

Algoritmo

CENTRO FEDERAL DE
EDUCAÇÃO TECNOLÓGICA DE
CAMPOS

**Curso Técnico de
Informática Industrial**

Apostila de Algoritmos

2007

EMENTA:

Estudo das formas para representação do pensamento lógico através de técnicas de desenvolvimento de algoritmos. Representação e manipulação de dados. Construções de algoritmos sequenciais, condicionais e com estruturas de repetição. Manipulação de estruturas de dados homogêneas e heterogêneas e utilização de sub-rotinas.

OBJETIVOS:

Fornecer elementos e técnicas que capacitem o aluno a construir algoritmos, através da identificação dos passos ou ações necessários para transformar um conjunto de dados de entrada em informações de resultado, promovendo dessa forma, um ambiente de prática da lógica de programação.

CONTEÚDO PROGRAMÁTICO:

1. Algoritmo Básico
 - 1.1. Pascal
2. Conceitos Básicos
 - 2.1. Lógica(problemas, solução e resultado)
3. Definição de Algoritmo e Pseudocódigo
4. Algoritmo cotidiano X Algoritmos Computacionais
5. Conceitos básicos do funcionamento do computador e da memória
6. Definição de variáveis, constantes e tipos primitivos
7. Estruturas Sequenciais e Estruturas de Seleção
 - 7.1. Comando de Atribuição
 - 7.2. Operadores aritméticos e funções matemáticas
 - 7.3. Comandos de entrada e saída
 - 7.4. Operadores relacionais e operadores lógicos
 - 7.5. Estrutura de seleção simples
 - 7.6. Estrutura de seleção composta
 - 7.7. Estrutura de múltipla escolha
8. Estrutura de repetição
 - 8.1. Conceito de contador e acumulador
 - 8.2. Repetição com teste no início (enquanto)
 - 8.3. Repetição com teste no fim (repita)
 - 8.4. Repetição com variável de controle (para)
9. Tipos Estruturados Homogêneos (vetores e matrizes)

BIBLIOGRAFIA BÁSICA (LIVROS TEXTOS):

Manzano, J. A.; Oliveira, J. F.: “Algoritmos - Lógica para desenvolvimento de Programação”. Érica. SP. pp. 265. 1996.

Forbellone, A. L. V; Eberspächer, H. F. “ Lógica de Programação”. São Paulo, Ed. McGraw-Hill, 1993.

Lages, G.: “Algoritmos e Estruturas de Dados”. São Paulo, Ed. LTC,1988.

Tremblay, B.: “Ciência dos Computadores. Uma abordagem Algorítmica”. São Paulo, Ed. McGraw-Hill, 1985.

Algoritmo

CRONOGRAMA DAS AULAS: Cada aula equivale a 2 períodos (sujeito a alterações)

Aula 1-18 - Noções de Lógica. Fatores a serem considerados na construção dos Algoritmos.

- Método para construir um algoritmo. Exercícios de fixação.
- Tipos de Informações, Dados (tipos primitivos de dados), Constantes x Variáveis, - Variáveis: uso, nomenclatura, declaração e atribuição.
- Operadores matemáticos, funções matemáticas.
- Instruções (comandos) básicos: entrada e saída, blocos de programas.
- Português estruturado. Exercícios de fixação. Algoritmos sequenciais.
- Estruturas de controle - Algoritmos com seleção
- Desvio condicional simples. - Desvio condicional composto e aninhados.
- Múltiplas opções - Operadores lógicos . Exercícios de fixação.
- Algoritmos propostos - exercícios. Esclarecimento de dúvidas. Prova 1.

Aula 19-30 - Estruturas de repetição - Algoritmos com repetição.

- Tipos de Laços de repetição
- Contador, acumulador e Exercícios.
- Algoritmos propostos e exercícios. Esclarecimento de dúvidas. Prova 2.

Aula 31-45 - Estruturas de Dados Homogêneas

-Matrizes de uma dimensão (vetores). Algoritmos propostos. - Matrizes de mais uma dimensão. Algoritmos propostos.

- Estruturas de Dados Heterogêneas
- Registros. Exercícios
- Subalgoritmos
- Utilização de funções e procedimentos. Variáveis globais e locais. - Esclarecimento de dúvidas. Prova 3.

Algoritmo

Introdução

Noções de Lógica

Lógica é a forma correta de organizar os pensamentos e demonstrar o raciocínio de maneira correta. A utilização da lógica é a melhor forma de solucionar problemas e atingir objetivos. Sempre que se quer pensar, falar ou escrever corretamente, deve-se colocar os pensamentos em ordem. Exemplo:

- Todo mamífero é animal
- Todo cavalo é mamífero
- Portanto, todo cavalo é animal

A lógica é muito importante em nossa vida, no dia - a - dia. Veja os exemplos abaixo:

- a) A gaveta está fechada.
A bala está na gaveta.
Preciso primeiro abrir a gaveta, para depois pegar a bala.
- b) Moramos em três pessoas.
Nenhum de nós dois quebrou o vaso de porcelana.
Quem quebrou o vaso?

Algoritmos

Algoritmo é a forma organizada de expressar uma sequência de passos que visam atingir um objetivo definido. Algoritmo é a lógica necessária para o desenvolvimento de um programa.

Apesar do nome estranho, os algoritmos são muito comuns no nosso cotidiano, como por exemplo, em uma receita de bolo. Nela estão escritos os ingredientes necessários e a sequência de passos ou ações a serem cumpridos para que se consiga fazer um determinado tipo de bolo.

Em um modo geral, um algoritmo segue um determinado padrão de comportamento, com objetivo de alcançar a solução de um problema.

Padrão de comportamento: imagine a sequência de números: 1, 6, 11, 16, 21, 26, ... Para determinar qual será o sétimo elemento dessa série, precisamos descobrir qual é a sua regra de formação, isto é, qual é o seu padrão de comportamento.

Como a sequência segue uma certa constância, facilmente determinada, somos capazes de determinar qual seria o sétimo termo ou outro termo qualquer.

Descrevemos então uma atividade bem cotidiana: trocar uma lâmpada. Apesar de parecer óbvia demais, muitas vezes fazemos este tipo de atividade inconscientemente, sem percebermos os pequenos detalhes. Vejamos como seria descrevê-la passo a passo:

- pegar uma escada;
- posicionar a escada embaixo da lâmpada;
- buscar uma lâmpada nova;
- subir na escada;
- retirar a lâmpada velha;
- colocar a lâmpada nova.

Algoritmo

Para se trocar a lâmpada, é seguida uma determinada sequência de ações, representadas através desse algoritmo. Como isso pode ser seguido por qualquer pessoa, estabelece-se aí um padrão de comportamento. A sequencialização tem por objetivo reger o fluxo de execução, determinando qual ação vem a seguir.

O algoritmo anterior tem um objetivo bem específico: trocar uma lâmpada. E se a lâmpada não estiver queimada? O algoritmo faz com ela seja trocada do mesmo modo, não prevendo essa situação. Para solucionar este problema, podemos efetuar um teste seletivo, verificando se a lâmpada está ou não queimada:

- pegar uma escada;
- posicionar embaixo da lâmpada;
- buscar uma lâmpada nova;
- ligar o interruptor;
- se a lâmpada não acender, então:
 - subir na escada;
 - retirar a lâmpada velha;
 - colocar a lâmpada nova.

Dessa forma, algumas ações estão ligadas à condição (lâmpada não acender). No caso da lâmpada acender, as três linhas:

- subir na escada;
- retirar a lâmpada velha;
- colocar a lâmpada nova.

não serão executadas.

Em algumas situações, embora o algoritmo resolva o problema proposto, a solução pode não ser a mais eficiente. Exemplo: três alunos devem resolver um determinado problema:

- O aluno A conseguiu resolver o problema executando 35 linhas de programa. - O aluno B resolveu o problema executando 10 linhas de programa
- O aluno C resolveu o problema executando 54 linhas de programa.

Obviamente, o algoritmo desenvolvido pelo aluno B é menor e mais eficiente que os demais. Isso significa que há código desnecessário nos demais programas.

Dessa forma, podemos otimizar o algoritmo anterior, uma vez que buscamos a escada e a lâmpada sem saber se serão necessárias:

- ligar o interruptor;
- se a lâmpada não acender, então:
 - pegar uma escada;
 - posicionar a escada embaixo da lâmpada;
 - buscar uma lâmpada nova;
 - subir na escada;
 - retirar a lâmpada velha;
 - colocar a lâmpada nova.

Podemos considerar ainda que a lâmpada nova pode não funcionar. Nesse caso devemos trocá-la novamente, quantas vezes for necessário, até que a lâmpada acenda:

- ligar o interruptor;
- se a lâmpada não acender, então:
 - pegar uma escada;
 - posicionar a escada embaixo da lâmpada;

Algoritmo

- buscar uma lâmpada nova;
- subir na escada;
- retirar a lâmpada velha;
- colocar a lâmpada nova;
- se a lâmpada não acender, então:
 - retirar a lâmpada;
 - colocar outra lâmpada;
 - se a lâmpada não acender, então: ...

Observamos que o teste da lâmpada nova é efetuado por um conjunto de ações: - se a lâmpada não acender então:

- retire a lâmpada
- coloque outra lâmpada

Em vez de escrevermos várias vezes este conjunto de ações, podemos alterar o fluxo sequencial de execução do programa, de forma que, após executar a ação “coloque outra lâmpada”, voltemos a executar a ação “se a lâmpada não acender”.

Precisa-se então determinar um limite para tal repetição, para garantir que ela cesse quando a lâmpada finalmente acender:

- enquanto a lâmpada não acender, faça:
 - retire a lâmpada
 - coloque outra lâmpada

Uma versão final do algoritmo, que repete ações até alcançar o seu objetivo: trocar a lâmpada queimada por uma que funcione, é apresentada abaixo.

- ligar o interruptor;
- se a lâmpada não acender, então:
 - pegar uma escada;
 - posicionar a escada embaixo da lâmpada;
 - buscar uma lâmpada nova;
 - subir na escada;
 - retirar a lâmpada velha;
 - colocar a lâmpada nova;
 - enquanto a lâmpada não acender, faça:
 - retirar a lâmpada;
 - colocar outra lâmpada.

Até agora, estamos efetuando a troca de uma única lâmpada. Todo o procedimento poderia ser repetido 10 vezes, por exemplo, no caso de quisermos trocar 10 lâmpadas.

Inicialmente, tínhamos um pequeno conjunto de ações que deveriam ser executadas (estrutura sequencial). Através de uma condição, incluímos posteriormente uma estrutura de seleção. Na necessidade de repetir um determinado trecho do algoritmo, construiu-se no final uma estrutura de repetição.

Fatores a serem levados em consideração na construção de um algoritmo

1. Complexidade

Percebeu-se, na medida em que colocávamos situações novas no problema a ser resolvido, que ia

Algoritmo

aumentando a complexidade do algoritmo. Esse certamente é o maior problema envolvido na construção de algoritmos. A complexidade pode ser vista como um sinônimo de variedade (quantidade de situações diferentes que um problema pode apresentar), as quais devem ser previstas na sua solução.

Já que conviver com a complexidade é um mal necessário, é saudável fazer o possível para diminuí-la ao máximo, a fim de controlar o problema e encontrar sua solução.

Deve-se diferenciar O que de Como. Muitos programadores aumentam a complexidade de um devido problema desnecessariamente. A forma errada de interpretação de um problema pode levar a respostas irrelevantes à solução almejada ou até mesmo a nenhuma solução, gerando algoritmos mais complexos do que o necessário.

Exemplo: digamos que se pergunte a um leigo a respeito de um relógio: -

Como é um relógio?

= É um instrumento com três ponteiros concêntricos.

Como a descrição não é relevante, poderíamos indagar: -

Um relógio com 2 ponteiros é possível?

= É... pode ser!

Poderíamos ainda indagar:

- E um relógio com apenas 1 ponteiro não poderia ser uma possibilidade? =

Bem... Pode ser com 3, 2 ou 1 ponteiro.

- E sem ponteiro pode?

= Ah!, Sim! Pode ser digital

Já a pergunta: “O que é um relógio?”, poderia resultar na resposta: - É um instrumento cuja finalidade é marcar o decorrer do tempo.

Ou seja, algumas variáveis podem aumentar ou diminuir a complexidade de um sistema quando forem bem ou mal utilizadas.

2. Legibilidade

Mede a capacidade de compreensão de um algoritmo por qualquer observador (que não o construiu); a clareza com que sua lógica está exposta. Quanto mais legível for um algoritmo, menor será sua complexidade.

3. Portabilidade

Devido a quantidade enorme de linguagens de programação existentes, não será adotada nenhuma linguagem específica para trabalhar os algoritmos (ex: C, pascal, Java, etc.). Isso porque a solução do problema fica ligada a características e recursos da linguagem na qual ela foi concebida.

Utilizaremos uma pseudo-linguagem (linguagem fictícia) que visa a permitir a representação dos algoritmos através da língua portuguesa (português estruturado). Esses algoritmos poderão ser convertidos facilmente para qualquer linguagem de programação usual (Basic estruturado, C, pascal, Java).

4. Técnica de resolução por método cartesiano

A famosa frase de Descartes “Dividir para conquistar” é muito importante dentro da programação. É um método que ataca um problema grande, de difícil solução, dividindo-o em problemas menores, de solução

Algoritmo

mais fácil. Se necessário, pode-se dividir novamente as partes não compreendidas. Esse método pode ser esquematizado em passos:

1. Dividir o problema em partes
2. Analisar a divisão e garantir a coerência entre as partes.
3. Reaplicar o método, se necessário

5. Planejamento reverso

Consiste em, a partir do resultado final, determinar quais são os componentes básicos. Ou seja, a partir da saída desejada, devemos poder determinar, reversamente, quais são os componentes da entrada de dados necessários.

Método para construir um algoritmo

Utilizando os conceitos já desenvolvidos, esquematizaremos um método para construir um algoritmo logicamente correto:

1. Ler atentamente o enunciado

Deve-se reler o enunciado de um exercício quantas vezes for necessário, até compreendê-lo completamente. A maior parte da resolução de um exercício consiste na compreensão completa do enunciado.

2. Retirar a relação das entradas de dados do enunciado

Através do enunciado, descobrimos quais são os dados que devem ser fornecidos ao programa, via teclado, a partir dos quais são desenvolvidos os cálculos. Obs. Pode haver algum algoritmo que não necessite da entrada de dados (pouco comum).

3. Retirar do enunciado, a relação das saídas das informações

Através do enunciado podemos descobrir quais são as informações que devem ser mostradas para compor o resultado final, objetivo do algoritmo.

4. Determinar o que deve ser feito para transformar as entradas nas saídas especificadas

Nessa fase é que teremos a construção do Algoritmo propriamente dito. Devemos determinar qual sequência de passos ou ações é capaz de transformar um conjunto de dados nas informações de resultado. Para isso, utilizamos os fatores descritos anteriormente, tais como legibilidade, portabilidade, método cartesiano e planejamento reverso, e finalmente podemos construir o algoritmo.

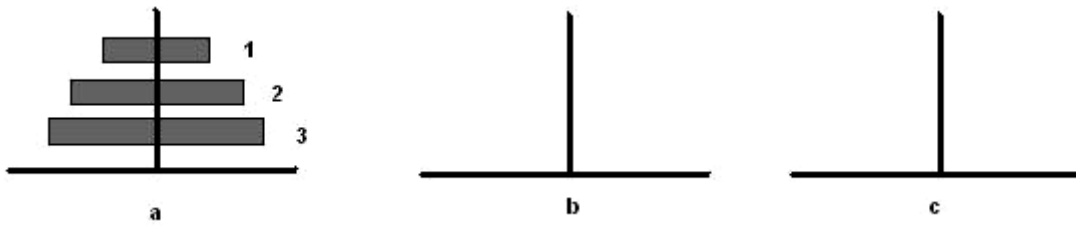
Exercícios de Fixação

1. Um homem quer atravessar um rio com um barco que pode carregar ele mesmo e apenas mais uma de suas três cargas: um lobo, um carneiro e um maço de alfafa. O que o homem deve fazer para atravessar o rio sem perder nenhuma de suas cargas?

