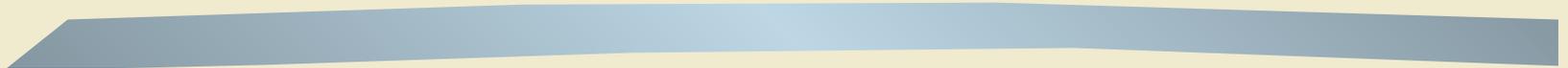


# ESTRUTURAS DE CONTROLE

## ESTRUTURAS DE REPETIÇÃO



# Estruturas de Controle

- ESTRUTURA SEQUENCIAL
- ESTRUTURAS CONDICIONAIS
  - Estrutura Condicional Simples
  - Estrutura Condicional Composta
  - Seleção entre duas ou mais Seqüências de Comandos
- **ESTRUTURA DE REPETIÇÃO**
  - Estrutura de Repetição em Algoritmos
  - Estrutura de Repetição em Linguagem C
  - Comando de controle de laço

# Estrutura de Repetição

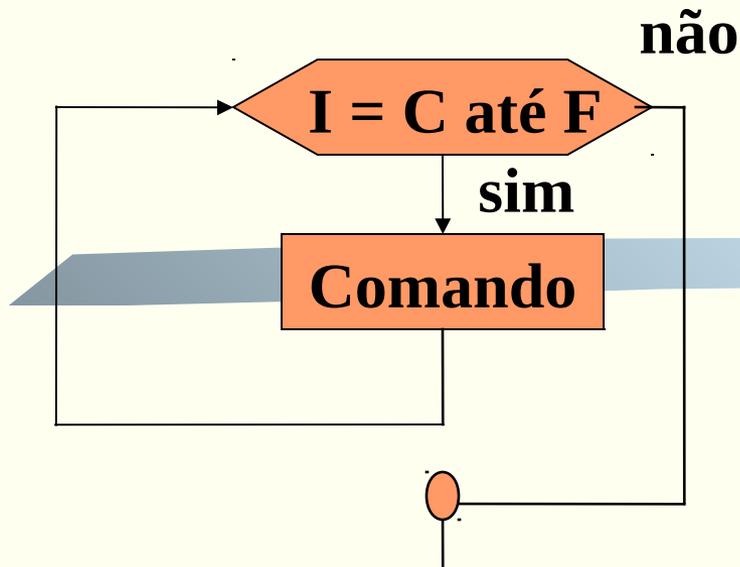
- Uma estrutura de repetição é utilizada quando um comando ou um bloco de comandos deve ser repetido.
- A quantidade de repetições pode ser fixa ou pode depender de uma determinada condição.
- O teste da condição pode ocorrer no início ou no final da estrutura de repetição.

# Estrutura de Repetição em Algoritmo

- Três tipos de estruturas de repetição serão consideradas na elaboração de Algoritmos:
  - ✓ Repetição Contada
  - ✓ Repetição com teste no início
  - ✓ Repetição com teste no final

# Repetição Contada

Fluxograma



Linguagem Algorítmica

```
para I = 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

# Estrutura PARA

```
para i = valor_inicial até valor_final faça  
    Instrução_1;  
    i = i+tamanho_passo;  
fim-para
```

