**Estruturas de Repetição Parte 3**

**Tipos de Estruturas já vistos:**

Enquanto: Onde sempre que começa o ciclo novamente é analisado se a condição imposta (teste lógico) é verdadeira ou falsa.

Enquanto nao arrumar o quarto Faca

Castigo

FimEnquanto

Liberado

Repita: Onde sempre que acaba o ciclo é analisado se a condição imposta (teste lógico) é verdadeira ou falsa.

Repita

Castigo

Ate arrumar o quarto

Liberado

Note que entre a estrutura de repetição Enquanto e a Repita há uma inversão de estruturas lógicas (no Enquanto usamos operador lógico “não” que não é usado no Repita).

**Estrutura de Repetição “Para”**

Sintaxe:

Para variavel <- inicio ate fim [passo salto] faca

Bloco

FimPara

Ex1:

Para C <- 1 ate 10 passo 1 faca

EscrevaL (C)

FimPara

Nesse exemplo acima atribuímos para a variável C o valor 1 ate 10, onde o passo é igual a 1, ou seja, a cada recomeço de ciclo, será adicionado +1 para a variável C (Como o valor do passo é igual a 1, ele não precisaria ser colocado, já que esse é seu valor padrão).

Ex2:

Algoritmo “ContaDe10Ate1”

Var

C : Inteiro

Inicio

Para C <- 10 ate 1 passo -1 faca

Escreval (C)

FimPara

FimAlgoritmo

Algoritmo ”Somar5Valores”

Var

S, C , N: Inteiro

Inicio

S <- 0

Para C <- 1 ate 5 faca

EscrevaL (“Digite um valor: “)

Leia (N)

S <- S + N

FimPara

EscrevaL (“A soma dos valores inseridos é “, S)

Fimalgoritmo

A estrutura de repetição “Para” não pode ser usada para criar estruturas de repetição onde o usuário determina a quantidade de vezes que ele vai repetir a estrutura (sem contar quantas vezes foram repetidas).

Só devemos usar a estrutura de repetição “Para” quando sabemos a quantidade de vezes que determinado conjunto de comandos deve ser repetido.

Ex2:

Algoritmo “MostrarValoresPares”

Var

C, V: Inteiro

Inicio

Escreval (“Digite um valor: “)

Leia (V)

Para C <- 0 ate V Passo 2 Faca

EscrevaL (C)

FimPara

Fimalgoritmo

Note que como o contador de repetições começa em 0 e sempre será atribuído para a variável contadora +2, então sempre será exibido apenas os números pares.

Ex3:

Algoritmo “ContadorRegresNPares”

Var

C, V: Inteiro

Inicio

Escreval (“Digite um valor: “)

Leia (V)

Se (V % 2 = 1) Entao

V <- V -1

FimSe

Para C <- V ate 0 Passo -2 Faca

EscrevaL (C)

FimPara

Fimalgoritmo

Note que nesse exemplo fizemos uma condição que se caso o resto da divisão do valor que o usuário inseriu por 2 seja = 1 será tirar 1 do valor, pois isto indica que o número é ímpar, o que iria fazer com que a contagem do programa exibisse apenas números ímpares ao invés de pares.

Ex4:

Algoritmo “ValoresEntre0E10”

Var

C, N, V, N0E10, SImp: Inteiro

Inicio

N0E10 <- 0

SImp <- 0

EscrevaL (“Quantos valores você deseja inserir? “)

Leia (V)

Para C <- 1 ate V Faca

Escreval (“Insira um valor: “)

Leia (N)

Se (N >= 0) E (N <= 10) Entao

N0E10 <- N0E10 + 1

Se (N % 2 = 1) Entao

SImp <- SImp + N

FimSe

FimSe

FimPara

EscrevaL (“Dentre os “, V, “ valores inseridos, “, N0E10, “ estão entre 0 e 10.”)

EscrevaL (“A soma de todos os valores ímpares inseridos que estão entre 0 e 10 é “, SImp)

Fimalgoritmo

Ex4:

Algoritmo “CombinacoesD3ValoresC2Digitos”

Var

C1, C2: Inteiro

Inicio

Para C1 <- 1 ate 3 Faca

Para C2 <- 1 ate 3 Faca

EscrevaL (C1, C2)

FimPara

FimPara

Fimalgoritmo

Nesse código, o computador vai ficar mudando a variável de C2 até que seu valor seja 3 enquanto o C1 permanece com o valor 1, após isso, o C1 recebe o valor 2 e o C2 volta a fazer sua contagem de 1 até 3, esse loop acaba quando tanto C1 quanto C2 estiverem com o valor 3 (3, 3).