2. Elabore um algoritmo que mova 3 discos de uma torre de Hanói, que consiste em 3 hastes (a-b-c), uma das quais serve de suporte para os três discos de tamanhos diferentes (1-2-3), os menores sobre os maiores. Pode-se

Algoritmo

mover um disco de cada vez para qualquer haste, sendo que nunca deve ser colocado um disco maior sobre um menor. O objetivo é transferir os três discos da haste A para haste C.



Mova <disco n> da haste <n1> para haste <n2>

-
-
-
-
-
-
-

Tipos de Informações

Podemos classificar os tipos de informações a serem processadas, a grosso modo, em dados e instruções.

Dados

São as informações a serem processadas por um computador. Consideremos 3 tipos de dados: numéricos (inteiros e reais), caracteres e lógicos.

Tipos primitivos de dados:

1.a) **Inteiro**: toda e qualquer informação numérica que pertença ao conjunto dos números inteiros (negativa, nula ou positiva). Exemplos: 39, 0, -56 entre outros.

a) Ele tem 15 irmãos.

b) A temperatura desta noite será de -2 graus.

1.b) **Real**: toda e qualquer informação numérica que pertença ao conjunto dos números reais (negativa, nula ou positiva, inteiro ou fracionário). Exemplos: -4, 3, 0, 35, 1,23

a) Ela tem 1,73 metro de altura.

b) Meu saldo bancário é de - R\$ 121,07.

2) **Caractere**: São caracterizadas como tipos caracteres, as seqüências contendo letras, números e símbolos especiais. Uma seqüência de caracteres deve ser indicada entre aspas (""). Este tipo de dado é também conhecido como alfanumérico, string, literal ou texto. Exemplos: "Rua Alfa, 52 Apto 1", "Fone: 574-9988", "04387-456", " ", "7".

a) Constava na prova: "Use somente caneta!".

3) **Lógico**: São caracterizados como tipos lógicos, os dados com valores verdadeiro e falso, sendo que este tipo de dado poderá representar apenas um dos dois valores. Ele é chamado por alguns de tipo booleano, devido à contribuição do filósofo e matemático inglês George Boole na área da lógica matemática. Os dados do tipo lógico, poderão ser apresentados e delimitados pelo caractere ponto (.) ou não: falso, .f., verdadeiro ou .v.

Exemplo: você pode entrar.

A tabela abaixo resume os tipos de dados mais comuns e sua definição nas linguagens mais conhecidas:

Tipo	Exemplo	Basic (VB)	Pascal/Delphi	Java/C++
Inteiro	2 45 100	Integer	Integer	int
Real	2,456 101,19	Single	Real	float
Caractere / String	"A" "4" "Algoritmos"	Byte / String	Char / String	Char / String
Lógico	Verdadeiro Falso .V. .F.	True / False	True / False	True / False

Exercícios de Fixação

Determine qual é o tipo primitivo de informação presente nas sentenças abaixo:

a) A placa "Pare!" tinha dois furos de bala.

b) Josefina subiu 5 degraus para pegar uma maçã boa.

c) Alberta levou 3 horas e meia para chegar ao hospital onde concebeu uma garota.

d) Julia escreveu em sua casa: "Preserve o meio ambiente", e ficou devendo R\$ 20,59 ao vendedor de tintas.

e) José recebeu sua 18^a medalha na nataçao por ter alcançado a marca de 57,3 segundos nos 100 metros rasos.

Constantes x Variáveis:

Entende-se por uma informação constante, aquela que não sofre nenhuma alteração no decorrer do tempo. Por exemplo, na fórmula a seguir, o que é constante e o que é variável?

RESULTADO ← ENTRADA + 2.25

O uso de Variáveis:

Tem-se como definição de variável tudo aquilo que é sujeito a variações. Variáveis são espaços de memória que são alocados para armazenar informações. Por que precisa-se de variáveis?

R: Para armazenar valores que serão utilizados posteriormente. Ex.: em um cálculo complexo, resultados intermediários podem ser armazenados e posteriormente processados para se obter o resultado final.

Imagine que a memória de um computador é um grande armário com diversas gavetas, onde cada gaveta pode conter somente um valor (numérico, caractere ou lógico). Deve-se concordar que é necessário identificar cada gaveta para sua utilização posterior, através de um nome.

Nomenclatura e declaração das variáveis:

O **nome** de uma variável faz referência ao endereço de memória onde essa variável se encontra. Existem algumas regras para a utilização das variáveis:

Nome de variável pode ter 1 ou mais caracteres

O primeiro caractere de uma variável **sempre** deve ser uma letra

Nenhuma variável pode ter espaço em branco em seu nome

Nenhum nome de variável pode ser uma palavra reservada a uma instrução da linguagem usada

Poderão ser utilizados somente letras, números e sublinhado (embora algumas linguagens aceitem o cedilha e acentuação).

Procure sempre utilizar variáveis que tenham sentido (Mnemônicos). Ex: alt_homens ao invés de x.

A declaração de uma variável também depende da linguagem. A variável X será declarada do tipo inteiro e a variável Y do tipo real. Observe a definição em cada linguagem:

Linguagem	Variável X	Variável Y
Visual Basic / Basic	Dim X as integer	Dim Y as single
Pascal / Delphi	X: integer;	Y: real;
Java / C++	Int X;	Float Y;

Nos algoritmos, não será cobrada a distinção entre inteiro e real. Utilizaremos a declaração das variáveis simplesmente como numérica, na forma apresentada abaixo:

```
variáveis
numérico a,b,c      -> 1, 3, 5.4, 8.72, 0
caracter d           -> a, b, teste, Juca
lógico entra         -> verdadeiro ou falso
```

Algoritmo

Atribuição de variáveis:

O comando de atribuição (=, := ou \leftarrow) permite fornecer um valor a uma certa variável, onde o tipo de informação deve ser compatível com o tipo de variável utilizada, ou seja, somente poderemos atribuir “Pedro” a uma variável do tipo caractere. Exemplos:

$a \leftarrow \text{“mesa”}$

$b := 2 + 5 - XB$

$c = -5.4 - b$

Quando uma variável é declarada (criada) qual o seu valor inicial?

R: Não se pode saber o valor inicial de uma variável, pois na memória existem várias informações armazenadas, muitas delas estão sendo utilizadas pelo computador, mas podem existir espaços em que foram armazenadas informações anteriormente mas que não estão mais em uso. Esses espaços, mesmo com informação estão disponíveis para novas variáveis. Uma variável pode ser criada em um desses espaços, e seu valor será um "lixo" qualquer da memória. Mas uma variável pode ser criada em um espaço vazio da memória, nesse caso seu valor será nulo. Para resolver esse problema do valor inicial da variável, algumas linguagens inicializam suas variáveis com 0 (zero) ou nulo.

Suponha que fosse atribuído os seguintes valores às seguintes variáveis:

$A := \text{“mesa”}$

$B := 0$

$C := 2$

$D := -5.4$

$E := \text{“João”}$

$F := 5.656$

Veja abaixo como poderia ficar a Memória do computador:

1	2	Mesa	0	endereço
C:\		125		
BA		TXT		
-5,4	xYz		2	
João	30		5656	

Utilizado

Não vazio, não utilizado (lixo)

Vazio, não utilizado

Operadores Aritméticos:

Unários ou binários, têm uma ordem de prioridade que é alterada somente com utilização de parênteses.

Operador	Operação	Tipo	Prioridade matemática
-	Inversão de sinal	Unário	1
$^$, **	Exponenciação	Binário	2
/	Divisão	Binário	3
*	Multiplicação	Binário	3
+	Adição	Binário	4
-	Subtração	Binário	4

Funções matemáticas

Somente será permitida a utilização das seguintes funções nos algoritmos:

inteiro(x): retorna o valor inteiro de x. Se por exemplo, $x=3,452$, retornará 3

abs(x): retorna o valor absoluto de x. Se por exemplo, $x=-2,45$, retornará 2,45 (não existe no AMBAP)

resto(x,y): retorna o resto da divisão de x por y. Por exemplo, resto(4,3) retornará 1

Expressões matemáticas ou fórmulas matemáticas:

Uma expressão matemática apresentada como: $x = \{43 \cdot [55 : (30+2)]\}$
nos algoritmos deve ser apresentada como: $x := (43 * (55 / (30+2)))$

Uma fórmula que matematicamente é apresentada como: $\text{Área} = \frac{b \cdot h}{2}$

nos **algoritmos** deve ser apresentada como: $\text{Area} := (b * h) / 2$

Instruções básicas ou Comandos básicos

As instruções ou comandos básicos são o conjunto de palavras-chave de uma determinada linguagem de programação. Variam de linguagem para linguagem mas possuem o mesmo significado, assim como as palavras na língua falada (português, inglês, etc.). Esses comandos, colocadas de forma estratégica, formarão os blocos de programas e, como são **palavras reservadas**, **não** devem ser utilizadas como nomes de variáveis.

Daqui para frente você terá contato com os comandos básicos em português estruturado que serão utilizados nos algoritmos, tais como: **início, fim, var, variáveis, programa, faça enquanto, se... então, senão, ler, escrever, repita, numérico, caractere, lógico**, etc.

Comandos de Entrada e Saída de dados:

Assim como o ato de respirar envolve a entrada de ar, processamento pelo organismo e saída de ar, todo programa executado em um computador (salvo algumas exceções) consiste de três etapas ou pontos de trabalho: a **entrada de dados**, o seu **processamento** e a **saída** dos mesmos.

A entrada de dados será feita através da instrução **ler**
leia <variável>

Exemplo:
leia A

A saída de dados será feita através da instrução **escrever**

escreva <variável>

Exemplo:
escreva (A)
escreva ("Valor de A:", A , "A Soma de B com C é :", B+C)
escreva ("Você pesa ", X , " quilos e tem ", idade, "anos.")

Aqui colocamos em prática os conceitos aprendidos anteriormente, nos fatores levados em consideração na construção de um algoritmo. Sempre que estamos diante de um problema, este deve ser primeiro resolvido por nós, para que depois seja resolvido pelo computador. Isto é, ele deve ser muito bem compreendido para que depois se busque a solução no computador.

O computador deve ser utilizado para agilizar os processos que executados mecanicamente seriam muito lentos. Um exemplo bem simples: Numa turma com 50 alunos, o professor deve calcular a média de cada aluno baseado em 3 provas, sendo que a primeira tem peso 1, a segunda peso 2 e a terceira peso três. Para cada aluno o professor deve fazer o seguinte cálculo: $(\text{nota } 1 + \text{nota } 2 * 2 + \text{nota } 3 * 3) / 6$.

Algoritmo

Este processo seria muito lento, e com a utilização do computador e um pequeno algoritmo (programa), seria necessário somente digitar os valores de entrada para aparecer na tela a média:

```
Digite Nota 1: 10.0
Digite Nota 2: 9.0
Digite Nota 3: 8.0
Média = 8.67
```

Exercício: faça um algoritmo, cuja execução do mesmo corresponda com o que está apresentado na tela acima.

Blocos

Um bloco pode ser definido como um conjunto de ações de função bem definida. No nosso caso, o algoritmo será visto como um bloco. Seus delimitadores serão as palavras início e fim (veja o exemplo abaixo, à direita).

Exemplo de um programa em português estruturado:

Português estruturado com definição das variáveis
--> estrutura parecida com pascal (utilizada no VisuAlg)

variaveis

```
a,b,soma :numerico
```

inicio

```
Leia (A)
```

```
Leia (B)
```

```
Soma := A + B
```

```
escreva ("A soma de A com B é:" , soma)
```

```
fim
```

Para testes práticos em laboratório, será utilizado o ambiente Visualg por ser de sintaxe simples e clara, o que facilita a sua compreensão.

Exercícios de Fixação

1) Considerando as variáveis abaixo, assinale N para numérico, C para texto e L para Lógico:

() 1000

() "12"

() .v.

() "Casa 8"

() - 456

() - 4.56

() "4.56"

() .verdadeiro.

() "cinco"

() "5"

() 456

() 45.8976

Algoritmo

2) Supondo que as variáveis Nb, Na, Nmat e Sx sejam utilizadas para armazenar a nota de um aluno, o nome do aluno, o número de matrícula do aluno e o sexo do aluno, respectivamente. Declare-as corretamente:

3) Encontre os erros da seguinte declaração de variáveis:

```
variaveis
    endereço, nfilhos, valor$, xpto, c, peso :numerico
    idade, x :caractere
    2lâmpada :logico
```

4) Seguindo a prioridade dos operadores, qual é o resultado das seguintes operações

a) $4 ** 2/4 + 17 - 5/2,5$

b) $3 ** 3 * 2 + \text{abs}(4-6-1)/4 + 11$

c) $3 * \text{inteiro}(7/2)^2 + \text{abs}(4-6)/4 + 1 + \text{resto}(4,2)$

5) Assinale com um x os nomes válidos para variáveis abaixo:

<input type="checkbox"/> Alpha	<input type="checkbox"/> BJ153	<input type="checkbox"/> A:B
<input type="checkbox"/> 5x	<input type="checkbox"/> K7	<input type="checkbox"/> Inps
<input type="checkbox"/> E(13)	<input type="checkbox"/> X-Y	<input type="checkbox"/> FGTS
<input type="checkbox"/> lBeta	<input type="checkbox"/> Notas/2	<input type="checkbox"/> PE&E
<input type="checkbox"/> X	<input type="checkbox"/> Notas	<input type="checkbox"/> int
<input type="checkbox"/> ABC	<input type="checkbox"/> Médias	<input type="checkbox"/> AeQ*

6) Utilizando o seguinte trecho de algoritmo, e considerando a entrada dos valores 3 para x e -2 para y:

```
variaveis
    x,y,z : numerico
inicio
    leia (x)
    escreva (x, "ao cubo = ", x^3 )
    leia (y)
    escreva (x+y)
    z := x / y
    escreva (z)
    z := inteiro(z)
    escreva (z)
    z := z+1
    x := abs(y-z)
    x := resto(x,z)
    escreva (x)
```

Explique o que está acontecendo em cada linha e qual será o resultado de cada ação executada.

Algoritmos propostos

- 1) Apresente o seguinte algoritmo:
Ler 2 valores, no caso variáveis A e B.
Efetuar a soma das variáveis A e B implicando seu resultado na variável X.
Apresentar o valor da variável X após a soma dos dois valores indicados.
- 2) A fórmula para calcular a área de uma circunferência é: $A = \pi R^2$. Sendo que o valor de $\pi = 3.14159$ basta estabelecer:
Ler 1 valor para raio, no caso a variável R.
Efetuar o cálculo da área, elevando o valor de R ao quadrado e multiplicando por π .
Apresentar o valor da variável A.
- 3) Leia duas variáveis A e B. A seguir, calcule a soma entre elas e atribua à variável SOMA. A seguir escreva o valor desta variável.
- 4) Leia duas variáveis A e B. A seguir, calcule o produto entre elas e atribua à variável PROD. A seguir mostre a variável PROD com mensagem correspondente.
- 5) Leia 2 variáveis A e B, que correspondem a 2 notas de um aluno. A seguir, calcule a média do aluno, sabendo que a nota A tem peso 3,5 e a nota B tem peso 7,5.
- 6) Leia 3 variáveis A e B e C, que são as notas de um aluno. A seguir, calcule a média do aluno, sabendo que a nota A tem peso 2, a nota B tem peso 3 e a nota C tem peso 5.
- 7) Leia 4 variáveis A,B,C e D. A seguir, calcule e mostre a diferença do produto de A e B pelo produto de C e D ($A*B-C*D$).
- 8) Escreva um algoritmo que leia o número de um funcionário, seu número de horas trabalhadas, o valor que recebe por hora e calcule o salário desse funcionário. A seguir, mostre o número e o salário do funcionário.
- 9) Escrever um algoritmo que leia o nome de um vendedor, o seu salário fixo e o total de vendas efetuadas por ele no mês (em dinheiro). Sabendo que este vendedor ganha 15% de comissão sobre suas vendas efetuadas, informar o seu salário no final do mês.
- 10) Escrever um algoritmo que calcule e mostre o fatorial de 5.
- 11) Escrever um algoritmo que escreva os números ímpares entre 10 e 20.
- 12) Escrever um algoritmo que leia o código de uma peça 1, o número de peças 1, o valor unitário de cada peça 1, o código de uma peça 2, o número de peças 2, o valor unitário de cada peça 2 e calcule e mostre o valor a ser pago
- 13) Escrever um algoritmo para calcular o volume de uma esfera sendo fornecido o valor de seu raio. ($\frac{4}{3} * \pi * R^3$)
- 14) Escrever um algoritmo que leia três valores: A, B e C. A seguir, calcule e mostre:
 - a) a área do triângulo que tem A por base e C por altura.
 - b) a área do círculo de raio C.
 - c) A área do trapézio que tem a e b por bases e c por altura.

