

AULA DE ALGORITMO - 03

Profa. M. Sc. Valéria Maria Volpe

Estruturas de Repetição

2

□ Estruturas de Repetição

- ▣ As estruturas de repetição servem para executar as mesmas ações (bloco de comandos) um número finito de vezes.
- ▣ Compete ao programador criar a lógica que permitirá a repetição, assim como, sua finalização.
- ▣ O controle da repetição é feito usando uma expressão lógica que irá resultar verdadeiro ou falso.
- ▣ Temos as seguintes estruturas de repetição:
 - Com teste no início
 - Com teste no final
 - Com variável de controle

Estruturas de Repetição

3

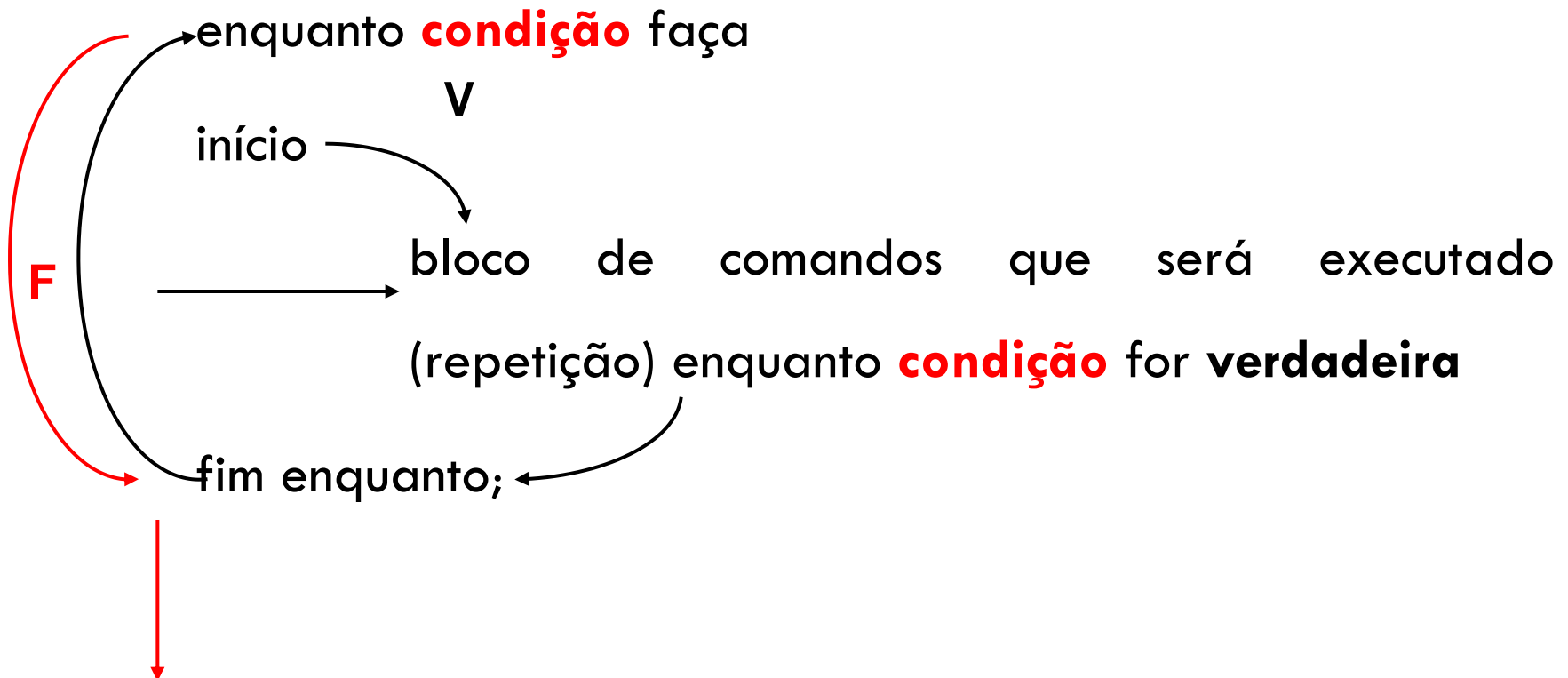
□ Estrutura de repetição com teste no início

- Esta estrutura consiste em uma estrutura de controle do fluxo de execução do algoritmo/programa, que permite repetir várias vezes **o mesmo bloco de comandos**, porém sempre verificando **ANTES** de cada execução se é **PERMITIDO** executar o mesmo trecho mais uma vez.

