

ESTRUTURAS DE CONTROLE

ESTRUTURAS DE REPETIÇÃO



Introdução à Ciência da Computação

Estruturas de Controle

- ESTRUTURA SEQUENCIAL
- ESTRUTURAS CONDICIONAIS
 - Estrutura Condicional Simples
 - Estrutura Condicional Composta
 - Seleção entre duas ou mais Sequências de Comandos
- ESTRUTURA DE REPETIÇÃO
 - Repetição com Teste no Início
 - Repetição com Teste no Final
 - Repetição Contada

Estruturas de Controle

- ESTRUTURA SEQUENCIAL
- ESTRUTURAS CONDICIONAIS
 - Estrutura Condicional Simples
 - Estrutura Condicional Composta
 - Seleção entre duas ou mais Sequências de Comandos
- ESTRUTURA DE REPETIÇÃO
 - Repetição com Teste no Início
 - Repetição com Teste no Final
 - Repetição Contada

Estruturas

- ESTRUTURAS

- ESTRUTURAS

- Estrutura Com
- Estrutura Com
- Seleção entre d
- Comandos

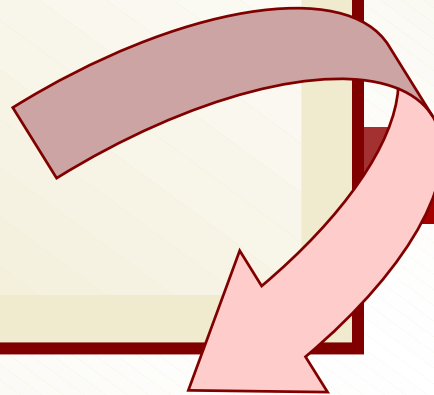
- ESTRUTURA DE REPETIÇÃO

- Repetição com Teste no Início
- Repetição com Teste no Final
- Repetição Contada

- Permite que uma sequência de comandos seja executada **repetidamente** até que uma determinada **condição de interrupção** seja satisfeita.

Estruturas de Controle

- ESTRUTURA SEQUENCIAL
- ESTRUTURAS CONDICIONAIS
 - Estrutura Condicional Simples
 - Estrutura Condicional Composta
 - Seleção entre duas ou mais Sequências de Comandos
- ESTRUTURA DE REPETIÇÃO
 - **Repetição com Teste no Início**
 - Repetição com Teste no Final
 - Repetição Contada