Algoritmo

- d) A área do quadrado que tem lado b.
- e) a área do retângulo que tem lados a e b.

- 15) Escrever um algoritmo que leia 3 valores A, B e C. A seguir, encontre o maior dos três valores e o escreva com a mensagem “É o maior”. Utilize a fórmula: $\text{MaiorAB} = (a + b + \text{abs}(a - b)) / 2$
- 16) Escrever um algoritmo para determinar o consumo médio de um automóvel sendo fornecidos a distância total percorrida pelo automóvel e o total de combustível gasto.
- 17) Escrever um algoritmo que leia dois pontos quaisquer no plano, $p1(x1,y1)$ e $p2(x2,y2)$ e calcule a distância entre eles.

$$d = \sqrt{(x2-x1)^2 + (y2-y1)^2}$$

- 18) Escrever um algoritmo que calcule e mostre a quantidade de litros de combustível gastos em uma viagem, utilizando-se um automóvel que faz 12 KM/L. Deve-se fornecer o tempo gasto na viagem e a velocidade média durante a mesma. Assim pode-se obter distância percorrida:
Distância = Tpo * vel
Litros_usados = distância/12.
- 19) Dois carros (x e y) saem de uma mesma cidade. O carro X sai com velocidade constante de 60 Km/h e o 2º com velocidade constante de 90 Km/h. A seguir faça a leitura de uma distância (em Quilômetros) e calcule quanto tempo leva para o carro mais veloz tomar essa distância do outro carro.
- 20) Considerando 3 carros (X,Y e g) que saem de uma cidade A. A seguir, considerando que o caminho a percorrer pelo carro X é de 360 Km, pelo carro Y é de 270 Km e pelo carro g é de 310 KM, faça a leitura da velocidade dos 3 carros e mostre o tempo em horas que cada um deles levará para percorrer seu percurso.
- 21) Escrever um algoritmo que leia um valor em Reais. A seguir, calcule o menor número de notas possíveis no qual o valor pode ser decomposto. As notas consideradas são de 100, 50, 20, 10, 5, 2 e 1. A seguir mostre o valor lido e a relação de notas necessárias.
- 22) Escrever um algoritmo que leia o tempo de duração em segundos de um determinado evento em uma fábrica e informe-o expresso no formato horas:minutos:segundos.
- 23) Escrever um algoritmo que leia a idade de uma pessoa em dias e informe-a em anos, meses e dias.
- 24) Leia a hora inicial e a hora final de um jogo. A seguir calcule a duração do jogo, sabendo que o mesmo pode começar em um dia e terminar em outro, tendo uma duração máxima de 24 horas.
- 25) Com relação ao exercício anterior, calcule o tempo considerando os minutos.

Algoritmos com Seleção - Estruturas de Controle

Já vimos entrada, processamento e saída, que compõem os algoritmos puramente sequenciais, bem como a utilização de variáveis, constantes e operadores aritméticos. Porém, esses recursos são muito limitados dentro da programação, e haverá momentos em que um valor deverá ser tratado para que possa ser efetuado um processamento mais adequado.

Através das estruturas básicas de controle do fluxo de execução - seqüenciação, seleção, repetição - e da combinação delas, poderemos criar algoritmos para solucionar qualquer problema.

Imagine a seguinte situação: queremos um programa que mostre a média de um aluno. Até aí muito simples, mas além de calcular a média, queremos simplesmente que o programa apresente uma mensagem dizendo se o aluno está Aprovado, no caso da sua nota ter sido maior do que 5.

Desvio condicional simples

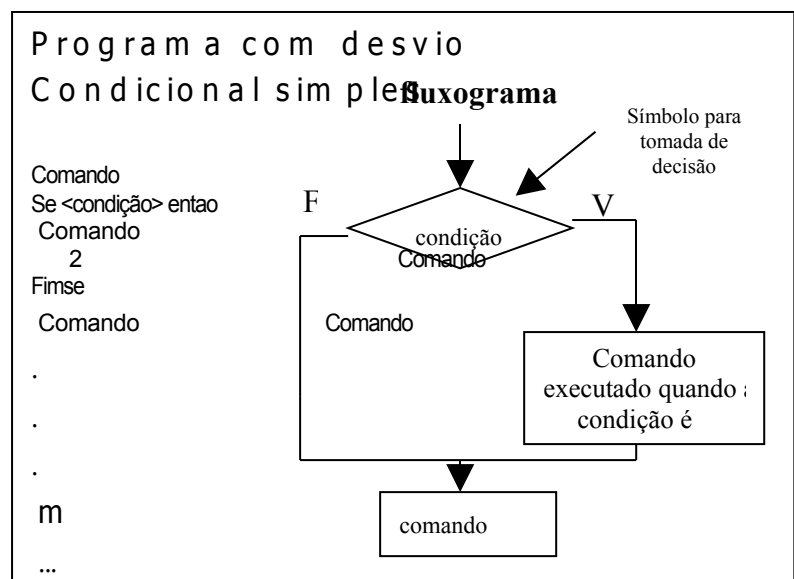
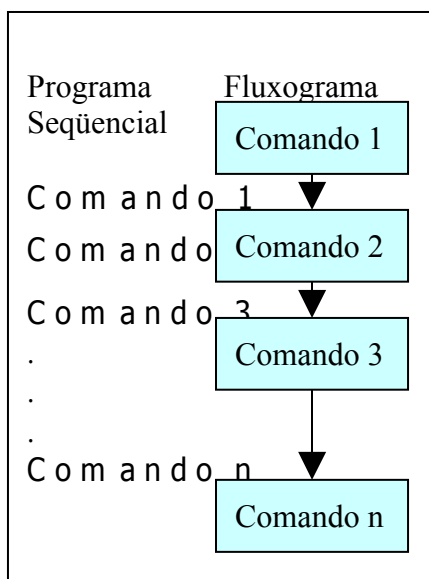
Para solucionar este problema, será necessário trabalhar uma nova instrução não vista até agora: se... então e fimse. Essa instrução tem por finalidade representar a tomada de uma decisão. Sendo a condição Verdadeira, serão executadas todas as instruções entre se... então e fimse. No caso da condição ser falsa, o programa segue o fluxo normal após o comando fimse, não executando as linhas de programa que ficam entre se... então e fimse.

```
se (<condição>) entao
    <comandos para condição verdadeira somente>
fimse
...
```

comandos

Fluxograma ou diagrama de blocos

O Fluxograma mostra, através de desenhos e setas, a ordem em que os comandos do programa devem ser realizados pelo computador. Compare o funcionamento de um algoritmo sequencial e um algoritmo de seleção, através dos fluxogramas de cada um deles:



Operadores relacionais

Nos testes condicionais dos algoritmos, são utilizados os seguintes operadores: >, <, =, >=, <= e <>.

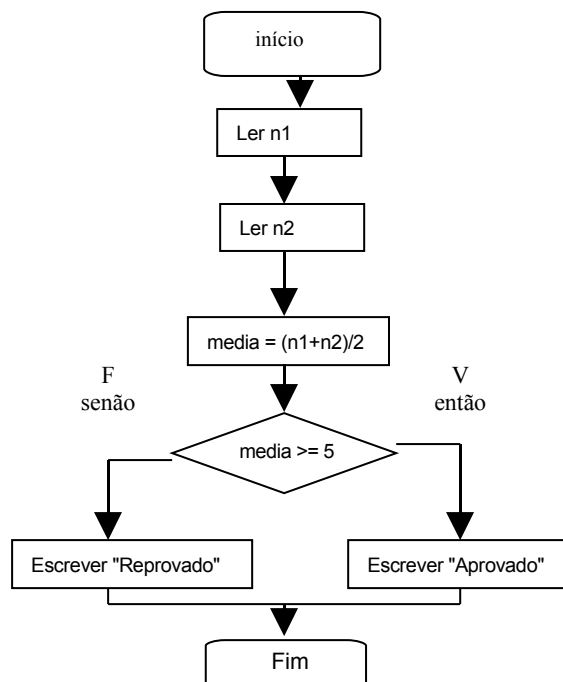
Desvio condicional composto

Exemplo: considerando o caso anterior onde um algoritmo deve calcular a média de um aluno e mostrar uma mensagem dizendo se esse aluno está aprovado (no caso da nota ter sido igual ou superior a Cinco), uma cláusula *senão* deve ser incluída no algoritmo no caso do aluno ser reprovado (nota inferior a cinco).

comandos

```
se (<condição>) entao
    <comandos para condição verdadeira>
senao
    <comandos para condição falsa>
fimse
```

comandos



Exercício:

Ler dois valores (A e B)

Efetuar a soma dos valores A e B e implicar o valor da soma em X

Verificar se X é maior ou igual a 10, caso sim mostre X+5, senão mostre X-7

Desvio condicional encadeado (ou aninhado)

Existem casos em que é necessário se estabelecerem verificações de condições sucessivas. Quando uma ação é executada, ela poderá ainda estabelecer novas condições, isso significa condições dentro de condições. Esse tipo de estrutura poderá ter diversos níveis de condição, sendo chamados de aninhamentos ou encadeamentos.

O segundo encadeamento pode ser tanto para uma condição verdadeira quanto uma condição falsa.

comandos

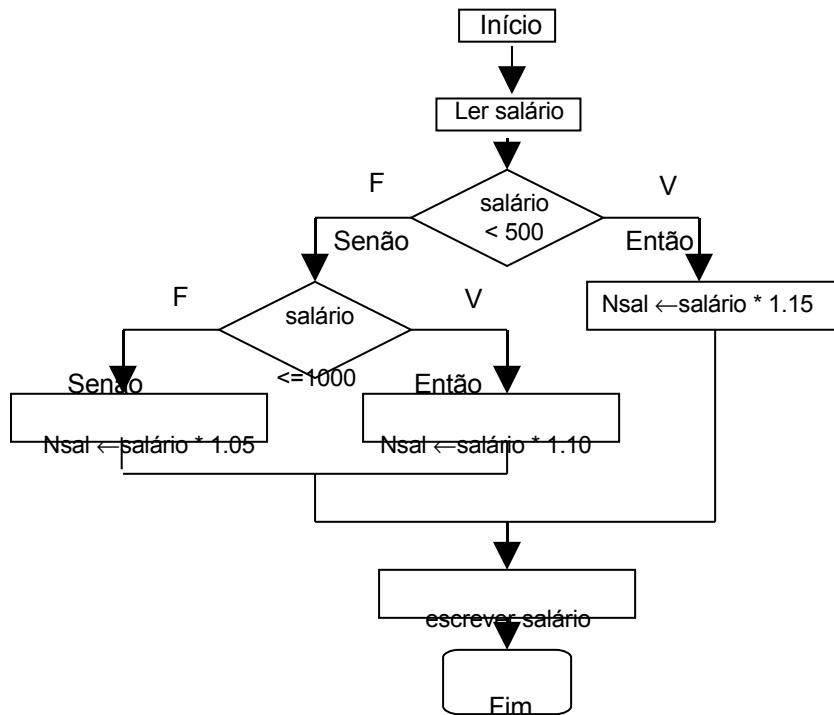
```
se (<condição1>) entao
    <comandos para condição1 verdadeira>
senao
    se (<condição2>) entao
        <comandos para condição2 verdadeira, porém condição1 Falsa>
    senao
        <condição1 e condição2 falsas>
    fimse
fimse
```

comandos

Considere o problema a seguir, onde se estabelecem 3 condições para se calcular o reajuste de salário de um funcionário:

- Para salário < 500, o reajuste será de 15%
- Para salário ≥ 500 mas ≤ 1000 , o reajuste será de 10%
- Para salário > 1000, o reajuste será de 5%

Fluxograma:



Faça o algoritmo:

Operadores Lógicos

Pode ser que se necessite, em um determinado momento, trabalhar com o relacionamento de duas ou mais condições, ao mesmo tempo na mesma instrução se, efetuando assim testes múltiplos. Nesses casos se torna necessário trabalhar com operadores lógicos ou booleanos. Os operadores lógicos são três: e, ou, não .

Operador Lógico: e

Também chamado de conjunção. Nesse caso, todas as expressões condicionais componentes de uma conjunção devem ser verdadeiras para que a expressão resultante tenha valor .verdadeiro. Abaixo é apresentada a tabela de decisão para o operador e:

Condição 1	Condição 2	Resultado
Verdadeira	Verdadeira	V
Verdadeira	Falsa	F
Falsa	Verdadeira	F
Falsa	Falsa	F

Operador Lógico: ou

Também chamado de disjunção. Nesse caso quando qualquer uma das expressões componentes da disjunção for verdadeira, a expressão resultante terá valor .verdadeiro. Abaixo é apresentada a tabela de decisão para o operador ou:

Condição 1	Condição 2	Resultado
Verdadeira	Verdadeira	V
Verdadeira	Falsa	V
Falsa	Verdadeira	V
Falsa	Falsa	F

Operador Lógico: não

Também chamado de negação. Nesse caso, a negação apenas inverte o valor verdade da expressão. Se x for falso, não x é verdade. Abaixo é apresentada a tabela de decisão para o operador não

Condição	Resultado
Verdadeira	F
Falsa	V

Exercício de Fixação

1) Para os valores (a=5) (b=7) e (c=9), assinale com X a sequência de execução dos algoritmos abaixo

a) Se (a=5) e (b=7) entao ()
 c := c+1 ()
senao ()
 c := c-1 ()
fim_se ()

b) Se (a=5) e (b<6) entao ()
 c := 1 ()
 senao ()
 Se (c=8) entao ()
 c := 2 ()
 senao ()
 c := 3 ()
 fimse ()
 fimse ()

2) Com base nas tabelas-verdade apresentadas anteriormente, assinale com V ou F, indicando quais valores são verdadeiros e quais valores são falsos. Utilize os valores: X=1 A=3 B=5 C=8 D=7 E=6

- a) nao (x>3) ()
- b) (x<1) e nao (b>d) ()
- c) nao (D<0) e (C> 5) ()
- d) nao (X>3) ou (C< 7) ()
- e) (A > B) ou (C> B) ()
- f) (X>=2) ()
- g) (X< 1) e (B>=D) ()
- h) (D >3) ou (C> 5) ()
- i) nao (D>3) ou nao (B<7) ()
- j) (A>B) ou nao (C>B) ()

3) Indique a saída dos trechos de programa em português estruturado, mostrado abaixo. Para as saídas, considere os seguintes valores: A=2, B=3, C=5, D=9 e E=6.

a)
se nao (D > 5) **entao**
 X := (A + B) * D
senão
 X := (A - B) / c
Fimse
escreva X

Resposta: _____

b)
se (A > 2) e (B < 7) **entao**
 X := (A + 2) * (B - 2)
senao
 X := (A + B) / D * (C + D)
fimse
escreva X

Resposta: _____

c)

se (A = 2) ou (B < 7) **entao**

 X := (A + 2) * (B - 2)

senao

 X := (A + B) / D * (C + D)

fim_se

escreva X

Resposta: _____

d)

se (A >= 2) ou (C <=1) **entao**

 X := (A + D)/2

senao

 X := D * C

fimse

escreva X

Resposta: _____

e)

se nao (A > 2) ou nao (B < 7) **entao**

 X:= A + E

senao

 X:= A / E

fimse

escreva X

Resposta: _____

f)

se nao (A > 3) e não (B < 5) **entao**

 X := A + D

senao

 X := D / B

fimse

escreva X

Resposta: _____

g)

se (C >=2) e (B <= 7) **entao**

 X := (A + D)/2

senao

 X := D * C

fimse

escreva X

Resposta: _____

h)

se (A > 2) ou nao (B < 7) **entao**

 X := A + B - 2

senao

 X := A - B

fimse

escreva X

Resposta: _____

Algoritmos Propostos

1. Faça um algoritmo que leia 4 variáveis A,B,C e D. A seguir, se B for maior do que C e se D for maior do que A e a soma de C com D for maior que a soma de A e B e se C e D, ambos, forem positivos e se a variável A for par escrever a mensagem “valores aceitos”, senão escrever “valores não aceitos”.
2. Desenvolva um algoritmo que leia 3 valores inteiros e efetue o cálculo das raízes da equação de báscara. Se não foi possível calcular as raízes, mostre uma mensagem correspondente.
Obs.: não é possível divisão por 0 (zero) e raiz quadrada de número negativo;
3. Escreva um algoritmo que leia um valor qualquer. A seguir, mostre uma mensagem dizendo em qual dos seguintes intervalos: h0,25i (25,50i, (50,75i, (75,100i o número está. Obs: por exemplo (25,... indica valores maiores que 25 Ex: 25,00001.
4. Com base na tabela abaixo, escreva um algoritmo que leia o código de um item e a quantidade deste item. A seguir, calcule e mostre o valor da conta a pagar.