Estruturas de Repetição

4

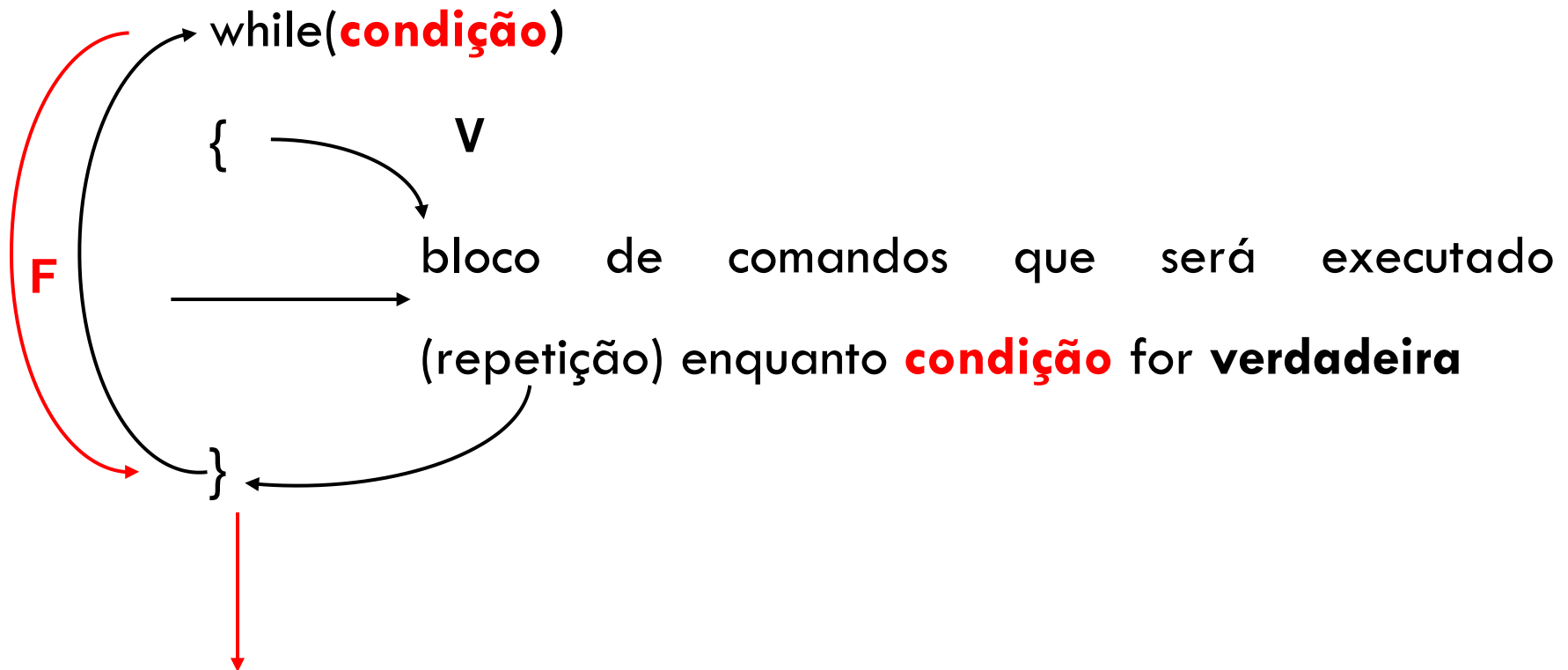
□ Sintaxe – Algoritmo



Estruturas de Repetição

5

□ Sintaxe – Linguagem C



Estruturas de Repetição

6

□ Exemplo:

- 1) João mede 1.4 m de altura e Pedro mede 1.25m. Sabe-se que João cresce 3 cm ao ano e Pedro 5 cm. Quantos anos serão necessários para Pedro ficar mais alto que João. Faça um algoritmo/programa para resolver o problema.

□ Ações:

- Atualizar a altura de João e de Pedro
- Contar um ano a cada atualização

□ Variáveis

- Altura João
- Altura Pedro
- Anos

Estruturas de Repetição

7

Algoritmo Altura João e Pedro

Início

var

real: alturaJoao, alturaPedro;

inteiro: anos;

alturaJoao \leftarrow 1.4;

alturaPedro \leftarrow 1.25;

anos \leftarrow 0;

enquanto alturaPedro \leq alturaJoao **faça**

início

alturaJoao \leftarrow alturaJoao + 0.03;

alturaPedro \leftarrow alturaPedro + 0.05;

anos \leftarrow anos + 1;

fim enquanto;

escreva("Serão necessário ", anos, "anos para Pedro ficar mais alto que João");

Fim.

