



Curso de Formação Profissional em Técnico em Informática

DISCIPLINA DE LÓGICA DE PROGRAMAÇÃO:

Ementa:

- Introdução
- Tipos de Dados
- Operadores e expressões
- Variáveis
- Constantes
- Comentários
- Entrada e saída de dados
- Seleção e repetição
- Funções e procedimentos

Avaliação

- ◆ Provas escritas
- ◆ Resolução de listas de exercícios
- ◆ Trabalhos

Nome do Aluno: _____

Prof.: Arley Rodrigues

Disciplina: Introdução à Informática

Site: www.portalgigaweb.com.br

LÓGICA DE PROGRAMAÇÃO

INTRODUÇÃO À LÓGICA DE PROGRAMAÇÃO

O QUE É LÓGICA?

É a arte de pensar corretamente. Corrigir o raciocínio. Colocar ordem no pensamento.

A lógica de programação é necessária para pessoas que desejam trabalhar com desenvolvimento de sistemas e programas, ela permite definir a seqüência lógica para o desenvolvimento.

A Lógica de programação é a técnica de encadear pensamentos para atingir determinado objetivo.

O QUE É ALGORITMO?

Algoritmo é uma seqüência de passos que visam atingir um objetivo pré-definido e bem definido. O algoritmo formado e bem definido que pode ser seguido por qualquer pessoa, e o objetivo será alcançado com certeza.

SEQÜÊNCIA LÓGICA

São passos executados até atingir um objetivo ou solução de um problema.

Instruções são um conjunto de regras ou normas definidas para a realização ou emprego de algo. Em informática, é o que indica a um computador uma ação elementar a executar.

Os algoritmos são descritos em uma linguagem chamada pseudocódigo.

PROGRAMAÇÃO

O que é Programação: É o processo de escrita, teste e manutenção de um programa de computador. O programa é escrito em uma linguagem de programação. Os programas de computadores nada mais são do que algoritmos escritos numa linguagem de computador (Pascal, C, Delphi, Java, PHP entre outras) e que são interpretados e executados por uma máquina, no caso um computador.

Os algoritmos são descritos em uma linguagem chamada **pseudocódigo**. O algoritmo deve ser fácil de se interpretar e fácil de codificar. Ou seja, ele deve ser o intermediário entre a linguagem falada e a linguagem de programação.

OBSTÁCULOS DE UM PROGRAMADOR

O maior problema que envolve a construção de algoritmo é a complexidade. Portanto, quando tentamos construir um, o objetivo principal é vencer a complexidade do problema a ser solucionado.

O QUE É COMPLEXIDADE?

Complexidade representa a quantidade de situações diferentes que um problema pode apresentar. Existem várias maneiras de reduzir a complexidade, ou evitar que esta cresça. Vamos, no momento tentar somente evitar que ela cresça.

O QUE É QUESTIONAMENTO ERRÔNEO?

Um dos maiores responsáveis, talvez o maior, pelo aumento desmedido da complexidade é a forma errônea de auto questionar-se a respeito de um dado problema.

Como é um sorvete? Deliciosa sobremesa gelada, com formato esférico, normalmente acompanhada de um cone comestível, em diversas cores e sabores.

Gordura vegetal hidrogenada, leite em pó, aromatizantes, corantes e estabilizantes.

O QUE É LEGIBILIDADE?

Legibilidade é a capacidade de compreensão do algoritmo por qualquer observador. É a clareza com que a lógica do mesmo algoritmo está exposta. Quanto mais legível for o algoritmo menor sua complexidade.

DICAS PARA CRIAÇÃO DE UM BOM ALGORITMO

Criar primeiro um algoritmo simples. (estrutura seqüencial)

Melhorar este algoritmo criando teste seletivo. (condição)

Observar a colocação do teste seletivo. (estrutura de seleção)

Provar que existem várias formas de resolver o mesmo algoritmo.

Melhorar este algoritmo criando um teste repetitivo. (estrutura de repetição)

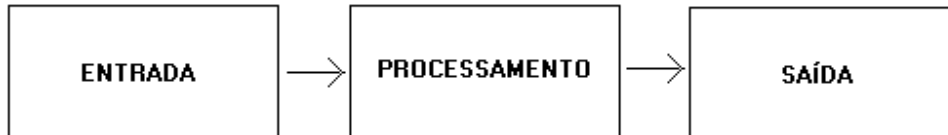
Exercício: Criar um algoritmo para trocar uma lâmpada queimada.

MÉTODOS PARA CONSTRUÇÃO DE ALGORITMO

- Ler atentamente o enunciado
- Retirar do enunciado a relação das entradas de dados
- Retirar do enunciado a relação das saídas de dados
- Determinar o que deve ser feito para transformar as entradas determinadas na saídas especificadas
- Construir o algoritmo
- Usar somente um verbo por frase
- Imaginar que você está desenvolvendo um algoritmo para pessoas que não trabalham com informática
- Usar frases curtas e simples
- Ser objetivo
- Executar o algoritmo

O desenvolvimento do raciocínio lógico requer que estudemos uma série de problemas “lógicos”. Muitas vezes é difícil “explicar” a solução destes exercícios, mas trabalhar com lógica representa solucionar e vencer desafios. Considero os desafios abaixo interessantes o bastante para motivá-los a se interessar pelo raciocínio lógico.

Para montar um algoritmo, precisamos primeiro dividir o problema apresentado em três fases fundamentais. Onde temos:

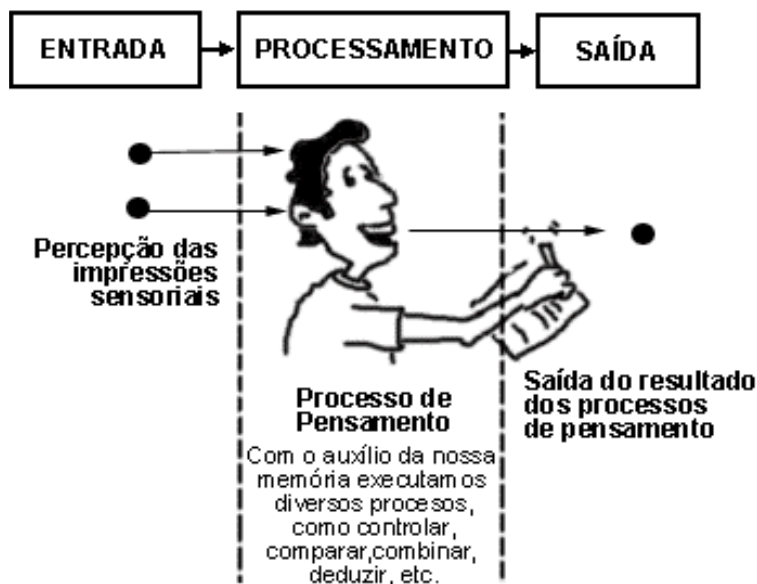


ENTRADA: São os dados de entrada do algoritmo

PROCESSAMENTO: São os procedimentos utilizados para chegar ao resultado final

SAÍDA: São os dados já processados

ANALOGIA HOMEM E MÁQUINA



EXEMPLO

Imagine o seguinte problema: Calcular a média final dos alunos da 3ª Série. Os alunos realizarão quatro provas: P1, P2, P3 e P4. Onde:

$$\text{Média Final} = (P1+P2+P3+P4)/4$$

Para montar o algoritmo proposto, faremos três perguntas:

a) Quais são os dados de entrada?