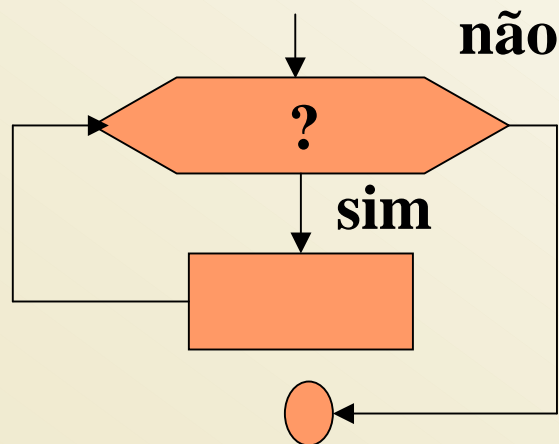


Repetição com Teste no Início

Repetição com Teste no Início

formas de representação no algoritmo

Fluxograma

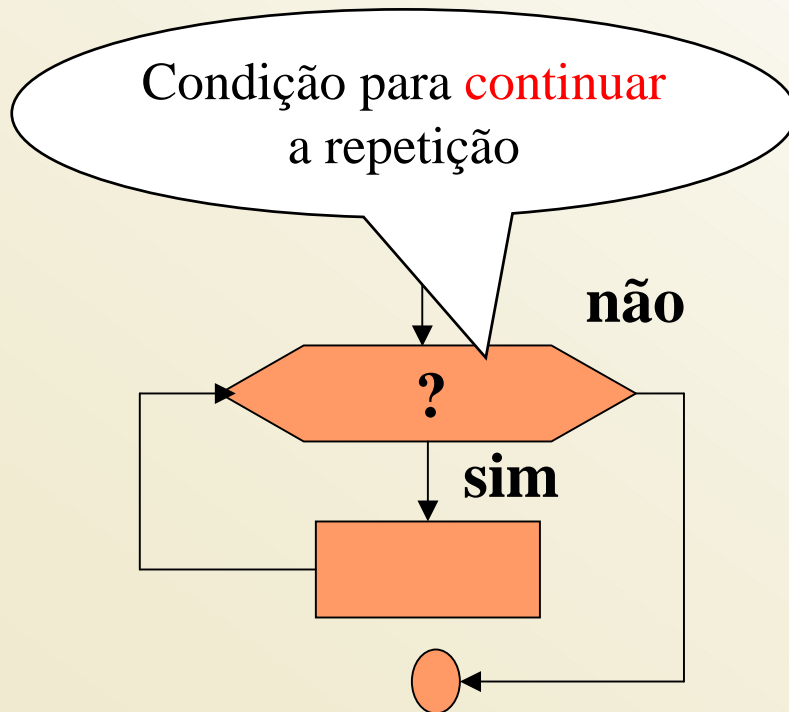


Linguagem Algorítmica

```
enquanto <condição>  
| faça comando  
fim enquanto
```

Repetição com Teste no Início

formas de representação no algoritmo



Condição para **continuar** a repetição

enquanto <condição>
| faça comando
fim enquanto

Repetição com Teste no Início

```
while <expressão lógica>  
    comando;
```


Repetição com Teste no Início **Exemplo**

Desenvolver algoritmo para o problema:

Ler a variável SEXO.

Enquanto o valor fornecido não for correto
(M,m,F,f) exibir mensagem de erro e solicitar
novamente a leitura.

Quando o valor fornecido estiver correto, ler a
variável ANO.

Escrever ANO e SEXO

Repetição com Teste no Início

Exemplo

alg TESTE1

Início

declarações

ler (SEXO)

enquanto SEXO <> (f, F, m, M)

início

escrever ("ERRO")

ler (SEXO)

fim

ler (ANO)

escrever (ANO,SEXO)

fim

Repetição com Teste no Início

Exemplo

programa TESTE1
Início
declarações
ler (SEXO)
enquanto SEXO <> (f, F, m, M)
 início
 escrever ("ERRO")
 ler (SEXO)
 fim
fim enquanto
ler (ANO)
escrever (ANO, SEXO)
fim

laço {

Esta variável de controle deve ter um valor conhecido. Neste caso ela foi lida antes.

A variável de controle deve ter seu valor modificado dentro do "laço". Neste caso a variável foi lida novamente.

Repetição com Teste no Início Contador/Acumulador **Exemplo**

Desenvolver um programa para resolver o seguinte problema:

- Ler um conjunto de valores correspondentes aos pontos que alunos obtiveram em um teste.
- Quando o valor fornecido for um número negativo, isto é um sinal que não existem mais pontos para serem lidos.
- Contar quantos alunos fizeram o teste.
- Escrever o total de alunos

Repetição com Teste no Início Contador/Acumulador **Exemplo**

```
programa TESTE2
declarações
inicio
  N ← 0
  ler (PONTOS)
  enquanto PONTOS ≥ 0
    início
      N ← N + 1
      ler (PONTOS)
    fim
  fim enquanto
  escrever (N)
fim
```

Repetição com Teste no Início Contador/Acumulador **Exemplo**

programa TESTE2
declarações
início

$N \leftarrow 0$

ler (PONTOS)

enquanto PONTOS ≥ 0

início

$N \leftarrow N + 1$

ler (PONTOS)

fim

fim enquanto

escrever (N)

fim

N
contador/acumulador

N deve ser zerado e então incrementado em 1 dentro do comando de repetição

Repetição com Teste no Início Contador/Acumulador **Exemplo**

Desenvolver um programa para resolver o seguinte problema:

- Ler um conjunto de valores correspondentes aos pontos que alunos obtiveram em um teste.
- Quando o valor fornecido for um número negativo, isto é um sinal que não existem mais pontos para serem lidos.
- Contar e escrever quantos alunos fizeram o teste.
- Contar e escrever quantos alunos tiveram nota baixa ($\text{PONTOS} < 50$)
- Contar e escrever quantos alunos tiveram nota alta ($100 \leq \text{PONTOS} \leq 150$)
 - **OBS: O intervalo válido das notas é: $[0,150]$**

Repetição com Teste no Início Contador/Acumulador **Exemplo**

Desenvolver um programa para resolver o seguinte problema:

