

Disciplina: TCC-00.326 Prog. de Computadores III

Professor: Luis Martí Orosa

Exercícios de Fixação Pseudocódigo e Estruturas Básicas de Controle

1. Construa um algoritmo (pseudocódigo e fluxograma) que:
 - a. Leia o nome de três pessoas (dado do tipo caractere).
 - b. Escreva os nomes na ordem inversa da que foram informados.

variáveis

caractere: Nome1, Nome2, Nome3

início

Ler Nome1

Ler Nome2

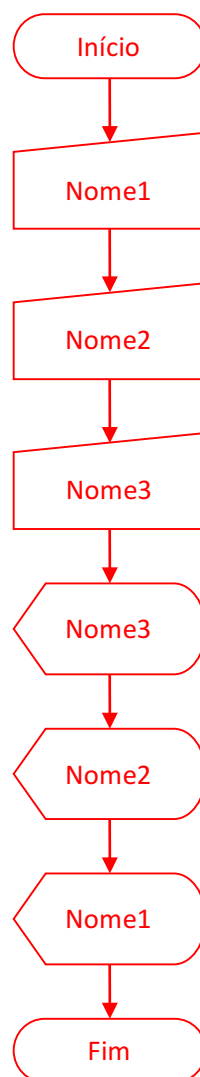
Ler Nome3

Mostrar Nome3

Mostrar Nome2

Mostrar Nome1

fim



2. Os funcionários da empresa 'Dois Irmãos' receberão um reajuste de salário. Construa um algoritmo (pseudocódigo e fluxograma) que:

- Leia o nome de um funcionário (dado do tipo caractere).
- Leia o valor atual de seu salário (dado do tipo real).
- Leia o percentual de reajuste (assuma que as entradas serão valores reais entre 0 e 100).
- Calcule o novo salário.
- Exiba o nome do funcionário e o novo valor do salário.

variáveis

caractere: Nome

real: SalarioAtual, Reajuste, NovoSalario

início

Ler Nome

Ler SalarioAtual

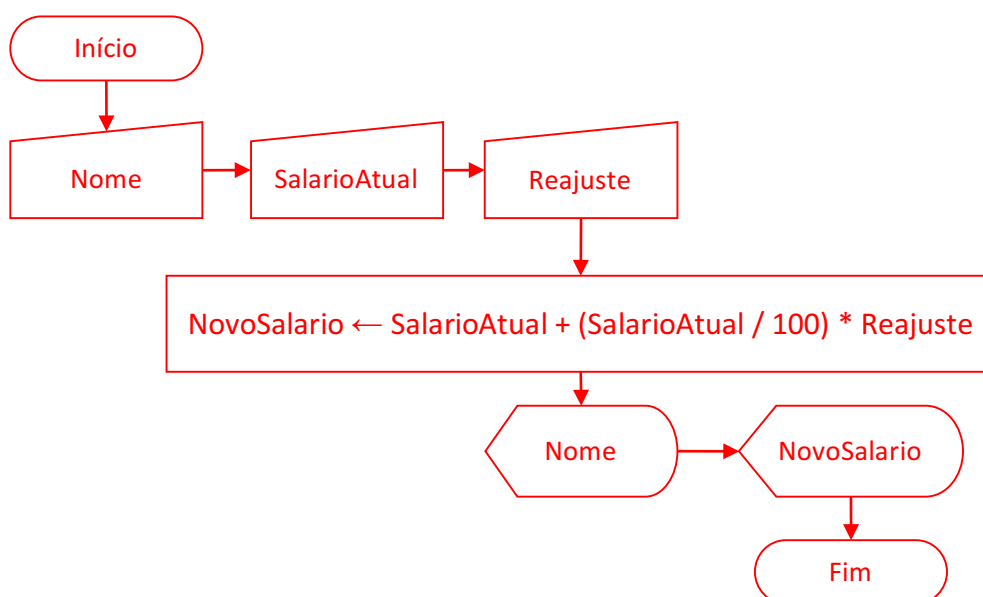
Ler Reajuste

$\text{NovoSalario} \leftarrow \text{SalarioAtual} + (\text{SalarioAtual} / 100) * \text{Reajuste}$

Mostrar Nome

Mostrar NovoSalario

fim



3. Construa um algoritmo (pseudocódigo e fluxograma) que:

- a. Leia um número inteiro.
- b. Informe se o número lido é negativo ou positivo. Considere o valor zero como sendo positivo.