R: Os dados de entrada são P1, P2, P3 e P4

b) Qual será o processamento a ser utilizado?

R: O procedimento será somar todos os dados de entrada e dividi-los por 4 (quatro)

c) Quais serão os dados de saída?

R: O dado de saída será a média final

ALGORITMO

Receba a nota da prova1

Receba a nota de prova2

Receba a nota de prova3

Receba a nota da prova 4

Some todas as notas e divida o resultado por 4

Mostre o resultado da divisão

Após desenvolver um algoritmo ele deverá sempre ser testado. Este teste é chamado de TESTE DE MESA, que significa, seguir as instruções do algoritmo de maneira precisa para verificar se o procedimento utilizado está correto ou não.

Exercício 1

1. Faça um algoritmo para trocar uma lâmpada queimada
2. Faça um algoritmo para chupar uma bala
3. Faça um algoritmo para somar dois números
4. Faça um algoritmo para somar dois números e multiplicar o resultado pelo primeiro número
5. Faça um algoritmo para "Calcular o estoque médio de uma peça", sendo que $ESTOQUEMÉDIO = (QUANTIDADE\ MÍNIMA + QUANTIDADE\ MÁXIMA) / 2$
6. Construa um algoritmo para pagamento de comissão de vendedores de peças, levando-se em consideração que sua comissão será de 5% do total da venda e que você tem os seguintes dados:

Exercícios 2

1. Um homem precisa atravessar um rio com um barco que possui capacidade de carregar apenas ele mesmo e mais uma de suas três cargas, que são: um lobo, um bode e um maço de alfafas. O que o homem deve fazer para atravessar o rio sem perder suas cargas?

Obs.: O bode se ficar só, come a alfafa

O lobo se ficar só, come o bode

2. Três padres e três índios querem atravessar um rio em uma canoa. Porém a mesma só comporta duas pessoas. Como devem proceder para que os seis cheguem a outra margem do rio?

Obs.: Não pode ficar, em hipótese nenhuma, mais índio do que padre, pois os índios são canibais e ficando em maioria comerão o (s) padres (s) que estão em minoria.

3. Um sitiante tem que transportar 1 galinha, 1 raposa e um saco de milho de uma margem à outra do rio. Apenas o sitiante e um dos "acompanhantes" pode ser atravessar com o barco em cada viagem. A galinha não pode ser deixada sozinha com o milho (pois o come) nem com a raposa (pois é comida). Como o sitiante realiza a travessia sem perder nenhum de seus bens?

4. 3 Pessoas estão presas e recebem o seguinte desafio: será colocado um chapéu sobre a cabeça de cada uma delas. Cada pessoa poderá ver os chapéus dos 2 outros, mas não o seu próprio. Há 3 chapéus brancos e 2 vermelhos. Se a pessoa acertar a cor do seu chapéu, será liberada. Se errar, será morta. O primeiro prisioneiro, que tem as duas vistas boas, garante: "eu não tenho como saber a cor do meu chapéu". O segundo prisioneiro, que tem 1 vista boa, garante: "eu não tenho como saber a cor do meu chapéu". O terceiro prisioneiro, que é cego, garante: "eu sei a cor do meu chapéu: ele é branco."

Admitindo que os três prisioneiros sejam igualmente inteligentes, qual o raciocínio utilizado pelo terceiro prisioneiro para acertar a cor do seu chapéu?

5. Há 5 pessoas que pretendem atravessar uma ponte. Eles têm que atravessar a ponte em um prazo de 30 segundos no máximo, sendo que os cinco levam, respectivamente, 1, 3, 6, 8 e 12 segundos na travessia. 2 pessoas podem atravessar a ponte de cada vez, MAS irão fazê-lo na velocidade do mais lento. Em cada travessia, uma das pessoas tem que transportar um lampião (que fica aceso por apenas 30 segundos, e todos têm que atravessar com o lampião aceso). É possível atravessar todos com o lampião aceso?

6. 3 jesuítas e 3 canibais têm que atravessar um rio. No barco, cabem apenas 2 pessoas por travessia (e o barco não retorna sozinho...). Em nenhum momento pode-se ter menos missionários que canibais nas margens, pois os primeiros serão devorados pelos segundos. Como se pode realizar a travessia?

7. Em uma sala temos 3 lâmpadas incandescentes e fora da mesma 3 interruptores (um para cada lâmpada). Você pode entrar na sala apenas uma vez. Como descobrir qual interruptor acende cada lâmpada?

8. Imagine-se fechado numa sala, onde existem apenas duas portas. Uma conduz à vida e outra à morte. Consigo estão duas pessoas. Uma só diz a verdade e a outra a mentira, e você não sabe qual é qual. Tem direito apenas a uma pergunta. Que pergunta você faria para descobrir qual é a porta da vida e sair sem problemas?

9. DESAFIO DE EINSTEIN

No final do século passado, Einstein propôs um problema que, segundo ele, 98% das pessoas não seriam capazes de resolver. Há 5 casas de diferentes cores. Em cada casa mora uma pessoa de uma diferente nacionalidade. Esses 5 proprietários bebem diferentes bebidas, fumam diferentes tipos de cigarro e têm um certo animal de estimação. Nenhum deles tem o mesmo animal, fuma o mesmo cigarro ou bebe a mesma bebida.