Estruturas de Repetição

8

```
#include <stdio.h>
#include <conio.h>
main()
{
    float alturaJoao = 1.4, alturaPedro = 1.25;
    int anos = 0;
    while (alturaPedro <= alturaJoao)
    {
        alturaJoao = alturaJoao + 0.03;
        alturaPedro = alturaPedro + 0.05;
        anos = anos + 1;
    }
    printf("Serão necessário %d anos para Pedro ficar mais alto que João", anos);
    getch();
}
```


Estruturas de Repetição

9

□ Exercícios

- 2) Faça um algoritmo para mostrar os números de 1 a 100.
- 3) Faça um algoritmo para mostrar os N° pares entre 1 a 100.
- 4) Faça um algoritmo para mostrar os N° pares entre 1 a 100, contando de 1 em 1.
- 5) Faça um algoritmo para ler 100 N° inteiro e mostrar somente os impares lidos.
- 6) Faça um algoritmo para ler um número inteiro e calcular o somatório:

$$\sum_{i=1}^n \frac{1}{i} = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$$

Solução Exercício 2

10

Algoritmo Contar de 1 a 100;

Início

var

inteiro: contador;

contador \leftarrow 1;

enquanto contador \leq 100 **faça**

início

 escreva(contador);

 contador \leftarrow contador + 1;

fim enquanto;

Fim.

Solução Exercício 3

11

Algoritmo Contar Pares de 1 a 100;

Início

var

inteiro: contador;

contador \leftarrow 2;

enquanto contador \leq 100 **faça**

início

 escreva(contador);

 contador \leftarrow contador + 2;

fim enquanto;

Fim.

Solução Exercício 4

12

Algoritmo Contar de 1 a 100;

Início

var

inteiro: contador;

contador \leftarrow 1;

enquanto contador \leq 100 **faça**

início

se contador MOD 2 = 0 então

início

escreva(contador);

fim se;

contador \leftarrow contador + 1;

fim enquanto;

Fim.

Solução Exercício 5

13

Algoritmo Ler 100 números mostrar ímpar;

Início

var

inteiro: contador, num;

contador \leftarrow 1;

enquanto contador \leq 100 **faça**

início

leia(num);

se num MOD 2 \neq 0 então

início

escreva(num, “ é ímpar”);

fim se;

contador \leftarrow contador + 1;

fim enquanto;

Fim.

Solução Exercício 6

14

Algoritmo Somatório;

Início

var

inteiro: i, n;

real: somatorio;

somatorio \leftarrow 0.0;

i \leftarrow 1;

leia(n);

enquanto **i** \leq **n** **faça**

início

somatorio \leftarrow **somatorio** + (1.0 / **i**);

i \leftarrow **i** + 1;

fim enquanto;

escreva("Somatório = ", **somatorio**);

Fim.

Estruturas de Repetição

15

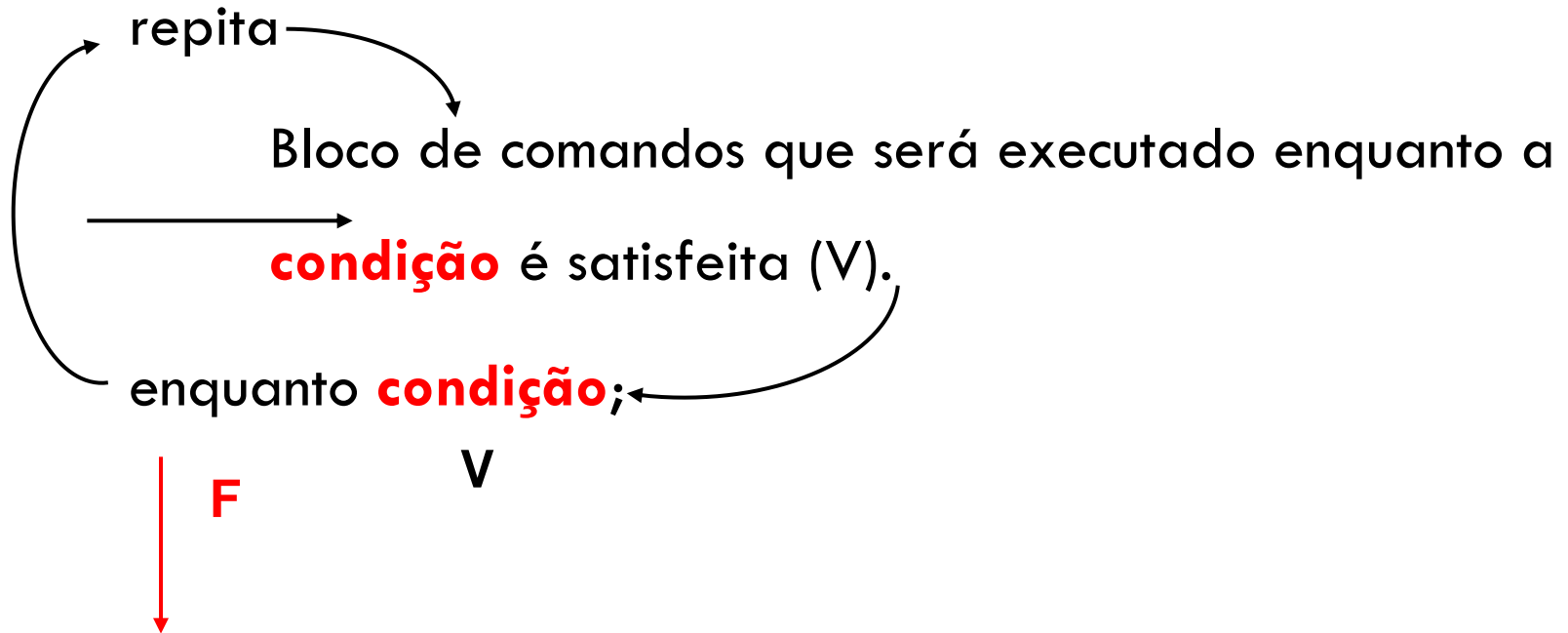
□ Estrutura de repetição com teste no final