CÓDIGO	ESPECIFICAÇÃO	PREÇO UNITÁRIO
1	Cachorro Quente	R\$ 4.00
2	X-Salada	R\$ 4.50
3	X-Bacon	R\$ 5.00
4	Torrada simples	R\$ 2.00
5	Refrigerante	R\$ 1.50

5. Ler 4 valores (N1, N2, N3 e N4) referentes às quatro notas de um aluno. A seguir, se a média ponderada com pesos 2,3,4 e 1 para as respectivas provas for maior ou igual a 7, mostre uma mensagem: "Aluno foi aprovado". Se a nota for inferior a 5, mostrar uma mensagem: "Aluno foi reprovado". Se a média ficou entre 5 e 7, fazer a leitura de uma variável (EX) correspondente a nota de exame, recalculando a média (somando a nota do exame com a nota anterior e dividindo por 2) e mostrar se o aluno foi ou não aprovado.
6. Considerando: $a = 2$, $b = 5$ e $C = 6$, mostre o resultado da execução dos algoritmos abaixo.

SOMA DE A, B E C=

```

a := b * c - a * b
b := b ^ 2 - (a + 6)
se (b > a) e não (b <= C) entao
    b := c * 2 - (a + 2)
    c := b - a * c * 2
senão
    c := c + 2 * a + 2
    b := c * 2 - a * 2
fimse
x := a + b + c
escreva ( "SOMA DE A, B E C=", x)

```

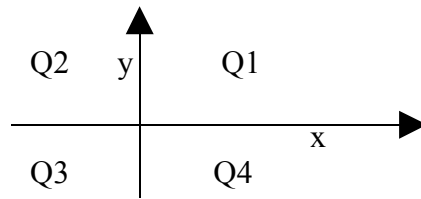
SOMA DE A, B E C=

```

a := b * c + a * b
b := b ^ 2 + (a + 6)
se (b > a) e não (b <= C) entao
    b := c * 2 - (a + 2)
    c := b - a * c * 2
senao
    c := c + 2 * a + 2
    b := c * 2 - a * 2
fimse
x := a + b + c
escreva ( "SOMA DE A, B E C=", x)

```

7. Faça um algoritmo que leia 2 valores (x e y), que devem representar as coordenadas de um ponto em um plano. A seguir, determine qual o quadrante ao qual pertence o ponto, ou se está sobre um dos eixos cartesianos ou na origem ($x=y=0$).



8. Faça um algoritmo que leia 3 valores e ordene-os em ordem crescente. No final, mostre os valores em ordem crescente e a seguir, como foram lidos.
9. Faça um algoritmo que leia 3 valores e mostre estes valores ordenados por ordem decrescente.
10. Faça um algoritmo que leia 3 valores (A, B e C) e verifique se eles formam ou não um triângulo. Em caso positivo, calcule o perímetro do triângulo e em caso negativo, calcule a área do trapézio que tem A e B como base e C como altura. Para qualquer um dos casos, deve ser apresentada uma mensagem correspondente.
11. Faça um algoritmo que lê 2 valores (A e B). Após a mensagem, o algoritmo deve mostrar uma mensagem “São Múltiplos” ou “Não são Múltiplos”, correspondente aos valores lidos.
12. Escrever um algoritmo que lê 3 comprimentos de lado (A, B e C) e os ordena em ordem decrescente, de modo que o lado A representa o maior dos 3 lados. A seguir, determine o tipo de triângulo que estes três lados formam, com base nos seguintes casos, sempre escrevendo uma mensagem adequada:
- se $A > B + C$, não formam triângulo algum
 - se $A^2 = B^2 + C^2$, então formam triângulo retângulo
 - se $A^2 > B^2 + C^2$, então formam triângulo obtusângulo
 - se $A^2 < B^2 + C^2$, então formam triângulo acutângulo
 - se os três lados forem iguais, então formam triângulo equilátero
 - se dois dos lados forem iguais, então forma um triângulo isósceles
13. Leia a hora inicial e a hora final de um jogo. A seguir calcule a duração do jogo, sabendo que o mesmo pode começar em um dia e terminar em outro, tendo uma duração máxima de 24 horas. Utilize agora o teste condicional se... então...
14. Leia a hora inicial, minuto inicial, hora final e minuto final de um jogo. A seguir calcule a duração do jogo, mostrando a seguinte mensagem:
- “O jogo durou xxx horas e yyy minutos”.
15. Qual é a ordem de execução dos operadores de cada uma das expressões abaixo:
- a) se $(a > b + c * d)$ e $(a + b * c ^ d - 2 > (x + y))$
 - b) se $(a + b - c * d) > (a - b)$ ou $(c > 2 * 3 * a)$
 - c) se não $((a + b) < (c * d))$ ou $(a > c - d)$ e $(2 > 3 * a)$
 - d) se não $(a > b + c * d)$ ou $(b * x > y)$ ou $(a > b)$ e $(a > c)$ e $(a > d)$
 - e) se $(a + b > c)$ ou $(c + d > a)$ ou $(a + b > 10)$ e $(c > d)$ e não $(a > c)$

Algoritmos com Repetição - Laços de Repetição

Em vários momentos, na programação, se torna necessário repetir um trecho de um programa um determinado número de vezes. Nesse caso, pode ser criado um laço de repetição que efetue o processamento de um determinado trecho, tantas vezes quantas forem necessárias. Os laços de repetição também são conhecidos por loopings.

Supondo que se queira executar um determinado trecho de um programa três vezes (3x). Por exemplo, suponha que um programa deva ler dois valores e mostrar a média entre esses valores. Certamente a melhor técnica não seria repetir simplesmente um mesmo trecho três vezes, como é mostrado abaixo:

```
variaveis
  media,n1,n2 :numerico
inicio
  Leia (n1)
  Leia (n2)
  media := (n1+n2)/2
  escreva ("Média:", media)

  Leia (n1)
  Leia (n2)
  media := (n1+n2)/2
  escrever ("Média:", media)

  Leia (n1)
  Leia (n2)
  media := (n1+n2)/2
  escreva ( "Média:", media )
fim
```

Para esse e outros casos similares, existem comandos apropriados para efetuar a repetição de determinados trechos de programa tantas vezes quantas forem necessárias. A principal vantagem é que o programa passa a ter um tamanho muito menor, podendo ser ampliada a amplitude de processamento como for necessária. Existem 2 tipos de repetições:

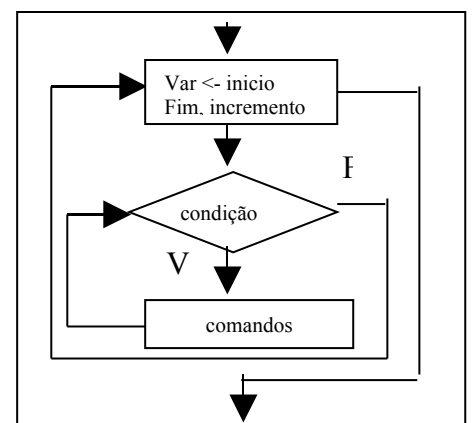
- definidas e garantidas, ou seja, um mesmo trecho de programa será executado um número pré-determinado de vezes e;
- indefinida e sem garantia, ou seja, um trecho de programa poderá ser executado indefinidamente, não havendo garantia do término da execução do algoritmo.

Repetição utilizando estrutura Para / Próximo

A estrutura do Para/fimpara é garantida, e sua sintaxe em português estruturado é apresentada no seguinte formato:

```
para <var> de <início> ate <fim> passo <incremento>
  <comandos>
fimpara
```

Veja o exemplo anterior, aplicando-se nele o laço de repetição Para / fimpara.



```

variaveis
    media,n1,n2,i :numerico
inicio
    para i de 1 ate 3 passo 1 faca// (nesse caso passo 1 é opcional)
        Leia (n1)
        Leia (n2)
        media := (n1+n2)/2
        escreva ("Média:", media)
    fimpara
fim

```

O algoritmo, que tinha que 17 linhas, passou a ter 10 linhas, com a mesma funcionalidade e ainda com a vantagem de se ampliar o número de repetições de 3 para 300, com a maior facilidade.

Repetição utilizando estrutura Enquanto / Fimenquanto

Embora também possa ser utilizada quando se tem um número pré-determinado de repetições a executar, como no exemplo acima, essa estrutura é mais indicada quando é necessário repetir um determinado trecho de programa indefinidamente. Para ilustrar, suponha que um determinado valor deva ser lido indefinidamente, até que seja digitado zero (condição de parada). Nesse caso devemos usar a estrutura Enquanto :

```

variaveis
    val :numerico
inicio
    val:= 1
    enquanto val <> 0 faca
        escreva ("Valor:")
        leia (val)

        <comandos1>
    fimenquanto
    <comandos2>
fim

```

Nesse caso, quando for digitado 0 (zero) para val, o fluxo do programa segue até chegar no comando fimenquanto. Ao retornar na linha faça enquanto, é verificado que o valor não é diferente de zero e o controle do programa passa para a linha comandos₂. A partir daí o fluxo do programa segue normalmente. Veja abaixo como ficaria a utilização do laço enquanto para o exercício das 4 repetições:

```

variaveis
    media,n1,n2,i :numerico
inicio
    i:= 1
    faça enquanto i <= 3
        Leia (n1)
        Leia (n2)
        media := (n1+n2)/2
        escrever ("Média:", media)
        i:= i+1
    fimenquanto
fim

```

Usando o comando repita ... até <condição>, neste caso o comando verifica a condição desejada no fim do mesmo. Veja o exemplo:

```

variaveis

```

```

    media,n1,n2,i :numerico
inicio
    i:= 1
    repita
        Leia (n1)
        Leia (n2)
        media := (n1+n2)/2
        escreva ("Média:", media)
        i:= i+1
    ate i > 3
fim

```

Embora existam outros tipos de laços de repetição, além do para e repita, somente trabalharemos com estes nos algoritmos, pois são encontrados em praticamente todas as linguagens de programação e resolvem qualquer problema que necessite repetição.

Contador

É uma variável de controle, inteira, que serve para controlar quantas vezes um determinado trecho de programa foi executado. Considere, por exemplo, um programa que leia 10 valores, podendo eles serem somente negativos ou positivos (desconsidere os valores nulos). A seguir, considere que o programa deva mostrar a quantidade de valores positivos digitados. Nesse caso, devemos fazer um teste a cada leitura, e, no caso do valor lido ser positivo, adicionar +1 para uma variável tipo contador (contp=contp+1).

Exercício.: faça o algoritmo acima (ler 100 valores), mostrando no final a quantidade de números negativos e positivos digitados.

Acumulador

É uma variável de controle, inteira, que serve para acumular valores. Considere que um programa, além de ler 100 valores e mostrar a quantidade de números positivos, deva mostrar a média dos valores positivos digitados. Resolva este problema.

Algoritmos Propostos

- 1) O que é e para que serve uma variável tipo **CONTADOR** e uma variável tipo **ACUMULADOR**.
- 2) Leia 5 valores para uma variável A. A seguir mostre quantos valores digitados foram pares, quantos valores digitados foram ímpares, quantos foram positivos e quantos foram negativos.
- 3) Mostre os números pares entre 1 e 100, inclusive.
- 4) Calcule e mostre a soma dos números pares entre 1 e 100, inclusive.
- 5) Calcule e mostre a média dos números pares entre 1 e 100, inclusive.
- 6) Calcule e mostre a soma dos números ímpares entre 1 e 100, inclusive
- 7) Leia 2 valores: X e Y. A seguir, calcule e mostre a soma dos números ímpares entre eles.
- 8) Leia 10 valores. Para cada valor lido, mostre uma mensagem dizendo se ele é par ou ímpar e se ele é positivo ou negativo.
- 9) Faça um algoritmo que leia um conjunto não determinado de pares de valores M e N (parar quando algum dos valores for menor ou igual a zero). Para cada par lido, mostre:

- a) a sequência do menor até o maior, inclusive;
- b) a soma dos inteiros consecutivos do menor até o maior, inclusive.

- 10) Apresente o quadrado de cada um dos números pares entre 1 e 1000, inclusive.
- 11) Apresente todos os números divisíveis por 5 que sejam maiores do que 0 e menores ou iguais a 200.
- 12) Leia 1 variável X inúmeras vezes (parar quando o valor digitado for ≤ 0). Para cada valor lido mostre a sequência de 1 até o valor lido.
- 13) Faça um algoritmo que leia 1 valor para uma variável N. A seguir, mostre a tabuada de N:
 $1 \times N = N \quad 2 \times N = 2N \quad \dots \quad 10 \times N = 10N$
- 14) Escreva um algoritmo que leia 10 valores quaisquer. A seguir, mostre quantos deles estão dentro do intervalo (10,20) e quantos estão fora do intervalo, mostrando essas informações.
- 15) Faça um algoritmo que leia um conjunto de 15 valores, um de cada vez, acompanhados de um código 1 ou 2. O valor representa o número de cobaias utilizadas em uma das 15 experiências feitas e os códigos 1 e 2 representam respectivamente coelhos e ratos. Mostre no final, o total de cobaias utilizadas, o total de coelhos, total de ratos, e o percentual de coelhos e ratos.
- 16) Escreva um algoritmo que leia 50 valores, um de cada vez. Mostre então o maior valor lido.
- 17) Faça um algoritmo que leia 5 conjuntos de 2 valores, o primeiro representando o número de um aluno e o segundo representando a sua altura em centímetros. Encontrar o aluno mais alto e mais baixo e mostrar seus números e suas alturas, dizendo se o aluno é o mais alto ou é o mais baixo.
- 18) Escreva um algoritmo que leia um número e diga se este número é perfeito ou não. Um número perfeito é aquele que é igual à soma de seus divisores. Ex: $1+2+3=6$, $1+2+4+7=28$, etc...
- 19) Escrever um algoritmo que lê um valor para X e calcula e mostra os 20 primeiros termos da série:
1º termo: $1/x$ 2º termo: $1/x + 1/x^2$ 3º termo: $1/x + 1/x^2 + 1/x^3$
- 20) Mostre os números entre 1000 e 2000 que quando divididos por 11 dão resto igual a 5.
- 21) Supondo que a população de um país A seja 90 milhões de habitantes, crescendo com uma taxa anual de 3.5 % e que a população de um país B seja 140 milhões de habitantes, crescendo a uma taxa anual de 1 %. Faça um algoritmo que calcule e mostre quantos anos serão necessários para que a população do país A ultrapasse a população do país B.

Algoritmos com Fatoriais (considere a entrada de valores positivos)

- 20) Faça um algoritmo que leia um valor para M. Se este valor for positivo, calcule e mostre o seu fatorial.
- 21) Faça um algoritmo que leia um conjunto não determinado de valores para M (parar quando o valor digitado for $= 0$). Para cada valor lido, se este valor for par, mostre quantos divisores ele possui. Se for ímpar, mostre o fatorial de M.

Algoritmos com números primos (considere a entrada de valores positivos)

- 22) Escrever um algoritmo que leia um valor qualquer. A seguir, mostre se o valor lido é primo ou não é primo.