1. O inglês vive na casa vermelha
2. O sueco tem cachorros como animais de estimação
3. O dinamarquês bebe chá
4. A casa verde fica à esquerda da casa branca
5. O dono da casa verde bebe café
6. A pessoa que fuma Pall Mall cria pássaros

7. O dono da casa amarela fuma Dunhill
8. O homem que vive na casa do centro bebe leite
9. O norueguês vive na primeira casa
10. O homem que fuma Blends vive ao lado de quem tem gatos
11. O homem que cria cavalos vive ao lado de quem fuma Dunhill
12. O homem que fuma Bluemaster bebe cerveja
13. O alemão fuma Prince
14. O norueguês vive ao lado da casa azul
15. O homem que fuma Blends é vizinho do que bebe água

A questão é: qual a nacionalidade do homem que cria peixe?

O QUE É A INFORMAÇÃO

Informação é a matéria-prima que faz com que seja necessária a existência dos computadores, pois eles são capazes de manipular e armazenar um grande volume de dados que vão gerar as informações.

Lembre-se: dados = informação; informática = informação automática; computação = ato de computar (calcular)

INFORMAÇÃO CONSTANTE:

É aquela que não sofre nenhuma variação em seu conteúdo no decorrer do tempo.

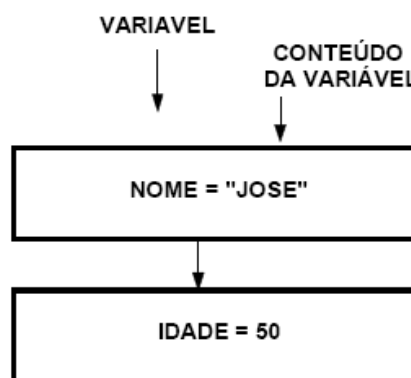
Ex.: Número PI, valor da gravidade, divisor da média de três números

$$\frac{N1+N2+N3}{3}$$

← CONSTANTE

INFORMAÇÃO VARIÁVEL:

É aquela que pode sofrer variação no seu conteúdo no decorrer do tempo. É a representação simbólica dos elementos de um certo conjunto. Cada variável corresponde a uma posição de memória, cujo conteúdo pode se alterado ao longo do tempo durante a execução de um programa. Embora uma variável possa assumir diferentes valores, ela só pode armazenar um valor a cada instante.



FORMAÇÃO DE IDENTIFICADORES:

Nomes dados às informações variáveis.

Devem começar por um caractere alfabético

Podem ser seguidos por mais caracteres e/ou numéricos

Não é permitido o uso de caracteres especiais

TIPOS DE VARIÁVEIS

As variáveis e as constantes podem ser basicamente de três tipos: Numéricas, caracteres ou lógicas.

Numéricas: Específicas para armazenamento de números, que posteriormente poderão ser utilizados para cálculos. Podem ser ainda classificadas como Inteiras ou Reais.

Caracteres: Formada por conjunto de caracteres alfanuméricos, deve ser colocado sempre entre aspas.

Lógicas: Armazenam somente dados lógicos que podem ser Verdadeiro ou Falso.

As variáveis só podem armazenar valores de um mesmo tipo, de maneira que também são classificadas como sendo numéricas, lógicas e literais.

Informação inteira: Ex.: 15, 0, 9, -3

Informação real: Ex.: 1,67, 0,00, -2,45

Informação caractere: Ex.: "O menino disse não"

Informação lógica: Ex.: Verdadeira ou falsa.

DECLARAÇÃO DE VARIÁVEIS:

Precisamos sempre definir as variáveis especificando qual o material dos objetos que por lá podem ser armazenados.

Exemplo:

Inteiro: AULA

Real: ABC, XPTO, D123

Caractere: NOME, ENDERECO

Lógico: NOTA, RESPOSTA

RESPOSTA é um local de memória que só pode conter valores do tipo lógico, ou seja, VERDADEIRO ou FALSO.

AULA: É um local de memória que só pode conter valores inteiros: 543, 12333, -987777

Observação importante: Não podemos definir duas variáveis com o mesmo nome em um mesmo algoritmo. **NUNCA**

Exercícios 3

1. Defina três identificadores válidos para cada tipo primitivo.
2. Defina agora três identificadores não válidos para cada tipo primitivo.
3. Defina as variáveis para calcular a média de três notas.
4. Cite um exemplo que use uma constante.

EXPRESSÕES MATEMÁTICAS

Chamamos de operadores matemáticos ou aritméticos o conjunto de símbolos que representa as operações básicas da matemática, a saber:

+ adição; - subtração; / divisão; ** potenciação; // radiciação

Exemplos: - 2 + 2; ABC/5; aula * ABC + XPTO - 8

FUNÇÕES MATEMÁTICAS QUE SERÃO USADAS

MOD (resto da divisão)

DIV (quociente inteiro da divisão)

INT (X): resulta a parte inteira de um número X

FRAC (X): resulta a parte fracionária de X

ABS (X): resulta o valor absoluto de X.

Exemplos:

15 DIV 7 resulta 2

15MOD 7 resulta 1

INT (34.567) resulta 34

FRAC (546.34) resulta 34

ABS(-34) resulta 34

PRIORIDADES: Na resolução das expressões aritméticas, as separações e funções matemáticas guardam entre si uma hierarquia.

1. Parênteses mais internos
2. Funções matemáticas
3. ** //
4. * /
5. + -

Exercícios 4

1. $3 ** 2 - 4 / 2 + ABS(5 - 3 * 5) / 2$
2. $(FRAC(A/B) + ABS@ ** 3$ onde $A = 5$; $B = 10$ e $C = -4$
3. $3 + (27 // 3) * ((3 MOD D + 0.5) * 2)$ onde $D = 1.5$

OPERADORES RELACIONAIS COM EXPRESSÕES LÓGICAS

= (igual a)

ou <> (diferente)

> (maior)

>= (maior igual)

< (menor)

<= (menor igual)

Exemplo

$2 * 4 = 24 / 3$

$$8 = 8$$

$$V$$

OPERADORES LÓGICOS

E – conjunção
 OU – disjunção
 NÃO – negação

$$2 < 5 \text{ e } 15 / 3 = 5$$

$$V \text{ e } V = V$$

$$V \text{ e } V = V$$

$$V$$

TABELA DOS OPERADORES LÓGICOS

P	Q	P e Q	P ou Q
V	V	V	V
V	F	F	V
F	V	F	V
F	F	F	F

PRIORIDADES: Na resolução das expressões lógicas, as operações e funções lógicas guardam entre si uma hierarquia.