- Ler um conjunto de valores referentes aos pontos que alunos obtiveram em uma prova.
- Quando o valor for negativo, isto é um sinal de que os dados não devem ser lidos.
- Contar e escrever quantos alunos fizeram o teste.
- Contar e escrever quantos alunos tiveram nota baixa ($\text{PONTOS} < 50$)
- Contar e escrever quantos alunos tiveram nota alta ($100 \leq \text{PONTOS} \leq 150$)

São necessários
3 contadores

programa TESTE2

declarações

inicio

$N \leftarrow 0$

$ALTAS \leftarrow 0$

$BAIXAS \leftarrow 0$

ler (PONTOS)

enquanto PONTOS ≥ 0

início

se PONTOS ≥ 100

então $ALTAS \leftarrow ALTAS + 1$

senão se PONTOS < 50

então $BAIXAS \leftarrow BAIXAS + 1$

fim-se

fim-se

$N \leftarrow N + 1$

ler (PONTOS)

fim

fim enquanto

escrever (N,BAIXAS, ALTAS)

fim

Fazer crítica para a entrada
de dados maior que 150
Ou arrumar a condição do
primeiro SE-Então

programa TESTE2

declarações

início

N ← 0

ALTAS ← 0

BAIXAS ← 0

ler (PONTOS)

enquanto PONTOS ≥ 0

faça início

se PONTOS ≥ 100

então **ALTAS** ← **ALTAS** + 1

senão se PONTOS < 50

então **BAIXAS** ← **BAIXAS** + 1

fim-se |

fim-se

N ← **N** + 1

ler (PONTOS)

fim

fim enquanto

escrever (N,BAIXAS, ALTAS)

fim

3 contadores

Repetição com Teste no Início

Exemplo

Desenvolver programa para resolver o problema:

- Ler o valor de dois números inteiros N e M, sendo N menor ou igual a M.
- Calcular e exibir a soma de todos os números ímpares positivos entre N e M, inclusive.
- Exemplos: $N = 1, M = 10$

$$N = -7, M = +7$$

```
programa REPET10
declarar N,M,NRO,SOMA inteiros
início
  solicitar entrada do limite inferior do intervalo, ler (N)
  solicitar entrada do limite superior do intervalo, ler (M)
  se N < M
    então início
      se N for número par
        então  $NRO \leftarrow N+1$ 
      senão  $NRO \leftarrow N$ 
      fim-se
       $SOMA \leftarrow 0$ 
      enquanto ( $NRO \leq M$ )
        faça início
          se  $NRO > 0$ 
            então  $SOMA \leftarrow SOMA + NRO$ 
          fim-se
           $NRO \leftarrow NRO + 2$ 
        fim
      fim-enquanto
      escrever (N,M, SOMA)
    fim
  senão escrever (INTERVALO INCORRETO)
fim-se
fim-programa
```

Fazer crítica para entrada de dados negativos (quando N e M são negativos) para tornar o programa mais eficiente

programa REPET10

declarar N,M,NRO,SOMA inteiros

início

solicitar entrada do limite inferior do intervalo, ler (N)

solicitar entrada do limite superior do intervalo, ler (M)

se $N < M$

então início

se N for número par

então $NRO \leftarrow N$

senão $NRO \leftarrow N + 1$

fim-se

$SOMA \leftarrow 0$

enquanto $(NRO \leq M)$

faça início

se $NRO > 0$

então $SOMA \leftarrow SOMA + NRO$

fim-se

$NRO \leftarrow NRO + 2$

fim

fim-enquanto

escrever (N,M, SOMA)

fim

senão escrever (INTERVALO INCORRETO)

fim-se

fim-programa

SOMA
acumulador

SOMA deve ser zerado

SOMA deve ser então
incrementado dentro do
comando de repetição

```
programa REPET10
declarar N,M,NRO,SOMA inteiros
início
  solicitar entrada do limite inferior do intervalo, ler (N)
  solicitar entrada do limite superior do intervalo, ler (M)
  se N < M
    então início
      se N for número par
        então  $NRO \leftarrow N+1$ 
        senão  $NRO \leftarrow N$ 
      fim-se
       $SOMA \leftarrow 0$ 
      enquanto ( $NRO \leq M$ )
        faça início
          se  $NRO > 0$ 
            então  $SOMA \leftarrow SOMA + NRO$ 
          fim-se
           $NRO \leftarrow NRO + 2$ 
        fim
      fim-enquanto
      escrever (N,M, SOMA)
    fim
  senão escrever (INTERVALO INCORRETO)
fim-se
fim-programa
```