- ▣ Esta estrutura permite que ações (bloco de comandos) sejam executadas enquanto a condição é satisfeita.
- ▣ Esta estrutura consiste em uma estrutura de controle do fluxo de execução do algoritmo/programa, que permite repetir diversas vezes o **mesmo bloco de comandos**, porém sempre verificando **DEPOIS** de cada execução se é **PERMITIDO** executar o mesmo trecho mais uma vez
- ▣ Devido a sintaxe da estrutura, o bloco de comandos será executado pelo menos uma vez, pois o teste ocorre no final do bloco da repetição.

Estruturas de Repetição

16

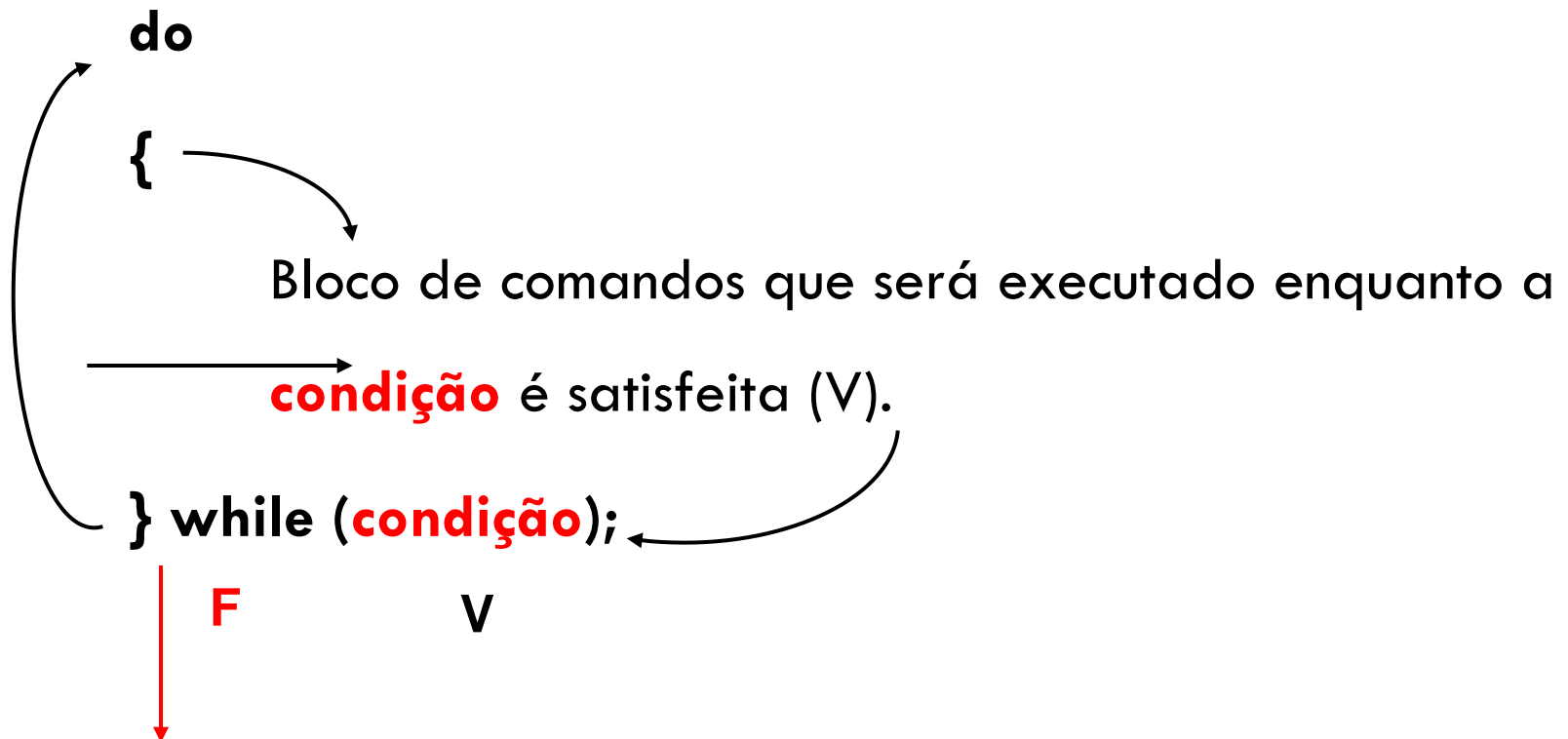
□ Sintaxe – Algoritmo



Estruturas de Repetição

17

□ Sintaxe – Linguagem C



Estruturas de Repetição

18

□ Exemplo:

- 1) João mede 1.4 m de altura e Pedro mede 1.25m. Sabe-se que João cresce 3 cm ao ano e Pedro 5 cm. Quantos anos serão necessários para Pedro ficar mais alto que João. Faça um algoritmo/programa para resolver o problema.