Teste o algoritmo utilizando o Método Chinês com as entradas 5, -3 e 0. Mantenha anotado o valor de cada variável durante os passos da execução.

variáveis

inteiro: Numero

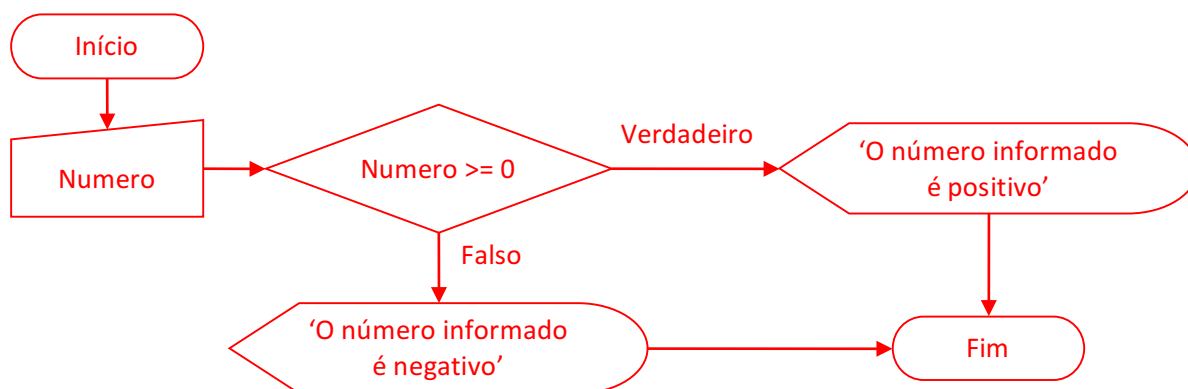
início

```

1  Ler Numero

2  se Numero >= 0 então
3  Mostrar 'O número informado é positivo'
4  se não
5  Mostrar 'O número informado é negativo'
6  fim se
fim

```



O teste de mesa neste caso é trivial. A única variável (`Numero`) assume o valor lido e o mantém até o fim da execução. O importante é acompanhar o desenvolver do algoritmo para observar que com as entradas 5 e 0 as linhas 1, 2, 3 e 6 são executadas, enquanto que com a entrada -3 as linhas 1, 2, 4, 5 e 6 são executadas.

4. Incremente o algoritmo (pseudocódigo e fluxograma) do exercício anterior modificando a saída. Neste caso, informe se o número lido é positivo, negativo ou igual a zero. Teste o algoritmo utilizando o Método Chinês com as entradas -16, 0 e 4. Mantenha anotado o valor de cada variável durante os passos da execução.