Se o limite inferior (N) for par o primeiro número é o próximo a partir do limite

```
programa REPET10
declarar N,M,NRO,SOMA inteiros
início
  solicitar entrada do limite inferior do intervalo, ler (N)
  solicitar entrada do limite superior do intervalo, ler (M)
  se N < M
    então início
      se N for número par
        então  $NRO \leftarrow N+1$ 
        senão  $NRO \leftarrow N$ 
      fim-se
       $SOMA \leftarrow 0$ 
      enquanto  $(NRO \leq M)$ 
        início
          se  $NRO > 0$ 
            então SOMA
          fim-se
           $NRO \leftarrow NRO + 2$ 
        fim
      fim-enquanto
      escrever (N,M, SOMA)
    fim
  senão escrever (INTERVALO INCORRETO)
fim-se
fim-programa
```

Se o limite inferior (N) for ímpar o primeiro número é o próprio limite

```
programa REPET10
declarar N,M,NRO,SOMA inteiros
início
  solicitar entrada do limite inferior do intervalo, ler (N)
  solicitar entrada do limite superior do intervalo, ler (M)
  se N < M
    então início
      se N for número par
        então NRO ← N+1
        senão NRO ← N
      fim-se
      SOMA ← 0
      enquanto (NRO ≤ M)
        início
          se NRO > 0
            então SOMA ← SOMA + NRO
          fim-se
          NRO ← NRO + 2
        fim
      fim-enquanto
      escrever (N,M, SOMA)
    fim
  senão escrever (INTERVALO INCORRETO)
fim-se
fim-programa
```

Verifica se o
número é
positivo


```
programa REPET10
declarar N,M,NRO,SOMA inteiros
início
  solicitar entrada do limite inferior do intervalo, ler (N)
  solicitar entrada do limite superior do intervalo, ler (M)
  se N < M
    início
      se N for número par
        então NRO ← N+1
        senão NRO ← N
      fim-se
      SOMA ← 0
      enquanto (NRO ≤ M)
        início
          se NRO > 0
            então SOMA ← SOMA + NRO
          fim-se
          NRO ← NRO + 2
        fim
      fim-enquanto
      escrever (N,M, SOMA)
    fim
  senão escrever (INTERVALO INCORRETO)
  fim-se
fim-programa
```

Calcula a soma
dos impares
positivos até M

Repetição com Teste no Início

Exemplo

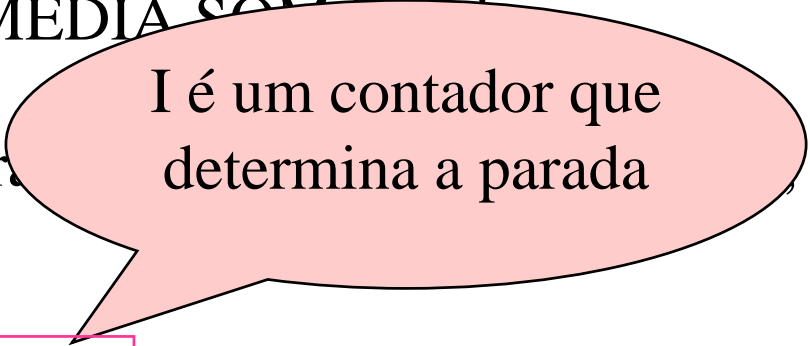
Desenvolver programa para resolver o problema:

- Ler N números reais (o valor de N também deve ser lido), sendo impressa a média desses números.

```
programa REPET11
declarar N, I inteiro
          NUM,SOMAMEDIA real
inicio
  solicitar a entrada da quantidade de números, ler (N)
  SOMA ← 0
  I ← 1
  enquanto I ≤ N
    início
      solicitar a entrada do número, ler (NUM)
      SOMA ← SOMA + NUM
      I ← I + 1
    fim
  fim enquanto
  MEDIA ← SOMA / N
  escrever (MEDIA)
fim do programa
```