■ Ações:

- Atualizar a altura de João e de Pedro
- Contar um ano a cada atualização

■ Variáveis

- Altura João
- Altura Pedro
- Anos

Estruturas de Repetição

19

Algoritmo Altura João e Pedro

Início

var

real: alturaJoao, alturaPedro;

inteiro: anos;

alturaJoao \leftarrow 1.4;

alturaPedro \leftarrow 1.25;

anos \leftarrow 0;

repita

alturaJoao \leftarrow alturaJoao + 0.03;

alturaPedro \leftarrow alturaPedro + 0.05;

anos \leftarrow anos + 1;

enquanto alturaJoao \geq alturaPedro;

escreva("Serão necessário ", anos, " anos para Pedro ficar mais alto que João");

Fim.

Estruturas de Repetição

20

```
#include <stdio.h>
#include <conio.h>
main()
{
    float alturaJoao = 1.4, alturaPedro = 1.25;
    int anos = 0;
    do
    {
        alturaJoao = alturaJoao + 0.03;
        alturaPedro = alturaPedro + 0.05;
        anos = anos + 1;
    } while (alturaJoao >= alturaPedro);
    printf("Serão necessário %d anos para Pedro ficar mais alto que João", anos);
    getch();
}
```

Estruturas de Repetição

21

□ Exercícios

2) Foi feita uma pesquisa com os alunos da FATEC, com o objetivo de calcular a média de idade dos alunos e encontrar a idade do aluno mais velho. Para isso, um grupo de alunos (amostra) digitou sua idade ao termino da pesquisa as informações acima foram mostradas a todos. Faça um algoritmo para resolver o problema.

□ **Ações:**

- Ler idade
- Calcular a media de idade
- Encontram a idade mais velha
- Mostram os resultados.

□ **Variáveis:**

- Idade; Somaldade; ContPessoa; IdadeMaisVelho; Medialdade; - inteiro
- Resp – caracter

Solução do Exercício 2

22

Algoritmo Pesquisa FATEC;

Início

var

caracter: resp;

inteiro: idade, somaidade, contpessoas, idademaivelho, mediaidade;

contpessoas \leftarrow 0;

somaidade \leftarrow 0;

idademaivelho \leftarrow 0;

repita

leia(idade);

somaidade \leftarrow somaidade + idade;

contpessoas \leftarrow contpessoas + 1;

se idade > idademaivelho então

início

idademaivelho \leftarrow idade;

fim se;

leia(resp)

enquanto resp = 'S' OU resp = 's';

mediaidade \leftarrow somaidade / contpessoas;

escreva("Média de idade = ", **mediaidade**, " Idade mais velho = ", **idademaivelho**);

Fim.

Estruturas de Repetição

23

□ Exercícios

- 3) Faça um algoritmo para mostrar os números de 1 a 100
- 4) Faça um algoritmo para mostrar os N° pares entre 1 a 100.
- 5) Faça um algoritmo para mostrar os N° pares entre 1 a 100, contando de 1 em 1.
- 6) Faça um algoritmo para ler 100 N° inteiro e mostrar somente os impares lidos.
- 7) Faça um algoritmo para ler um número inteiro e calcular o somatório:

$$\sum_{i=1}^n \frac{1}{i} = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$$

Exercício para entregar

24

- Uma companhia de teatro deseja dar uma série de espetáculos. A direção calcula que a R\$ 48,00 o ingresso, serão vendidos 350 ingressos, e que as despesas fixas serão R\$ 5.600,00. Diminuindo-se R\$ 2,00 do preço dos ingressos espera-se que as vendas aumentarão em 17 ingressos. Faça um **algoritmo/programa em C** que escreva uma tabela de valores de lucros esperados em função do preço do ingresso, fazendo-se variar esse preço de R\$ 22,00 a R\$ 48,00 de R\$ 2,00 em R\$ 2,00. Escreva, ainda, o lucro máximo esperado, o preço do ingresso e a quantidade de ingressos vendidos para a obtenção desse lucro.