```
para i = valor_inicial até valor_final faça  
    Instrução_1;  
    Instrução_2;  
    .....  
    Instrução_N;  
    i = i+tamanho_passo;  
fim-para
```

# Repetição Contada

## Exemplo

**Desenvolver algoritmo para o problema:**

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

Algoritmo média\_notas

var

inteiro: i,N;

real: S, X;

inicio

ler N;

S = 0;

para I = 1 até N

faça ler X;

S = S + X;

fim-para

escrever (S/N);

fim

quantidade de notas

nota

# Repetição Contada

## Exemplo

**Desenvolver algoritmo para o problema:**

Ler um número inteiro  $\geq 0$  e calcular seu fatorial.

Algoritmo fatorial

inteiro : N, FAT, I;

inicio

ler N

FAT = 1

Inicializa o multiplicador

para I = 1 até N

| faça FAT = FAT \* I

fim para

escrever (FAT)

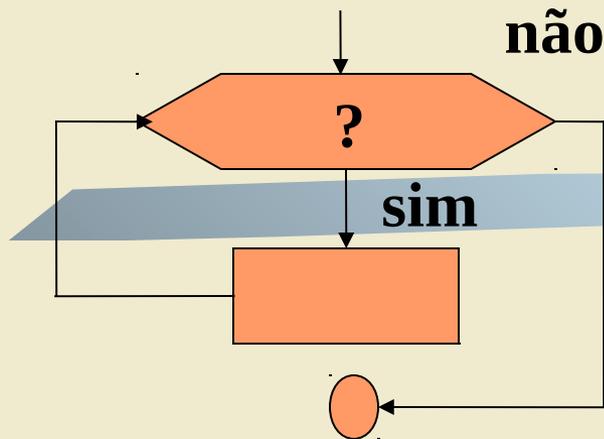
fim

# Repetição com Teste no Início

formas de representação no algoritmo

Fluxograma

Linguagem Algorítmica



enquanto

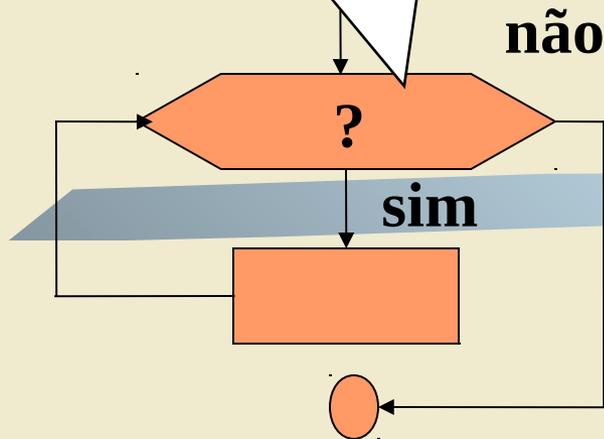
| <condição>

faça comando;

fim-enquanto

# Repetição com Teste no Início formas de representação no

Condição para **continuar** a repetição



Condição para **continuar** a repetição

enquanto

| <condição>

faça comando;

fim-enquanto

# Estrutura ENQUANTO

- Estrutura ENQUANTO

```
enquanto (condição) faça  
    instrução_1;  
fim-enquanto
```

```
enquanto (condição) faça  
    Instrução_1;  
    Instrução_2;  
    ....  
    instrução_n;  
fim-enquanto
```

# Repetição com Teste no Início Exemplo

Algoritmo TESTE1

var

caracter : sexo;

início

escreve(“sexo:”);

ler(sexo);

enquanto ((sexo != 'f')&&(sexo !='F')&&(sexo != 'm')  
&&(sexo != 'M'))

faça escreve(“erro”);

ler (sexo);

fim enquanto

escrever (sexo)

fim

# Repetição com Teste no Início Exemplo

Algoritmo TESTE1

var

caracter : sexo;

inicio

escreve("sexo: ");

ler(sexo);

**laço** { enquanto ((sexo != 'f') && (sexo != 'F') && (sexo != 'm') && (sexo != 'M'))

faça escreve("erro");

ler (sexo);

fim enquanto

escrever (sexo)

fim

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**

programa TESTE2

var

inteiro pontos,n;

inicio

n=0

ler(pontos)

enquanto ( pontos  $\geq$  0)

faça início

n=n+1

ler(pontos)

fim

fim enquanto

escrever (n)

fim 03/12/16

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

programa TESTE2

var

inteiro pontos,n;

inicio

n=0

ler(pontos)

enquanto ( pontos  $\geq$

faça início

n = n + 1

ler(pontos)

fim

fim enquanto

escrever (n)

fim 03/12/16



**n**  
contador/acumulador

The diagram shows a pink rounded rectangle representing a variable. Inside the rectangle, the letter 'n' is written in bold black font. Below 'n', the text 'contador/acumulador' is written in black font. A blue arrow points from the code line 'n = n + 1' to the bottom-left corner of the rectangle. Another blue arrow points from the right side of the rectangle to the right edge of the slide.

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 ( $100 \leq \text{PONTOS} \leq 150$ )

Repetição com Teste no Início

# Contador/Acumulador **Exemplo**

**Desenvolver um programa para resolver o seguinte problema:**

- Ler um conjunto de dados referentes aos pontos que foram obtidos em um teste.
- Quando o valor for zero ou negativo, isto indica que o aluno não tem mais pontos para ser considerado.
- Contar e escrever a quantidade de alunos que passaram o teste.
- Contar e escrever a quantidade de alunos que tiveram nota baixa ( $PONTOS < 50$ ).
- Contar e escrever quantos alunos tiveram nota alta ( $100 \leq PONTOS \leq 150$ ).

São necessários  
3 contadores

programa TESTE2

var inteiro: n,altas,baixas, pontos;

inicio

n= 0;

altas= 0;

baixas= 0;

ler(pontos)

enquanto(pontos  $\geq$  0)

faça início

se (pontos  $\geq$  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

programa TESTE2

var inteiro: n,altas,baixas, pontos;

inicio

**n = 0**

**altas = 0**

**baixas = 0**

ler(pontos)

enquanto(pontos  $\geq$  0)

faça início

se (pontos  $\geq$  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.

Calcular e exibir a soma de todos os números ímpares positivos entre N e M

programa REPET10

var inteiro: n,m,nro,soma;

início

ler (n)

ler (m)

se (n < m)

então início

se (n %2==0)

então nro = n+1

senão nro = n

fim-se

soma= 0

enquanto (nro <= m)

faça 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

ALGORÍTMO

