**Avaliação de Diagnóstico**

**Habilidades em algoritmos**

**Objetivo do Trabalho:** avaliar o nível geral da turma em relação ao domínio de conceitos e técnicas de elaboração de algorítmos.

Resolver em sala, individualmente, e apresentar os resultados ao professor, até o início da aula do dia **19/02/2019,** quando haverá um debate em sala sobre o trabalho.

# **Parte 1 – *Quiz***

1. Algoritmo é uma sequência finita de instruções organizadas de forma lógica para a resolução de um determinado problema.
2. **Verdadeiro**
3. Falso
4. Qual identificador não é válido?
5. nota1 b) idade c) **#upis** d) salario\_hora
6. O que é tipo de dado?
7. **Faixa de valores válidos para a variável**
8. Um conjunto de símbolos aritméticos
9. Uma área de armazenamento na memória
10. Uma forma de descrever objetos
11. As expressões compostas são resolvidas da esquerda para direita.
12. Verdadeiro
13. **Falso – são resolvidas de acordo com as precedências dos operadores**
14. O que é uma estrutura condicional?
15. Comando de atribuição
16. **Comando de decisão**
17. Comando de repetição
18. Chamada de função
19. Comando de seleção é adequado para a programação de menus
20. **Verdadeiro**
21. Falso
22. São comando de repetição, exceto
23. for (para...faça) b) while (enquanto...faça)

c) do...while (repita...até) d) **goto (vá para)**

1. Qual comando de repetição possui teste no final?
2. for (para...faça) b) while (enquanto...faça) c) **do...while (repita...até)**
3. O que é um vetor?
4. Uma matriz unidimensional
5. Uma estrutura de dados homogêneos
6. Uma coleção de elementos indexados
7. **Todas estão corretas**
8. Subrotina é bloco de instruções que realiza uma tarefa específica, com nome e parâmetros de entrada.
9. **Verdadeiro**
10. Falso
11. Uma função pode retornar
12. Nenhum valor b) **um único valor** c) dois valores d) três ou mais valores

# **Parte 2 – Construção de Algoritmos**

# **Algoritmos aplicando estruturas sequenciais**

1. Algoritmo que leia pelo teclado dois números inteiros e determina o maior múltiplo do primeiro informado, menor que ou igual ao segundo informado (exemplo: qual é o maior múltiplo de 6 menor que ou igual a 40? Resposta: 36).

**Algoritmo** “Maior\_Multiplo”

**Var** Num, Limite, Mult : **Inteiro**

**Inicio**

**Escreva**(“Informe o número base: “)

**Leia**(Num)

**Escreva**(“Informe o número limite: “)

**Leia**(Limite)

Mult <- (Limite **div** Num) \* Num

**EscrevaL**(“Maior múltiplo de “,Num, “menor que ou igual a “,

Limite, “ é “, Mult)

**FimAlgoritmo.**

1. Algoritmo que leia um valor inteiro positivo, menor que 1000 e determine, a soma dos dígitos que formam seu valor. (Ex.: valor 397 tem soma de dígitos igual a 19 [3+9+7] ).

Dica: use os operadores **Div** e **Mod** para resolver este problema.

**Algoritmo** “Soma\_Digitos”

**Var** Soma**,**numero, unidade, dezena, centena : **inteiro**

**Inicio**

**Escreva** (“Informe um número inteiro menor que 1000: “)

**Leia** (numero)

Unidade <- numero **Mod** 10

Dezena <- numero **Div** 10 **Mod** 10

Centena <- numero **Div** 100 **Mod** 10

Soma <- Unidade+Dezena+Centena

**EscrevaL** (“Soma dos dígitos do número “, numero, “ = “, Soma)

**FimAlgoritmo**

**Algoritmos aplicando estruturas de decisão**

1. Algoritmo que receba o nome e três notas (entre 0 e 10) de um aluno e implementa uma função para calcular a média ponderada considerando: primeira nota peso 2, segunda nota peso 3 e terceira nota peso 2. Informar o nome, a média e se aluno foi aprovado (media ≥ 6,0) ou não.

**Algoritmo** “Aprovado”

**Funcao** mediaPond (n1, n2, n3: **real**): **real**

**Var** mP:**real**

**Inicio**

mP <- (2\*n1+3\*n2+2\*n3)/7

Retorne mP

**FimFuncao**

**Var** Nome : **caracter**

Nota1, nota2, nota3, media: **real**

**Inicio**

**Escreva**(“Digite o nome do aluno”)

**Leia**(nome)

**Escreva**(“Digite as três notas do aluno “)

**Leia**(nota1, nota2, nota3)

media <- mediaPond (nota1, nota2, nota3)