Problema se $N = 0!!$
Fazer crítica

```
programa REPET11
declarar N, I inteiro
          NUM, MEDIA SOMA
inicio
  solicitar a entrada do N, ler (N)
  SOMA  $\leftarrow$  0
  I  $\leftarrow$  1
  enquanto I  $\leq$  N
    faça início
      solicitar a entrada do número, ler (NUM)
      SOMA  $\leftarrow$  SOMA + NUM
      I  $\leftarrow$  I + 1
    fim
  fim enquanto
  MEDIA  $\leftarrow$  SOMA / N
  escrever (MEDIA)
fim do programa
```



programa REPET11
declarar N, I inteiro

I é um contador que
determina a parada

NUM, MEDIA, SOMA real

inicio

solicitar a entrada da qual

I é declarado
como inteiro

SOMA \leftarrow 0

I \leftarrow 1

I é inicializado

enquanto $I \leq N$

faça início

solicitar a entrada do número, ler (NUM)

SOMA \leftarrow SOMA + NUM

I \leftarrow I + 1

I é incrementado de 1

fim

fim enquanto

MEDIA \leftarrow SOMA /N

escrever (MEDIA)

fim do programa

programa REPET12
declarar N, I inteiro

NUM, MEDIA, SOMA real

inicio

solicitar a entrada da quantidade de números, ler (N)

SOMA \leftarrow 0

I \leftarrow 0

enquanto $I \leq N-1$

faça início

I \leftarrow I + 1

solicitar a entrada do número, ler (NUM)

SOMA \leftarrow SOMA + NUM

fim

fim enquanto

MEDIA \leftarrow SOMA / N

escrever (MEDIA)

fim do programa

I é um contador que
delimitar a parada

I é inicializado

I é incrementado de 1

Outra maneira de contar
os números

```

programa REPET11
declarar N, I inteiro
        NUM,MEDIA,SOMA real
inicio
    solicitar a entrada da quantidade de números, ler (N)
    SOMA ← 0
    I ← 1
    enquanto I ≤ N
        faça início
            solicitar a entrada do número, ler (NUM)
            SOMA ← SOMA+ NUM
            I ← I +1
        fim
    fim enquanto
    MEDIA ← SOMA/N
    escrever (MEDIA)
fim do programa

```

A leitura do
número é feita
antes do
incremento

A leitura do
número é feita
depois do
incremento

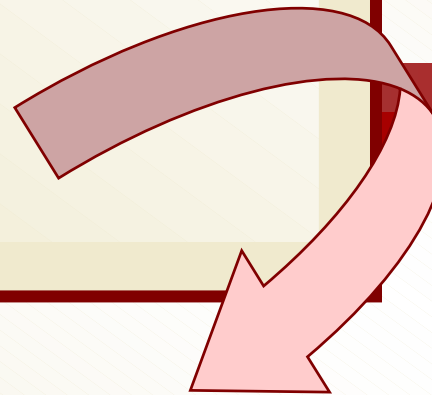
```

programa REPET12
declarar N, I inteiro
        NUM,MEDIA,SOMA real
inicio
    solicitar a entrada da quantidade de números, ler(N)
    SOMA ← 0
    I ← 0
    enquanto I ≤ N-1
        faça início
            I ← I +1
            solicitar a entrada do número, ler (NUM)
            SOMA ← SOMA + NUM
        fim
    fim enquanto
    MEDIA ← SOMA /N
    escrever (MEDIA)
fim do programa

```

Estruturas de Controle

- ESTRUTURA SEQUENCIAL
- ESTRUTURAS CONDICIONAIS
 - Estrutura Condicional Simples
 - Estrutura Condicional Composta
 - Seleção entre duas ou mais Sequências de Comandos
- ESTRUTURA DE REPETIÇÃO
 - Repetição com Teste no Início
 - **Repetição com Teste no Final**
 - Repetição Contada