```
programa REPET10
var inteiro: n,m,nro,soma;
início
  ler (n)
  ler (m)
  se (n < m)
    então início
      se se (n %2==0)
        então nro = n+1
        senão nro = n
      fim-se
      soma= 0
      enquanto (nro <= m)
        faça início
          se nro >0
            então som
          fim-se
          nro = 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

ALGORÍTMO

```
programa REPET10
var inteiro: n,m,nro,soma;
início
  ler (n)
  ler (m)
  se (n < m)
    então início
      se (n %2==0)
        então nro = n+1
        senão nro = n
      fim-se
      soma= 0
      enquanto (nro <= m)
        faça início
          se nro >0
            então soma =
          fim-se
          nro = 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
var inteiro: n,m,nro,soma;
início
```

SOMA  
acumulador

```
ler (n)
ler (m)
se (n < m)
```

```
então início
```

```
se (n %2==0)
```

```
então nro =
```

```
senão nro =
```

```
fim-se
```

```
soma= 0
```

```
enquanto (nro <= m)
```

```
faça 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
```

ALGORITMO

SOMA dever ser zerado

SOMA deve ser então incrementado dentro do comando de repetição

programa REPET10

var inteiro: n,m,nro,soma;

início

ler (n)

ler (m)

se (n < m)

então início

se (n %2==0)

então nro = n+1

senão nro = n

fim-se

soma= 0

enquanto (nro <= m)

faça 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

ALGORITMO

programa REPET10

var inteiro: n,m,nro,soma;

início

ler (n)

ler (m)

se (n < m)

então início

se (n %2==0)

então nro = n+1

senão nro = n

fim-se

soma= 0

enquanto (nro <= m)

faça 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
var inteiro: n, i;
    real: num,media;
inicio
ler(n)
media = 0
i = 1
enquanto(i ≤ n)
    faça início
        ler(num);
        media = media + num;
        i = i + 1;
    fim
fim enquanto
media = media /n;
escrever (media)
fim
```

programa REPET11

var inteiro: n, i;

real: num, media;

inicio

ler(n)

media = 0

i = 1

enquanto( $i \leq n$ )

faça início

ler(num);

media = media + num;

i = i + 1;

fim

fim enquanto

media = media / n;

escrever (media)

fim

i é um contador que  
determina a parada

```
programa REPET11
var inteiro: n, i;
  real: num,media;
inicio
ler(n)
media = 0
i = 1
enquanto(i ≤ n)
  faça início
    ler(num);
    media ← media + num;
    i ← i + 1;
  fim
fim enquanto
media ← media /n;
escrever (media)
fim
```

i é um contador que determina a parada

i é declarado como inteiro

i é inicializado

i é incrementado de 1

