

Introdução à Algoritmos

Notas de aula

Prof^a. Cynthia Guerra Braga Agosto 2012

"Pensamentos levam a objetivos, objetivos se tornam ações, ações viram hábitos, hábitos formam o caráter, e o caráter determina nosso destino".

Tryon Edwards (1809-1894)

"O homem por si só é um ser lógico, e essa lógica é um dom que não se aprende de uma hora para outra. Resulta, em geral, de experiências do dia a dia"

"Não existem meios de se ensinar lógica, podemos sim, através de um grande esforço, bastante dedicação e exercícios, aperfeiçoar a lógica da pessoa para que esta possa formar <u>sua própria estrutura de pensamento</u>, a qual é de fundamental importância para a arte de programar."

É correto afirmar que todo ser humano é um "resolvedor de problemas" nato, sendo que cada um tem a sua própria lógica de resolução.

Ao contrário do que se pode pensar, o conceito de algoritmo não foi criado para satisfazer às necessidades da computação. Pelo contrário, a programação de computadores é apenas um dos campos de aplicação dos algoritmos. <u>O algoritmo sempre existe, mesmo que apenas no seu pensamento.</u>

A resolução dos algoritmos contribuem para que os alunos comecem a pensar de forma ordenada e clara; adquirirem destreza para Diagnosticar e apresentar soluções aos problemas de engenharia; Ter a capacidade de encontrar soluções inovadoras para diversos problemas do cotidiano.

Orlando Bueno, coordenador do centro Paulista de Neuropsicologia da UNIFESP trata inteligência como sendo "a CAPACIDADE DE SE COMPREENDER O MUNDO À SUA VOLTA, DE <u>RESOLVER PROBLEMAS</u> E DE SE ADAPTAR AO MEIO AMBIENTE POR MEIO DO <u>RACIOCÍNIO</u>"

Este é o desafio, "aprender a aprender", quebrar paradigmas antigos como não sei, não posso e outros tantos impregnados na nossa mente por um novo padrão baseado em vou pensar, vou analisar, vou resolver. Lógica e algoritmos serão degraus para aprender a aprender.

1. Introdução

a) Lógica

Lógica é uma parte da filosofia que estuda o fundamento, a estrutura e as expressões humanas do conhecimento. A lógica foi criada por Aristóteles no século IV a.C. para estudar o pensamento humano e distinguir interferências e argumentos certos e errados.

É o estudo formal sistemático dos princípios da <u>inferência</u> válida e do pensamento correto. Já que o pensamento é a manifestação do conhecimento, e que o conhecimento busca a verdade, é preciso estabelecer algumas regras para que essa meta possa ser atingida. Assim, a lógica é o ramo da filosofia que cuida das regras do bem pensar, ou do pensar correto, sendo, portanto, um *instrumento do pensar*.

Inferência: ato ou processo de derivar conclusões lógicas de premissas conhecida ou decididamente verdadeiras

b) Lógica matemática

A Lógica Matemática, em síntese, pode ser considerada como a ciência do raciocínio e da demonstração. Este importante ramo da Matemática desenvolveu-se no século XIX, sobretudo através das idéias de George Boole, matemático inglês (1815 - 1864), criador da Álgebra Booleana, que utiliza símbolos e operações algébricas para representar <u>proposições</u> e suas interrelações.

c) Lógica de Programação

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

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

Exemplo

Seu pai lhe diz: se você tirar 10 em Física e Matemática, lhe darei um presente. Você sabe que não basta tirar 10 apenas em Física ou apenas em Matemática. Para ganhar o presente, é necessário tirar 10 nas duas disciplinas. Se por outro lado ele dissesse: se você tirar 10 em Física ou Matemática, lhe darei um presente; aí bastaria tirar 10 em uma das matérias.

d) Exemplos:

♣ Com quatro, faço todos os números de 1 a 10. Use e abuse de operações elementares, desde que não tenha mais do que 4 quatros e tenha, obrigatoriamente, 4 quatros

Solução:

$$1 = \frac{44}{44} \quad 2 = \frac{4}{4} + \frac{4}{4} \quad 3 = \frac{4+4+4}{4} \quad 4 = \frac{4-4}{4} + 4 \quad 5 = \frac{4*4+4}{4}$$
$$6 = \frac{4+4}{4} + 4 \quad 7 = 4+4-\left(\frac{4}{4}\right) \quad 8 = 4+4+(4-4) \quad 9 = 4+4+\frac{4}{4} \quad 10 = \frac{44-4}{4}$$

- Complete as sequências
 - a) 1, 2, 6, 39, ____
 - b) 2, 10, 12, 16, 17, 18, 19, ____
 - c) 3, 6, 10, 15, 21, 28, 36, 45, 55, ____

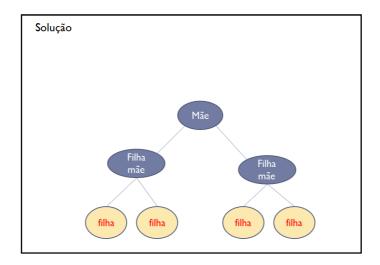
Solução:

- a) 2 = 1*1 + 1
 - 6 = 2*2 + 2
 - 39 = 6*6 + 3
 - X = 39*39 + 4

1525

- b) A sequência é composta pelos números que começam com a letra "D" Resp. **200**
- C) 3+3=6

Entram em um restaurante para jantar três mulheres. Cada uma com duas filhas. Só existiam
 7 lugares. Nenhuma ficou de pé. Como isso é possível?



♣ Tenho 3 camisas:A, B e C. Uma é verde, uma é branca e a outra é azul, não necessariamente nessa ordem. Somente uma das afirmações abaixo é verdadeira:

A é Verde,

B não é Verde e

C não é Azul.

Quais as cores de A, B e C nessa ordem?

Α	В	С	1ª	2ª	3 ª
Azul	Verde	Branco	F	F	V
Azul	Branco	Verde	F	V	V
Branco	Azul	Verde	F	V	V
Branco	Verde	Azul	F	F	F
Verde	Azul	Branco	V	V	V
Verde	Branco	Azul	V	V	F



- 1) Um ladrão entrou em uma loja e roubou um terço das jóias que lá se encontravam. Outro ladrão entrou e roubou dois terços do que encontrou, deixando 12 jóias para trás. Quantas pedras havia inicialmente na loja?
- 2) Um agente de viagens atende três amigas. Uma delas [e loura, outra é morena e a outra é ruiva. O agente sabe que uma delas se chama Bete, outra se chama Elza e a outra se chama Sara. Sabe ainda que cada uma delas fará uma viagem a um pais diferente da Europa. Uma delas irá a Alemanha, outra a França e a outra a irá a Espanha. Ao agente de viagens que irá identificar o nome e o destino de cada uma, elas deram as seguintes informações:

A loura: "Não vou à França nem à Espanha"

A morena: "Meu nome não é Elza nem Sara"

A ruiva: "Nem eu nem Elza vamos a França"

O agente de viagens concluiu acertadamente que:

- a) A loura é Sara e vai a Espanha
- b) A ruiva é Sara e vai a França
- c) A ruiva é Bete e vai a Espanha
- d) A morena é Bete e vai a Espanha
- e) A loura é Elza e vai a Alemanha
- 3) Depois de um assalto a um banco, quatro testemunhas deram quatro diferentes descrições do assaltante segundo quatro características, a saber: estatura, cor de olhos, tipo de cabelos e usar ou não bigode.

