**Laços Repetidos**

O controle de repetição permite que uma sequência de comandos seja executada repetidamente até que uma determinada condição de interrupção seja satisfeita.

**Sintaxe:**

• *Interrupção no início:*

## repita

## se condição

## então

## abandone

## fim-se

sequência B de comandos

fim-repita

A sequência B de comandos será repetida até que a condição seja satisfeita. Quando isto ocorrer, a repetição é interrompida e a sequência de comandos que vier logo após a expressão “fim-repita” passa a ser executada.

• *Interrupção no interior*

repita

sequência A de comandos

se condição

então

abandone

fim-se

sequência B de comandos

fim-repita

As sequências A e B de comandos serão repetidas até que a condição seja satisfeita. Quando isto ocorrer, a repetição é interrompida e a sequência de comandos que vier logo após a expressão “fim-repita” passa a ser executada.

**Exemplo:**

ALGORITMO exemplo da estrutura repita - fim-repita

1. [Início]
2. [Declaração de Variáveis]

A,B,C: inteiro

1. [Inicialização de Variáveis]

**A** 10

B A/10

C B\*5

1. [Processamento]

repita

B  B+1

se C ≤B

então

abandone

fim-se

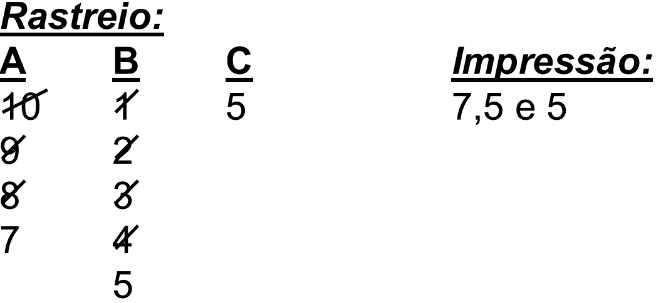
A  A -1

fim-repita

1. [Impressão]

escreva (A,”, “,B,” e “,C)

1. [Fim]



# Laços Repetidos com Teste no Final

**Sintaxe:**

## repita

comando(s)

até <condição>

O teste é feito depois de executar o(s) comando(s). O(s) comando(s) serão executados(s) pelo menos uma vez. Quando a condição é verdadeira, a sequência de comandos que vier logo após a expressão “até <condição>” passa a ser executada. Se a condição for falsa, o(s) comando(s) serão executados novamente, até que a condição se torne verdadeira.

**Exemplo:**

ALGORITMO exemplo da estrutura repita - até

1. [Início]
2. [Declaração de Variáveis]

X,Y,Z: real

1. [ Inicialização de Variáveis]
   1.  40
   2.  10
   3.  Y

1. [Processamento]

repita

Z  Z\*2

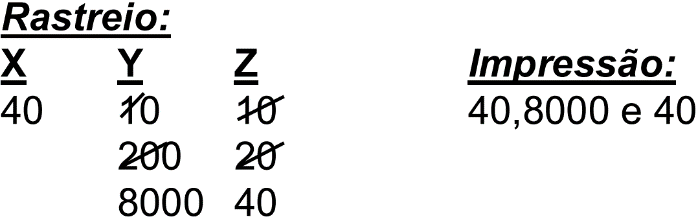
Y  Y\*Z

## até Z ≥ X

1. [Impressão]

escreva (X,”, “,Y,” e “,Z)

1. [Fim]



# Laços Contados

Para algumas aplicações, o controle condicional de laços imposto pela construção ”enquanto”, é desnecessariamente complicado. Em muitos casos, pode-se desejar executar um laço, um número fixo de vezes, onde este número já é conhecido. Para aplicações como estas, existe uma forma modificada de construção repetitiva. Esta construção está descrita abaixo:

**Sintaxe:**

para variável = valor-inicial até valor-final [**de** <inc/dec> **em** <inc/dec>] faça

comando(s)

## fim-para

Esta estrutura é usada quando se deseja repetir um trecho de programa um número conhecido de vezes. O conjunto de comandos delimitados pelas cláusulas “para” e ”fimpara” será executado cada vez que a variável receber um valor, começando com o <*valorinicial*> e indo até o <*valor-final*>, sendo incrementada de <*inc*> ou decrementada de *<dec>*. Caso se omita o incremento, pois o mesmo é opcional, a variável será incrementada de um em um.

**Obs.:** Incremento de um em um não precisa constar na estrutura. Para qualquer incremento diferente de um, é obrigatório constar.

**Exemplo1:**

ALGORITMO exemplo da estrutura para

1. [Início]
2. [ Declaração de Constantes]

TOTALREP  5

1. [Declaração de Variáveis]

NOME : literal

I : inteiro

1. [Processamento]

para I=1até TOTALREP faça

escreva (“Entre com o nome: “) leia(NOME)

escreva (I,” - “,NOME)

fim-para

1. [Fim]

