



CENTRO PAULA SOUZA - ETEC

LUIS EDUARDO RAMOS TEZIN

LISTA DE EXERCÍCIOS 3:

Estruturas de repetição

São José dos Campos

Abril / 2019

LUIS EDUARDO RAMOS TEZIN

LISTA DE EXERCÍCIOS 3:
Estruturas de repetição

Projeto apresentado ao professor
Gildarcio Gonçalves na escola ETEC Centro
Paula Souza. Área de Concentração:
Programação e Algoritmos.
Instrutor: Gildarcio Gonçalves

São José dos Campos
Abril / 2019

SUMÁRIO

Questão 1 4

Questão 2 4

Questão 3 5

Questão 4 5

Questão 5 6

Questão 6 6

Questão 7 7

Questão 8 8

Questão 9 9

Questão 10 10

Questão 1

```
algoritmo "Soma_Impares"
// Função :
// Autor : Luis Eduardo Ramos Tezin
// Data : 11/04/2019
// Questão 1
// Dado um número N, inteiro e positivo, obter a soma dos N primeiros ímpares.
// Seção de Declarações
var
    count, n, soma: Inteiro
inicio
    escreva("Digite um valor inteiro: ")
    leia(n)
    count <- 0
    soma <- 0
    enquanto (count <= n) faca
        se (count mod 2 <> 0) entao
            soma <- soma + count
        fimse
        count <- count + 1
    fimenquanto
    escreva("Soma dos ímpares: ", soma)
finalgoritmo
```

Questão 2

```
algoritmo "Soma_Sequencial"
// Função :
// Autor : Luis Eduardo Ramos Tezin
// Data : 11/04/2019
// Questão 2
// Some os números de 1 a 100 e imprima o valor.
// Seção de Declarações
var
    x, soma: Inteiro
inicio
    soma <- 0
    para x de 1 ate 100 faca
        soma <- soma + x
    fimpara
    escreva("Soma dos valores: ", soma)
finalgoritmo
```

Questão 3

```
algoritmo "Tabuada"
// Função :
// Autor : Luis Eduardo Ramos Tezin
// Data : 16/04/2019
// Questão 3
// Faça um programa que mostre as tabuadas dos números de 1 a 10 (Utilizar Repetição Aninhada).
// Seção de Declarações
var
    count1, count2, resultado: Inteiro
inicio
    para count1 de 1 ate 10 faca
        escreval("Tabuada do ", count1)
        para count2 de 1 ate 10 faca
            resultado <- count1 * count2
            escreval(count1, " x ", count2, " = ", resultado)
        fimpara
    fimpara
fimalgoritmo
```

Questão 4

```
algoritmo "Multiplos7"
// Função :
// Autor : Luis Eduardo Ramos Tezin
// Data : 11/04/2019
// Questão 4
// Faça um algoritmo que imprima os múltiplos positivos de 7, inferiores a 1000.
// Seção de Declarações
var
    count: Inteiro
inicio
    para count de 1 ate 1000 faca
        se (count mod 7 = 0) entao
            escreval(count, " é multiplo de 7.")
        fimse
    fimpara
fimalgoritmo
```

Questão 5

```
algoritmo "Crescimento"
// Função :
// Autor : Luis Eduardo Ramos Tezin
// Data : 11/04/2019
// Questão 5
// Chico tem 1,50 metros e cresce 3 centímetros por ano, enquanto Zé tem 1,10
// metros e cresce 4 centímetros por ano.
// Construa um algoritmo que calcule e imprima quantos anos serão necessários
// para que Zé seja maior que Chico.
// Seção de Declarações
var
    chico, ze: Real
    anos: Inteiro
inicio
    anos <- 0
    chico <- 1.50
    ze <- 1.10
    enquanto (ze <= chico) faça
        chico <- chico + 0.03
        ze <- ze + 0.04
        anos <- anos + 1
    fimenquanto
    escreva("Zé passará Chico em ", anos, " anos.")
fimalgoritmo
```

Questão 6

```
algoritmo "Massa"
// Função :
// Autor : Luis Eduardo Ramos Tezin
// Data : 11/04/2019
// Questão 6
// Um determinado material radioativo perde metade de sua massa a cada 50 segundos.
// Dada a massa inicial, em gramas, faça um programa que determine o tempo
// necessário para que essa massa se torne menor que 0,05 gramas.
// Seção de Declarações
var
    massa: Real
    tempo: Inteiro
inicio
    tempo <- 0
    escreva("Digite o valor da massa(g) do material: ")
    leia(massa)
    repita
        massa <- massa / 2
        tempo <- tempo + 50
    ate (massa < 0.05)
    escreva("A massa terá menos de 0,05 gramas em ", tempo, " segundos.")
fimalgoritmo
```

Questão 7

algoritmo "Stop"

// Função :

// Autor : Luis Eduardo Ramos Tezin

// Data : 11/04/2019

// Questão 7

// Faça um programa que leia vários números reais, até que o usuário digite

// o valor 0 (zero). Ao finalizar, o programa deverá informar:

// a. A soma de todos os valores informados;

// b. A quantidade de valores informados;

// c. O valor médio entre todos os valores;

// d. O maior valor, dentre os que foram digitados pelo usuário;

// e. O menor valor, dentre os que foram digitados pelo usuário;