variáveis

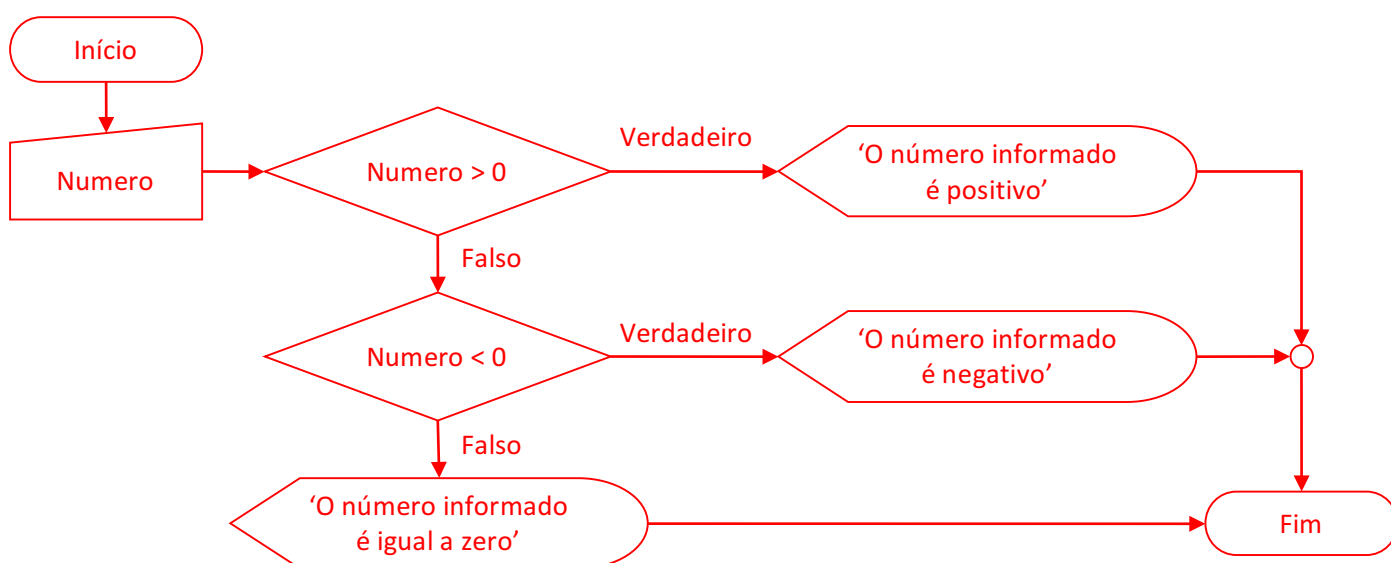
inteiro: Numero

início

```

1  Ler Numero
2
3  se Numero > 0 então
4      Mostrar 'O número informado é positivo'
5  se não
6      se Numero < 0 então
7          Mostrar 'O número informado é negativo'
8      se não
9          Mostrar 'O número informado é igual a zero'
10 fim se
    fim se
    fim

```



O teste de mesa neste caso é trivial. A única variável (`Numero`) assume o valor lido e o mantém até o fim da execução. O importante é acompanhar o desenvolver do algoritmo para observar que com a entrada -16 as linhas 1, 2, 4, 5, 6, 9 e 10 são executadas. Com a entrada 0 as linhas 1, 2, 4, 5, 7, 8, 9 e 10 são executadas. Finalmente, com a entrada 4 as linhas 1, 2, 3 e 10 são executadas.

5. Construa um algoritmo (pseudocódigo e fluxograma) que determine e mostre o maior número entre vários fornecidos pelo usuário. A condição de parada é a entrada de um valor 0 (zero), ou seja, o algoritmo deve ficar escolhendo o maior até que a entrada seja igual a 0 (zero). Utilize a estrutura ENQUANTO ... REPETIR ... no algoritmo. Assuma que valores negativos nunca serão informados pelo usuário.