- 23) Leia 100 valores. A seguir, para cada valor, mostre se o valor é primo ou não. Em caso negativo, mostre quantos divisores o número possui.
- 24) Leia uma quantidade de valores para X indeterminadamente (parar quando o valor digitado para x for igual a zero). Para cada valor lido, mostrar se este valor é primo ou não.
- 25) Faça um algoritmo que gere e mostre os valores primos entre 1 e 1000. Mostre também a soma destes valores.
- 26) Escreva um algoritmo que calcula e mostra a soma dos números primos entre 92 e 1478. 27)

Mostre nas telas ao lado a execução dos seguintes algoritmo (os cálculos devem aparecer):

a) `x := 0`
`para i de 1 até 3 passo 2 faca`
`para j de 6 até i passo-3 faca`
`s := s - 1`
`x := x + i * j * s`
`fimpara`
`fimpara`
`escreva ("Soma = ", x)`

Cálculos (m e m ó r i a)

Soma =

b) `C := 0`
`Para i de 1 até 3 faca`
`para j de 6 até i passo-3 faca`
`C := C+i`
`escreva (C)`
`fimpara`
`fimpara`
`escreva ("C=", C)`

Cálculos

C=

c) `A := 12`
`B := 27`
`D := 0`
`enquanto A <= B`
`Para i de 1 até int(A/6) faca`
`D := D+A`
`Próximo`
`A := A+10`
`fimenquanto`
`escreva ("D=", D)`

Cálculos

D=

Considerando a entrada dos valores 4, 5, 2, 3, 0, -1, 5 para X, mostre na tela a execução dos algoritmos:

d) `s := 0`
`x := 1`
`enquanto x <> 0`
`leia x`
`s := s + x`
`fimenquanto`
`escrever ("Soma = ", s)`

`escrever "Soma = ", s`
`ate x=0`

e) `s:=0`
`x:=1`
`repita`
`leia x`
`Se x = 0 então`
`exit`
`fimse`
`s := s + x fim_enquanto`

Cálculos

Soma =

Cálculos

Soma =

Estruturas de dados

Temos os seguintes tipos de dados básicos: numéricos (inteiros e reais), lógicos e caracteres. Baseando-se nestes tipos básicos podemos construir tipos compostos. Os tipos compostos dividem-se em dois: homogêneos (vetores e matrizes) e heterogêneos (registros).

Tipos de dados Homogêneos - VETORES e MATRIZES

Os tipos homogêneos são conjuntos do mesmo tipo básico. A utilização desse tipo de estrutura de dados recebe diversos nomes, tais como: variáveis indexadas, compostas, arranjos, tabelas em memória, arrays (do inglês) vetores e matrizes. Para simplificar utilizaremos somente os nomes vetores e matrizes. Os vetores são uma lista de elementos do mesmo tipo. Quando os vetores possuem mais de uma dimensão se tornam matrizes.

Todas as variáveis e constantes ocupam um espaço em memória. Este espaço ocupado é um espaço linear, ou seja, se temos a seguinte declaração :

```
variaveis
```

```
  a,b : numerico
```

provavelmente (dependendo do compilador da linguagem utilizada) estas variáveis ocuparão espaços subsequentes em memória. Baseado na necessidade de utilização de tipos como vetores e utilizando esta linearidade do armazenamento em memória fica fácil para as linguagens de programação implementarem estes tipos.

Vetores:

Quando possuímos uma ordem e um índice de acesso aos elementos de um conjunto então temos caracterizado um vetor. Para definirmos um vetor a sintaxe utilizada será a seguinte :

matriz identificadorhíndicei exemplo: Matriz numerico X[3]

Para se ter idéia de como utilizar vetores em uma determinada situação, considere o seguinte problema: Calcular a média geral da turma segundo a tabela abaixo:

Aluno	Nota 1	Nota 2	Média
1	4.0	6.0	5.0
2	6.0	7.0	6.5
3	9.0	8.0	7,5
4	3.0	9.0	6.0
5	4.0	6.0	5.0
6	7.0	7.0	7.0
7	8.0	7.0	7.5

Agora basta escrever um programa para efetuar as 7 médias, onde serão utilizadas as médias finais de cada aluno, sendo que a média do primeiro aluno está na variável MD1, do segundo em MD2, do terceiro em MD3, assim por diante. Então tem-se:

MD1 := 5

MD2 := 6.5

MD7 := 7.5

Com o conhecimento adquirido até o presente momento, seria então elaborado um programa que efetuar a leitura de cada nota, a soma das mesmas e uma divisão do valor da soma por 7.

```
variaveis
    MD1, MD2, MD3, MD4, MD5, MD6, MD7, soma, med :numerico
inicio
    Soma := 0
    Leia MD1, MD2, MD3, MD4, MD5, MD6, MD7
    soma := MD1+ MD2 + MD3 + MD4 + MD5 + MD6 + MD7
    med := soma /7
    escrever "Média = " , med
fim
```

Como seria o cálculo utilizando um Vetor MD, com 7 elementos?

Nesse caso, teremos então uma única variável indexada (vetor) contendo todos os valores das 8 notas. Isso é apresentado da seguinte forma:

```
MD[ 1 ] := 5
MD[ 2 ] := 6.5
MD[ 3 ] := 7.5
MD[ 4 ] := 6.0
MD[ 5 ] := 5.0
MD[ 6 ] := 7.0
MD[ 7 ] := 7.5
```

O vetor pode ser declarado da seguinte forma:

```
variaveis
    MD[7]: numerico
```

Faça o exercício para calcular a média da turma utilizando o vetor.

Algoritmos Propostos

- 1) Conceitue Vetor.
- 2) Como se faz a leitura de um vetor A que tenha 10 elementos (A[10])
- 3) Faça um algoritmo que leia um vetor V[6]. Conte a seguir, quantos valores de V são negativos e mostre essa informação.
- 4) Faça um algoritmo que leia um vetor A[10]. No final, mostre todas as posições do vetor que armazenam um valor menor ou igual a 10 e o valor armazenado na posição.
- 5) Faça um algoritmo que leia um vetor X[10]. Substitua, a seguir, todos os valores nulos do vetor X por 1 e escreva novamente o vetor X.
- 6) Faça um algoritmo que leia um vetor C[50]. Encontre a seguir o maior elemento de C e mostre-o.
- 7) Faça um algoritmo que leia um vetor N[20]. A seguir, encontre o menor elemento do vetor N e a sua posição dentro do vetor, mostrando: "O menor elemento de N é", M, "e sua posição dentro do vetor é:",P.

- 8) Faça um algoritmo que leia um vetor $N[20]$. Troque a seguir, o 1º elemento com o último, o 2º elemento com o penúltimo, etc., até trocar o 10º com o 11º. Mostre o vetor modificado.
- 9) Faça um algoritmo que leia um vetor $K[30]$. Troque a seguir, todos os elementos de ordem ímpar do vetor com os elementos de ordem par imediatamente posteriores. Mostre o vetor modificado.
- 10) Faça um algoritmo que leia um vetor $D[60]$. A seguir, troque o 1º elemento com o 31º, o 2º com o 32º, etc. Mostre no final o vetor modificado.
- 11) Faça um algoritmo que leia um vetor $S[20]$ e uma variável A . A seguir, mostre o produto da variável escalar A pelo vetor.
- 12) Faça um algoritmo que leia um vetor $F[20]$ e uma variável A . A seguir, crie um vetor $Gh20i$ que é o produto da variável A pelo vetor F . Mostre o vetor G no final.
- 13) Faça um algoritmo que leia dois vetores: $F[20]$ e $G[20]$. Calcule e mostre, a seguir, o produto dos valores de F por G .
- 14) Faça um algoritmo que leia um vetor $K[10]$ e um vetor $N[10]$. A seguir, crie um vetor M que seja a diferença entre o vetor K e N ($M=K-N$). Mostre a seguir o vetor M .
- 15) Faça um algoritmo que leia um vetor $F[20]$. A seguir ordene este vetor F em ordem crescente e mostre então o vetor ordenado.
- 16) Faça um algoritmo que leia um vetor $G[5]$ e a seguir leia 15 números de alunos e de vetores $R[5]$, que seriam as respostas dos alunos para as questões, da letra A até a letra E, sendo que deve ser lido um número de aluno e um vetor de respostas por vez. Para cada aluno, mostre o número de acertos e a sua nota, sendo que todas as questões tem o mesmo peso.
- 17) Faça um algoritmo que leia um vetor $G[5]$ e a seguir leia um número indefinido de alunos e de vetores $R[5]$, que seriam as respostas dos alunos para as questões, sendo que deve ser lido um número de aluno e um vetor de respostas por vez. Para cada aluno, mostre o número de acertos e a sua nota, sendo que todas as questões tem o mesmo peso. Mostre, ainda, a mensagem “Aprovado”, se o aluno ficou com nota acima ou igual a 5 e “Reprovado” se o aluno ficou com a nota menor do que 5.
- 18) Faça um algoritmo que leia um vetor $G[5]$ e a seguir leia um número indefinido de alunos e de vetores $R[5]$, que seriam as respostas dos alunos para as questões, sendo que deve ser lido um número de aluno e um vetor de respostas por vez. Para cada aluno, mostre o número de acertos e a sua nota, sendo que todas as questões tem o mesmo peso. Mostre, ainda, a mensagem “Aprovado”, se o aluno ficou com nota acima ou igual a 7, “Reprovado” se o aluno ficou com a nota menor do que 5 e “Em exame”, se o aluno ficou com a média entre 5 e 7.
- 19) Faça um algoritmo que leia 2 vetores $A[10]$ e $B[10]$. A seguir, Crie um vetor C que seja a intersecção de A com B e mostre este vetor C . Obs.: Intersecção é quando um valor estiver nos dois vetores. Considere que não há elementos duplicados em cada um dos vetores.
- 20) Faça um algoritmo que leia um vetor $G[13]$ que é o gabarito de um teste da loteria esportiva, contendo os valores 1 quando for coluna 1, 0 quando for coluna do meio e 2 quando for coluna 2. Ler a seguir, para 10 apostadores, o número do cartão de cada apostador e um vetor $R[13]$ que seriam as respostas dos apostadores. Para cada apostador, mostre o número de acertos.
- 21) Com relação ao exercício anterior, calcule e mostre o percentual dos apostadores que fizeram de 10 a

13 pontos e o percentual dos apostadores que fizeram menos do que 10 pontos.

- 22) Faça um algoritmo que leia um vetor S[20]. A seguir, compacte este vetor S, retirando todos os valores nulos ou negativos e mostre então o vetor compactado.
- 23) Faça um algoritmo que leia um vetor T[15]. Crie, a seguir, um vetor Fat[15] que contém os fatoriais do vetor T. Mostre a seguir o vetor Fat.
- 24) Faça um algoritmo que leia um vetor U[15]. Crie, a seguir, um vetor Primo[15] que conterà todos os valores primos do vetor U. Mostre o vetor Primo[15] no final.
- 25) Faça um algoritmo que leia um vetor V[10] e o escreva. Crie, a seguir, um vetor COMP que conterà somente os valores não repetidos de V. Mostre então o vetor COMP.
- 26) Faça um algoritmo que leia um vetor V[10] e o escreva. Mostre a seguir, todos os valores distintos que aparecem no vetor.
- 27) Faça um algoritmo que leia um conjunto de 30 valores. Para cada valor lido, coloque em um vetor P ou I, conforme os valores forem pares ou ímpares. O tamanho dos vetores P e I é de 10 posições. Cada vez que encher um dos vetores, (P ou I) esvazie-o, mostrando os valores que estavam no vetor. Cada vetor P ou I pode ser preenchido quantas vezes forem necessárias. No final, mostre os valores que restaram em cada um dos vetores.
- 28) Faça um algoritmo que leia um vetor V[10] e um vetor X[10]. A seguir, crie um vetor Y[20] que conterà os valores dos vetores V e X em ordem crescente. Obs.: o método para fazer é livre.
- 29) Faça um algoritmo que leia 5 valores para um vetor V[10] e ordene estas primeiras 5 posições do vetor em ordem crescente. Leia, a seguir, 5 valores para A, um de cada vez, e para cada valor lido, insira-o dentro do vetor de forma que o mesmo continue ordenado em ordem crescente.
- 30) Faça um algoritmo que gere os 10 primeiros números primos acima de 100 e armazena-os em um vetor X[10]. Mostre então o vetor X.

Obs: A resolução dos exercícios 25 a 29 se encontram no Anexo I, no final da apostila

Matrizes:

A matriz mais comum é a de duas dimensões (linha e coluna), por se relacionar diretamente com a utilização de tabelas. Trabalharemos somente com matrizes de 2 dimensões, por serem mais comuns, mas podem ser necessárias, em algum momento, matrizes de 3 ou mais dimensões.

Uma matriz de 2 dimensões estará sempre fazendo menção a linhas e colunas e será representada por seu nome e tamanho. Dessa forma, uma matriz Tabelah8,5i indica que tem 8 linhas e 5 colunas.

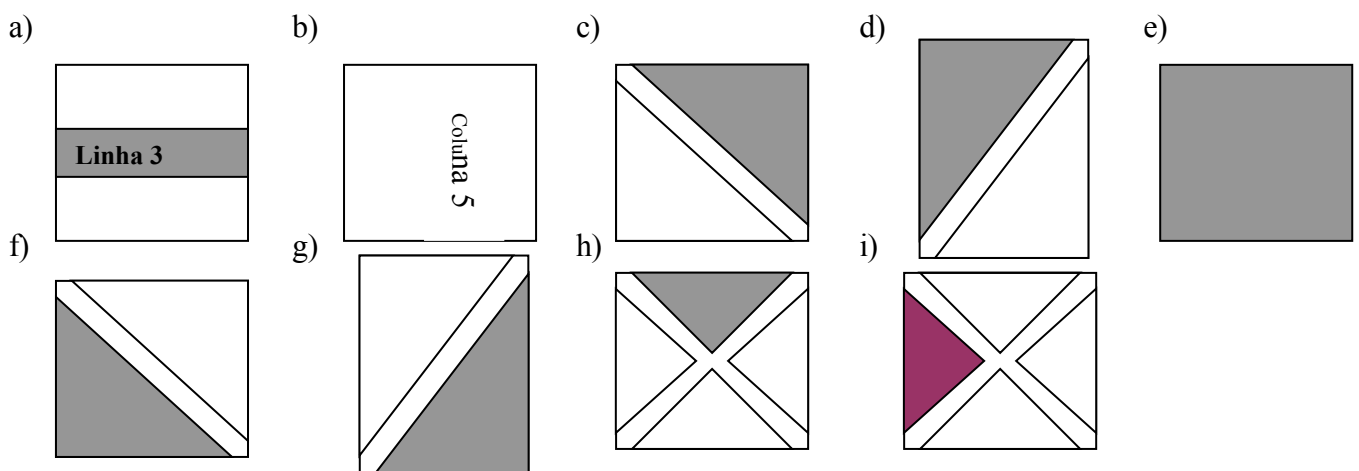
Matriz Tabela		Colunas				
		↓				
		1	2	3	4	5
Linhas →	1					
	2					
	3					
	4					
	5					

A matriz pode ser declarado da seguinte forma:

```
variaveis  
  M[10,10]: numerico
```

Exercícios

- 1) Faça um algoritmo que leia uma matriz $M[6,6]$ e uma matriz $N[6,6]$. A seguir, calcule o produto de M por N , colocando os resultados em uma matriz $PROD[6,6]$.
- 2) Faça um algoritmo que leia uma matriz $S[6,6]$ e um valor A . A seguir, multiplique a matriz pelo valor A , colocando o resultado em um vetor $V[36]$. Mostre o vetor $V[36]$.
- 3) Leia uma matriz $M[7,7]$. A seguir, calcule e mostre a soma dos elementos de cada área hachuriada:



- 4) Leia uma matriz $M[5,5]$ e crie 2 vetores $Sl[5]$ e $Sc[5]$ que contenham respectivamente as somas das linhas e das colunas de M .
- 5) Leia uma matriz $M[6,5]$. Após, divida os 5 elementos de cada linha da matriz pelo maior elemento de cada uma das 6 linhas. Coloque o resultado em uma matriz $S[6,5]$.
- 6) Escreva um algoritmo que lê uma matriz $M[5,5]$. Substitua, a seguir, todos os valores negativos da matriz pelo seu módulo. Exemplo: substitua -2 por 2, -16 por 16, assim por diante.
- 7) Escreva um algoritmo que lê uma matriz $M[6,6]$. A seguir, troque os elementos da primeira coluna com os elementos da segunda coluna, os da terceira coluna com a quarta coluna e os elementos da quinta coluna com os elementos da sexta coluna.
- 8) Repita o exercício 4, trocando os elementos das linhas, ao invés das colunas.
- 9) Leia uma matriz $M[5,5]$. A seguir, ordene os elementos da matriz M e mostre como ficou a Matriz ordenada, linha por linha.
- 10) Execute o algoritmo abaixo, preenchendo a matriz $M[2,6]$:

```
variaveis
```

```

M[2,6] :numerico
fat,i :numerico
inicio
para i de 1 até 6 faca
    leia M[1,i]
    se M[1,i] > 0 então
        fat := 1
        para j de 1 até M[1,i]
            fat := fat * j
        fimpara
        M[2,i] := fat
    senao
        M[2,i] := i * i
    fimse
    M[2,i] := M[2,i] - 10
fimpara
fim

```

Na leitura da Matriz, considere os valores: -5 7 2 1 4 0

M [2,6] =

	1	2	3	4	5	6
1						
2						

Declaração de Variáveis

Variável é uma identificação dada a uma posição de memória. Utilizada para armazenar dados. Todas as variáveis definidas pelo usuário devem ser declaradas separadamente em um bloco especial denominado variaveis. Podem ser definidas duas classes distintas de variáveis, simples e compostas.