Testemunha 1: "Ele é alto, olhos verdes, cabelos crespos e usa bigode."

Testemunha 2: "Ele é baixo, olhos azuis, cabelos crespos e usa bigode."

Testemunha 3: "Ele é de estatura mediana, olhos castanhos, cabelos lisos e usa bigode."

Testemunha 4: "Ele é alto, olhos negros, cabelos crespos e não usa bigode."

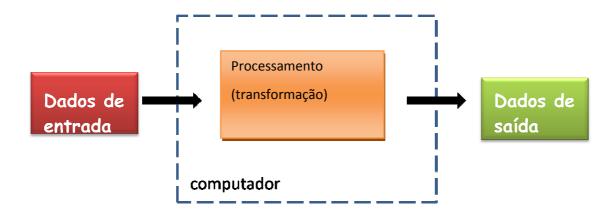
Cada testemunha descreveu corretamente <u>uma e apenas uma</u> das características do assaltante, e cada característica foi corretamente descrita por uma das testemunhas.

Assim, o assaltante é:

- a) baixo, olhos azuis, cabelos lisos e usa bigode.
- b) alto, olhos azuis, cabelos lisos e usa bigode.
- c) baixo, olhos verdes, cabelos lisos e não usa bigode.
- d) estatura mediana, olhos verdes, cabelos crespos e não usa bigode.
- e) estatura mediana, olhos negros, cabelos crespos e não usa bigode.

2. Algoritmo

a) <u>Resolução de Problemas pelo Computador</u>



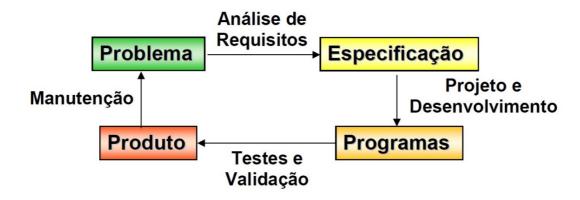
O computador é uma ferramenta que permite a realização do processamento de dados.

- b) Passos para resolução de problemas:
- 1. Entendimento do Problema
- 2. Criação de uma seqüência de operações para solução do problema
- 3. Execução desta seqüência
- 4. Verificação da adequação da solução

O computador desempenha apenas uma parte deste processo (3º passo).

c) Fases de Desenvolvimento de Sistemas

O processo de desenvolvimento de sistemas de programação é dividido em 4 fases:



c.1) Análise e Especificação de Requisitos

Um sistema de programação deve satisfazer as necessidades de seus usuários, as quais são expressas na forma de requisitos.

Requisito = ação que deve ser executada pelo sistema. (Ex: registrar as notas dos alunos, calcular a média final, etc.)

O levantamento destes requisitos e o seu refinamento (detalhamento) devem ser realizados junto com o usuário e registrado em um documento.

O sucesso do sistema depende de 3 fatores:

- ✓ Quão bem o sistema captou os requisitos expressos;
- ✓ Quão bem os requisitos captaram as necessidades;
- ✓ Quão bem as necessidades refletem a realidade.

c.2) Projeto e Desenvolvimento do Sistema

■ A partir do documento de análise de requisitos, projeta-se o sistema de programação:



Este processo é dividido em 3 etapas:

- ✓ Projeto Preliminar: definição da estrutura modular do software, as interfaces e as estruturas de dados utilizadas;
- ✓ Projeto Detalhado: descrição detalhada de cada módulo definido no projeto preliminar (algoritmo);
- ✓ Codificação: migração das instruções do algoritmo para uma linguagem de programação previamente definida (programas).

c.3) Teste e Validação

- Tem por objetivo garantir que o sistema satisfaça os requisitos expressos.
- Consiste da realização de alguns tipos de testes com o intuito de encontrar erros.

A inexistência de erros não representa a adequação operacional do sistema.

d) Conceito de Algoritmo

Um algoritmo pode ser definido como uma sequência finita de passos (instruções) para resolver um determinado problema. Sempre que desenvolvemos um algoritmo estamos estabelecendo um padrão de comportamento que deverá ser seguido (uma norma de execução de ações) para alcançar o resultado de um problema.

d.1) Propriedades do Algoritmo

- Composto por ações simples e bem definidas (não pode haver ambigüidade, ou seja, cada instrução representa uma ação que deve ser entendida e realizada).
- Seqüência ordenada de ações
- Conjunto finito de passos

d.2) Definições:

- Cada linha do algoritmo podemos chamar de uma instrução, logo, podemos dizer que um algoritmo é um conjunto de instruções.
- Instrução: indica a um computador uma ação elementar a ser executada

Exemplos:

Alg 1) Troca de pneu do carro.

1: desligar o carro

2: pegar as ferramentas (chave e macaco)

3: pegar o estepe

4: suspender o carro com o macaco

5: desenroscar os 4 parafusos do pneu furado

6: retirar o pneu

7: colocar o estepe

8: enroscar os 4 parafusos

9: baixar o carro com o macaco

10: guardar as ferramentas

Alg 2) Trocar lâmpadas

Se (lâmpada estiver fora do alcance)
Pegar escada;
Pegar a lâmpada;
Se (lâmpada estiver quente)

pegar pano;

tirar a lâmpada queimada;

Colocar lâmpada boa;

Alg 3) Bolo simples

- 1. Pegar os ingredientes;
- 2. Se (roupa branca)

colocar avental;

3. Se (tiver batedeira)

bater os ingredientes na batedeira; senão

bater os ingredientes à mão;

- 4. Colocar a massa na forma;
- 5. Colocar a forma no forno;
- 6. Aguardar tempo necessário;
- 7. Retirar o bolo;

Alg 4) Pegar um ônibus

- 1. ir até a parada;
- 2: enquanto ônibus não chega faça;

esperar ônibus

- 3: fim-enquanto
- 4: subir no ônibus
- 5: pegar passagem
- 6: se não há passagem então

pegar dinheiro

- 7: fim-se
- 8: pagar o cobrador
- 9: troco = dinheiro passagem
- 10: enquanto banco não está vazio faça

ir para o próximo

- 11: fim-enquanto
- 12: sentar
- 13::::

e) Partes de Um Algoritmo

Um algoritmo quando programado num computador é constituído pelo menos uma das 3 partes, sendo elas:

- 1. Entrada de dados;
- 2. Processamento de dados;
- 3. Saída de dados;

Entrada de dados

Na parte de entrada, são fornecidas as informações necessárias para que o algoritmo possa ser executado. Estas informações podem ser fornecidas no momento em que o programa está sendo executado ou podem estar embutidas dentro do mesmo.