variáveis

inteiro: NumAtual, MaiorNum

início

```

1     MaiorNum ← 0
2     NumAtual ← -1

3     enquanto NumAtual <> 0 repetir
4         Ler NumAtual
5         se NumAtual > MaiorNum então
6             MaiorNum ← NumAtual
7         fim se
8     fim enquanto

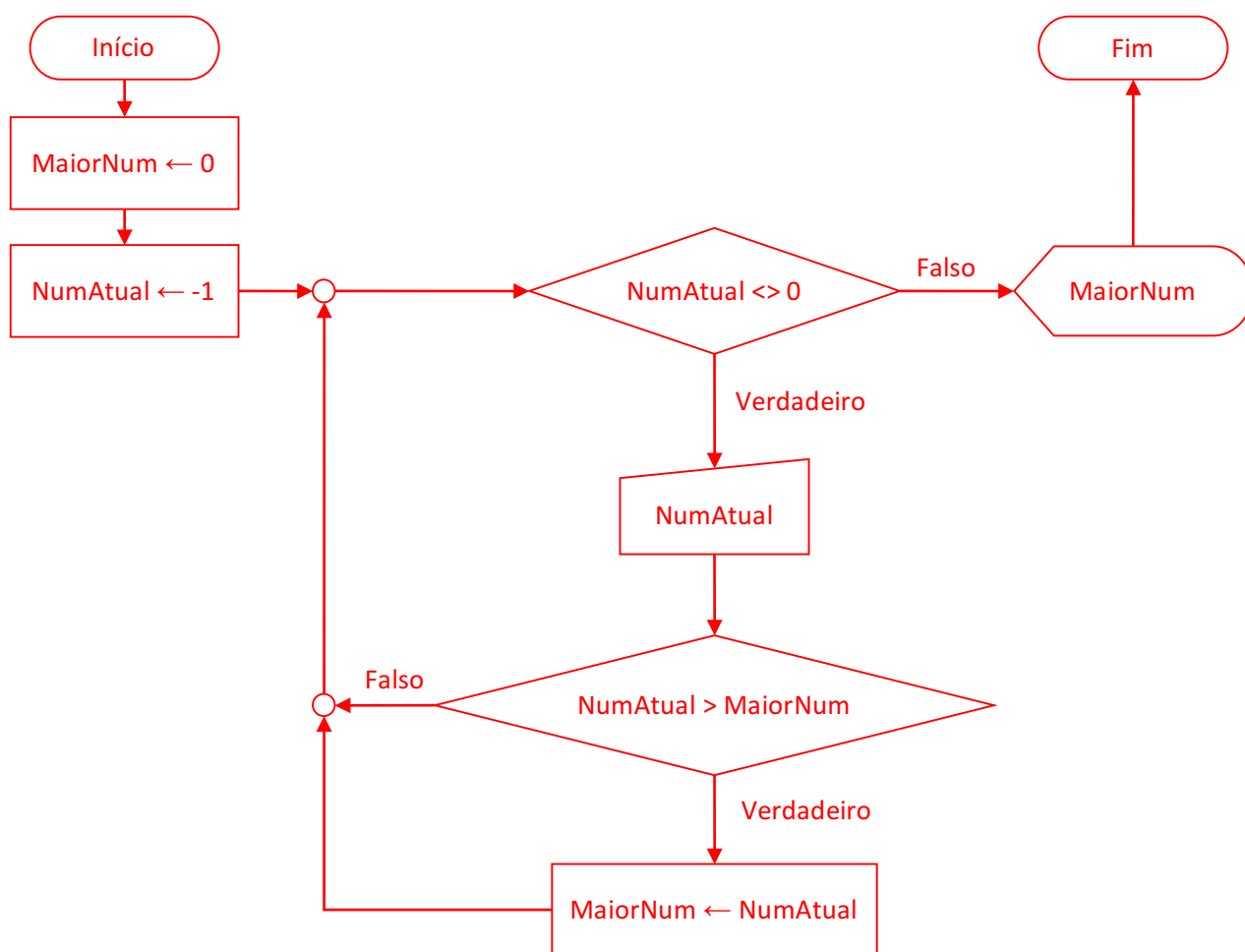
9     Mostrar MaiorNum
fim

```

A variável `MaiorNum` precisa ser inicializada com um valor menor que qualquer valor que possa ser informado pelo usuário. Neste caso, foi inicializada com 0 (zero) na Linha 1 do pseudocódigo.

A variável `NumAtual` precisa ser inicializada com qualquer valor diferente de 0 (zero), para que haja a possibilidade de entrar do laço ENQUANTO... REPETIR... Neste caso foi escolhido -1 (veja a Linha 2 do pseudocódigo). Note que o primeiro valor atribuído a `NumAtual` não afeta a escolha do maior número dentre os informados pelo usuário, pois o valor de inicialização é sobrescrito por um valor informado pelo usuário na Linha 4 do pseudocódigo, antes de ser feita a comparação com o maior número armazenado até o momento (Linha 5).

Outras soluções (algoritmos) equivalentes podem ser construídas. Teste sua solução fazendo o teste de mesa.



6. Construa um algoritmo (pseudocódigo) que leia 500 valores inteiros e positivos e:
- Encontre e mostre o maior valor.
 - Encontre e mostre o menor valor.
 - Calcule e mostre a média dos números lidos.

Utilize a estrutura ENQUANTO ... REPETIR ... no algoritmo.

variáveis

inteiro: Cont, NumAtual, MaiorNum, MenorNum, Soma, Media

início

```

1   MaiorNum ← -∞
2   MenorNum ← +∞
3   Soma ← 0

4   Cont ← 0
5   enquanto Cont < 500 repetir
6     Ler NumAtual

7     se NumAtual > MaiorNum então
8       MaiorNum ← NumAtual
9     fim se

10    se NumAtual < MenorNum então
11      MenorNum ← NumAtual
12    fim se

13    Soma ← Soma + NumAtual
14    Cont ← Cont + 1
15  fim enquanto

16  Media ← Soma / Cont

17  Mostrar MaiorNum
18  Mostrar MenorNum
19  Mostrar Media
fim

```

Assim como no exercício anterior, a variável `MaiorNum` precisa ser inicializada com um valor menor que qualquer valor que possa ser informado pelo usuário, para que este valor inicial possa ser “vencido” por qualquer outro número informado pelo usuário. A variável `MenorNum`, por sua vez, precisa receber um valor maior que qualquer valor que possa ser informado pelo usuário, para que o valor inicial também possa ser “vencido”.

