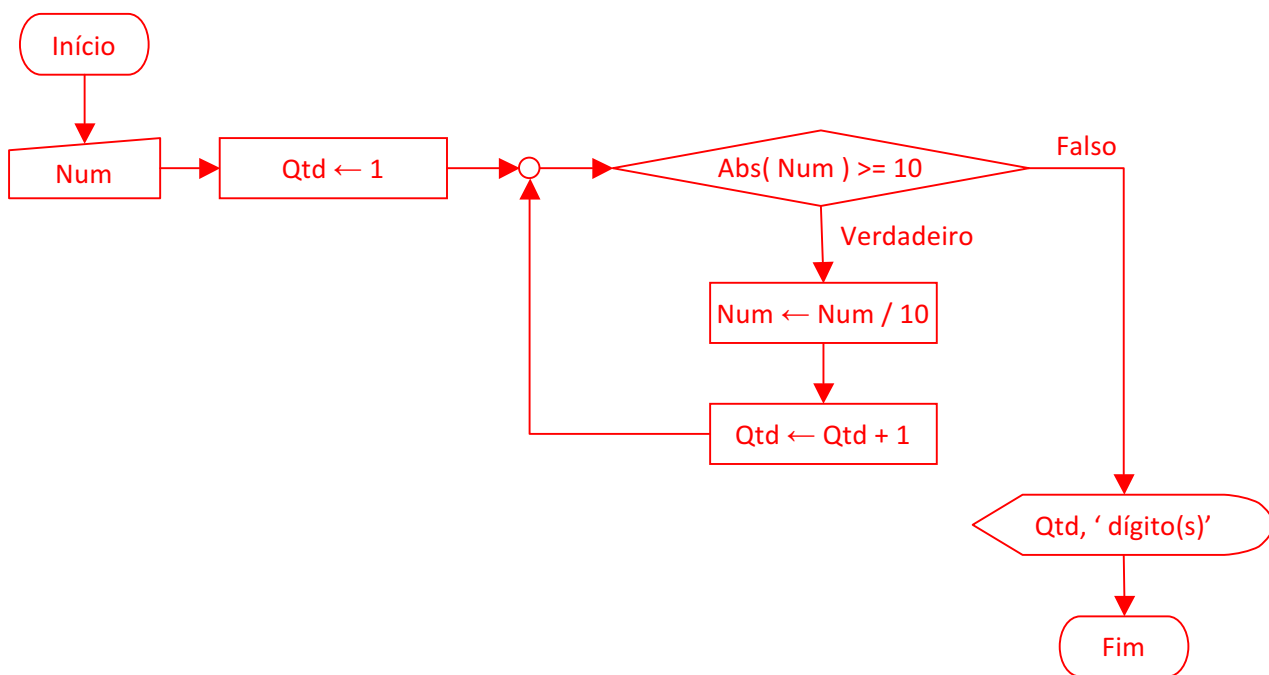
**início**

```

1  Ler Num

2  Qtd ← 1
3  enquanto Abs( Num ) >= 10 repetir
4      Num ← Num / 10
5      Qtd ← Qtd + 1
6  fim enquanto

7  Mostrar Qtd, ' dígito(s)'
fim
    
```





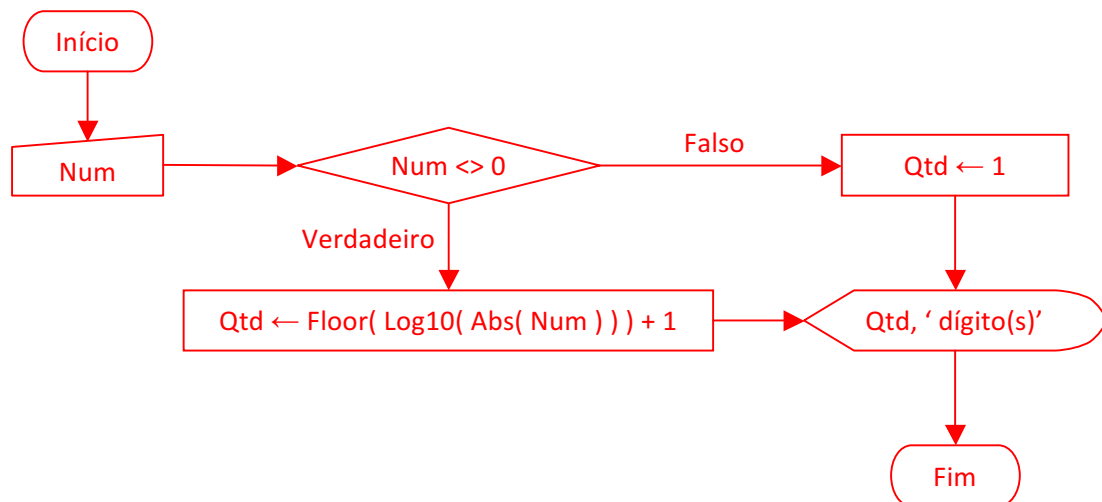
## Solução 2

### variáveis

**inteiro:** Num, Qtd

### início

```
1   Ler Num
2
3   se Num <> 0 então
4     Qtd ← Floor( Log10( Abs( Num ) ) ) + 1
5   se não
6     Qtd ← 1
7   fim se
8
9   Mostrar Qtd, ' dígito(s) '
10  fim
```



11. Considere os algoritmos abaixo. Eles lêem um código repetidamente e imprimem o código lido até que o código lido seja igual a -1. O código -1 não deve ser impresso. Responda:

a) Qual das duas soluções é a correta?

A primeira solução.

b) Como a solução incorreta poderia ser corrigida?