Normalmente a entrada dedados é feita pelo teclado ou como nos caixa eletrônico por tela sensível ao toque.

Processamento de dados

Na parte do processamento são avaliadas todas as expressões algébricas, relacionais e lógicas, assim como todas as estruturas de controle existentes no algoritmo (condição e/ou repetição).

Saída de dados

Na parte de saída, todos os resultados do processamento (ou parte deles) são enviados para um ou mais dispositivos de saída, como: monitor, impressora, ou até mesmo a própria memória do computador.

Exemplo:

Alg 5) Calcular Área de uma Circunferência.

Fórmula matemática: $A = \P * R^2$

1-) Entrada de dados

O valor de ¶ e o raio R da circunferência

2-) Processamento de dados

 $A \leftarrow \P * R^2$

3-) Saída de dados

Escreva, imprima A

Explicando Ala 5

Objetivo: calcular a área de uma circunferência dada por: $A = \P^*R^2$.

Para calcular a área é necessário saber os valores do raio R e do \underline{pi} (\P).

Considerando que o valor de \P é constante o mesmo poderá ser gravado (definido) dentro do próprio algoritmo pelo programador, e a entrada para o processamento desse algoritmo consistirá nesse valor de \P juntamente com o valor do raio R (que deverá ser informado pelo usuário via teclado, por exemplo).

O processamento do algoritmo será a realização do cálculo $\P R^2$ e a atribuição do resultado dessa expressão para a variável \underline{A} .

A parte da saída consistirá na escrita do valor de A no monitor

3. Algoritmos Computacionais

a) Algoritmo - revisão

Conjunto de regras e operações bem definidas e ordenadas, destinadas à solução de um problema ou de uma classe de problemas, em um número finito de etapas.

O algoritmo representa o caminho de solução para um problema.

b) Algoritmo - resumo

- Um algoritmo representa uma seqüência de regras.
- Essas regras devem ser executadas em ordem preestabelecida.
- Cada algoritmo possui um conjunto finito de regras.
- Essas regras devem possuir um significado e ser formalizadas segundo algumas convenções, para que todas as pessoas envolvidas na sua criação possam entendê-lo da mesma forma.

c) Algoritmos Computacionais

- Diferem dos algoritmos gerais por serem executados pelo computador.
- Diferem dos programas por serem desenvolvidos em linguagens NÃO reconhecidas pelo computador. Ex: pseudolinguagem, fluxograma, etc...
- Auxiliam o usuário na concepção da solução de um problema, independentemente da linguagem de programação que será utilizada.

Limitações:

 Os algoritmos computacionais devem ser expressos nos termos do conjunto de instruções entendidas pelo computador.

d) <u>Diretrizes para a Elaboração de Algoritmos</u>

- <u>Identificação do Problema:</u> determinar o que se quer resolver ou qual objetivo a ser atingido;
- <u>Identificação das "entradas do sistema</u>": quais informações estarão disponíveis (serão fornecidas);
- Identificação das "saídas do sistema": quais informações deverão ser geradas/calculadas como resultado;

<u>Definir os passos a serem realizados</u>: determinar a seqüências de ações que leve à solução do problema (transforme as entradas nas saídas):

- Identificar as regras e limitações do problema;
- Identificar as limitações do computador;
- Determinar as ações possíveis de serem realizadas pelo computador.

<u>Concepção do algoritmo</u>: registrar a seqüência de comandos, utilizando uma das formas de representação de algoritmos.

<u>Teste da solução</u>: execução manual de cada passo do algoritmo, seguindo o fluxo estabelecido, para detectar possíveis erros.

Exemplo:

Exemplo: Calcular a média final dos alunos, sendo que foram realizadas 4 provas de mesmo peso no período.

- Quais os dados de entrada? P1, P2, P3 e P4
- Quais os dados de saída? Média
- Quais os passos a serem realizados? *Obter notas, calcular média (somar notas e dividir o resultado por 4) e apresentar média*

e) Formas de Representação de Algoritmos

A descrição de um algoritmo de forma clara e fácil de ser seguida ajuda no seu desenvolvimento, depuração (localização e correção de erros) e futura migração para uma linguagem de programação.

Para facilitar este trabalho, são utilizadas ferramentas específicas de representação da lógica de programação (seqüência de ações a serem realizadas) que são: *Descrição Narrativa (portugol); Fluxograma; Diagramas de Chapin ou Nassi Shneiderman; Pseudolinguagem*

f.1) Descrição Narrativa

Especificação verbal dos passos em linguagem natural.

Desvantagens:

- A linguagem natural é imprecisa (possibilita ambigüidades).
- Proporciona maior trabalho na codificação.

Sugere-se sua utilização apenas para comentar algoritmos e/ou programas, esclarecendo ou realçando pontos específicos.

f.2) Fluxograma

"Uma figura vale mais que mil palavras" -(atribuída a Fred R. Barnard – executivo de vendas em 1921 -que disse ser um provérbio chinês)

Um fluxograma é a representação gráfica da definição, análise ou solução de um problema onde símbolos são utilizados para representar operações, dados, fluxos, equipamentos etc..

Fluxograma é uma ferramenta originalmente criada na indústria de computadores para mostrar os diversos passos envolvidos em um programa. Provavelmente é o método mais antigo criado para representar uma seqüência de atividades ou comandos.

É constituído por um diagrama que utiliza figuras geométricas (quadrados, retângulos, losangos, círculos etc.) conectados por linhas e flechas, onde cada figura representa um passo no programa e as flechas representam a sequência em que os passos ocorrem.

A utilização de fluxograma apresenta algumas vantagens:

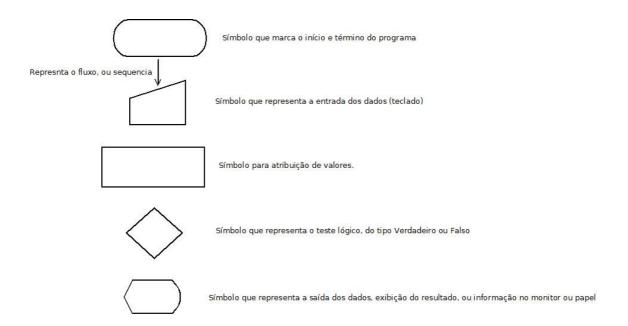
- São fáceis de fazer e compreender (melhor que um texto descritivo).
- Demonstram claramente o fluxo do programa ou atividade usando símbolos padronizados.
- Facilmente atualizados ou modificados.