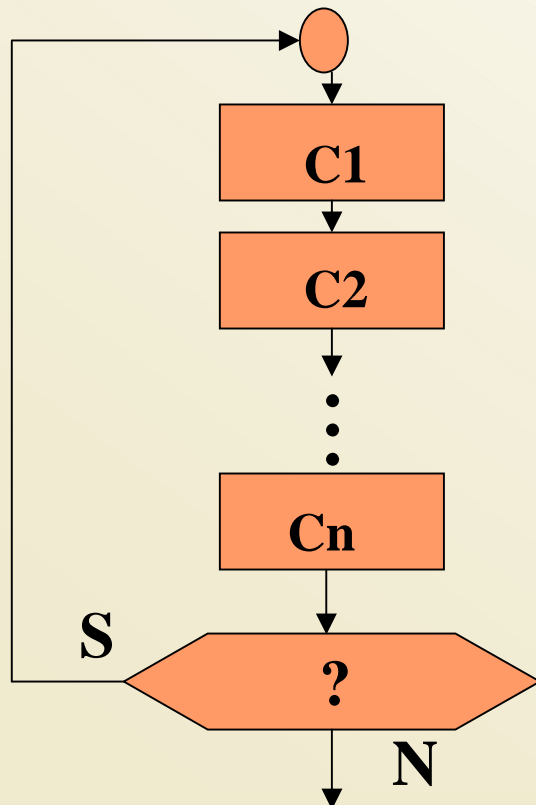


Repetição com Teste no Final

Repetição com Teste no Final

formas de representação no algoritmo

Fluxograma



Linguagem Algorítmica

faça

| comando 1

| comando 2

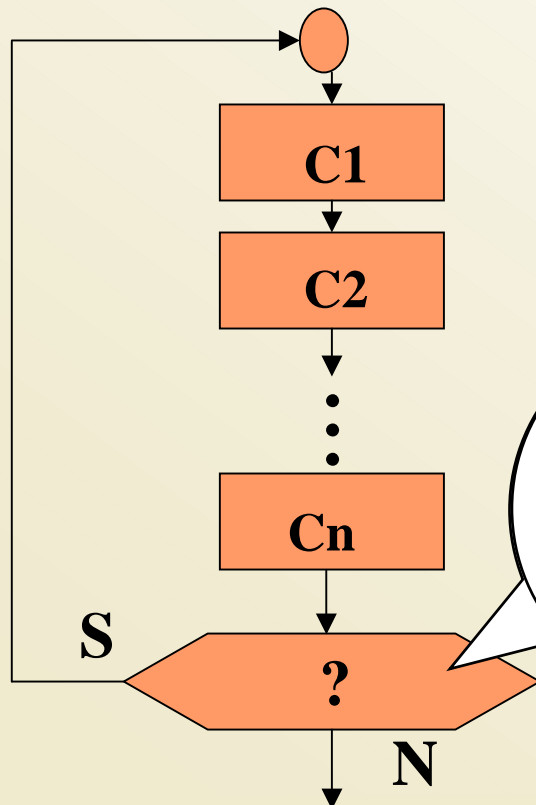
| comando n

enquanto <condição>

Repetição com Teste no Final

formas de representação no algoritmo

Fluxograma



Linguagem Algorítmica

faça

comando 1

comando 2

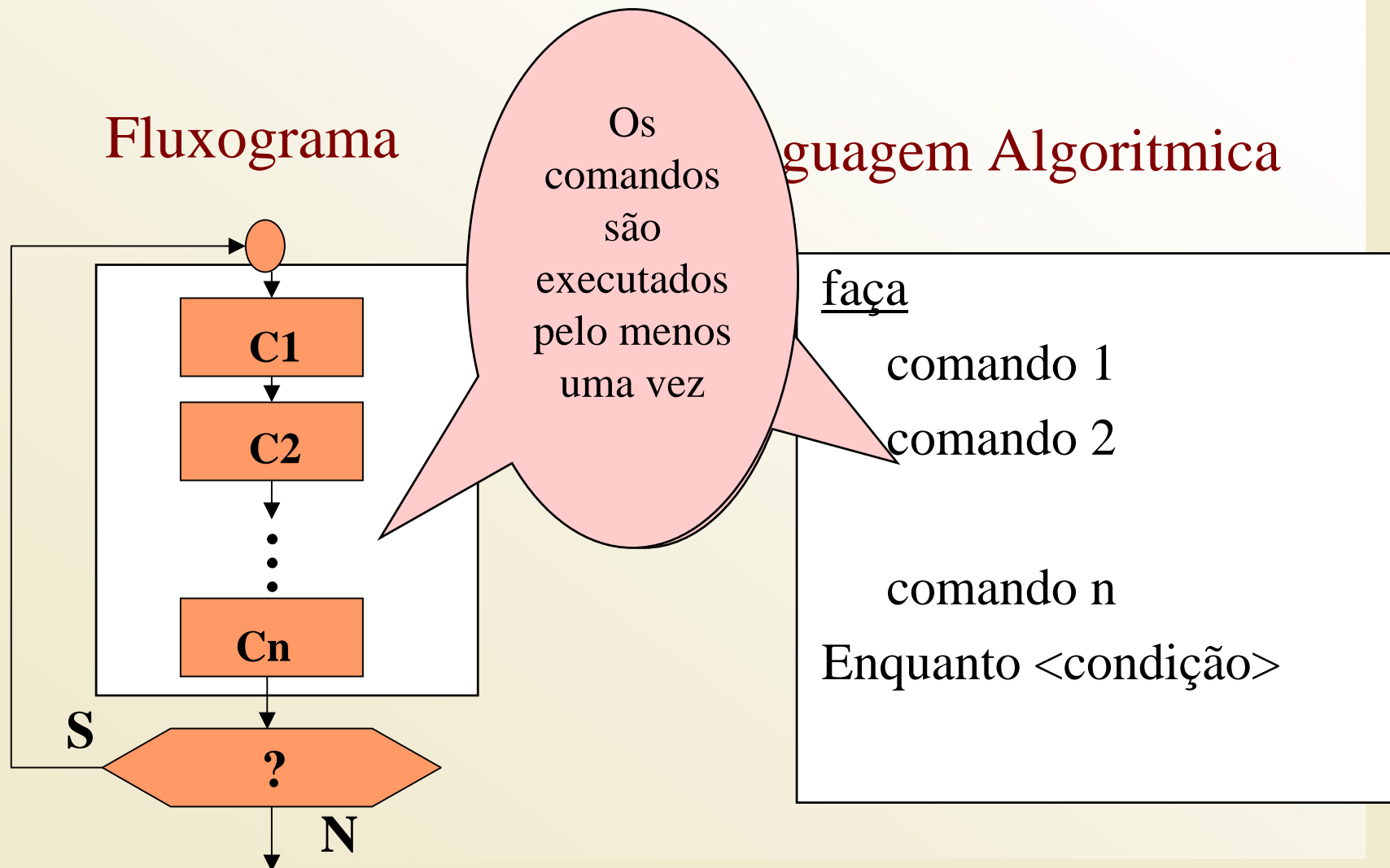
comando n

enquanto <condição>

Condição
de **parada**
da
repetição

Repetição com Teste no Final

formas de representação no algoritmo



Repetição com Teste no Final

do

<seqüência de comandos>

while <expressão lógica>;

Repetição com Teste no Final

Exemplo

Desenvolver algoritmo para o problema:

Calcular e exibir o valor da série.

$$S = \frac{1}{1} + \frac{3}{2} + \frac{5}{3} + \frac{7}{4} + \dots + \frac{99}{50}$$

programa TESTE4

declarações

inicio

S ← 0

Inicializa a somatória

N ← 1