// Seção de Declarações

var

n, soma, count, media, maior, menor: Real

inicio

count <- 0

escreva("Digite um valor (0 para sair): ")

leia(n)

soma <- n

menor <- n

maior <- n

enquanto (n <> 0) faca

se (n > maior) entao

maior <- n

fimse

se (n < menor) entao

menor <- n

fimse

escreva("Digite um valor (0 para sair): ")

leia(n)

soma <- soma + n

count <- count + 1

fimenquanto

media <- soma / count

escreval("Soma dos valores: ", soma)

escreval("Quantidade de valores: ", count)

escreval("Média aritmética: ", media)

escreval("Maior valor: ", maior)

escreval("Menor valor: ", menor)

fimalgoritmo

Questão 8

```
Algoritmo "semnome"
// Função :
// Autor : Luis Eduardo Ramos Tezin
// Data : 11/04/2019
// Questão 8
// Faça um programa que receba duas notas de N alunos a partir do usuário. N é o número de alunos que um professor tem,
// então o programa deverá pedir para o usuário este valor. O programa deverá:
// a. Imediatamente após ler as duas notas de cada aluno, informar a nota média do aluno e se ele está aprovado
// (até 3), em exame (entre 3 e 7) ou aprovado (de 7 para cima).
// b. Após terminar de calcular a média de todos os alunos, mostrar:
// i. A nota média total da turma; ii. O número total de alunos
// aprovados, em exame e reprovados.
// Seção de Declarações
var
    media, media_total, media_somatorio, n1, n2: Real
    aluno, n, aprovados, reprovados, exame: Inteiro
inicio
    escreva("Digite o número de alunos: ")
    leia(n)
    aprovados <- 0
    reprovados <- 0
    exame <- 0
    para aluno de 1 ate n faca
        escreval("Digite a nota 1 do aluno ", aluno)
        leia(n1)
        escreval("Digite a nota 2 do aluno ", aluno)
        leia(n2)
        media <- (n1 + n2) / 2
        media_somatorio <- media_somatorio + media
        se (media >= 7) entao
            escreval("O aluno ", aluno, " está aprovado com média ", media)
            aprovados <- aprovados + 1
        senao
            se (media >= 3) e (media < 7) entao
                escreval("O aluno ", aluno, " está de exame com média ", media)
                exame <- exame + 1
            senao
                escreval("O aluno ", aluno, " está reprovado com média ", media)
                reprovados <- reprovados + 1
            fimse
        fimse
    fimpara
    media_total <- media_somatorio / n
    escreval("Média total da turma: ", media_total)
    escreval("Alunos aprovados: ", aprovados)
    escreval("Alunos reprovados: ", reprovados)
    escreval("Alunos em exame: ", exame)
fimalgoritmo
```


Questão 9

```
algoritmo "Loja_Carro"
// Função :
// Autor : Luis Eduardo Ramos Tezin
// Data : 12/04/2019
// Questão 9
// Faça um programa que
// receba o valor de um carro. O programa deverá calcular e mostrar
// uma tabela de informações ao vendedor contendo: Preços finais x
// Quantidade de Parcelas x Valor de cada parcela. Considere o
// seguinte:
// a. Utilize estrutura de repetição para construir a tabela;
// b. O preço final para compra à vista tem desconto de 20%;
// c. A quantidade de parcelas pode ser 6, 12, 18, 24, 30, 36, 42, 48, 54 e 60;
// d. Os percentuais de acréscimo para cada quantidade de parcelas encontra-se na tabela a seguir:
//
// Parcelas Acréscimo sobre o preço final
// 6 3%
// 12 6%
// 18 9%
// 24 12%
// 30 15%
// 36 18%
// 42 21%
// 48 24%
// 54 27%
// 60 30%
// Seção de Declarações
var
    valor, preco_final, valor_parcela, acrescimo: Real
    parcelas: Inteiro
início
    escreva("Valor do carro: ")
    leia(valor)
    parcelas <- 0
    acrescimo <- 0
    enquanto (parcelas <= 60) faça
        se (parcelas = 0) então
            preco_final <- valor - (valor * (20/100))
        senão
            preco_final <- valor + (valor * (acrescimo/100))
            valor_parcela <- preco_final / parcelas
        fimse
        escreval(preco_final, " x ", parcelas, " x ", valor_parcela:6:2)
        parcelas <- parcelas + 6
        acrescimo <- acrescimo + 3
    fimenquanto
fimalgoritmo
```

Questão 10

```
algoritmo "Fibonacci"
// Função :
// Autor : Luis Eduardo Ramos Tezin
// Data : 13/04/2019
// Questão 10
// A série de Fibonacci é formada pela sequência: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...
// Escreva um programa que gere a série de FIBONACCI até o N-ésimo termo.
// Seção de Declarações
var
    n, antecessor1, antecessor2, atual, count: Inteiro
inicio
    escreva("Digite um valor inteiro: ")
    leia(n)
    count <- 1
    para count de 1 ate n faca
        se (count = 1) entao
            antecessor1 <- 1
            atual <- antecessor1
        senao
            se (count = 2) entao
                antecessor2 <- antecessor1
                atual <- antecessor2
            senao
                atual <- antecessor1 + antecessor2
                antecessor1 <- antecessor2
                antecessor2 <- atual
            fimse
        fimse
        escreval(atual)
    fimpara
fimalgoritmo
```