Variáveis Simples

Sintaxe:

variaveis

```

<expr1>, <expr2>, ... , <exprN> : numerico
<expr1>, <expr2>, ... , <exprN> : logico
<expr1>, <expr2>, ... , <exprN> : caracter

```

Variáveis Compostas

Sintaxe:

variaveis

```

vetor[10], x, matriz[5,5] : numerico
<expr1>[ <expr2>, <expr3>], ... : numerico
<expr1>[ <expr2>, <expr3>], ... : caracter
<expr1>[ <expr2>, <expr3>], ... : logico

```

Exemplo:

```

// Algoritmo para criar e imprimir um vetor de 100
posições. //

```

variaveis

```

i : numerico
vet[100]: numerico

```

inicio

limpar

```

para i de 1 ate 3 faca
    posicionar (10,10)

```



```

        escreva ("Entre com um numero = ")
        posicionar (10,40)
        leia vet[i]
    fimpara
para i de 1 ate 3 faca
    escrever vet[i]
fimpara
fim

```

Falso

Constante lógica de valor FALSO.

Exemplo:

```

// Algoritmo para mostrar o valor de uma constante lógica
falsa. //-----
---variaveis
    f :logico
inicio
    f := falso
    escreva ( "constante f = ", f)
fim

```

Verdadeiro

Constante lógica de valor VERDADEIRO.

Exemplo:

```

// Algoritmo que demonstra uma constante lógica
verdadeira. //-----
-----variaveis
    v :logico
inicio
    v := verdadeiro
    escreva ("constante v = ", v)
fim

```

Operadores

Segue abaixo a lista de operadores da linguagem. A prioridade é a mesma utilizada nos algoritmos:

Potenciação: ^ Multiplicação: * Divisão: /

Soma: + Subtração: -

Negação: NAO Multiplicação lógica: E Adição lógica: OU

Maior: > Menor: < Igual: = Diferente: <> Maior igual: >= Menor igual: <=

Concatenação de Caracteres: +

Exemplo:

```

// Concatenar dois campos tipo caracter em um
terceiro. //
-----
variaveis
pre_nome, sobrenome, nome : caracter
inicio
    limpar
    leia (pre_nome)
    leia (sobrenome)
    nome := pre_nome + " " + sobrenome
    escreva ("aluno = ", nome)
fim

```

Comandos

Escreva

Exibe no vídeo um texto, o conteúdo de uma variável ou o valor de uma expressão.

ESCREVA (<variavel>,<"texto">,<expressao arit.>)

Leia

Lê uma entrada do teclado e armazena o dado lido em uma variável.

LEIA (<variavel>)

Exemplo:

```
// Algoritmo para leia uma variável nome, número e imprimir: nome e número.
variaveis
    nome :caracter
    numero :numerico
inicio
    leia (nome)
    leia (numero)
    escreva ("Nome = ",nome," Numero = ",numero)
fim
```

Se

Estrutura que decide entre duas seqüências de instruções qual será executada, ou apenas se uma dada seqüência de instruções será ou não executada.

```
SE <expressao> ENTAO
    <bloco-de-instrucoes1...>
SENAO
    <bloco-de-instrucoes2...>
FIMSE
```

Escolha ... Caso

Estrutura que decide entre vários blocos de instruções qual aquele que deverá ser executado ou se nenhum deles deverá ser executado.

```
ESCOLHA <variável>
    CASO <expressao>
        <bloco-de-instrucoes1...>
    OUTROCASO
        <bloco-de-instrucoes2...>
FIMESCOLHA
```

Para

Estrutura que repete a execução de um bloco de instruções um determinado número de vezes.

```
PARA <variavel> DE <expressao1> ATE <expressao2> FACA
    <bloco-de-instrucoes>
fimpara
```

REPITA

Repete um bloco de instruções enquanto a avaliação de uma expressão fornecida resultar verdadeiro ao fim da primeira execução.

```
REPITA
    <bloco-de-intrucoes>
ATE <expressao>
```

OBSERVAÇÕES

1. Usar comentários nos programas de forma a esclarecer a lógica
2. Usar nome elucidativos nos identificadores de variáveis, constantes, etc.
3. Manter sempre a mesma Identação, de no mínimo 3 caracteres
4. Usar espaços e parênteses em expressões de forma a facilitar a leitura das mesmas. Ex.: `X1 := sqr(b) - 4 * A * C` , e não `X1:=sqr(b)-4*a*c`.
5. Pular algumas linhas em branco, entre trechos do programa de forma a facilitar a leitura

Exercícios Complementares

Algoritmos com repetição

1. Escreva um algoritmo para repetir a leitura de um número enquanto o valor fornecido for diferente de 0. Para cada número fornecido, imprimir se ele é NEGATIVO ou POSITIVO. Quando o número 0 for fornecido a repetição de ser encerrada sem imprimir mensagem alguma.

OBS: Utilize uma estrutura de repetição com teste de saída no final (Faça/Enquanto).

[Para os dados de entrada abaixo]	[Deve ser gerada a seguinte saída]
4	Positivo
-1	Negativo
2	Positivo
6	Positivo
-7	Negativo
0	

2. Escreva um algoritmo para ler uma quantidade indeterminada de valores inteiros. Para cada valor fornecido escrever uma mensagem que indica se cada valor fornecido é PAR ou IMPAR. O algoritmo será encerrado imediatamente após a leitura de um valor NULO ou NEGATIVO.

[Para os dados de entrada abaixo]	[Deve ser gerada a seguinte saída]
11	Ímpar
3	Ímpar
2	Par
10	Par
5	Ímpar
-2	

3. Ler uma quantidade indeterminada de duplas de valores (2 valores de cada vez). Escrever para cada dupla uma mensagem que indique se ela foi informada em ordem crescente ou decrescente. A repetição será encerrada ao ser fornecido para os elementos da dupla valores iguais.

[Para os dados de entrada abaixo]	[Deve ser gerada a seguinte saída]
5 4	Decrescente
7 2	Decrescente
3 8	Crescente
2 2	

4. Escreva um algoritmo para repetir a leitura de uma senha até que ela seja válida. Para cada leitura da senha incorreta informada escrever a mensagem "SENHA INVÁLIDA". Quanto a senha for informada corretamente deve ser impressa a mensagem "ACESSO PERMITIDO" e o algoritmo encerrado. Considere que a senha correta é o valor 2002

[Para os dados de entrada abaixo]	[Deve ser gerada a seguinte saída]
2200	Senha Inválida
1020	Senha Inválida
2022	Senha Inválida
2002	Acesso Permitido

5. Escreva um algoritmo para ler as coordenadas (X,Y) de uma quantidade indeterminada de pontos no sistema cartesiano. Para cada ponto escrever o quadrante a que ele pertence. O algoritmo será encerrado quando pelo menos uma de duas coordenadas for NULA (nesta situação sem escrever mensagem alguma).

[Para os dados de entrada abaixo]	[Deve ser gerada a seguinte saída]
2 2	primeiro
3 -2	quarto
-8 -1	terceiro
-7 1	segundo
0 2	

6. Para que a divisão entre 2 números possa ser realizada, o divisor não pode ser nulo. Escreva um algoritmo para ler 2 valores e imprimir o resultado da divisão do primeiro pelo segundo.

OBS: O algoritmo deve validar a leitura do segundo valor (que não deve ser nulo). Enquanto for fornecido um valor nulo a leitura deve ser repetida. Utilize a estrutura FAÇA/ENQUANTO na construção da repetição de validação.

7. Altere a solução do exercício anterior, para que seja impressa a mensagem **Valor inválido** caso o segundo valor informado seja ZERO.

8. Altere a solução do exercício 6 para que seja impressa a mensagem **Valor inválido** caso o segundo valor informado seja 0.

9. Escreva um algoritmo para ler as notas da 1a. e 2a. avaliações de um aluno, calcular e imprimir a média semestral. Faça com que o algoritmo só aceite notas válidas(uma nota válida deve pertencer ao intervalo h0,10i. Cada nota deve ser validada separadamente. Deve ser impressa a mensagem "Nota inválida" caso a nota informada não pertença ao intervalo h0,10i.

10. Reescreva o algoritmo para o problema 9 para que no final seja impressa a mensagem **Novo cálculo (1.sim 2.não)** solicitando ao usuário que informe um código (1 ou 2) indicando se ele deseja ou não executar o algoritmo novamente. Se for informado o código 1 deve ser repetida a execução de todo o algoritmo para permitir um novo cálculo, caso contrário ele deve ser encerrado.

11. Reescreva o algoritmo do exercício 9 validando a resposta do usuário para a pergunta **Novo Cálculo (1.sim 2.não)?** (aceitar apenas os código 1 ou 2).

12. Escreva um algoritmo para ler 2 notas de um aluno, calcular e imprimir a média final. Logo após escrever a mensagem "Calcular a média de outro aluno 1.Sim 2.Não?" e solicitar um resposta. Se a resposta for 1, o algoritmo deve ser executado novamente, caso contrário deve ser encerrado imprimindo a quantidade de alunos aprovados.

13. Reescreva o algoritmo do exercício 12, para que seja impresso no final, a quantidade de alunos aprovados, reprovados ou que ficaram em exame.

14. Reescreva o algoritmo do exercício 13, para que seja impresso no final, o percentual de alunos aprovados, reprovados ou em exame em relação a quantidade total de alunos cujas notas foram informadas.

15. Escreva um algoritmo que verifique a validade de uma senha fornecida pelo usuário. A senha válida é o número 1234.

OBS: Se a senha informada pelo usuário for inválida, a mensagem "ACESSO NEGADO" deve ser impressa e repetida a solicitação de uma nova senha até que ela seja válida. Caso contrário deve ser impressa a mensagem "ACESSO PERMITIDO" junto com um número que representa quantas vezes a senha foi informada.

16. A Federação Gaúcha de Futebol contratou você para escrever um programa para fazer uma estatística do resultado de vários GRENAIS. Escreva um algoritmo para ler o número de gols marcados pelo Inter, o número de gols marcados pelo GRpMIO em um GRENAL, imprimindo o nome do time vitorioso ou a palavra EMPATE. Logo após escrever a mensagem "Novo GRENAL 1.Sim 2.Não?" e solicitar uma resposta. Se a resposta for 1, o algoritmo deve ser executado novamente solicitando o número de gols marcados pelos times em uma nova partida, caso contrário deve ser encerrado imprimindo:

-Quantos GRENAIS fizeram parte da estatística.

-O número de vitórias do Inter.

-O número de vitórias do Grêmio.

-O número de Empates.

- Uma mensagem indicando qual o time que venceu o maior número de GRENAIS (ou NIO HOUVE VENCEDOR).

17. Um Posto de combustíveis deseja determinar qual de seus produtos tem a preferência de seus clientes. Escreva um algoritmo para ler o tipo de combustível abastecido (codificado da seguinte forma: 1.álcool 2.Gasolina 3.Diesel 4.Fim).

Caso o usuário informe um código inválido (fora da faixa de 1 a 4) deve ser solicitado um novo código (até que seja válido). Ao ser informado o código do combustível, o seu respectivo nome deve ser impresso na tela. O programa será encerrado quando o código informado for o número 4 escrevendo então a mensagem: "MUITO OBRIGADO" e a quantidade de clientes que abasteceram cada tipo de combustível.

18. Escreva um algoritmo para imprimir os número de 1 a 10 utilizando uma estrutura FAoA/ENQUANTO e um contador.

19. Escreva um algoritmo para escrever a palavra PROGRAMACAO 5 vezes utilizando uma estrutura de repetição e um contador.

20. Escreva um algoritmo para ler um número inteiro e escrevê-lo na tela 10 vezes utilizando uma repetição e um contador.

21. Escreva um algoritmo que imprima na tela os 10 primeiros números inteiros maiores que 100 utilizando uma estrutura de repetição.

22. Escreva um algoritmo que imprima os números ímpares existentes de entre 1(inclusive) e 9(inclusive).

23. Escreva um algoritmo que imprima os números de 100 a 200 de 10 em 10 utilizando uma estrutura de repetição.

24. Escreva um algoritmo para ler um valor N (validar para aceitar apenas valores positivos). Imprima a seguir "PROGRAMA" N vezes.

25. Escreva um algoritmo para ler um valor N (validar para aceitar apenas valores positivos) e imprimir os N primeiros números inteiros.

26. Escreva um algoritmo que imprima a tabuada do 8 utilizando uma estrutura de repetição.

27. Ler 10 valores e escrever quantos destes valores são negativos.

28. Ler 10 valores, calcular e escrever a média aritmética destes valores.

29. Ler 2 valores (considere que o primeiro valor lido sempre será menor que o segundo), calcular e escrever a soma dos números inteiros existentes entre eles(incluindo os 2 valores lidos na soma).

30. Reescreva o exercício anterior considerando que o primeiro pode ser maior que o segundo e vice-versa.

31. Escreva um algoritmo que forneça quantos números devem existir em sequência a partir do 1 (1,2,3,4,...) para que a sua soma ultrapasse a 100 o mínimo possível.

32. Ler um valor A e um valor N. Imprimir a soma dos N números a partir de A(inclusive). Caso N seja negativo ou ZERO, deverá ser lido um novo N(apenas N).

Valores para teste:

A N SOMA

3 2 7 (3+4)

4 5 30 (4+5+6+7+8)

33. Ler um valor X e um valor g (se g for menor que X deve ser lido um novo valor para g). Contar quantos números inteiros devemos somar em sequência (a partir do X inclusive) para que a soma ultrapasse a g o mínimo possível. Escrever o valor final da contagem.

Ex:

X g Reposta

3 20 5 (3+4+5+6+7=25)

2 10 4 (2+3+4+5=14)

30 40 2 (30+31=61)

34. A seguinte sequência de números 0 1 1 2 3 5 8 13 21... é conhecida como série de Fibonacci. Nessa sequência, cada número, depois dos 2 primeiros, é igual à soma dos 2 anteriores. Escreva um algoritmo que mostre os 20 primeiros números dessa série.

35. Ler um valor (só deve aceitar valores positivos), calcular e escrever seu respectivo fatorial.

Fatorial de $n = n * (n-1) * (n-2) * (n-3) * \dots * 1$

Ex: Fatorial de $4 = 4 * 3 * 2 * 1 = 24$

36. Ler um número indeterminado de dados, contendo cada um, a idade de um indivíduo. O último dado, que não entrará nos cálculos, contém o valor de idade negativa. Calcular e imprimir a idade média deste grupo de indivíduos.
37. Escreva um algoritmo para calcular e escrever o valor de S.

38. Escreva um algoritmo para calcular e escrever o valor de S.

$$S = 1 + 3/2 + 5/4 + 7/8 + \dots + 99/?$$

39. Escreva um algoritmo para calcular e escrever o valor de S.

$$S = 1/2 + 2/4 + 3/8 + 4/16 + 5/32 + \dots + 20/?$$

40. Escreva um algoritmo que calcule e escreva a soma dos 30 primeiros termos da série:

$$\frac{1000}{1} - \frac{997}{2} + \frac{994}{3} - \frac{991}{4} +$$

41. Elabore um algoritmo para resolver o valor de S, onde:

$$S = 1/1 - 2/4 + 3/9 - 4/16 + 5/25 - 6/36 \dots - 10/100$$

42. Escrever um algoritmo que lê um valor N inteiro e positivo e que calcula e escreve o valor de E.

$$E = 1 + 1/1! + 1/2! + 1/3! + 1/N!$$

43. Escreva um algoritmo que lê um valor n inteiro e positivo e que calcula a seguinte soma:

$$S := 1 + 1/2 + 1/3 + 1/4 + \dots + 1/n$$

O algoritmo deve escrever cada termo gerado e o valor final de S.