- 1 – NÃO
- 2 – E OU

PRIORIDADES ENTRE TODOS OS OPERADORES

1. Substituição dos valores das variáveis
2. Parênteses mais internos
3. Funções matemáticas
4. Operadores aritméticos
5. Operadores relacionais
6. Operadores lógicos

Exercícios 5

1. NÃO $2 ** 3 < 4 ** 2$ OU $ABS (INT (-15 / 2)) < 10$

Resposta: V

2. $3 * (C / 4 + 5) < -8 * 3 + (15 \text{ MOD } 8 - 3)$ OU $5 ** 2 > INT (C * 0.7)$; onde $C = 20$

Resposta: V

3. NÃO $L \text{ OU } V \text{ E } ABS (C) \geq A \text{ DIV } A$ onde $L = F$; onde $C = -3.5$ e $A < > 0$

Para as questões 4 e 5 admita $A = 9$, $B = 3$, $C = 4$ e $D = 2$

4. $A ** 3 / B + 5 - C * D > C * D + A - B$ OU $A // 2 / D < 18 - A$

5. $5 + A * B \geq 16 // 4 - D \text{ E } 6 / A * C / (A - B) = 234$ OU $A / 4 - (7 + 5 * C) < A ** 2 - 3 * B$

COMANDOS E ESTRUTURAS NA LÓGICA DE PROGRAMAÇÃO

COMANDO DE ATRIBUIÇÃO:

Fornece um valor a uma certa variável, sendo que este valor seja compatível com o tipo de variável que foi declarada.

Exemplo:

Lógico: A, B;

Inteiro: X, AULA;

A ← Verdadeiro;

X ← 15;

AULA ← 8 + 13 DIV 5;

B ← 5 = 3;

COMANDO DE ENTRADA DE DADOS

Comando utilizado para enviar dados.

Exemplo:

Leia (X);

Leia (X, AULA, B);

COMANDOS DE SAÍDA DE DADOS

Comando utilizado para mostrar dados ou informações.

Exemplo:

Escreva (X);

Escreva (X, AULA, B);

Escreva ("O resultado do peso do aluno é de ", PESO, "quilos.");

Para representar um algoritmo em pseudocódigo, utilizaremos um formato padrão que facilita muito a passagem, posteriormente, da solução desenvolvida, para uma linguagem de programação estruturada. O formato-padrão pode ser visto a seguir:

Exemplo

Início

|

| *declaração de variáveis*

|

| *seqüência de ações*

|

|

| *resultado*

|

Fim.

ESTRUTURAS DE CONTROLE – SEQÜENCIAL

Estrutura de controle (ou **fluxo de controle**) refere-se à ordem em que instruções, expressões são executadas. Temos como estrutura de controle: Seqüencial, de Seleção e de Repetição

A Estrutura Seqüencial possui a característica de ter cada comando executado após, e somente após, o término do passo anterior. Quando concluída determinada linha de comando, seu comando posterior será sempre executado, necessariamente. Cada comando é executado uma vez. Os comandos são sempre executados de cima para baixo e da esquerda para a direita. O conjunto desses comandos é denominado bloco de instruções.

BLOCOS

Um algoritmo pode ser visto como um bloco. Ele serve para definir os limites nos quais as variáveis declaradas em seu interior são conhecidas e processadas. Para delimitar um bloco (algoritmo) utilizamos os delimitadores: início e fim. Obs.: Como no algoritmo não admite caracteres especiais, então não colocaremos acento agudo na palavra início.

Início

|

| *declaração de variáveis*

|

| *comando a;*

| *comando b;*

| *comando c;*

| .

| .

| *comando n;*

Fim.

*A melhor forma de prever o futuro é criá-lo.
Peter Drucker*

Exercício resolvido

1. Construa um algoritmo que calcule a média aritmética entre quatro notas quaisquer fornecidas pelo usuário.

Resolução Nº 1

início

Real: N1, N2, N3, N4;
Leia (N1, N2, N3, N4);
Escreva ((N1 + N2 + N3 + N4) / 4);

fim.

Resolução Nº 2

início

Real: N1, N2, N3, N4, MEDIA;
Leia (N1, N2, N3, N4);
MEDIA ← ((N1 + N2 + N3 + N4) / 4);
Escreva (MEDIA);

fim.

Resolução Nº 3

início

Real: N1, N2, N3, N4, MEDIA;
Leia (N1, N2, N3, N4);
MEDIA ← ((N1 + N2 + N3 + N4) / 4);
Escreva ("A média aritmética do aluno nas quatro notas tiradas é", MEDIA);

fim.

Exercícios 6 (Classe)

1. Faça um algoritmo para calcular o peso ideal de uma pessoa sabendo sua altura.

DADOS:

Para homens (72.7 * altura) – 58;

Para mulheres (62.1 * altura) – 44.7;

Questão Desafio

2. Faça um algoritmo para ler nove números inteiros quaisquer. Tirar a média aritmética dos primeiros, depois a média dos outros três e por fim a média dos três últimos. Escreva as três médias e a média das três médias.

Exercícios 7

1. Faça um algoritmo para ler dois números e imprimir a soma dos números lidos.

2. Faça um algoritmo para ler três números e imprimir a soma, média e produto dos números lidos.

3. Faça um algoritmo para ler os catetos de um triângulo retângulo e calcular e imprimir a sua hipotenusa.

5. Dois homens vão fazer uma viagem de automóvel. Entretanto, o carro faz 15 Km/litro de combustível. Faça um algoritmo para ler a quantidade de quilômetros que os homens irão rodar e mostrar quantos litros, no mínimo, precisam para fazer a viagem?

4. Faça um algoritmo para ler o preço de compra de uma mercadoria e calcular o seu preço de venda para que possa ser obtido um lucro de 30 %.

Questões Desafio

5. Faça um algoritmo para ler duas variáveis caracter e concatená-las em uma terceira.

6. Faça um algoritmo para ler duas variáveis inteiras e trocar o conteúdo lido de uma pela outra.

ESTRUTURAS DE CONTROLE – SELEÇÃO SIMPLES

SELEÇÃO: Permite a escolha de um grupo de ações e estruturas a ser executado quando determinadas condições, representadas por expressões lógicas, são ou não satisfeitas.

Este tipo de estrutura permite executar uma seqüência de comandos de acordo com o resultado de uma comparação (condição) ou de uma decisão. Após a avaliação da condição, uma das alternativas é executada. Sendo executada a alternativa <entao>, quando a condição for satisfeita (verdadeira), ou a alternativa <senao>, quando a alternativa não for satisfeita (falso).

SIMPLES

```

inicio
| declaração de variáveis
|
| se < condição > entao;
|   comando a;
|   comando b;
|   .
|   .
|   comando n;
| fimse;
|   .
|   .
|   comando n;
fim.