```
programa REPET12
var inteiro: n, i;
  real: num,media;
inicio
ler(n);
media = 0;
i = 0
enquanto(i ≤ n-1)
  faça início
    i = i + 1
    ler(num)
    media = media + num
  fim
fim enquanto
media = media /n
escrever (media)
fim do programa
```

I é um contador que delimitar a parada

I é inicializado

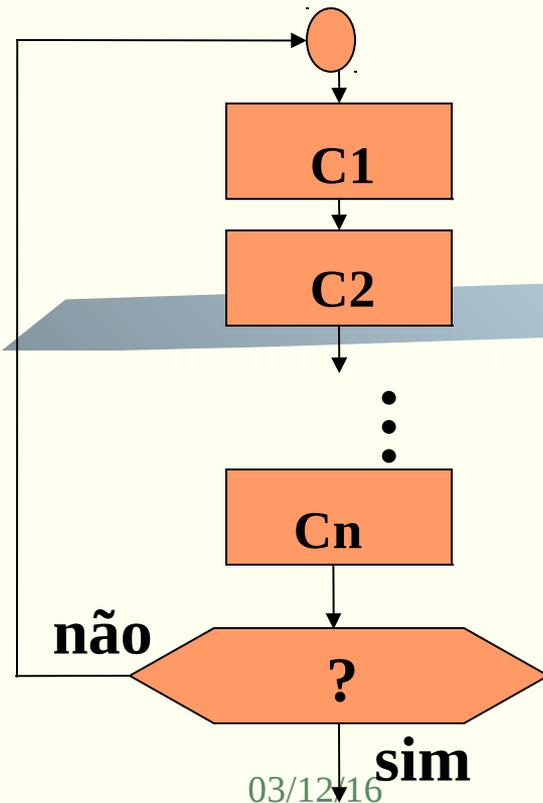
I é incrementado de 1

Outra maneira de contar os números

# Repetição com Teste no Final

formas de representação no algoritmo

Fluxograma



Algoritmo

repita

comando 1

comando 2

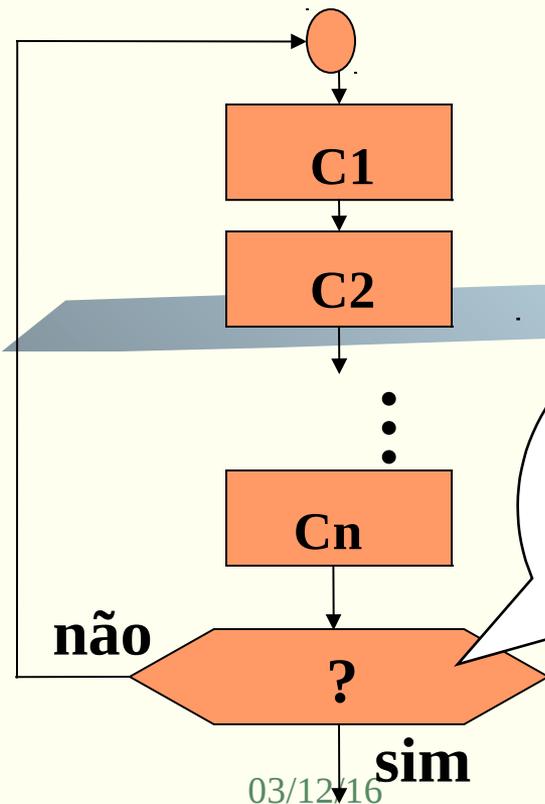
comando n

até <condição>

# Repetição com Teste no Final

formas de representação no algoritmo

Fluxograma



Algoritmo

Condição de **parada** da repetição

repete

comando 1  
comando 2

Condição de **parada** da repetição

comando n  
até <condição>

# Estrutura REPITA

repita

instrução\_1;

instrução\_2;

....

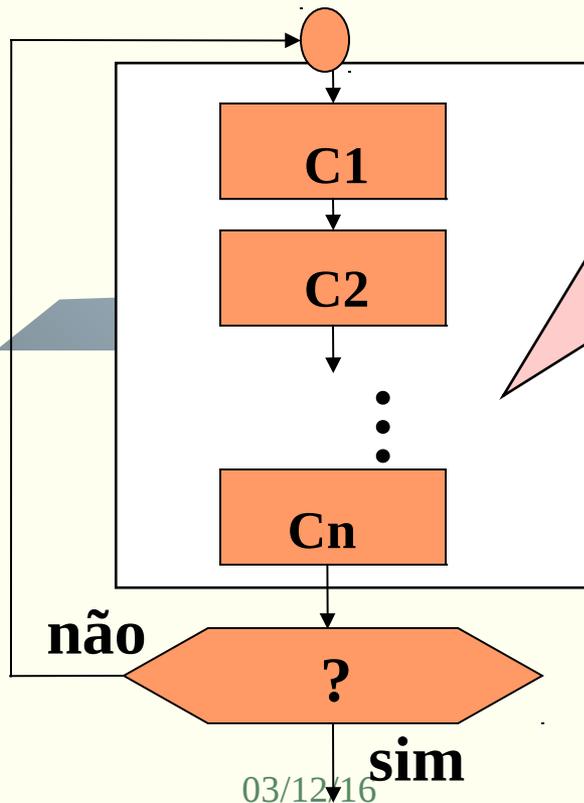
instrução\_n;

até (condição)

# Repetição com Teste no Final

formas de representação no algoritmo

Fluxograma



Os comandos são executados pelo menos uma vez

Algoritmo

repita

comando 1

comando 2

comando n

até <condição>

# 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

var inteiro: s;

inicio

s = 0

n = 1

d = 1

**repete**

s = s +  $\frac{n}{d}$

n = n+2

d = d+1

**até que** (d > 50)

escrever (s)

fim

Inicializa a somatória

Inicializa o numerador

Inicializa o denominador

Calcula os termos da série

Calcula o próximo numerador

Calcula o próximo denominador