44. Elabore um algoritmo que calcule e escreva o valor H, sendo que ele é determinado pela seguinte série:

$$H = 1/1 + 3/2 + 5/3 + 7/4 + \dots 99/50.$$

45. Escreva um algoritmo que calcule e escreva a soma dos dez primeiros termos da série:

$$2/500 - 5/450 + 2/400 - 5/350 + \dots$$

46. Escreva um algoritmo para calcular e escrever o n-ésimo termo da série abaixo, onde o valor de N é será informado pelo usuário.

$$5, 6, 11, 12, 17, 18, 23, 24, \dots$$

Exemplo:

-para N=3 deve ser impresso o valor 11

-para N=5 deve ser impresso o valor 17

-para N=8 deve ser impresso o valor 24

47. Ler 2 valores A e B. Se A for igual a B devem ser lidos novos valores para A e B. Se A for menor que B calcular e imprimir a soma dos números ímpares existentes entre A(inclusive) e B(inclusive). Se A for maior que B calcular e imprimir a média aritmética dos múltiplos de 3 existentes entre A(inclusive) e B(inclusive).

OBS: Considere que só serão informados valores inteiros positivos.

48. Suponha que exista um prédio de 20 andares onde existam três elevadores, identificados pelos número 1, 2 e 3. Para otimizar o sistema de controle dos elevadores, foi realizado um levantamento no qual cada usuário respondia qual o elevador que utilizava com mais frequência. Escreva um algoritmo que leia o número de usuários do prédio, leia as respostas de cada usuário calcule e imprima qual o elevador mais freqüentado.

49. No planeta Alpha vive a criatura Blobs, que come precisamente 1/2 de seu suprimento de comida disponível todos os dias. Escreva um algoritmo que leia a capacidade inicial de suprimento de comida (em Kg), e calcule quantos dias passarão antes de Blobs coma todo esse suprimento até atingir um quilo ou menos.

50. Considere uma sequência de números que tenha a propriedade que começa com o quarto deles, e cada número é a soma do anterior mais duas vezes o anterior ao anterior a ele, menos o anterior dele. Por exemplo, o 8o. deles é (o 7o. deles) mais duas vezes (o 6o. deles) menos (o 5o. deles). Suponha que os primeiros três números na sequência são: 1, 2, 3, escreva um algoritmo que encontre o 15o deles.

51. Ler um número inteiro N (só aceitar valores positivos. Caso o N não seja positivo deverá ser lido um novo valor para o N) e escrever os divisores de N.

52. Ler um número inteiro N (só aceitar valores positivos-Caso o N não seja positivo deverá ser lido um novo valor para o

N) e escrever se é ou não PRIMO.

53. Escrever os números de 1 a 20 com seus respectivos divisores.

54. Escreva um algoritmo que imprima todos os números primos existentes entre N1 e N2, onde N1 e N2 são números positivos fornecidos pelo usuário.

55. Calcular e escrever a maior nota entre N notas lidas. O valor de N será informado pelo usuário.

56. Ler um número indeterminado de notas. O algoritmo deverá ser encerrado ao ser fornecido um valor negativo para a nota. Calcular e escrever a menor nota entre as notas lidas.

57. Ler um número indeterminado de pares de valores. Cada par representa a idade de um aluno e a sua respectiva nota. Calcular e escrever:

- A idade do aluno que obteve a maior nota.

- A nota do aluno mais velho.

Obs: O algoritmo será encerrado imediatamente após o usuário fornecer uma idade negativa (sem fornecer a respectiva nota).

58. Calcular e escrever o maior entre N valores lidos do teclado. O valor de N será fornecido pelo usuário.

59. Ler um conjunto de dados contendo, cada um, uma nota. Determinar e escrever a maior e a menor nota da turma. O último dado que não será processado, contém, nota = 99.

60. Ler o código e a altura de cada aluno de uma turma de 10 alunos. Escrever o código e a altura do aluno mais alto.

61. Ler um conjunto de 10 pares de dados contendo, cada um, a altura e um código para masculino (1) e outro para feminino (2) de uma turma de alunos. Calcular e escrever:

- a maior e a menor altura da turma.

- a média de altura das mulheres.

62. Ler um número indeterminado de dados, contendo cada um o peso de um indivíduo. O último dado que não entrará nos cálculos, contém um valor negativo. Calcular e imprimir:

- A média aritmética das pessoas que possuem mais de 60 Kg.

- O peso do mais pesado entre aqueles que possuem menos de 60 Kg.

63. Escrever um algoritmo que calcula e mostra a média aritmética dos números pares compreendidos entre 13 e 73.

64. A prefeitura de uma cidade fez uma pesquisa entre seus habitantes, coletando dados sobre o salário e número de filhos. A prefeitura deseja saber:

- a) média do salário da população;

- b) média do número de filhos;

- c) maior salário;

- d) percentual de pessoas com salário até R\$100,00.

O final da leitura de dados se dará com a entrada de um salário negativo. (Use o comando ENQUANTO-FACA)

65. Chico tem 1,50 metro e cresce 2 centímetros por ano, enquanto gé tem 1,10 metro e cresce 3 centímetros por ano. Construa um algoritmo que calcule e imprima quantos anos serão necessários para que gé seja maior que Chico.

66. Faça um algoritmo que leia vários números inteiros e positivos e calcule o produto dos números pares. O fim da leitura será indicado pelo número 0.

67. Foi realizada uma pesquisa de algumas características físicas da população de uma certa região, a qual coletou os seguintes dados referentes a cada habitante para serem analisados:

- sexo (masculino e feminino)

- cor dos olhos (azuis, verdes ou castanhos)

- cor dos cabelos (louros, castanhos, pretos)

- idade

Faça um algoritmo que determine e escreva:

- a maior idade dos habitantes

- a quantidade de indivíduos do sexo feminino cuja idade está entre 18 e 35 anos inclusive e que tenham olhos verdes e cabelos louros.

O final do conjunto de habitantes é reconhecido pelo valor -1 entrada como idade.

68. Construa um algoritmo para receber uma sequência de números inteiros qualquer. Quando o número zero for entrado a repetição deve ser interrompida e deve ser mostrada a média dos números digitados.

69. Faça um algoritmo que calcule o faturamento de um cinema a cada sessão. Devemos considerar que os menores de 18 anos pagam meia, devido à carteirinha de estudante e que os maiores de 65 anos também pagam meia, devido à carteirinha de aposentado. O preço normal do ingresso irá variar conforme o dia da semana, portanto deve ser solicitada estas

informação. A quantidade de pessoas no cinema irá variar a cada sessão, portanto deve haver esta informação também. Além disto, o algoritmo pode ser executado mais de uma vez, ou seja, deve-se verificar ao final do cálculo de uma sessão se o usuário deseja verificar o faturamento de outra sessão.

`vetor[10], x, matriz[5,5] : numerico`70. Faça um algoritmo que dado um número você imprima em ordem decrescente seu valor variando de 2 em 2 até atingir zero. Quando o número for digitado verifique se ele é maior que 100 antes de realizar a impressão. Se ele for menor que 100 envie uma mensagem ao usuário de que o número digitado está fora do limite e solicite que um novo número seja digitado. Quando o usuário entrar com o número -1 o algoritmo deve ser finalizado.

71. Escreva Algoritmo que apresente todos fatoriais cujo resultado seja inferior a um dado valor que é lido pelo teclado.

72. Escreva um Algoritmo que apresente em ordem decrescente, os fatoriais desde um dado valor, até o fatorial de 1.

73. Elabore um algoritmo que obtenha o número inteiro que mais se aproxima da raiz quadrada de um número fornecido pelo usuário.

74. Um rainha requisitou os serviços de um monge e disse-lhe que pagaria qualquer preço. O monge, necessitando de

alimentos, perguntou a rainha se o pagamento poderia ser feito em grãos de trigo dispostos em um tabuleiro de damas, de forma que o primeiro quadrado tivesse apenas um grão, e os quadrados subsequentes, o dobro do quadrado anterior..A rainha considerou o pagamento barato o pediu que o serviço fosse executado, sem se dar conta de que seria impossível executar o pagamento. Escreva um algoritmo para calcular o número de grãos que o monge esperava receber.

75. Existem números de 4 dígitos (de 1000 a 9999) que obedecem à seguinte característica: se dividirmos o número em dois números de dois dígitos, um composto pela dezena e pela unidade, e outro pelo milhar e pela centena, somarmos estes dois novos números gerando um terceiro, o quadrado deste terceiro número é exatamente o número original de quatro dígitos. Por exemplo: 2025-> separando: 20 e 25 -> somando temos 45 -> $45^2 = 2025$. Escreva um programa para calcular todos os números que obedecem a esta característica.

76. Ler um número N qualquer menor ou igual a 50 e apresentar o valor obtido da multiplicação sucessiva de N por 3 (ou seja: $N*3$, $N*3*3$, $N*3*3*3$, ...), enquanto o produto for menor do que 250.

77. Escrever um algoritmo que escreva a soma dos números que não são múltiplos de 13 entre 100 e 200.

Lembrete:

Algoritmos não se aprendem:

- Copiando algoritmos
- Estudando algoritmos

Algoritmos só se aprendem:

- Construindo algoritmos
- Testando algoritmos

Algoritmos com Vetores e matrizes

1. Elaborar um algoritmo que lê um conjunto de 30 valores e os coloca em 2 vetores conforme estes valores forem pares ou ímpares. O tamanho do vetor é de 5 posições. Se algum vetor estiver cheio, escrevê-lo. Terminada a leitura escrever o conteúdo dos dois vetores. Cada vetor pode ser preenchido tantas vezes quantas for necessário.
2. Faça um algoritmo que leia um vetor $N[20]$. A seguir, encontre o menor elemento do vetor N e a sua posição dentro do vetor, mostrando: "O menor elemento de N é", M , "e sua posição dentro do vetor é:", P .
3. Escreva um algoritmo que leia dois vetores de 10 posições e faça a multiplicação dos elementos de mesmo índice, colocando o resultado em um terceiro vetor. Mostre o vetor resultante.
4. Faça um algoritmo que leia um vetor $K[30]$. Troque a seguir, todos os elementos de ordem ímpar do vetor com os elementos de ordem par imediatamente posteriores.
5. Faça um algoritmo que leia um vetor $S[20]$ e uma variável A . A seguir, mostre o produto da variável A pelo vetor.
6. Faça um algoritmo que leia dois vetores: $F[20]$ e $G[20]$. Calcule e mostre, a seguir, o produto dos valores de F por G .
7. Escreva um algoritmo que leia dois vetores de 10 posições e faça a multiplicação dos elementos de mesmo índice, colocando o resultado em um terceiro vetor. Mostre o vetor resultante.
8. Escreva um algoritmo que leia e mostre um vetor de 20 números. A seguir, conte quantos valores pares existem no vetor.
9. Escreva um algoritmo que leia um vetor de 100 posições e mostre-o ordenado em ordem crescente.
10. Escreva um algoritmo que leia um vetor de 20 posições e mostre-o. Em seguida, troque o primeiro elemento com o último, o segundo com o penúltimo, o terceiro com o antepenúltimo, e assim sucessivamente. Mostre o novo vetor depois da troca.
11. Escreva um algoritmo que leia 50 valores para um vetor de 50 posições. Mostre depois somente os positivos.
12. Escreva um algoritmo que leia um vetor inteiro de 30 posições e crie um segundo vetor, substituindo os valores nulos por 1. Mostre os 2 vetores.
13. Escreva um que leia um vetor G de 20 elementos caracter que representa o gabarito de uma prova. A seguir, para cada um dos 50 alunos da turma, leia o vetor de respostas (R) do aluno e conte o número de acertos. Mostre o nº de acertos do aluno e uma mensagem APROVADO, se a nota for maior ou igual a 6; e mostre uma mensagem de REPROVADO, caso contrário.
14. Escreva um algoritmo que leia um vetor de 13 elementos, que é o Gabarito de um teste da loteria esportiva, contendo os valores 1 (coluna 1), 2 (coluna 2) e 3 (coluna do meio). Leia, a seguir, para cada apostador, o número do seu cartão e um vetor de Respostas de 13 posições. Verifique para cada apostador o números de acertos, comparando o vetor de Gabarito com o vetor de Respostas. Escreva o número do apostador e o número de acertos. Se o apostador tiver 13 acertos, mostrar a mensagem "Ganhador".
15. Escrever um algoritmo que gera os 30 primeiros números primos a partir de 100 e os armazena em um vetor de $X[100]$ escrevendo, no final, o vetor X .
16. Escrever um algoritmo que lê 2 vetores de tamanho 10. Crie, a seguir, um vetor S de 20 posições que contenha os elementos dos outros 2 vetores em ordem crescente. Obs.: copie primeiro os valores para o vetor S para depois ordená-los
17. Escrever um algoritmo que lê 2 vetores $X[10]$ e $Y[10]$. Crie, a seguir, um vetor g que seja
 - a) a diferença entre X e Y ;
 - b) a soma entre X e Y ;
 - c) o produto entre X e Y ;Escreva o vetor g a cada cálculo.
18. Escrever um algoritmo que lê um vetor $K[15]$. Crie, a seguir, um vetor P , que contenha todos os números primos de K . Escreva o vetor P .
19. Escrever um algoritmo que lê um vetor $X[20]$. Escreva, a seguir, cada um dos valores distintos que aparecem em X dizendo quantas vezes cada valor aparece em X .
20. Faça um algoritmo que leia dois vetores de 200 posições de caracteres. A seguir, troque o 1º elemento de A com o 200º de B , o 2º de A com o 199º de B , assim por diante, até trocar o 200º de A com o 1º de B . Mostre os vetores antes e depois da troca.

21. Faça um algoritmo que leia um código numérico inteiro e um vetor de 50 posições de números. Se o código for zero, termine o algoritmo. Se o código for 1, mostre o vetor na ordem como ele foi lido. Se o código for 2, mostre o vetor na ordem inversa, do último até o primeiro.

22. Faça um algoritmo que leia um vetor (A) de 100 posições. Em seguida, compacte o vetor, retirando os valores nulos e negativos, colocando o resultado em um vetor B de 100 posições (deixe em branco as posições não utilizadas).

23. Faça um algoritmo que leia um vetor (A) de 100 posições. Em seguida, compacte o vetor, retirando os valores nulos e negativos, colocando o resultado em um vetor B de 100 posições (Defina o vetor B com o número exato de posições para que seja suficientemente grande para conter o vetor A sem os valores nulos).

24. Faça um algoritmo que leia um vetor de 500 posições de números e divida todos os seus elementos pelo maior valor do vetor. Mostre o vetor após os cálculos.

25. Faça um algoritmo que leia um vetor de 10 posições. Mostre então os 3 menores valores do vetor.

26. Faça um algoritmo que leia dois vetores (A e B) de 50 posições de números. O algoritmo deve, então, subtrair o primeiro elemento de A do último de B, acumulando o valor, subtrair o segundo elemento de A do penúltimo de B, acumulando o valor, e assim por diante. Mostre o resultado da soma final.

27. Uma locadora de vídeos tem guardada, em um vetor de 500 posições, a quantidade de filmes retirados por seus clientes durante o ano de 1993. Agora, esta locadora está fazendo uma promoção e, para cada 10 filmes retirados, o cliente tem direito a uma locação grátis. Faça um algoritmo que crie um outro vetor contendo a quantidade de locações gratuitas a que cada cliente tem direito.

28. Faça um algoritmo que leia um vetor A[10]. Preencha então um vetor B[10] com o fatorial de cada valor de A respeitando as posições, caso o referido valor for positivo ou nulo. Deixe os valores negativos intactos. Mostre o vetor B.

29. Faça um algoritmo que leia um vetor A[10]. Preencha então um vetor B[10] com o fatorial de cada valor de A respeitando as posições, caso o referido valor for positivo ou nulo. Substitua no final os valores negativos por 0 (zero). Mostre o vetor B.