A quantidade de informação que se coloca no fluxograma é suficiente para a compreensão do comando, instrução ou processo.

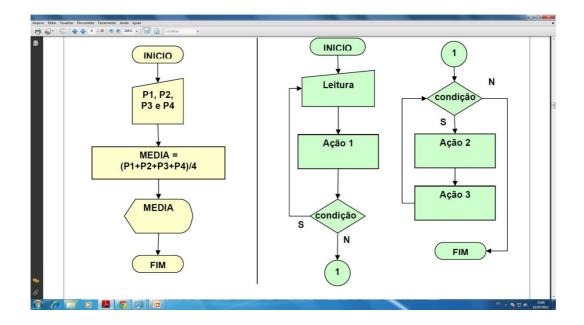
Desvantagens:

- Fluxogramas detalhados podem obscurecer a estrutura do programa.
- Permite transferências arbitrárias de controle

<u>Símbolos utilizados :</u>

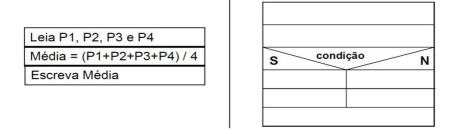


Exemplo:



f.3) Diagramas de Chapin ou Nassi_Shneiderman (Cartas N-S)

São ferramentas gráficas que representam a função principal de cada passo do algoritmo dentro de uma caixa.



f.4) Pseudolinguagem

- Linguagem especial para desenvolvimento de algoritmos, que utiliza expressões prédefinidas para representar ações e fluxos de controle.
- Funciona como uma linguagem simplificada de programação, logo, facilita a codificação futura.
- É uma descrição textual, estruturada e regida por regras; que descrevem os passos executados no algoritmo.
- Possui características similares às linguagens de programação:
 - ✓ Utiliza palavras-chaves (ex: escreva, se-então, etc.);
 - ✓ Identação (alinhamento dos blocos de comandos);
 - ✓ Possui um comando por linha;
 - ✓ Utiliza ";" como finalizador de comando;

Exemplo:

```
algoritmo_Media_Final

declare

inteiro P1, P2, P3, P4;

real media;

inicio

| leia(P1, P2, P3, P4);

media ← (P1+P2+P3+P4)/4;

escreva(media);

fim

fim_algoritmo
```

f) Regras para construção do Algoritmo

Para escrever um algoritmo precisamos descrever a seqüência de instruções, de maneira simples e objetiva. Para isso utilizaremos algumas técnicas:

- 1. Usar somente um verbo por frase
- 2. Imaginar que você está desenvolvendo um algoritmo para pessoas que não trabalham com informática
- 3. Usar frases curtas e simples
- 4. Ser objetivo
- 5. Evite usar palavras que tenham sentido dúbio
- 6. Procure dividir o problema em etapas menores.

g) Sintaxe do algoritmo

- A sintaxe de um algoritmo resume-se na regras para escrevê-lo corretamente
- Existem apenas três tipos de estruturas de programação que podem ser utilizadas para escrever qualquer programa: <u>Estrutura sequencial, de decisão, e de repetição.</u>
- As expressões que são escritas em estruturas de programação envolvem a utilização de dados. Os dados são números binários armazenados em sua memória

h.1) Estrutura de Dados (para manipulação das informações utilizadas no algoritmo.)

- O computador só conhece os valores que estão armazenados na sua memória.
- Portanto, a maioria das instruções está, de certa forma, associada ao processo de armazenamento ou transformação destes valores.

Na concepção de algoritmo, pode-se considerar:

- Memória = conjunto de posições;
- Cada posição recebe uma identificação (nome) e armazena um valor.

Abstração do Conceito de Memória

IDENTIFICADOR	IDADE	NOME	X1
VALOR	18	"João"	2.5

As posições de memória sempre armazenam um valor e seguem as seguintes premissas:

- Se armazenamos um novo valor em uma posição, o seu valor antigo será perdido;
- Se nenhum valor for atribuído a uma determinada posição, esta possui um "lixo" (as posições nunca estão vazias)

Identificador:

- Nome de uma posição da memória;
- É definido pelo programador;
- Recomenda-se o uso de nomes significativos.
- Exemplo: IDADE, NOME, VLR_SALARIO
- Contra-exemplo: X1, KCP, VAR_1, VAR_2

Regras para definição de identificadores:

- Deve começar com uma letra;
- Pode conter letras, números e o caracter especial "_".
- Exemplo: NOME, VLR_SALARIO, NOTA_1
- Contra-exemplo: 1ª NOTA, C&A, X-1

h.2) Constantes e Variáveis

Constantes: representam valores que não mudam no decorrer do algoritmo

Exemplo: "O nome é:", 24, PI, etc.

Variáveis: representam dados cujos valores são modificados ao longo da execução do algoritmo

h.3) Variável

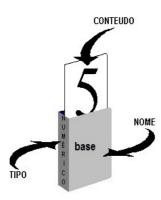
É possível manipular diversos tipos de dados em um algoritmo: <u>números inteiros e reais, valores</u> <u>lógicos, textos e etc...</u>

A manipulação destes dados é feita por meio de variáveis e valores constantes.

Uma variável é um local na memória principal, isto é, um endereço que armazena um conteúdo.

Em algoritmos, as <u>variáveis serão definidas no inicio</u>. O objetivo de se declarar uma variável no início do algoritmo é para que seja <u>alocada (reservada) uma área na memória para a variável.</u>

Uma variável pode ser vista como uma caixa com um rótulo ou nome colado a ela, que num dado instante guarda um determinado objeto. O conteúdo desta caixa não é algo fixo, permanente. Na verdade, essa caixa pode ter seu conteúdo alterado diversas vezes. Contudo, o conteúdo deve ser sempre do mesmo tipo.



h) <u>Tipos Primitivos de Dados</u>

A cada variável está associado um Tipo de Dados. O tipo de dados define quais os valores que a variável pode conter. Se, por exemplo, dissermos que uma variável é do tipo Inteiro, não poderemos lá colocar um valor Real ou um Caractere.