Na solução apresentada optei por usar $-\infty$ e $+\infty$ para inicializar `MaiorNum` e `MenorNum`, respectivamente (veja as Linhas 1 e 2 do pseudocódigo).

Outras soluções (algoritmos) equivalentes podem ser construídas. Teste sua solução fazendo o teste de mesa.

7. Reescreva o algoritmo desenvolvido no Exercício 5 utilizando a estrutura de repetição REPETIR ... ENQUANTO ... no lugar de ENQUANTO ... REPETIR ...

variáveis

inteiro: NumAtual, MaiorNum

início

1 MaiorNum ← 0

2 **repetir**

3 **Ler** NumAtual

4 **se** NumAtual > MaiorNum **então**

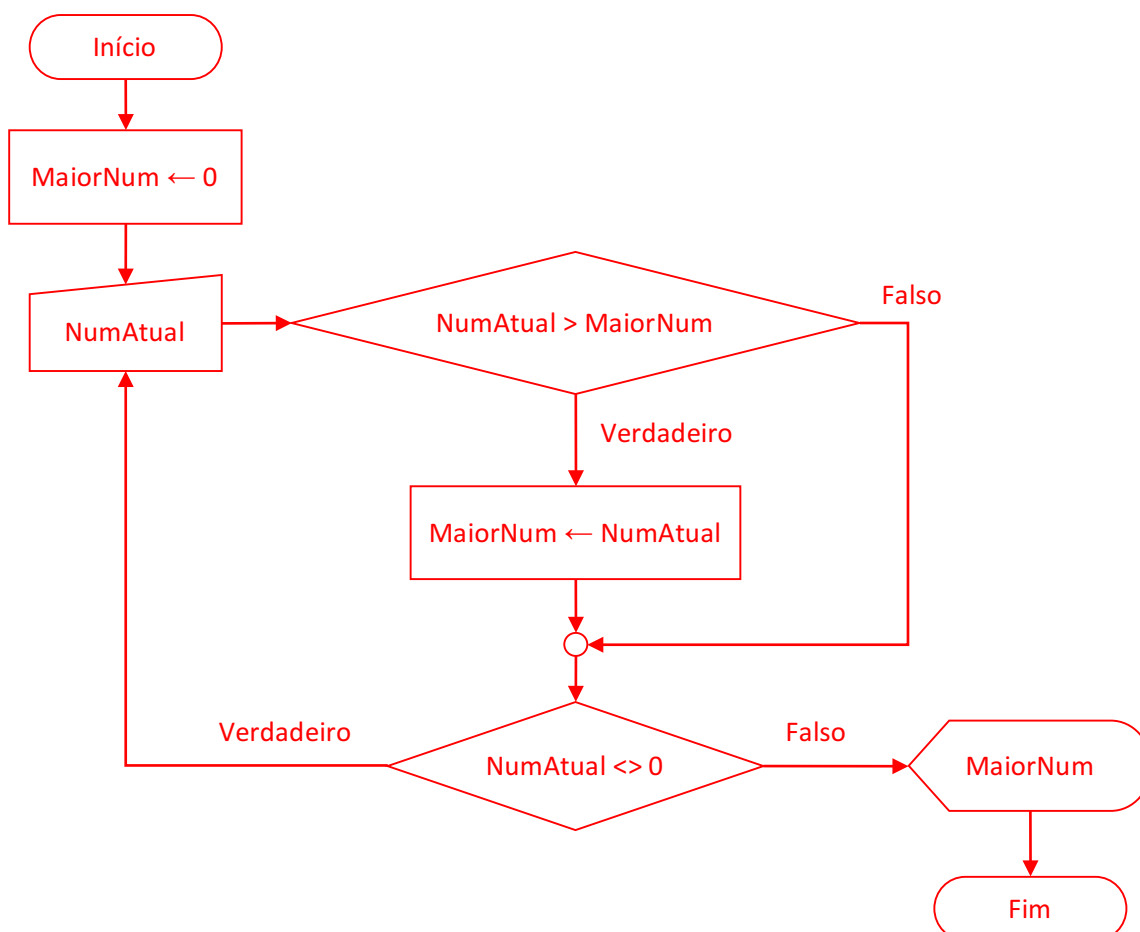
5 MaiorNum ← NumAtual

6 **fim se**

7 **enquanto** NumAtual <> 0

8 **Mostrar** MaiorNum

fim



8. Utilizando a estrutura PARA... ATÉ... REPETIR..., construa um algoritmo (pseudocódigo) que exiba os números ímpares entre 100 e 200.