**EscrevaL**(“Nome do aluno “, nome)

**Se** (media >= 7) **Entao**

**EscrevaL** (“Aluno aprovado com média: “, media)

**Senao**

**EscrevaL** (“Aluno reprovado com média: “, media)

**FimSe**

**FimAlgoritmo**

1. Algoritmo que, tendo como dados de entrada a altura (h) e o sexo de uma pessoa, chame uma função que calcule seu peso ideal, utilizando as seguintes fórmulas:

Homens: Peso Ideal = 72.7 \* h – 58 Mulheres: Peso Ideal = 62.1 \* h – 44.7

**Algoritmo** “Peso”

**Funcao** Peso\_Ideal (sex: **caractere**, alt: **real**):**real**

**Var** pIdeal:**real**

**Inicio**

**Se**(sex = “M”) **Entao**

pIdeal <- 72.7\*h-58

**Senao**

pIdeal <- 62.1\*h–44.7

**FimSe**

Retorne pIdeal

**FimFuncao**

**Var** sexo: **caractere**

h, peso : **real**

**Inicio**

**Escreva**(“Digite o sexo da pessoa (M ou F) : ”)

**Leia** (sexo)

**Escreva**(“ Digite a altura da pessoa : ”)

**Leia** (h)

Peso <- Peso\_Ideal(sexo, h)

**EscrevaL**(“ Peso ideal: ”, peso)

**FimAlgoritmo**

# **Algoritmos aplicando estruturas de repetição**

1. Algoritmo que leia dois números inteiros positivos e efetue a multiplicação de ambos sem usar o operador de multiplicação. A multiplicação deve ser implementada em uma função que recebe os dois números e devolva o produto entre embos.

**Algoritmo** “Mult”

**Funcao** Produto (a, b: **inteiro**): **inteiro**

**Var** P, i: **inteiro**

**Inicio**

P <- 0

**Para** i **de** 1 **ate** b **Faca**

P <- P+a

**FimPara**

**Retorne** P

**FimFuncao**

**Var** N1, N2, Prod : **inteiro**

**Inicio**

**Escreva**(“Digite dois números”)

**Leia**(N1,N2)

Prod <- Produto(N1, N2)

**EscrevaL** (N1, “ x ”, N2, “ = ”, Prod)

**FimAlgoritmo**

1. Algoritmo que carregue uma matriz 4x4 de valores inteiros. Depois mostre a soma da diagonal principal e o produto da diagonal secundária.

**Obs.:**

Os elementos da diagonal principal de uma matriz M possuem indices iguais: M[i, i];

Os elementos da diagonal segundaria possuem índices siméticos: M[i, N – 1 – i], onde N é tamanho da matriz;

**Algoritmo** “Matriz”

**Const** tamanho: 4

**Var**

M[tamanho, tamanho]: **inteiro**

lin, col, sPrinc, pSec: **inteiro**

**Inicio**

**Escreva** (“Carregue valores inteiros para matriz 4x4: “)

**Para** lin **de** 0 **ate** tamanho-1 **faca**

**Para** col **de** 0 **ate** tamanho-1 **faca**

**Leia** (M[lin, col])

**FimPara**

**FimPara**

sPrinc <- 0

pSec <- 1

**Para** lin **de** 0 **ate** tamanho-1 **faca**

sPrinc <- sPrinc + M[lin, lin]

pSec <- pSec \* M[lin, tamanho – 1 – lin]

**FimPara**

**EscrevaL** (“Soma da diagonal principal: “, sPrinc)

**EscrevaL** (“Produto da diagonal secundaria: “, pSec)

**FimAlgoritmo**

1. Algoritmo que calcule, através de uma função, o valor de A, dado pela série abaixo, onde N será lido pelo teclado:



Para este problema, é apresentado abaixo uma solução sem uso de função. Sua tarefa aqui é reescrever o algoritmo abaixo utilizando uma função que receba como parâmetro a quantidade de parcelas do somatório, calcule e retorne o resultado.

Algoritmo Somatorio

Var

A: real

N, i: inteiro

Inicio

Escreva (“Informe a quantidade de parcelas do somatório: “);

Leia (N);

A 🡨 0;

Para i de 1 ate N passo 1 faca

A 🡨 A + (N +1 – i) / i ;

FimPara

Escreval(“ A = “, A)

Fim.

**Algoritmo** “Somatorio”

**Funcao** Soma (N: **inteiro**): **real**

**Var** S: **real** i: **inteiro**

**Inicio**

S <- 0

**Para** i **de** 1 **ate** N **Faca**

S <- S + (N + 1 – i)/i

**FimPara**

**Retorne** S

**FimFuncao**

**Var** A: **Real** n: **inteiro**

**Inicio**

**Escreva**(“Informe a quantidade de parcelas do somatório: ”)

**Leia**(n)

A <- Soma(n)

**EscrevaL** (“ A = ”, A)

**FimAlgoritmo**