Antes de começarmos a utilizar uma variável, temos que declarar qual o seu tipo e nome. É feito do seguinte modo:

■ tipo <nome da variável>;

Ex:

float alt

Neste caso estamos a declarar que a variável *alt*, é do tipo *Tipo real ou float*. Se tivermos várias variáveis do <u>mesmo tipo</u> podemos junta-las numa só declaração:

- int h1, h2
- char sexo, nome

Ao declararmos o tipo de dados de uma variável, estamos a definir, não só, o tipo de valores que esta pode conter, mas também quais as operações que com elas podemos realizar.

Tipo de dados:

- Numéricos:
 - ✓ Inteiro (ex: 1, -5, 100);

✓ Real ou float (ex: 1.3, -3.5, 0.55).

Não-Numéricos:

- ✓ Booleano (lógico ex: True ou False);
- ✓ Caracter (alfanumérico ex: "A", '@', "x1").

Neste tipo, utilizamos "" como delimitadores de conteúdo. ('')

i) Resumo

- Algoritmo: Conjunto de regras e operações bem definidas e ordenadas, destinadas à solução de um problema
- Formas de representação: *Descrição Narrativa (portugol); <u>Fluxograma</u>; Diagramas de Chapin ou Nassi_Shneiderman; <u>Pseudolinguagem</u>.*
- Existem apenas três tipos de estruturas de programação que podem ser utilizadas para escrever qualquer programa: <u>Estrutura sequencial, de decisão, e de repetição.</u>
- Uma variável é um local na memória principal, isto é, um endereço que armazena um conteúdo.
- Em algoritmos, as <u>variáveis serão definidas no inicio</u>
- Variáveis podem ser Numéricas (inteiro ou real) ou Não numéricas

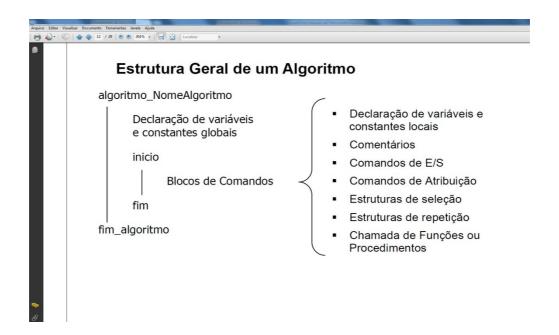
4. Estruturas de Controle: Sequencial

a) Estrutura Sequencial

- Algoritmos com comando seqüencial é aquele que, a partir do símbolo <u>Início</u>, permite a execução das instruções contidas nos símbolos subseqüentes sem desvio algum na direção até se alcançar o símbolo <u>FIM</u>.
- Conjunto de comandos que são executados numa seqüência linear, de cima para baixo, na mesma ordem que aparecem.

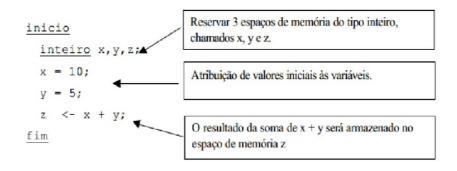
Sintaxe: comando1; comando2;; comandoN;

a.1) Estrutura Algoritmo Sequencial em <u>Pseudolinguagem</u>



Exemplos

a) Desenvolva um algoritmo em portugol para somar dois valores inteiros (10 + 5)



Notas: Todas as variáveis globais são declaradas no início do programa. Nenhuma variável pode ser usada antes de ser declarada

Obs:

- O inconveniente deste algoritmo é que sempre fornecerá o mesmo resultado, o que não é interessante. A maneira mais correta, seria que os <u>valores de x e y fossem fornecidos</u> <u>pelo usuário</u>, permitindo ao algoritmo que efetue a soma de dois números quaisquer.
- Neste exemplo podemos observar que o resultado da soma de x + y será armazenado em z. Como o usuário ficará sabendo da resposta ?
- É necessário usar comandos de escrita, para apresentar os resultados.
- Caso queira somar outros numerais diferentes de 10 e 5 preciso solicitar ao usuário os valores e armazená-los na memória do computador.

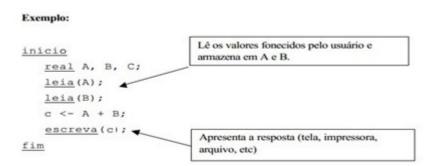
Comandos de entrada e saída de dados

Na construção de algoritmos, é conveniente que o usuário possa informar dados externos, para serem operados pelo programa. Assim, um programa pode receber um dado informado por um operador através de um comando de leitura.

Da mesma forma, pode ser necessário conhecer o resultado de determinada operação executada pelo computador, então será necessária uma forma de exibir os dados.

Cada linguagem tem uma forma específica para entrada e saída de dados.

Em algoritmos usaremos os comandos genéricos *leia()* e escreva(), para realizar a interface com o usuário.



Pergunta?

✓ Se os valores de A e B serão fornecidos pelo usuário via teclado por exemplo, como ele saberá o momento de digitar o valor de A e o valor de B?

Quando vamos a um caixa eletrônico para retirar dinheiro, após inserir o cartão com os dados do número da conta e agência e selecionarmos a opção "saque", qual o próximo passo?

Resp. Aparecerá na tela uma mensagem:

"Entre com sua senha"

Portanto para o usuário saber o momento de digitar um valor é necessário primeiramente uma mensagem solicitando o dado.

Ex: Fazer um programa para LER um número e no final, MOSTRAR esse número na tela.

- Uma das coisas que precisamos analisar no enunciado são os VERBOS. Observe que os verbos do exercício estão EM NEGRITO, para facilitar a explicação.
- Quando queremos fazer um programa para LER um número, significa que esse número será obtido por meio da leitura de dados VIA TECLADO, logo, precisamos fazer uso de um comando que obtém os dados via teclado. No algoritmo, o comando de leitura de dados via teclado é conhecido como LEIA.
- Outro detalhe: Observe que foi pedido para se ler um número, mas não foi especificado qual tipo do número a ler lido, se é número inteiro ou fracionário, logo, qualquer tipo de número lido é valido para este exercício.
- Ao lermos o exercício acima , observe que após a leitura do número pede-se para MOSTRAR o número lido na tela. Quando usamos o verbo MOSTRAR, significa que devemos exibir esse número lido na tela, por meio da instrução ESCREVA.

```
Algoritmo_numero
int num
inicio
escreva("Digite um número")
leia(num)
escreva("O numero digitado foi:", num)
```

- ✓ Observe que foi utilizado o tipo INTEIRO para a variável "num" mas, se quisesse, poderia utilizar o tipo REAL, para tipos fracionários, que o programa TAMBÉM estaria correto.
- ✓ Quando queremos exibir uma mensagem na tela ou impressora ela deve estar contida entre aspas dupla, caso contrário, o computador irá identificar a mensagem como Variável Indefinida.