Dica: a função $\text{mod}(X,Y)$ retorna o resto da divisão de X por Y.

Solução utilizando passo diferente de 1 (um)

variáveis

inteiro: NumAtual

início

```
1   para NumAtual ← 101 até 199 passo 2 repetir
2       Mostrar NumAtual
3   fim para
fim
```

Solução utilizando passo igual a 1 (um)

variáveis

inteiro: NumAtual

início

```
1   para NumAtual ← 100 até 200 repetir
2       se mod( NumAtual, 2 ) <> 0 então
3           Mostrar NumAtual
4       fim se
5   fim para
fim
```

9. Construa um algoritmo (pseudocódigo) que calcule a média de N números reais informados pelo usuário, onde o valor de N também é informado no início da execução. Utilize a estrutura PARA... ATÉ... REPETIR... Antes do término da execução do algoritmo, mostre a média calculada.

variáveis

inteiro: i, N
real: NumAtual, Soma, Media

início

```

1  repetir
2    Ler N
3  enquanto N <= 0

4    Soma ← 0
5  para i ← 1 até N repetir
6    Ler NumAtual
7    Soma ← Soma + NumAtual
8  fim para

9  se N > 0 então
10   Media ← Soma / N
11 se não
12   Media ← 0
13 fim se

14 Mostrar Media
fim

```

Neste exercício é importante notar que não sabemos *a priori* a quantidade de valores que serão informados pelo usuário. Saberemos esta quantidade (N) apenas em tempo de execução do algoritmo. Por conta deste detalhe, as Linhas 1 a 3 do pseudocódigo garantem que a quantidade de números informados pelo usuário será sempre maior ou igual a 0 (zero). Afinal, informar uma quantidade negativa de número não faz sentido.

Também, precisamos ser mais cuidadosos ao calcular a média. Precisamos testar na se a quantidade de valores é diferente de 0 (zero), a fim de evitar divisões por zero. Isso é feito na Linha 9 do pseudocódigo.

10. Tendo como dados de entrada a altura e o sexo de uma pessoa, construa um algoritmo (pseudocódigo e fluxograma) que calcule seu peso ideal, utilizando as seguintes fórmulas:

Para homens: $(72,7 * altura) - 58$

Para mulheres: $(62,1 * altura) - 44,7$

Ao perguntar o sexo, o algoritmo deverá garantir que as únicas entradas aceitas sejam 'M' para homens e 'F' para mulheres. Ou seja, caso um valor diferente de 'M' ou 'F' seja informado, o programa deverá requisitar ao usuário que informe o sexo novamente até a entrada ser válida. Dica: utilize a estrutura de repetição mais adequada dentre as três vistas em aula.