```

< condição > é uma expressão lógica que, quando inspecionada, pode gerar um resultado falso ou verdadeiro. Se < condição > for verdadeira, a cláusula que está entre o “se” e o “fimse” será executada. Se < condição > for falsa, a cláusula que está entre o “se” e o “fimse” na será executada.

IMPORTANTE → Todo “se” precisa de um “fimse”.

Exercício Resolvido

Faça um algoritmo para calcular o peso ideal de uma pessoa sabendo sua altura.
 Se HOMEM → $(72.7 * ALTURA) - 58$. Se MULHER → $(62.1 * ALTURA) - 44.7$

Resolução Nº 1

```

inicio
  real: ALT;
  caractere: SEXO;
  leia (SEXO, ALT);
  se SEXO = "MASCULINO" entao;
    escreva ("O peso ideal para um homem de ", ALT, "MT será de ", (72.7 * ALT) - 58, "Kg");
  fimse;
  se SEXO = "FEMININO";
    escreva ("O peso ideal para uma mulher de ", ALT, "mt será de ", (62.1 * ALT) - 44.7, "Kg");
  fimse;
fim.

```

Resolução Nº 2 → Acrescentando a variável PESOID

```

inicio
  real: ALT, PESOID;
  caractere: SEXO;
  leia (SEXO, ALT);
  se SEXO = "MASCULINO" entao;
    PESOID ← ((72.7 * ALT) - 58);
    escreva ("O peso ideal para um homem de ", ALT, "mt será de ", PESOID, "Kg");
  fimse;
  se SEXO = "FEMININO" entao;
    PESOID ← ((62.1 * ALT) - 44.7);
    escreva ("O peso ideal para uma mulher de ", ALT, "mt será de ", PESOID, "Kg");
  fimse;
fim.

```

Resolução Nº 3 → Acrescentando o teste com o sexo

início

```
real: ALT, PESOID;
caractere: SEXO;
leia (SEXO, ALT);
se SEXO = "MASCULINO";
    PESOID ← ((72.7 * ALT) – 58);
    Escreva ("O peso ideal para um homem de ", ALT, "mt será de ", PESOID, "Kg");
fimse;
se SEXO = "FEMININO" entao;
    PESOID ← ((62.1 * ALT) – 44.7);
    Escreva ("O peso ideal para uma mulher de ", ALT, "mt será de ", PESOID, "Kg");
fimse;
se SEXO # "MASCULINO" e SEXO # "FEMININO" entao;
    escreva ("Sexo digitado erradamente.");
fimse;
```

fim.

Resolução Nº 4 → Acrescentando o teste da altura

início

```
real: ALT, PESOID;
caractere: SEXO;
leia (SEXO, ALT);
se ALT <= 2.50 e ALT >= 0.60;
    se SEXO = "MASCULINO" entao;
        PESOID ← ((72.7 * ALT) – 58);
        Escreva ("O peso ideal para um homem de ", ALT, "mt será de ", PESOID, "Kg");
    fimse;
    se SEXO = "FEMININO" entao;
        PESOID ← ((62.1 * ALT) – 44.7);
        Escreva ("O peso ideal para uma mulher de ", ALT, "mt será de ", PESOID, "Kg");
    fimse;
    se SEXO # "MASCULINO" e SEXO # "FEMININO" entao;
        escreva ("Sexo digitado erradamente.");
    fimse;
fimse;
se ALT > 2.50 ou ALT < 0.60 entao;
    escreva("Este programa só reconhece altura maior que 0.60 mt e menor que 2.50 mt.");
fimse;
```

fim.

Exercícios 8

1. Faça um algoritmo para ler dois números e imprimir o maior, o menor ou então dizer se são iguais.
2. Faça um algoritmo para ler um número inteiro e dizer se o número lido é par ou ímpar.
3. Faça um algoritmo para ler dois números A e B e dizer se A é divisível por B.
4. Faça um algoritmo para ler dois números e imprimi-los em ordem crescente.
5. Faça um algoritmo para ler três números e imprimir o maior e o menor.

QUESTÕES DESAFIO

6. Faça um algoritmo para ler três números e imprimir se estes podem ou não formar um triângulo.
Obs.: Para formar os lados de um triângulo, cada um dos valores tem que ser menor que a soma dos outros dois.
7. Faça um algoritmo para ler três números e se estes podem formar um triângulo, se puderem formar então dizer qual tipo de triângulo: EQUILÁTERO, ISÓCELES, ESCALENO.
8. Faça um algoritmo que leia as três notas, as faltas e o nome de um aluno e imprima sua situação: APROVADO, REPROVADO POR FALTA ou REPROVADO POR MÉDIA.
Obs.: A média para aprovação é 6.0 e o limite de faltas é 17.

Exercícios 9

1. Faça um algoritmo para ler um valor real apresentando um salário e fazer com que este salário lido receba um aumento de acordo com a tabela:

FAIXA SALARIAL	AUMENTO
Até 300,00	50%
300,01 a 500,00	40%
500,01 a 700,00	30%
700,01 a 800,00	20%
800,01 a 1000,00	10%
Acima de 1000,00	5%

2. Faça um algoritmo que lendo a velocidade de um veículo, escreva em quanto tempo ele leva para percorrer a distância entre Sobra e Fortaleza.

Obs.: Distância Iguatu – Fortaleza = 380 Km

A velocidade lida será interpretada como sendo Km/h

3. Leia três palavras e escreva se das palavras lidas, as três são diferentes, as três são iguais ou pelo menos duas são iguais.

4. Leia o nome e a idade de três pessoas e escreva seus nomes em ordem crescente de idade.

5. Leia o conteúdo de uma variável lógica e escreva se é Verdadeiro ou Falso.

6. Leia o conteúdo de duas variáveis lógicas e escreva o resultado usando a tabela verdade do operador lógico E.

7. Leia o conteúdo de duas variáveis lógicas e escreva o resultado usando a tabela verdade do operador lógico OU.

8. Uma pessoa comprou 5 tipos diferentes de gêneros alimentícios. Faça um algoritmo que leia o nome do alimento, a quantidade comprada de cada um e o preço unitário escrevendo o nome deles e suas respectivas quantidades, além do preço total de cada um. Escrever também o valor total da compra sabendo que se a compra ultrapassar R\$ 100,00 terá um desconto de 10%. Se ultrapassar R\$ 50,00 e for menor do que R\$ 100,00 o desconto será de 5%. E abaixo de R\$ 50,00 não haverá desconto.