Estruturas de Repetição

25

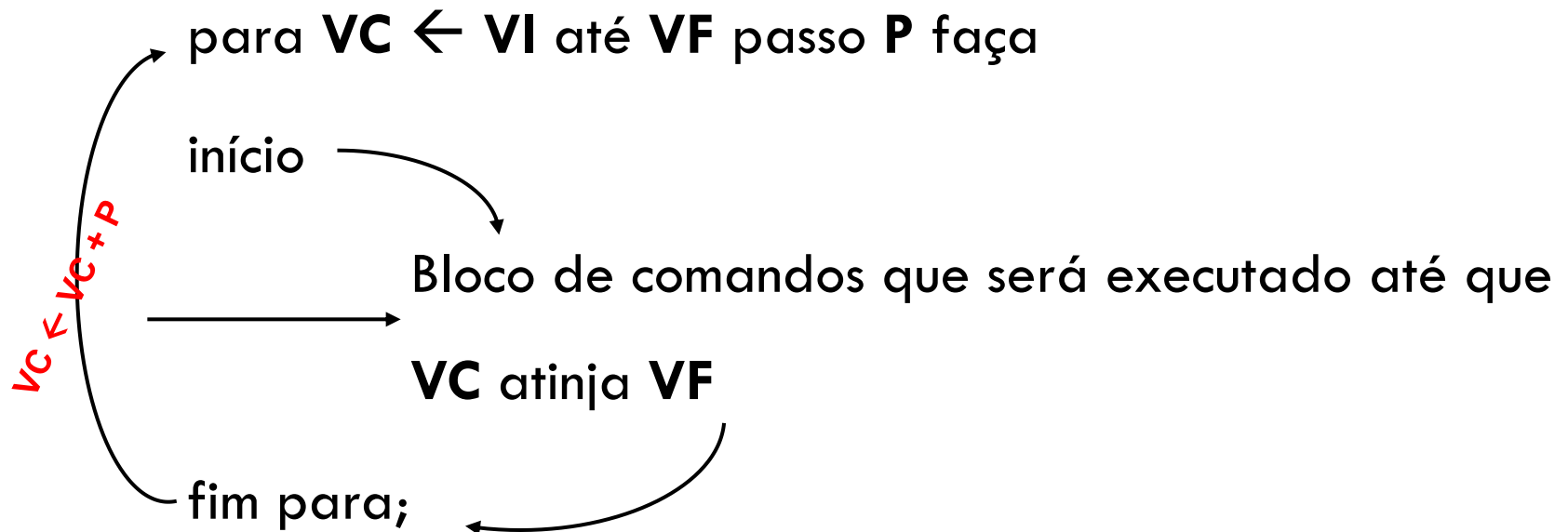
□ Estrutura de repetição com variável de controle

- Nas estruturas de repetição vistas até agora, podem ocorrer situações em que **não é possível prever** o número de vezes que a ação (bloco de comandos) será executada, sabe-se apenas que a repetição ocorrerá enquanto a condição for **satisfeita**.
- A estrutura de repetição com variável de controle repete a execução da mesma ação um **número definido de vezes**, pois possui limites fixos.
- Por este motivo, esta **estrutura só pode ser usada quando se sabe o número de vezes que se quer repetir** o mesmo bloco de comandos.

Estruturas de Repetição

26

□ Sintaxe – Algoritmo



Estruturas de Repetição

27

- Sabendo que:
 - ▣ VC – Variável de controle
 - ▣ VI – Valor iniciar atribuído a variável de controle
 - ▣ VF – valor final que a variável de controle pode atingir (é o limite)
 - ▣ P – passo – regra de contagem – como a variável de controle (VC) será atualizada a cada repetição