variáveis

caractere: Sexo

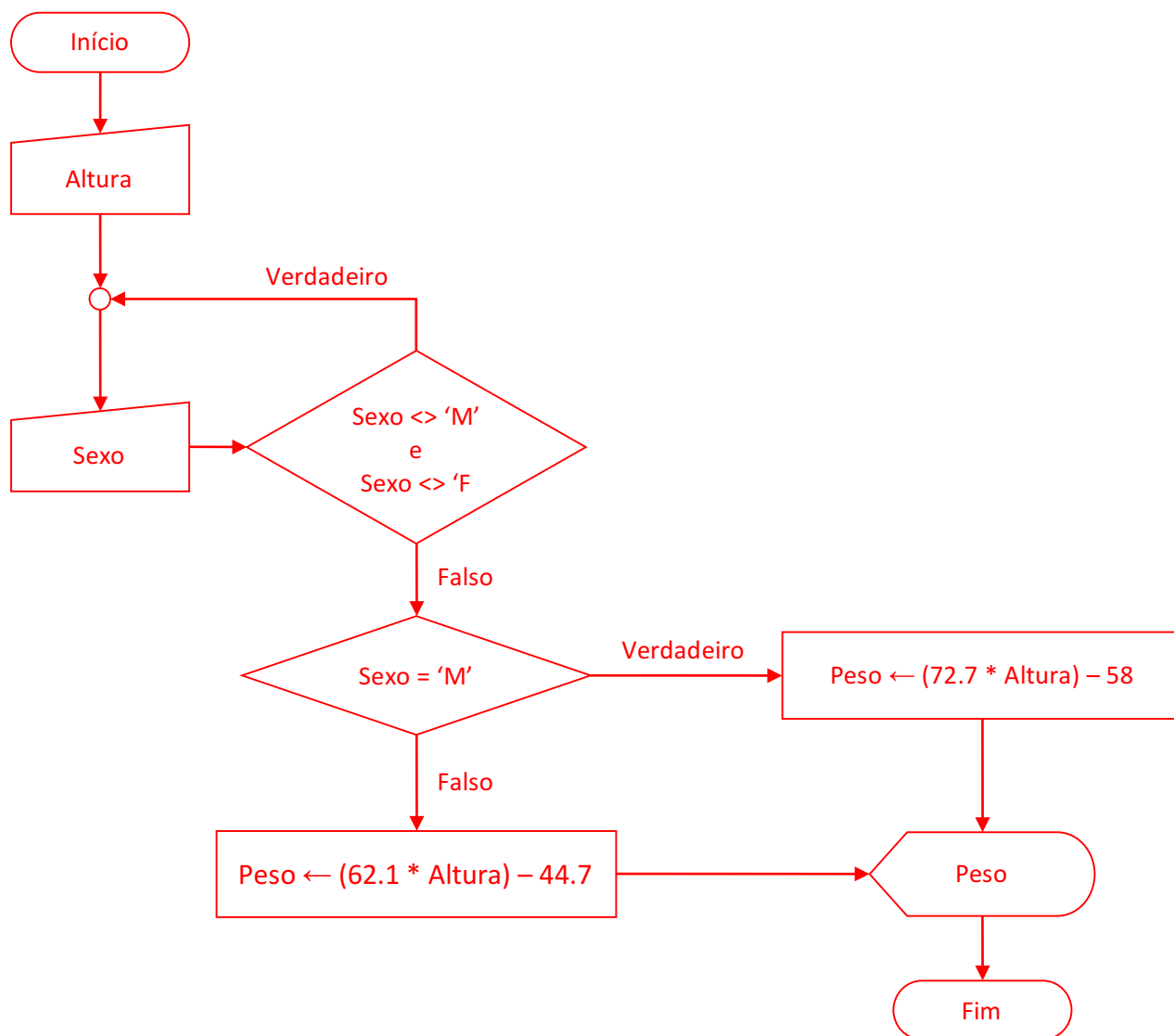
real: Altura, Peso

início

```

1  Ler Altura
2
3  repetir
4  Ler Sexo
5  enquanto Sexo <> 'M' e Sexo <> 'F'
6
7  se Sexo = 'M' então
8  Peso ← (72.7 * Altura) - 58
9  se não
10  Peso ← (62.1 * Altura) - 44.7
11  fim se
12
13  Mostrar Peso
14  fim

```



11. Dado o pseudocódigo a seguir, responda:

```

variáveis
  lógico: A, B, C

início
  Ler A
  Ler B
  Ler C

  se A então
    Mostrar 'Saída 1'
  se não
    se B então
      se C então
        Mostrar 'Saída 2'
      se não
        Mostrar 'Saída 3'
        Mostrar 'Saída 4'
      fim se
    fim se
    Mostrar 'Saída 5'
  fim se
  Mostrar 'Saída 6'
fim

```

- a. Se A = Verdadeiro, B = Verdadeiro, C = Falso, quais comandos “Mostrar” serão executadas?

Mostrar 'Saída 1'
Mostrar 'Saída 6'

- b. Se A = Falso, B = Verdadeiro, C = Falso, quais comandos “Mostrar” serão executados?

Mostrar 'Saída 3'
Mostrar 'Saída 4'
Mostrar 'Saída 5'
Mostrar 'Saída 6'

- c. Se A = Falso, B = Verdadeiro, C = Verdadeiro, quais comandos “Mostrar” serão executados?