A tela do computador após o programa ser executado:

Fim

```
"G:\pessoal\Int programap\( \text{O} \) \code\ler-num\bin\Debug\ler-num.exe"

Digite um numero: 6
0 numero digitado foi: 6

Process returned 0 (0x0) execution time: 3.214 s

Press any key to continue.
```

Observe a última linha do algoritmo:

✓ escreva("O numero digitado foi:", num)

O texto que queremos apresentar na tela está entre aspas. Além do texto queremos que apresente também a resposta do problema.

Para isto precisamos indicar que é para buscar na variável (num) o valor que está armazenado quando utilizamos o comando leia. Ou seja o comando leia guarda na memória um valor e para apresentá-lo além do comando escreva precisamos indicar onde está guardado esta número.

Ex: Construa um algoritmo (pseudo-código) para calcular a força exercida pela coluna de um líquido sobre a área da válvula de um reservatório. Conhece-se a altura h (m) do reservatório, o diâmetro d (m) da válvula e o peso específico gama do líquido (N/m3). A força F em Newtons é o peso da coluna do líquido e é calculada pela expressão:

$$F = pi * gama * \sqrt{d} * h_{\Delta}$$

✓ Análise

Objetivo: Calcular a força F

Dados de entrada: Os valores h, d e gama e pi= 3.14159265358

Dados de saída: O valor da força F

Fórmula matemática: $F = pi * gama * \sqrt{d} * h/4$

Restrições (limitações): nenhuma

Seqüência: Obter o valor de h, d, gama e pi.Calcular a Força F e apresentar o valor de F

Obs: Os valores de h,d, gama e pi serão fornecidos pelo usuário. O valor de pi é um valor constante definido pelo programador.

Resumo:

$$F = pi * gama * \sqrt{d} * \frac{h}{4}$$

- 1. Ler os valores h, d e gama
- 2. pi= 3.14159265358
- 3. Calcular F
- 4. Exibir o valor de F
 - Algoritmo representado em Pseudolinguagem:

```
Algoritmo_força_liqudido

real pi, gama, d, h, F;

inicio

escreva ("entre com o peso específico em N/m3 : ");

leia (gama);

escreva ("entre com o diâmetro do reservatório em metros: ");

leia (d);

escreva ("entre com a altura do reservatório em metros : ");

leia (h);

pi ← 3.14159265;

/* agora que temos todos os valores utilizados na expressão matemática para o

cálculo da força armazenados na memória podemos calcular a força. */

F ← pi * gama * sqrt(d) * (h/4);

escreva ("O peso da coluna do líquido = ", F);

Fim
```

✓ Comentários

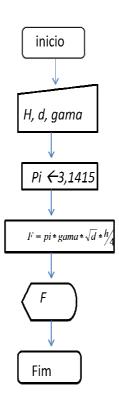
Na verdade, comentários não são comandos, mas são muito importantes para a documentação e entendimento dos algoritmos e programas de computador.

São compostos por textos explicativos em linguagem natural delimitados entre /* e */ (ou // textos)

Exemplo: /* Este algoritmo calcula a média aritmética das notas do semestre */

a.2) Estrutura seqüencial representada por Fluxograma

Os fluxogramas são uma apresentação do algoritmo em formato gráfico. Cada ação ou situação é representada por uma caixa. Tomadas de decisões são indicadas por caixas especiais, possibilitando ao fluxo de ações tomar caminhos distintos.



Exercícios resolvidos:

1) Elabore um algoritmo e represente-o na forma de fluxograma e pseudolinguagem, que calcule o alcance de um projétil, dada a velocidade inicial v_o e o angulo θ entre o cano do canhão e o solo. A formula a ser utilizada é:

$$S = \frac{vo^2}{g} * sen(2\theta)$$

✓ Análise:

Objetivo: Calcular o alcance de um projétil

Dados entrada:			
Usuário Velocidade inicial v_o e o angulo θ entre o cano do canhão e o solo			
Programador	Força da gravidade <i>g</i> = 9.18		

Dados de saída: Alcance do projétil (AI)

FM:
$$S = \frac{vo^2}{g} * sen(2\theta)$$

Restrições: nenhuma

Algoritmo_Alcance_projétil

real v, g, teta, S;

inicio

escreva ("entre com a velocidade inicial:");

leia (**v₀**);

escreva ("entre com o ângulo θ :");

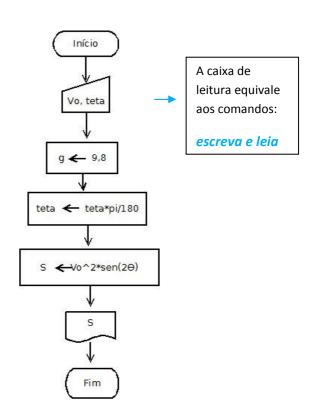
leia (teta);

g ← 9.8;

 $(v_0^2)/g * sen (2 * teta);$

escreva ("O alcance do projétil = ", \$);

FIM



2) Elabore um fluxograma que transforme uma temperatura fornecida em graus Celsius para a correspondente em graus fahrenheit . A fórmula de conversão de ${}^{o}F$ para ${}^{o}C$ é:

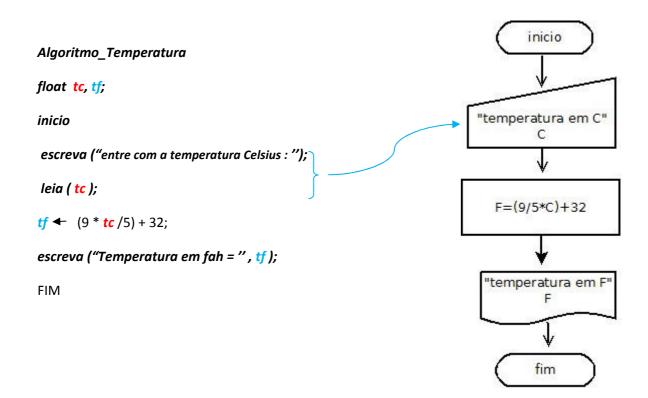
$$\frac{C}{5} = \frac{F - 32}{9}$$

✓ Análise:

Objetivo: Transformar temperatura de graus Celsius para graus Fahrenheit.