Estruturas de Repetição

28

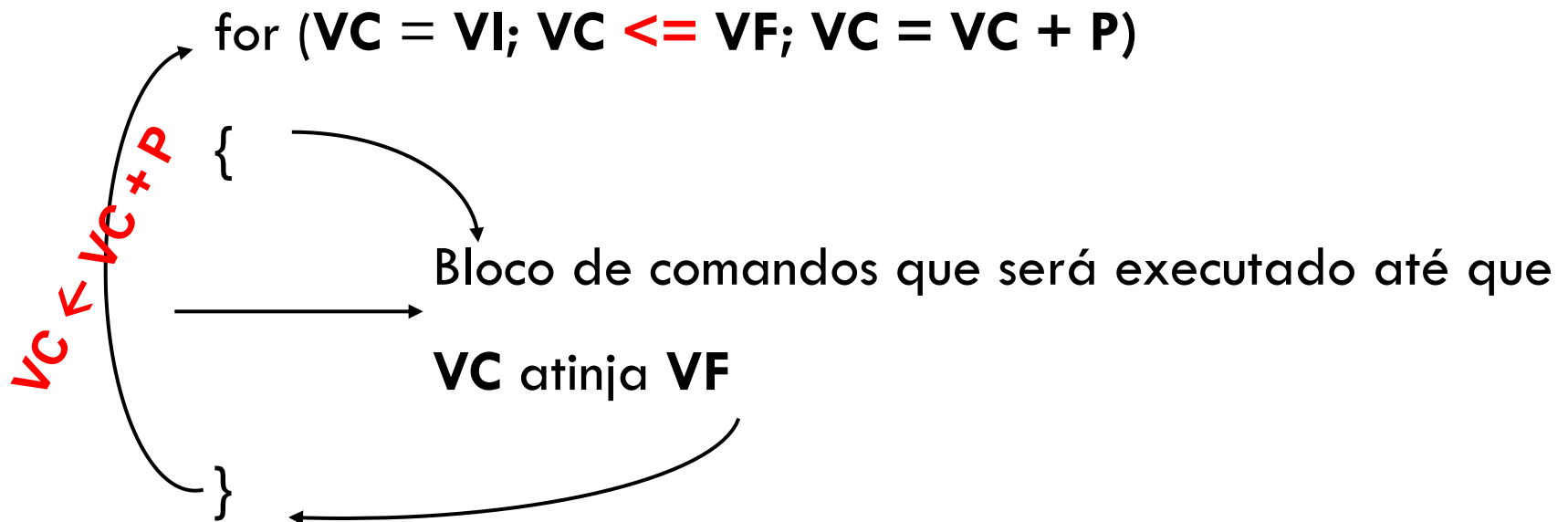
□ Contagem

- ▣ Pode-se ter contagem crescente e decrescente.
- ▣ Para isto deve-se seguir as seguintes orientações:
 - **contagem crescente:**
 - $VI \leq VF$
 - $P > 0$
 - **contagem decrescente:**
 - $VI \geq VF$
 - $P < 0$
- ▣ Sendo que **VC**, **VI**, **VF** e **P** são números inteiros

Estruturas de Repetição

29

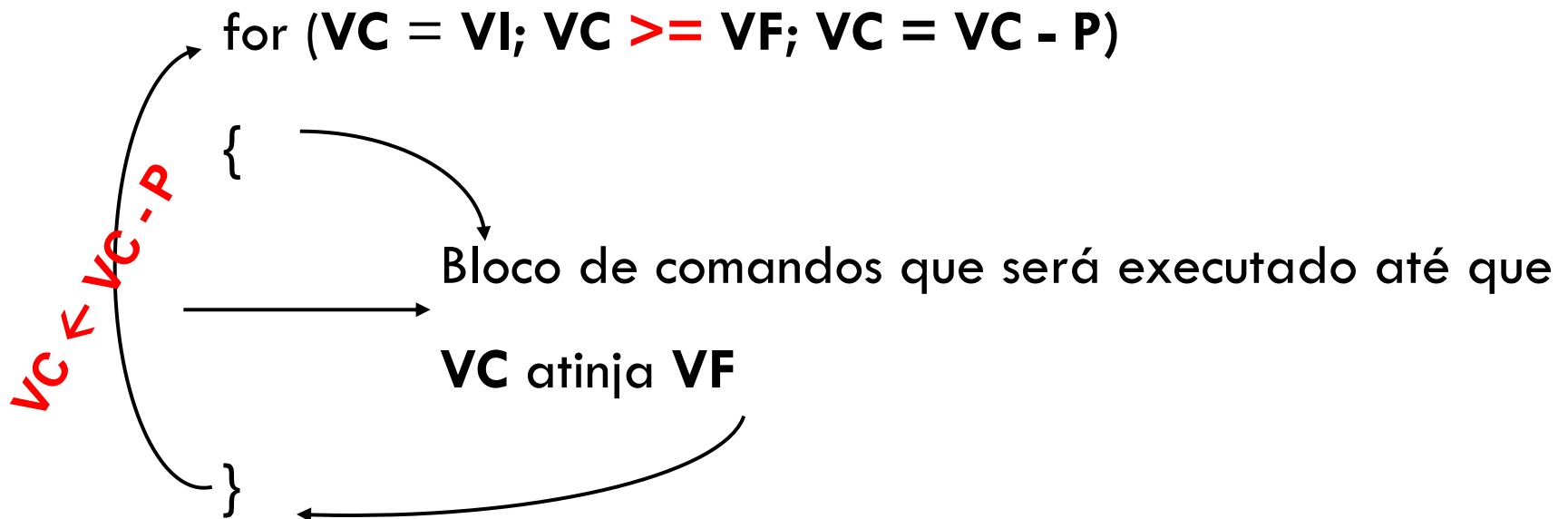
□ Sintaxe – Linguagem C – **Contagem Crescente**



Estruturas de Repetição

30

□ Sintaxe – Linguagem C – **Contagem Decrescente**



Estruturas de Repetição

31

□ Exemplo

1) Faça um algoritmo para mostrar os n° de 1 a 1500.