Mostrar 'Saída 2'
Mostrar 'Saída 5'
Mostrar 'Saída 6'

- d. Quais são os valores de A, B e C para que somente os comandos “Mostrar ‘Saída 5” e “Mostrar ‘Saída 6” sejam executados?

A = Falso, B = Falso e C tanto faz, pois ele não chega a ser testado quando o valor armazenado por B é igual a Falso.

- e. Quais são os valores de A, B e C para que somente o comando “Mostrar ‘Saída 2” seja executado?

Para que o comando “Mostrar ‘Saída 2” seja executado é preciso A = Falso, B = Verdadeiro e C = Verdadeiro. Entretanto, ele nunca será executado sozinho, pois nesse caso os comandos “Mostrar ‘Saída 5” e “Mostrar ‘Saída 6” também serão executados. Logo, não existe uma configuração de valores para A, B e C que faça o comando “Mostrar ‘Saída 2” ser o único a ser executado.

12. Escreva um algoritmo (pseudocódigo) que leia três valores inteiros e mostre-os em ordem decrescente. Dica: utilize estruturas de decisão encadeadas.

variáveis

inteiro: V1, V2, V3

início

1 **Ler** V1

2 **Ler** V2

3 **Ler** V3

4 **se** V1 > V2 **então**

5 **se** V1 > V3 **então**

6 **se** V2 > V3 **então**

7 **Mostrar** V1

8 **Mostrar** V2

9 **Mostrar** V3

10 **se não**

11 **Mostrar** V1

12 **Mostrar** V3

13 **Mostrar** V2

14 **fim se**

15 **se não**

16 **Mostrar** V3

17 **Mostrar** V1

```
18         Mostrar V2
19     fim se
20 se não
21     se V2 > V3 então
22         se V1 > V3 então
23             Mostrar V2
24             Mostrar V1
25             Mostrar V3
26         se não
27             Mostrar V2
28             Mostrar V3
29             Mostrar V1
30         fim se
31 se não
32     Mostrar V3
33     Mostrar V2
34     Mostrar V1
35 fim se
36 fim se
fim
```

13. Construa um algoritmo (pseudocódigo) que escreva uma contagem de 10 (dez) minutos, ou seja, mostre 0:00, e então 0:01, 0:02, ..., 0:58, 0:59, 1:00, 1:01, 1:02, ..., até 10:00. Dica: utilize a estrutura de repetição mais adequada dentre as três vistas em aula.

variáveis

inteiro: Min, Seg

início

```

1   para Min ← 0 até 9 repetir
2     para Seg ← 0 até 59 repetir
3       Mostrar Min, ':', Seg
4     fim para
5   fim para
6   Mostrar '10:00'
fim

```

14. Construa três algoritmos (pseudocódigos) que imprimam a tabuada do 5 (cinco). Em cada um deles utilize uma das estruturas de repetição vistas em aula. Dica: verifique se o resultado está correto fazendo um teste de mesa para cada uma das soluções apresentadas.

Solução com ENQUANTO ... REPETIR ...

variáveis

inteiro: Num, Resultado

início

```

1   Num ← 0
2   enquanto Num <= 10 repetir
3     Resultado ← 5 * Num
4     Mostrar '5 * ', Num, ' = ', Resultado
5     Num ← Num + 1
6   fim enquanto
fim

```


Solução com REPETIR ... ENQUANTO ...**variáveis****inteiro:** Num, Resultado**início**

```
1   Num ← 0
2   repetir
3       Resultado ← 5 * Num
4       Mostrar '5 * ', Num, ' = ', Resultado
5       Num ← Num + 1
6   enquanto Num <= 10
fim
```

Solução com PARA ... ATÉ ... REPETIR ...

variáveis

inteiro: Num, Resultado

início

```
1   para Num ← 0 até 10 repetir  
2       Resultado ← 5 * Num  
3       Mostrar '5 * ', Num, ' = ', Resultado  
4   fim enquanto  
fim
```

Obs.: Outras soluções (algoritmos) equivalentes podem ser construídas. O teste de mesa ajudará você a verificar se o algoritmo que você escreveu está correto. Em especial, ajudará no caso das duas primeiras estruturas de repetição. Com o teste de mesa você verifica se 5 é multiplicado apenas pelos números inteiros de 0 a 10.