Inicializa o numerador

D ← 1

Inicializa o denominador

faça

$$S \leftarrow S + \frac{N}{D}$$

Calcula os termos da série

N ← N+2

Calcula o próximo numerador

D ← D+1

Calcula o próximo denominador

enquanto D ≤ 50

escrever (S)

fim

Estruturas de Controle

- ESTRUTURA SEQUENCIAL
- ESTRUTURAS CONDICIONAIS
 - Estrutura Condicional Simples
 - Estrutura Condicional Composta
 - Seleção entre duas ou mais Sequências de Comandos
- ESTRUTURA DE REPETIÇÃO
 - Repetição com Teste no Início
 - Repetição com Teste no Fim
 - **Repetição Contada**

31

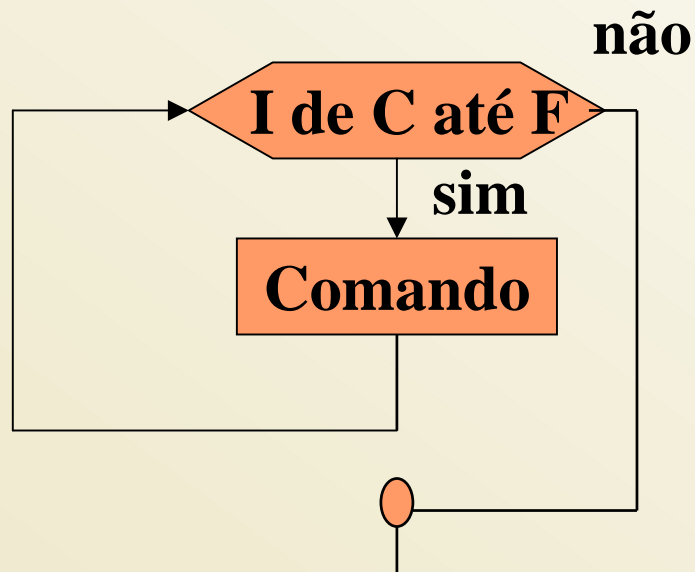
Repetição Contada

este comando deve ser usado quando se sabe a priori o número de repetições

Repetição Contada

formas de representação no algoritmo

Fluxograma



Linguagem Algorítmica

Para I de C até F
| faça comando
fim para

Repetição Contada

I - variável de controle

C - valor inicial da variável

F valor final da variável

- OBSERVAÇÕES

(teóricamente)