QUESTÕES DESAFIO

9. Uma pessoa precisa comprar 3 remédios. Porém tem somente R\$ 100,00. Faça um algoritmo que leia o nome e o preço de cada medicamento e escreva quais os medicamentos que ela pode comprar, se é que pode. (Para facilitar o algoritmo faça a compra por ordem de leitura.)

10. Faça um algoritmo que leia 9 números e escreva quantos números pares foram lidos.

ESTRUTURAS DE CONTROLE – SELEÇÃO COMPOSTA COM SE

início

declaração de variáveis

se < condição >;

comando a;

comando b;

senão

comando c;

comando d;

fimse;

comando e;

.

.

comando n;

fim.

Se < condição > for verdadeira, os comandos “a” e “b” serão executados.

Se < condição > for falsa, os comandos “c” e “d” serão executados.

Exercícios 10

1. Elabore um algoritmo que, dada a idade de um nadador, classifique-o em uma das seguintes categorias.

Infantil A = até 7 anos

Infantil B = 8 – 10 anos

Juvenil A = 11 – 13 anos

Juvenil B = 14 – 17 anos

Adulto = maiores de 17 anos

2. Refazer todos os exercícios passados na lista anterior, usando Estrutura de Seleção Composta.

SELEÇÃO ENCADEADA

Quando por algum motivo agrupamos várias seleções.

Heterogênea – Quando não conseguimos identificar um padrão lógico de construção.

```
se < cond1> entao;
    se < cond2> entao;
        com1;
        com2;
    fimse;
senão
    se < cond3> entao;
        com3;
        com4;
    senão
        com5;
        com6;
    fimse;
fimse;
```

Homogênea – Quando conseguimos identificar um padrão lógico de construção.

```
se < cond1> entao;
    se < cond2> entao;
        se < cond3> entao;
            se < cond4> entao;
                com1;
            fimse;
        fimse;
    fimse;
fimse;
```

```
se < cond1> entao;
    se < cond2> entao;
        com1;
        com2;
    fimse;
senão
    se < cond3> entao;
        com3;
        com4;
    fimse;
fimse;
```

Uso do “senão”

Podemos usar o “senão” se de duas formas

1ª Forma

```
se < cond1> entao;
    com1;
senão
    se < cond2> entao;
        com2;
    fimse;
fimse;
```

2ª Forma

```
Se < cond1> entao;
    Com1;
Senão se < cond2> entao;
    Com2;
    fimse;
fimse;
```

CONTADOR

Toda vez que precisamos contar algum movimento ou alguma repetição de um comando ou uma variável ou até mesmo constante usamos uma variável que chamamos de CONTADOR. Esta variável nada mais armazena do que o número de vezes que alguma coisa foi feita ou repetida.

Exemplo: Ler 10 números reais e dizer quantas vezes foi lido o número 2.

início

```
inteiro: CONTADOR;  
real: N1, N2, N3, N4, N5, N6, N7, N8, N9, N10;  
leia (N1, N2, N3, N4, N5, N6, N7, N8, N9, N10);  
CONTADOR ← 0;  
se N1 = 2;  
    CONTADOR ← CONTADOR + 1;  
fimse;  
se N2 = 2;  
    CONTADOR ← CONTADOR + 1;  
fimse;  
se N3 = 2;  
    CONTADOR ← CONTADOR + 1;  
fimse;  
se N4 = 2;  
    CONTADOR ← CONTADOR + 1;  
fimse;  
se N5 = 2;  
    CONTADOR ← CONTADOR + 1;  
fimse;  
se N6 = 2;  
    CONTADOR ← CONTADOR + 1;  
fimse;  
se N7 = 2;  
    CONTADOR ← CONTADOR + 1;  
fimse;  
se N8 = 2;  
    CONTADOR ← CONTADOR + 1;  
fimse;  
se N9 = 2;  
    CONTADOR ← CONTADOR + 1;  
fimse;  
se N10 = 2;  
    CONTADOR ← CONTADOR + 1;  
fimse;  
escreva ("O número 2 aparece ", CONTADOR, "vezes.");
```

fim.

Exercício de Classe

Uma empresa tem uma ficha para cada funcionário contendo seu nome, o número de horas trabalhadas em um determinado mês e o número de seus dependentes. Considerando que a empresa paga R\$ 6,80 por hora de trabalho, R\$ 5,20 por dependente e que desconta 8,7% do salário para pagar o INSS e 5% para ser descontado no I.R., fazer um algoritmo para ler o nome, número de horas trabalhadas, mês trabalhado e o número de dependentes de um determinado funcionário e escrever seu nome, mês trabalho, salário bruto, os valores descontados para imposto e finalmente seu salário líquido.

Estrutura de Controle: CASO

Outra estrutura de controle é o CASE (CASO), que se trata de uma seleção de multiplica escola, ou seja caso o conteúdo da variável X seja igual ao valor Vn, então o comando Cn será executado; caso contrário será inspecionado até se encontrar a igualdade ou terminarem os casos.

escolha X

```
| caso v1: c1;  
| caso v2: c2;  
| caso v3: c3;  
| caso v4: c4;
```

fimescolha;

ou

escolha X

```
| caso v1: c1;  
| caso v2: c2;  
| caso v3: c3;  
| caso v4: c4;  
| caso contrário: c5;
```

fimescolha;

Veja abaixo exemplo: Algoritmo que tendo como dados de entrada o preço de um produto e um código de origem, emite o preço junto de sua procedência. Caso o código de origem não seja nenhum dos especificados, o produto deve ser encarado como importado.

Código de origem:

1. Sul
2. Norte
3. Leste
4. Oeste
- 5 ou 6 – Nordeste
- 7,8 ou 9 – Sudeste
- 25 até 50 – Nordeste

Início

```

| {declaração de variáveis}
| Real: PRECO;
| Inteiro: ORIGEM;
| Leia( PRECO, ORIGEM);
| escolha ORIGEM
| | caso 1: escreva (PRECO, “ – produto do Sul”);
| | caso 2: escreva (PRECO, “ – produto do Norte”);
| | caso 3: escreva (PRECO, “ – produto do Leste”);
| | caso 4: escreva (PRECO, “ – produto do Oeste”);
| | caso 7, 8,9 : escreva (PRECO, “ – produto do Sudeste”);
| | caso 10 .. 20: escreva (PRECO, “ – produto do Centro-Oeste”);
| | caso 5,6,25 .. 50: escreva (PRECO, “ – produto do Nordeste”);
| | caso contrário: escreva (PRECO, “ – produto do importado”);
| | fimescolha;
|
Fim.
```

ESTRUTURAS DE REPETIÇÃO

I - Repetição com Teste no início.