30. Faça um algoritmo que leia um vetor A[10]. Inverta então os valores de A. troque o primeiro pelo último, segundo pelo penúltimo e assim por diante. Mostre o vetor A após as alterações.

31. Elaborar um algoritmo que lê duas matrizes M[4,6] e N[4,6] e cria uma matriz que seja:

- a) o produto de M por N;
- b) a soma de M com N;
- c) a diferença de M com N;

Escrever as matrizes calculadas.

32. Elaborar um algoritmo que lê uma matriz M[6,6] e um valor A e multiplica a matriz M pelo valor A e coloca os valores da matriz multiplicados por A em um vetor de V[36] e escreve no final o vetor V.

33. Escreva um algoritmo que leia um número inteiro A e uma matriz V 30x30 de números. Conte quantos valores iguais a A estão na matriz. Crie, a seguir, uma matriz X contendo todos os elementos de V diferentes de A. Mostre os resultados.

34. Escreva um algoritmo que lê uma matriz M[5,5] e calcula as somas:

- a) da linha 4 de M
- b) da coluna 2 de M
- c) da diagonal principal
- d) da diagonal secundária
- e) de todos os elementos da matriz M

Escrever essas somas e a matriz.

35. Escrever um algoritmo que lê uma matriz A[15,5] e a escreva. Verifique, a seguir, quais os elementos de A que estão repetidos e quantas vezes cada um está repetido. Escrever cada elemento repetido com uma mensagem dizendo que o elemento aparece X vezes em A.

36. Escrever um algoritmo que lê uma matriz M[12,13] e divida todos os 13 elementos de cada uma das 12 linhas de M pelo maior elemento em módulo daquela linha. Escrever a matriz lida e a modificada.

37. Escrever um algoritmo que lê uma matriz $M[10,10]$ e a escreve. Troque, a seguir:

a) a linha 2 com a linha 8

b) a coluna 4 com a coluna 10

Escreva a matriz assim modificada.

38. Escrever um algoritmo que lê uma matriz $M[10,10]$ e a escreve. Troque, a seguir:

a) a diagonal principal com a diagonal secundária

b) a linha 5 com a coluna 10.

Escreva a matriz assim modificada.

39. Escrever um algoritmo que lê uma matriz $M[12,13]$ e divida todos os 13 elementos de cada uma das 12 linhas de M pelo maior elemento em módulo daquela linha. Escrever a matriz lida e a modificada.

40. Escrever um algoritmo que lê uma matriz $M[5,5]$ e cria 2 vetores $SL[5]$ e $SC[5]$ que contenham, respectivamente, as somas das linhas e das colunas de M . Escrever a matriz e os vetores criados.

41. Escreva um algoritmo que lê uma matriz $M[5,5]$. Substitua, a seguir, todos os valores negativos da matriz pelo seu módulo. Exemplo: substitua -2 por 2, -16 por 16, assim por diante.

42. Escreva um algoritmo que lê uma matriz $M[6,6]$. A seguir, troque os elementos da primeira coluna com os elementos da segunda coluna, os da terceira coluna com a quarta coluna e os elementos da quinta coluna com os elementos da sexta coluna.

43. Repita o exercício anterior, trocando os elementos das linhas, ao invés das colunas.

44. Leia uma matriz $M[5,5]$. A seguir, ordene os elementos da matriz M e mostre como ficou a Matriz ordenada, linha por linha.

45. Faça um algoritmo que calcule a média dos elementos da diagonal principal de uma matriz 10 X 10 de números.

46. Faça um algoritmo que calcule a média dos elementos da diagonal secundária de uma matriz 10 X 10 de números.

47. Faça um algoritmo que gere a seguinte matriz:

```
1 1 1 1 1 1
1 2 2 2 2 1
1 2 3 3 2 1
1 2 3 3 2 1
1 2 2 2 2 1
1 1 1 1 1 1
```

48. Faça um algoritmo que gere a seguinte matriz:

```
1 3 3 3 3 2
3 1 3 3 2 3
3 3 1 2 3 3
3 3 2 1 3 3
3 2 3 3 1 3
2 3 3 3 3 1
```

49. Faça um algoritmo que leia uma matriz numérica 15 X 15 e calcule a soma dos elementos da diagonal secundária.

50. Faça um algoritmo que leia uma matriz 20x15 de números. calcule e mostre a soma das linhas pares da matriz.

51. Faça um algoritmo que leia uma matriz 20x20 de números e some cada uma das linhas, armazenando o resultado da soma em um vetor. A seguir, multiplique cada elemento pela soma da sua linha. Mostre a matriz resultante.

52. Faça um algoritmo que leia uma matriz 50x50 de números e encontre o maior valor da matriz. A seguir, multiplique cada elemento da diagonal principal pelo maior valor. Mostre a matriz após as multiplicações.

53. Faça um algoritmo que leia uma matriz 50x50 de números. A seguir, multiplique cada linha pelo elemento da diagonal principal daquela linha. Mostre a matriz após as multiplicações.

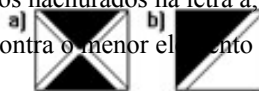
54. Faça um algoritmo que leia uma matriz de 60 linhas e 10 colunas. Depois de lê-la, some as colunas individualmente e acumule a soma na 61ª linha da matriz. Mostre o resultado de cada coluna no vídeo. (Lembrete: para guardar o resultado é necessário declarar uma matriz de 61 x 10.)

55. Na teoria dos sistemas, define-se como elemento minimax de uma matriz o menor elemento da linha onde se encontra o maior elemento da matriz. Escreva um algoritmo que leia uma matriz 10 X 10 de números e encontre seu elemento minimax, mostrando também sua posição.

56. Escrever um programa que lê uma matriz 17x17 e:

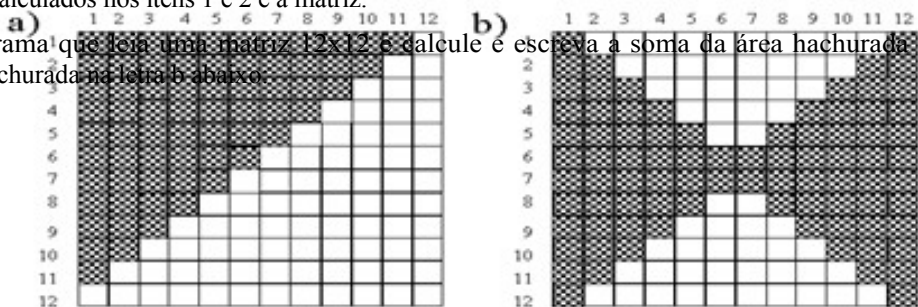
1. calcula a média aritmética dos elementos hachurados na letra a;

2. o maior elemento da linha onde se encontra o menor elemento da área hachurada na letra b;



Escreva os valores calculados nos itens 1 e 2 e a matriz.

57. Faça um programa que leia uma matriz 12x12 e calcule e escreva a soma da área hachurada na letra a e o maior elemento da área hachurada na letra b abaixo.



58. Faça um programa que leia uma matriz 12 x 12 e calcule e escreva:

- o menor elemento e a sua posição (índices) da área hachurada;
- a média dos elementos da área hachurada



59. Faça um programa lê uma matriz A 7 x 7 de números e cria 2 vetores ML[7] e MC[7], que contenham, respectivamente, o maior elemento de cada uma das linhas e o menor elemento de cada uma das colunas. Escrever a matriz A e os vetores ML e MC.

60. Execute o algoritmo abaixo, preenchendo a matriz M [2,6]:

```

variaveis
matriz numerico M [2,6]
2 1 4 0
numerico fat,i
inicio
para i de 1 até 6
    leia M[1,i]
    se M[1,i] > 0 então
        fat := 1
        para j de 1 até M[1,i]
            fat := fat * j
        fimpara
        M[2,i] := fat
    senão
        M[2,i] := i * i
    fim_se
    M[2,i] := M[2,i] - 10
fimpara
    
```

Na leitura da Matriz, considere os valores: -5 7

M [2,6]=

1	2	3	4	5	6

fim

Anexo I - Resolução de exercícios

Resolução de alguns dos algoritmos propostos nas páginas 15 e 16 da apostila

14)

```
variaveis
    numerico vol,R
inicio
    escrever "Digite Raio:"
    leia R
    vol:= 4/3 * 3.1415 * R^3
    escrever "Volume da esfera = ", vol
fim
```

15)

```
variaveis
    numerico A,B,C
inicio
    escrever "Digite A:"
    leia A
    escrever "Digite B:"
    leia B
    escrever "Digite C:"
    leia C
    escrever "Área do triangulo é: ", (A*C)/2
    escrever "Área do círculo é: ", 3.1415*C^2
    escrever "Área do trapézio é: ", (A+B)/2 * C
    escrever "Área do quadrado é: ", B * B
    escrever "Área do retângulo é: ", A*B
fim
```

17)

```
variaveis
    numerico dist,comb,consumo
inicio
    limpar
    escrever "Digite distância percorrida em Kms:"
    leia dist
    escrever "Digite combustível gasto na viagem em litros:"
    leia comb
    consumo := dist/comb
    escrever "O consumo médio do carro foi de ", consumo, " km(s) por litro"
fim
```

19)

```
variaveis
    numerico distancia,tpo,vel,litros_usados
inicio
    limpar
    escrever "Digite o tempo gasto na viagem em horas:"
    leia tpo
    escrever "Digite a velocidade média na viagem (km/h):"
    leia vel
    distancia := tpo*vel
    litros_usados :=distancia/12
    escrever "Foram gastos ", litros_usados, " litros de combustível"
fim
```

20)

```
variaveis
    numerico distancia,tpo
inicio
    limpar
    escrever "Digite a distancia em Kilômetros:"
    leia distancia
    escrever "leva ", distancia*2, " minutos p/ o carro mais veloz tomar esta distância."
fim
```



```

21)
variaveis
    numerico    vel_x,vel_y,vel_z,Htpx,Htpy,Htpz,Mtpx,Mtpy,Mtpz
inicio
    // Neste algoritmo está também mostrado o tempo em minutos,
    // o que seria opcional, pois o cálculo fica mais complexo
    // -----
    limpar
    escrever "Digite a velocidade do carro X (KM/h):"
    leia vel_x
    escrever "Digite a velocidade do carro Y (KM/h):"
    leia vel_y
    escrever "Digite a velocidade do carro Z (KM/h):"
    leia vel_z

    Htpx:=inteiro(360/vel_x)
    Mtpx:=(360/vel_x)-Htpx
    Mtpx:=Mtpx*60

    Htpy:=inteiro(270/vel_y)
    Mtpy:=(270/vel_y)-Htpy
    Mtpy:=Mtpy*60

    Htpz:=inteiro(310/vel_z)
    Mtpz:=(310/vel_z)-Htpz
    Mtpz:=Mtpz*60

    escrever "leva ", Htpx, " horas e ", Mtpx," minutos para o carro X percorrer 360 Km"
    escrever "leva ", Htpy, " horas e ", Mtpy," minutos para o carro Y percorrer 270 Km"
    escrever "leva ", Htpz, " horas e ", Mtpz," minutos para o carro Z percorrer 310 Km"
fim

```

```

26)
variaveis
    numerico dh, hi, hf, mi, mf, tpo_ini, tpo_fim, x, dur, durh, durm
inicio
    limpar
    escrever "Digite a hora de início do jogo:"
    leia hi
    escrever "Digite o minuto de início do jogo:"
    leia mi
    escrever "Digite a hora do fim do jogo:"
    leia hf
    escrever "Digite o minuto final do jogo:"
    leia mf
    tpo_ini:= hi*60+mi
    tpo_fim:= hf*60+mf

    x:= inteiro ((1440 - tpo_ini + tpo_fim)/1441 )
    dur:= (1440 - tpo_ini + tpo_fim) - x * 1440 durh
    := inteiro(dur/60)
    durm := dur - durh*60
    escrever "Durou ", durh, " hora(s) e ",durm, " minuto(s)"
fim

```

Resolução de alguns dos algoritmos propostos na página 33 da apostila

```

25)
variaveis
    matriz numerico V[10]
    matriz numerico Comp[10]
    numerico I,Aux,Ordem,Cont
inicio
    Para I de 1 ate 10
        LEIA V[I]
    fimpara
    Ordem := 0
    Faca enquanto (Ordem = 0)
        Ordem := 1
        Para I de 1 ate 9
            Se V[I] > V[I + 1] entao
                Aux := V[I]

```

```

26)
variaveis
    matriz numerico V[10]
    matriz numerico W[10]
    numerico I,Aux,Ordem,Cont
inicio
    Para I de 1 ate 10
        LEIA V[I]
    fimpara
    Ordem := 0
    Faca enquanto Ordem = 0
        Ordem := 1
        Para I de 1 ate 9
            Se V[I] > V[I + 1] entao
                AUX := V[I]

```

```

        V[I] := V[I + 1]
        V[I + 1] := Aux
        Ordem := 0
    Fim_Se
fimpara
Fim_Enquanto
Cont := 1
Para I de 2 ate 10
    Se V[I] > V[I - 1] entao
        Cont := Cont + 1
        Comp[Cont] := V[I]
    Fim_Se
fimpara
Para I de 1 ate Cont
    Escrever Comp[I]
fimpara
Fim

```

27)

```

variaveis
matriz numerico P[5]
matriz numerico I[5]
numerico V,J,K,L,ContP,ContI
inicio
ContP := 0
ContI := 0
Para J de 1 ate 15
    Leia V
    Se V / 2 = INTEIRO(V / 2) entao
        ContP := ContP + 1
        P[ContP] := V
    Senao
        ContI := ContI + 1
        I[ContI] := V
    Fim_Se
    Se ContP = 5 entao
        Para K de 1 ate 5
            Escrever P[K]
        fimpara
        ContP := 0
    Fim_Se
    Se ContI = 5 entao
        Para K de 1 ate 5
            Escrever I[K]
        fimpara
        ContI := 0
    Fim_Se
fimpara
Para L de 1 ate ContI
    Escrever I[L]
fimpara
Para L de 1 ate ContP
    Escrever P[L]
fimpara
Fim

```

29)

```

variaveis
matriz numerico V[10]
numerico I,J,A,Ordem,Aux,Posic
inicio
Para I de 1 ate 5
    Leia V[I]
fimpara
Ordem := 0
Faca enquanto Ordem = 0
    Ordem := 1
    Para I de 1 ate 4
        Se V[I] > V[I + 1] entao
            AUX := V[I]
            V[I] := V[I + 1]
            V[I + 1] := AUX
        Ordem := 0
    Fim_Se
    fimpara
Fim_Enquanto
Posic := 5
Para I de 1 ate 5
    Escrever "Digite A:"
    Leia A
Para J de 1 ate Posic AUX := V[J]

```

```

        V[I] := V[I + 1]
        V[I + 1] := AUX
        Ordem := 0
    Fim_Se
fimpara
Fim_Enquanto
Cont := 1
W[Cont] := V[1]
Para I de 2 ate 10
    Se V[I] > V[I - 1] entao
        Cont := Cont + 1
        W[Cont] := V[I]
    Fim_Se
fimpara
Para I de 1 ate Cont
    Escrever W[I]
fimpara
Fim

```

28)

```

variaveis
matriz numerico V[10]
matriz numerico X[10]
matriz numerico Y[20]
numerico I, Ordem, Aux
inicio
Para I de 1 ate 10
    Leia V[I]
    Leia X[I]
fimpara
Para I de 1 ate 10
    Y[I] := V[I]
    Y[I + 10] := X[I]
fimpara
Ordem := 0
Faca enquanto Ordem = 0
    Ordem := 1
    Para I de 1 ate 19
        Se Y[I] > Y[I + 1] entao
            AUX := Y[I]
            Y[I] := Y[I + 1]
            Y[I + 1] := AUX
        Ordem := 0
    Fim_Se
    fimpara
Fim_Enquanto
Para I de 1 ate 20
    Escrever Y[I]
fimpara
Fim

```

```

V[J] := A
A := AUX
Fim se
fimpara
Posic := posic + 1
V[posic] := A
Próximo
Escrever 'Matriz Numerico Y[20] ordenada'
Para I de 1 ate 10 faca
    Escrever V[I]
fimpara
fimalgoritmo

```