- **I**, **C** e **F** não devem ser modificadas
- a variável **I** fica indefinida ao terminar as repetições

Repetição Contada

```
for ( <inicialização>; <condição>; <incremento> )  
    <comando>;
```

A variável de controle assume o valor inicial e automaticamente é incrementada, a cada repetição, até o valor final

```
for ( int x = 0; x < 10; x++ ) {  
    printf( "%d", x );  
}
```

Repetição Contada

Exemplo

Desenvolver algoritmo para o problema:

Dado um conjunto de N números, calcular a média aritmética dos mesmos

programa TESTE6

declarações

inicio

ler (N)

$S \leftarrow 0$

para I de 1 até N

início

ler (X)

$S \leftarrow S + X$

fim

fim para

escrever (S/N) {Fazer crítica se $N = 0$ }

fim

quantidade de notas

nota

Repetição Contada

Exemplo

Desenvolver algoritmo para o problema:

Ler um número inteiro ≥ 0 e calcular seu fatorial.

programa TESTE7

declarações

inicio

| ler (N)

| FAT ← 1

Inicializa o acumulador

| para I de 1 até N

| | faça FAT ← FAT * I

| fim para

| escrever (FAT)

fim

Comando de Repetição

Exemplo

- Exibir a tabuada de qualquer número sendo que este número é fornecido pelo usuário.
- O programa deve permitir que sejam exibidas tabuadas de diversos números até que o usuário não deseje mais continuar.

programa TABUADA
declarar N, CON inteiro
C: caracter
CONTINUA : lógico

Utilizando Repetição com Teste no Início

```
Início
CONTINUA ← verdade
enquanto (CONTINUA = verdade) faça
  início
    solicitar o fornecimento do número, ler (N)
    CON ← 1
    enquanto (CON ≤ 10) faça
      início
        escrever(CON, ' x ', N, ' = ', CON * N)
        CON ← CON+1
      fim
    fim enquanto
    escrever('DESEJA FAZER OUTRA TABUADA ?(S/N)')
    ler (C)
    se C <> 's' ou C <> 'S'
      entao CONTINUA ← falso
    fim-se
  fim
fim enquanto
fim do programa
```


Utilizando Repetição com Teste no Final

programa TABUADA2
declarar N, CON inteiro
C: caracter
CONTINUA : lógico

Início

CONTINUA ← verdade

repita

solicitar o fornecimento do número, ler (N)

CON ← 1

repita

escrever(CON, ' x ', N, ' = ', CON * N)

CON ← CON+1

até que (CON > 10)

escrever('DESEJA FAZER OUTRA TABUADA ?(S/N)') ler (C)

se C <> 's' ou C <> 'S'

entao CONTINUA ← falso

fim-se

até que (CONTINUA = falso)

fim do programa

```
programa TABUADA3  
declarar N, CON inteiro  
C: caracter  
CONTINUA : lógico
```

Início

```
CONTINUA ← verdade
```

```
enquanto (CONTINUA = verdade) faça
```

```
  início
```

```
    solicitar o fornecimento do número, ler (N)
```

```
    para CON =1 até 10 faça
```

```
      escrever(CON, ' x ', N, ' = ', CON * N)
```

```
    fim para
```

```
    escrever('DESEJA FAZER OUTRA TABUADA ?(S/N)')
```

```
    ler (C)
```

```
    se C <> 's' ou C <> 'S'
```

```
      entao CONTINUA ← falso
```

```
    fim-se
```

```
  fim
```

```
fim enquanto
```

```
fim do programa
```

Aqui não é possível utilizar a estrutura de repetição contada pois esta exige que o número de repetições seja finito e determinado

Utilizando
Repetição
Contada

ESTRUTURAS DE CONTROLE

ESTRUTURAS DE REPETIÇÃO

fim