Enquanto <condição> faça

- . c1;
- . c2
- . cn

Fimenquanto;

Consiste numa estrutura de controle do fluxo lógico que permite executar diversas vezes um mesmo trecho do algoritmo, porém, sempre verificando antes de cada execução se é “permitido” repetir o mesmo trecho.

Para realizar a repetição com teste no início, utilizamos a estrutura **enquanto**, que permite um bloco ou uma ação primitiva seja repetida enquanto uma determinada <condição> for verdadeira.

Enquanto < condição> faça

```

| c1; {comando 1}
| c2; {comando 2}
| c3; {comando 3}
| .
| .
| .
| cn; {comando n}
Fimenquanto;
```

Obs: Quando o resultado de <condição> for falso, o comando será abandonado. Se já da primeira vez o resultado for falso, os comandos não são executados nenhuma vez.

Vejamos o exemplo abaixo:

Para inserir o calculo da média dos alunos em um laço de repetição utilizando a estrutura enquanto – que <condição> utilizaríamos?

A condição seria que a quantidade de médias calculadas fosse menor ou igual a 50; porém, o que indica quantas vezes a média foi calculada? A estrutura (enquanto) não oferece este recurso, portanto, devemos estabelecer um modo de contagem, o que pode ser feito através de um contador representado por uma variável com um dado valor inicial, o qual é incrementado a cada repetição.

Exemplo de contador:

```
Inteiro : CON; {contador}
CON ← 17; {inicializar o contador}
CON ← CON + 1; {incrementar o contador de 1}
```

Execute mais algumas vezes esta última ação, observando o que acontece com a variável com.

Aplicando estes conceitos no exemplo anterior, temos:

```
Início
| {declaração de variáveis}
| real: N1,N2,N3,N4; {notas bimestrais}
| real: MA; {média anual}
| inteiro: CONT; {contador}
| CONT ← 0; {inicializando o contador}
| enquanto CON < 50 faça {teste da condição}
| | leia (N1,N2,N3,N4);
| | MA ← (N1+N2+N3+N4)/4;
| | escreva (Média Anual=", MA);
| | | se MA >= 6 então
| | | | início {bloco verdade}
| | | | | escreva ("aluno Aprovado");
| | | | | escreva ("Parabéns!!!");
| | | | fim; {fim bloco verdade}
| | | senão
| | | | início {bloco falso}
| | | | | escreva ("Aluno Reprovado");
| | | | | escreva ("Estude Mais!!!");
| | | | fim; {fim bloco falso}
| | fimse;
| | CONT ← CONT + 1;
| fimenquanto;
fim.
```

Agora que aprendemos a utilizar laços de repetição, tente solucionar a questão abaixo:

Faça um algoritmo que calcule a média anual de uma turma de 50 alunos, lembre-se que as médias trabalhadas são anuais e de cada aluno. Para essa resolução utilize uma variável acumuladora, como por exemplo:

Exemplo (acumulador):

```
Inteiro: X; {contador}
Inteiro: ACM; {acumulador}
ACM ← 0; {inicializar o acumulador}
Leia(X); {ler o primeiro valor}
ACM ← ACM + X; {incrementar o acumulador de x}
```

Não esqueça de fazer sanfonado o algoritmo.

```
Início
| {declaração de variáveis}
| real: MA, { média anual de um dado aluno}
| real: ACM; {acumulador}
| real: MAT; {média anual da turma}
| inteiro: CONT; {contador}
| CONT ← 0; {inicializando o contador}
| ACM ← 0; {inicializando o acumulador}
| enquanto CON < 50 faça {teste da condição}
| | leia (MA); {lendo a média de um dado aluno}
| | ACM ← ACM + MA;
| | CONT ← CONT + 1;
| fimenquanto;
| MAT ← ACM / 50; {atribui a variável MAT, a soma ACM/50}
| escreva("Média anual da turma = ", MAT);
fim.
```

Construa um algoritmo que calcule a média de todos os números pares que forem fornecidos pelo usuário. O valor de finalização será a entrada do número 0. Observe que nada impede que o usuário forneça quantos números ímpares quiser, com a ressalva de que eles não poderão ser acumulados.

```

Início
| {declaração de variáveis}
| inteiro: N; { número fornecido pelo usuário}
| inteiro: CONT; {contador}
| inteiro: ACM; {acumulador}
| real: MNP; {média dos números pares}
| CONT←0; {inicializando o contador}
| leia(N);
| enquanto N<> 0 faça {teste da condição}
| | se (N mod 2 ) = 0 então {resto da divisão e igual a zero}
| | | início
| | | | ACM← ACM+N;
| | | | CONT←CONT+1;
| | | fim;
| | fimse;
| | leia (N);
| fimenquanto;
| se CONT > 0 então
| | início
| | | MNP ← ACM/CONT;
| | | escreva (Média = ", MNP);
| | fim;
| fimse;
fim.

```

II – Repetição com teste no final. (Outro tipo de laço com loop).

Repita

```

| c1; {comando 1}
| c2; {comando 2}
| c3; {comando 3}
| .
| .
| .
| cn; {comando n}
Até <condição>;

```

Observe que pela sintaxe da estrutura, percebemos que o bloco (C1 ... Cn) é executado pelo menos uma vez, independente da validade da condição. Isto ocorre porque a inspeção da condição é feita após a execução do bloco.

Veja o exemplo abaixo:

```

Início
| {declaração de variáveis}
| real: MA, {média de um dado aluno}
| ACM, {acumulador}
| MAT; {média anual da turma}
| inteiro: CONT; {contador}
| CONT← 0; {inicializando o contador}
| ACM← 0; {inicializando o acumulador}
| repita {início do comando Repeta}
| | leia(MA);
| | ACM← ACM+MA;
| | CONT←CONT+1;
| até CONT >= 50; {teste da condição}
| MAT←ACM/50;
| escreva ("Média anual da turma = ", MAT);
Fim.

```

III – Repetição com variável de controle.

Nas estruturas de repetição vistas até agora, ocorrem casos em que se torna difícil determinar quantas vezes o bloco será executado. Sabemos que ele será executado enquanto uma condição for satisfeita – enquanto –, ou até que uma condição seja satisfeita – repita. A estrutura **para** repete a execução do bloco um número definido

De vezes, pois ela possui limites fixos:

Para V de Vi até Vf passo p faça

| c1;

| c2;

| .

| .

| .

| cn;

Fimpara;

Onde:

- V é a variável de controle;
- Vi é o valor inicial da variável;
- Vf é o valor final da variável V, ou seja o valor até o qual ela pode chegar;
- P é o valor do incremento dado à variável.