Dados de entrada: Temperatura em Celsius Dados de saída: Temperatura em Fahrenheit

FM: $F = \left(\frac{9*C}{5}\right) + 32$ Restrições: nenhuma



3) Entrar com dois números inteiros e exibir a seguinte saída:

dividendo: divisor: quociente: resto:

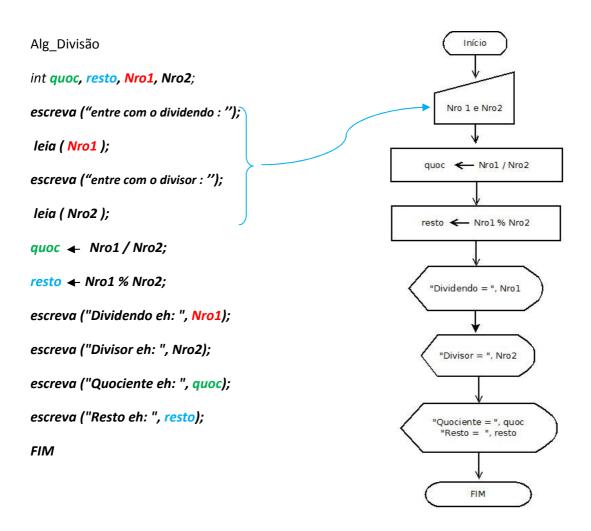
✓ Análise:

Objetivo: Mostrar o divisor, dividendo, quociente e resto da divisão de dois números

Dados de entrada: Número 1 e número 2

Dados de saída: dividendo, divisor, quociente e o resto

(O símbolo "%" ou a palavra "mod" significam o resto da divisão de dois numerais inteiros)



4) Entrar com nome e idade e exibir a seguinte saída:Seu Nome:Sua Idade:

✓ Análise:

Objetivo: Imprimir nome e idade

Dados de entrada: Nome e Idade

Dados de saída: Nome e Idade

FM: nenhuma

Restrições: nenhuma

```
Alg_cadastro

char nome;

int id;

escreva ("Digite o Nome:");

leia (nome);

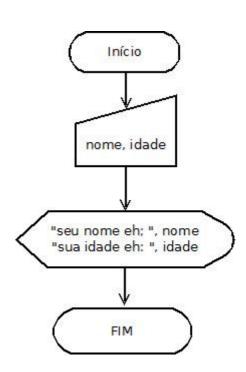
escreva ("Digite a Idade:");

leia (id);

escreva ("Seu nome: ", nome);

escreva ("Sua Idade: ", id);

FIM
```



- 1. Represente em pseudolinguagem o algoritmo que faça: ler nome, endereço e telefone e exibi-los.
- 2. Criar um algoritmo para ler dois números inteiros e imprimir a soma. Antes do resultado deverá aparecer a mensagem: "Soma= ". Representá-lo em fluxograma e pseudolinguagem.
- 3. Ler dois números inteiros e exibir o produto. Representá-lo em pseudolinguagem.
- 4. Entrar com quatro números e imprimir a média ponderada, sabendo-se que os pesos são respectivamente: 1, 2 , 3 e 4. Representá-lo em fluxograma e pseudolinguagem.
- 5. Fazer um algoritmo que possa entrar com o saldo de uma aplicação e exibir o novo saldo, considerando o reajuste de 1%. Representá-lo em fluxograma e pseudolinguagem.
- 6. Entrar coma base e a altura de um retângulo e exibir a seguinte saída:

perímetro:

área:

diagonal:

Obs: Representá-lo em pseudolinguagem.

- 7. Criar um algoritmo que efetue o calculo do salário liquido de um professor. Os dados a fornecer serão: valor da hora aula, numero de aulas dadas no mês e percentual de descontos do INSS. Representá-lo em fluxograma e pseudolinguagem.
- 8. Calcular e apresentar o volume de uma lata de óleo. Representá-lo em fluxograma e pseudolinguagem.
- 9. Ler dois números reais e imprimir o quadrado da diferença do primeiro valor pelo segundo e a diferença dos quadrados. Representá-lo em pseudolinguagem.
- Entrar com quatro números e quatro pesos e calcular e exibir a média ponderada.
 Representá-lo em fluxograma e pseudolinguagem.
- 11. Elabore um fluxograma que calcule quantas notas de 50, 10, 1 são necessárias para se pagar uma conta cujo valor é fornecido3 pelo usuário.

5. Estrutura de Controle: Decisão

a) Estrutura de Decisão

Conceito:

- É uma estrutura de controle de fluxo, executando um ou vários comandos se a condição testada for verdadeira e, em alguns casos, executando um ou vários comandos se for falsa.
- Permite a **escolha de um grupo de ações** para serem executadas de acordo com a aceitação ou não de certas condições.
- As condições que são testadas são do **tipo lógico** (só podem assumir 2 valores: **Verdadeiro** ou **Falso**).

b) Comando SE- Pseudolinguagem:

Executa um bloco de comandos específico se a condição testada for verdadeira e um outro bloco de comandos quando a condição for falsa.

```
Sintaxe: se <condição> então

bloco de comandos (condição Verdadeira);

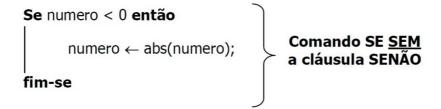
senão

bloco de comandos (condição Falsa);

fim-se

OBS: a cláusula SENÃO é opcional.
```

> Exemplos:



```
se cadastro[1].sexo = "F" então

escreva("Sra. ", cadastro[1].nome);

senão

escreva("Sr. ", cadastro[1].nome);

fim-se
```

c) Ninhos de SE:

Dentro de um comando de seleção podemos colocar qualquer tipo de comando, inclusive outros comandos de seleção.

A função SE Aninhada é um aperfeiçoamento da Função SE Simples. Permite a nós analisarmos mais do que duas condições para um valor ou elemento.

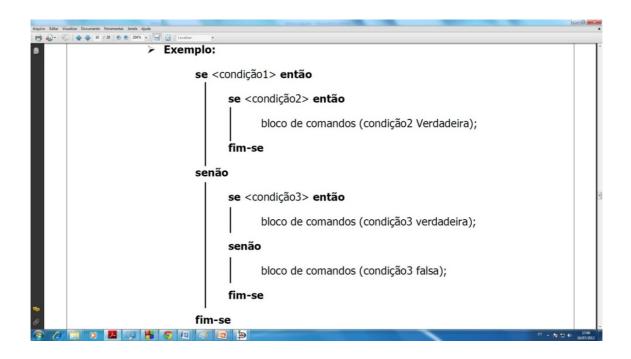
O caso típico da utilização do *SE Aninhada* é quando não temos apenas <u>uma</u> opção de resposta FALSA, ou seja varias opções. Ex:

Se Nota Final >= 90 Conceito ótimo

Se Nota Final >= 80 Conceito BOM

Se Nota Final >= 70 Conceito Regular

SE Nota Final <= 60 Conceito Ruim



d) A execução do comando SE

- Primeiro a condição é avaliada pela Unidade Aritmética e Lógica, sendo a condição uma expressão que possa retornar um valor lógico (verdadeiro ou falso).
- Se o valor retornado for <u>verdadeiro</u>, todos os comandos que se encontram entre o primeiro par de chaves serão executados. Após a execução dos comandos, o fluxo do algoritmo passa para o primeiro comando depois do fechamento do bloco do <u>senão</u>, representado pelo símbolo "}" (se existir); caso contrário, será executado o primeiro comando depois de "}" do próprio bloco.
- Se o valor retornado for falso, todos os comandos que se encontram entre o primeiro par de chaves serão ignorados e, se existir o comando <u>se-não</u>, serão executados os comandos que se encontram entre o segundo par de chaves, caso contrario, nada acontecerá e o fluxo do algoritmo seguirá para o próximo comando.