□ Ações:

- Contar 1 a 1500
- Mostrar o contador (VC)

□ Variável:

□ CONTADOR – inteiro

Estruturas de Repetição

32

Algoritmo Contar de 1 a 1500

Início

var

inteiro: CONTADOR;

para CONTADOR \leftarrow 1 até 1500 passo 1 faça

início

escreva (CONTADOR);

fim para;

Fim.

Estruturas de Repetição

33

```
#include <stdio.h>
#include <conio.h>
main()
{
    int CONTADOR;
    for (CONTADOR = 1; CONTADOR <=1500; CONTADOR = CONTADOR + 1)
    {
        printf("\n %d  \n", CONTADOR);
    }
    getch();
}
```

Estruturas de Repetição

34

□ Exercícios

- 2) Faça um algoritmo para mostrar os números de 1 a 100
- 3) Faça um algoritmo para mostrar os N° pares entre 1 a 100.
- 4) Faça um algoritmo para mostrar os N° pares entre 1 a 100, contando de 1 em 1.
- 5) Faça um algoritmo para ler 100 N° inteiro e mostrar somente os impares lidos.
- 6) Faça um algoritmo para ler um número inteiro e calcular o somatório:

$$\sum_{i=1}^n \frac{1}{i} = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$$

Solução Exercício 4

35

Algoritmo Mostrar Pares de 1 a 100;

Início

var

inteiro: CONTADOR;

para CONTADOR \leftarrow 1 até 100 passo 1 **faça**

início

se **CONTADOR** MOD 2 = 0 então

início

escreva(CONTADOR);

fim se;

fim para;

Fim.

Solução Exercício 5

36

Algoritmo ler número mostrar ímpar;

Início

var

inteiro: CONTADOR, num;

para CONTADOR \leftarrow 1 até 100 passo 1 **faça**

início

leia(num);

se num MOD 2 \neq 0 então

início

escreva(num);

fim se;

fim para;

Fim.

Comparação entre as estruturas de repetição com teste no início, final e variável de controle

Solução Exercício 6 – Repetição com teste no início

38

Algoritmo Somatório;

Início

var

inteiro: i, n;

real: somatorio;

somatorio \leftarrow 0.0;

i \leftarrow 1;

leia(n);

enquanto **i** \leq **n** **faça**

início

somatorio \leftarrow **somatorio** + (1.0 / **i**);

i \leftarrow **i** + 1;

fim enquanto;

escreva("Somatório = ", **somatorio**);

Fim.

Solução Exercício 6 – Repetição com teste no final

39

Algoritmo Somatório;

Início

var

inteiro: i, n;

real: somatorio;

somatorio \leftarrow 0.0;

i \leftarrow 1;

leia(n);

se n > 0 **então**

início

repita

somatorio \leftarrow somatorio + (1.0 / **i**);

i \leftarrow **i** + 1;

enquanto **i** <= **n**;

fim se;

escreva("Somatório = ", somatorio);

Fim.

Solução Exercício 6 – Repetição com variável de controle

40

Algoritmo Somatório;

Início

var

inteiro: i, n;

real: somatorio;

somatorio \leftarrow 0.0;

leia(n);

para i \leftarrow 1 até n passo 1 faça

início

somatorio \leftarrow somatorio + (1.0 / i);

fim para;

escreva("Somatório = ", somatorio);

Fim.

Exercício para entregar

41

- Uma loja utiliza o código V para compra à vista e P para compra a prazo. Faça um **algoritmo/programa em Linguagem C** que receba o código e o valor de 1500 compras.
- Calcule e mostre:
 - ▣ o valor total das compras à vista;
 - ▣ o valor total das compras a prazo;
 - ▣ o valor total das compras efetuadas;
 - ▣ o valor da primeira prestação das compras a prazo, sabendo que essas serão pagas em três vezes;
 - ▣ o valor da compra mais cara efetuada.

Bibliografia

42

□ Básica

ASCENCIO, A. F. G, CAMPOS, E. A. V. **Fundamentos da Programação de Computadores**: algoritmos, Pascal e C/C++ e Java. Longman, 2007.

FORBELLONE, L. V., EBERSPACHER, H. F. **Lógica de Programação**: a construção de algoritmos e estruturas de dados. Prentice Hall, 2005.

ZIVIANI, Nivio. **Projeto de Algoritmos com Implementações em Pascal e C**. 2.ed. Thomson Pioneira, 2004.

□ Complementar

FARRER, H et al. **Algoritmos estruturados**. 3 ed. Rio de Janeiro: LTC, 1999. 284 p.

MANZANO, J. A. N. G.; **Estudo dirigido de algoritmos**. 9. ed. São Paulo: Érica, 2004.

LOUNDON, L. **Algoritmos em C**. São Paulo: Ciência Moderna, 2000.

ASCENCIO, A. F. G.; CAMPOS, E.A.V. **Fundamentos da programação de computadores**: algoritmo, pascal e C++. São Paulo: Pearson Prentice Hall, 2002. 355 p