Exemplo: Elabore um algoritmo que imprima a tabuada do número 5.

Início

| inteiro: N, {número base}

| CONT, {contador}

| X; {variável de controle}

| leia(N);

| para CONT de 1 até 10 passo 1 faça

| | escreva(CONT, "X 5 = ", CONT*5);

| fimpara;

Fim.

Exemplo: Cálculo da média aritmética de uma turma fixa DE 50 ALUNOS.

Início

| real: MA, {média de um dado aluno}

| ACM, {acumulador}

| MAT, {média anual da turma}

| V; {variável de controle}

| ACM←0;

| para V de 1 até 50 passo 1 faça

| | leia(MA);

| | ACM←ACM+MA;

| fimpara;

| MAT←ACM/50;

| escreva("Média anual da turma = ", MAT);

Fim.

COMPARAÇÃO ENTRE ESTRUTURAS DE REPETIÇÃO

Podemos estabelecer dois "postulados" que relacionam as estruturas de repetição:

- Toda estrutura, **Enquanto**, pode ser convertida para **Repita** e vice-versa;
- Toda estrutura **Para** pode ser convertida em **Enquanto**, mas nem toda estrutura **Enquanto** pode ser convertida em **Para**.

Temos também a seguinte tabela comparativa que relaciona as características de cada estrutura de repetição:

Estrutura	Condição	Quantidade de Execuções	Condição de Existência
Enquanto	Início	?	Condição verdadeira
Repita	Final	Mínimo 1	Condição falsa
Para	Não tem	$(Vf - Vi + p) \text{ div } p$	$V < Vf$

Exemplo Elabore um algoritmo que, utilizando as três estruturas de repetição, imprima a tabuada do número 5.

- **Utilizando Enquanto:**

Início

| inteiro: CONT;

| CONT← 1;

| Enquanto CONT <= 10 faça

| | escreva(CONT, "X 5 = ", CONT*5);

| | CONT←CONT+1;

```
|   fimenquanto;
Fim.
```

- **Utilizando Repita:**

```
Início
|   inteiro: CONT;
|   CONT←CONT+1;
|   Repita
|   |   escreva(CONT, "x 5 = ", CONT*5);
|   |   CONT← CONT+1;
|   até CONT > 10;
Fim.
```

- **Utilizando Para:**

```
Início
|   inteiro: CONT;
|   Para CONT de 1 até 10 passo 1 faça
|   |   escreva (CONT, " x 5 = ", CONT*5);
|   fimpara;
Fim.
```

Agora que você já aprendeu a usar as estruturas de controle, elabore um algoritmo com o **Enquanto** e o **Repita**, para imprimir a tabuada de quaisquer números fornecidos pelo usuário, até encontrar como finalizador -1. Sabemos que o primeiro número base fornecido não é -1. Mãos a obra.

- **Utilizando Enquanto:**

```
Início
|   inteiro: CONT, N;
|   Leia(N);
|   Enquanto N <> -1 faça
|   |   CONT← 1;
|   |   enquanto CONT <= 10 faça
|   |   |   escreva(CONT, "x", N, "=", CONT*N);
|   |   |   CONT←CONT+1;
|   |   fimenquanto;
|   |   Leia(N);
|   fimenquanto;
Fim.
```

- **Utilizando Repita:**

```
Início
|   inteiro: N, CONT;
|   Leia(N);
|   Repita
|   |   CONT←1;
|   |   repita
|   |   |   escreva(CONT, "x", N, "=", CONT*N);
|   |   |   CONT←CONT+1;
|   |   até CONT > 10;
|   |   Leia(N);
|   até N= -1;
Fim.
```

Exercício (Desafio)

Uma empresa tem uma ficha para cada funcionário contendo seu nome, o número de horas trabalhadas em um determinado mês e o número de seus dependentes. Considerando que a empresa paga R\$ 6,80 por hora de trabalho, R\$ 5,20 por dependente e que desconta 8,7% do salário para pagar o INSS e 5% para ser descontado no I.R., fazer um algoritmo para ler o nome, número de horas trabalhadas, mês trabalhado e o número de dependentes de um determinado funcionário e escrever seu nome, mês trabalho, salário bruto, os valores descontados para imposto e finalmente seu salário líquido.

Exercício de Fixação

1. Faça um algoritmo para ler nove números inteiros quaisquer. Tirar a média aritmética deles, e escrever a média adquirida.
2. Faça um algoritmo para calcular o peso ideal de uma pessoa sabendo sua altura. DADO: para homens $(72,7 * \text{altura}) - 58$ e mulher $(62,1 * \text{altura}) - 44,7$.
3. Faça um algoritmo para ler dois números e imprimir o maior, o menor, o do meio ou então dizer se são iguais.
4. Faça um algoritmo para ler um número e dizer se ele é par ou ímpar.
5. Faça um algoritmo que leia 3 números quaisquer e escreva-os em ordem crescente.
6. Faça um algoritmo para ler três números e imprimir se estes podem ou não formar um triângulo.
7. Faça um algoritmo para ler três números e se estes podem formar um triângulo e dizer se o triângulo é "EQUILÁTERO", "ISÓSCELES", ou, 'ESCALENO".
8. Faça um algoritmo que leia as três notas de um aluno, o seu nome e imprima o nome do aluno com sua média e sua situação: APROVADO OU REPROVADO.
9. Elabore um algoritmo que leia um determinado número que represente um determinado mês do ano. Após a leitura escrever por extenso o nome do mês lido. Obs: deve ser um número de 1 a 12.
10. Escreva um algoritmo que leia 30 números e escreva sua soma.
11. Faça um algoritmo para ler o nome de 10 alunos e depois imprimi-los (escreve-los).
12. Elabora um algoritmo para imprimir a tabuada de 20.
13. Elabora um algoritmo para imprimir a tabuada de 20.
14. Elabore um algoritmo para escrever por extenso até 10.
15. Elabore um algoritmo que ao escrever um número ele irá dizer se esse número é primo. Lembre - se um número é primo quando ele é divisível por um por ele próprio.

DICA: Para aprender realmente lógica de programação ou qualquer outra matéria ou disciplina não existe segredo.

A fórmula é ...

APRENDER = PERSEVERANÇA + DEDICAÇÃO + VONTADE DE APRENDER + EXERCITAR.