Testando o código lido com uma estrutura SE ... ENTÃO ... FIM SE e executando o comando "Mostrar código" apenas se "codigo <> -1".

#### Pseudocódigo A

##### **variáveis**

inteiro: codigo

##### **início**

Ler codigo

enquanto codigo <> -1 repetir

Mostrar codigo

Ler codigo

fim enquanto

fim

#### Pseudocódigo B

##### **variáveis**

inteiro: codigo

lógico: repete

##### **início**

repete ← Verdadeiro

enquanto repete repetir

Ler codigo

Mostrar codigo

repete ← codigo <> -1

fim enquanto

fim

12. Faça um algoritmo (pseudocódigo e fluxograma) que calcula a série de Fibonacci para um número informado pelo usuário. A série de Fibonacci inicia com os números 1 e 1, e cada número posterior equivale à soma dos dois números anteriores.

Exemplo: caso o número 9 seja informado, o resultado será 1, 1, 2, 3, 5, 8, 13, 21, 34.

**variáveis**

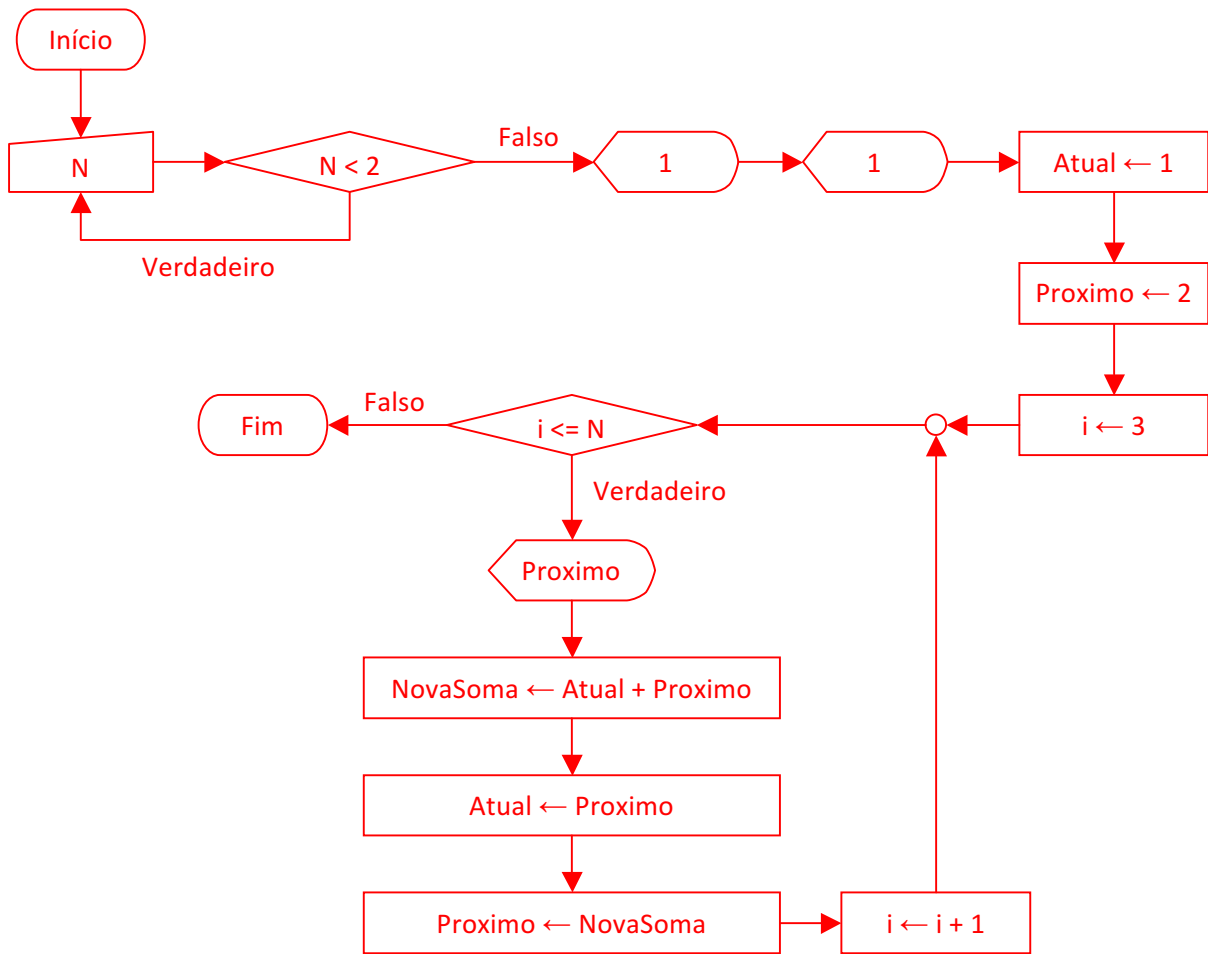
**inteiro:** i, N, Atual, Proximo, NovaSoma

**início**

```
1  repetir
2  Ler N
3  enquanto N < 2

4  Mostrar 1
5  Mostrar 1

6  Atual ← 1
7  Proximo ← 2
8  para i ← 3 até N repetir
9  Mostrar Proximo
10 NovaSoma ← Atual + Proximo
11 Atual ← Proximo
12 Proximo ← NovaSoma
13 fim para
fim
```



13. Determine a saída do seguinte algoritmo:

```
variáveis
  inteiro: a, b, c

início
  para a ← 2 até 8 passo 2 repetir
    para b ← a até 2 repetir
      para c ← 1 até a passo b repetir
        Mostrar a, b, c
      fim para
    fim para
  fim para
fim
```

A saída é "2 2 1". Note que o comando "Mostrar a, b, c" só é executado uma vez.