A Condição pode ser:

- Expressão relacional
- 1. A > B lê-se: o conteúdo da variável A é maior que o conteúdo da variável B?
- 2. A < 13 lê-se: o conteúdo da variável A é menor do que 13?
- 3. resp <>S lê-se: o conteúdo da variável resp é diferente da letra S?
- 4. resp=="Brasil" lê-se: o conteúdo da variável resp é igual a Brasil?
- Expressão lógica
- 1. A>=1 && A<9

Lê-se: O conteúdo da variável A é maior ou igual a 1 e menor que 9?

2. resp =="S" | resp == "s"

Lê-se: o conteúdo da variável resp é igual a S ou igual a s?

e) Operadores Aritméticos

Operadores	Usaremos
Soma	+
Subtração	-
Multiplicação	*
Divisão	/
Resto da divisão	%
Exponenciação	۸

f) Operadores Relacionais

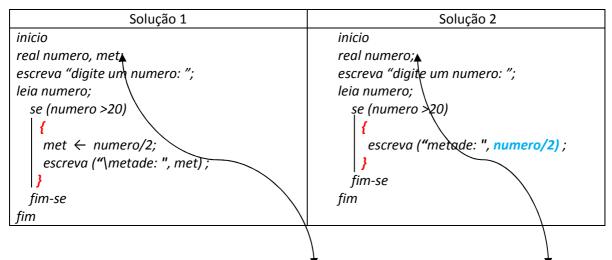
Operadores	Usaremos		
Igual	==		
Diferente	⇔		
Maior	>		
Menor que	<		
Maior ou igual a	>=		
Menor ou igual	<=		

g) Operadores Lógicos

Operadores	Usaremos
Conjunção "e"	&&
Disjunção "ou"	H
Negação "Não"	!

h) Exemplo:

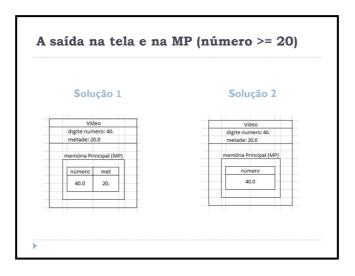
Ler um numero e se ele for maior do que 20, então imprimir a metade do numero

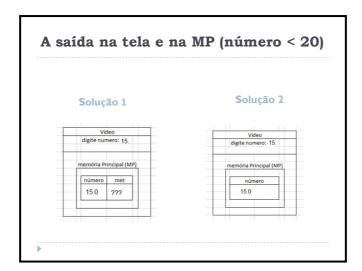


Observe as figuras abaixo onde são mostradas as diferenças das duas soluções. Note que na solução 1 foram criadas duas variáveis (*numero e met*) e na solução 2 uma <u>única</u> variável (*numero*), justamente para armazenar o número que o usuário digitará.

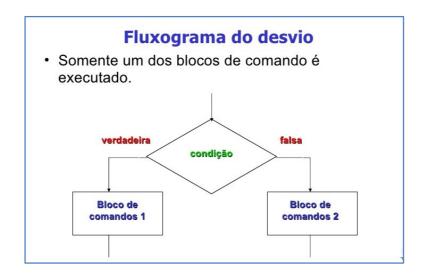
O comando escreva além de mostrar o conteúdo na tela aceita também cálculos matemáticos e outras operações lógicas, como no exemplo 2 (escreva "metade: ", numero/2;).

Portanto a única diferença destas soluções está na memória principal, onde na solução 1 reservamos dois espaços na memória e na solução 2 apenas 1.

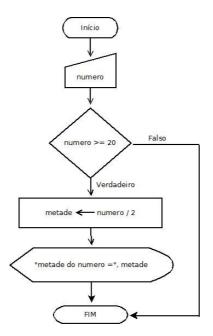




i) Comando SE- Fluxograma

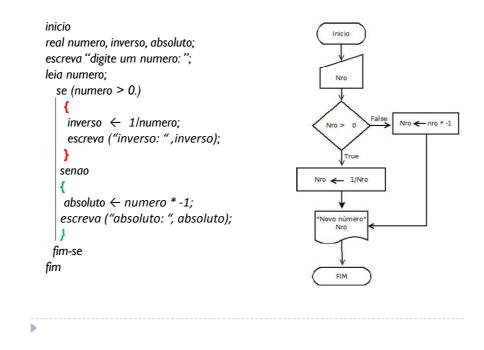


Exemplo: Ler um numero e se ele for maior do que 20, então imprimir a metade do numero

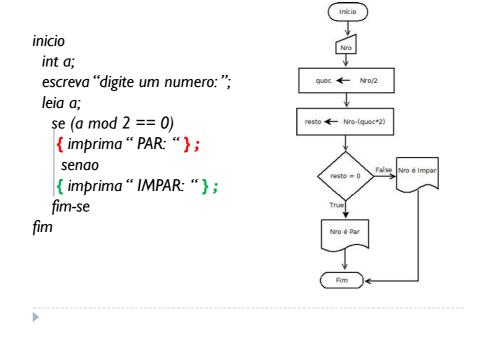


Exercícios Resolvidos

1) Ler um número e, se ele for positivo, imprimir seu inverso; caso contrario, imprimir o valor absoluto do número.



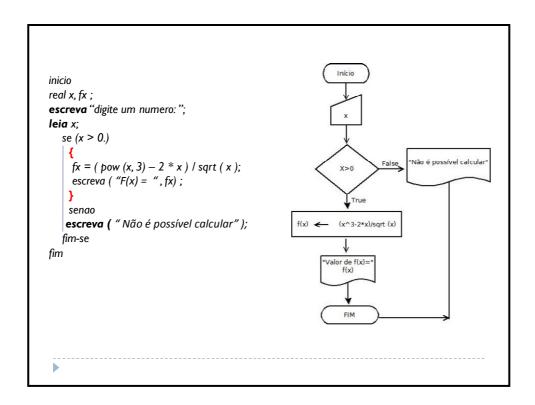
2) Ler um numero e imprimir se ele é par ou impar



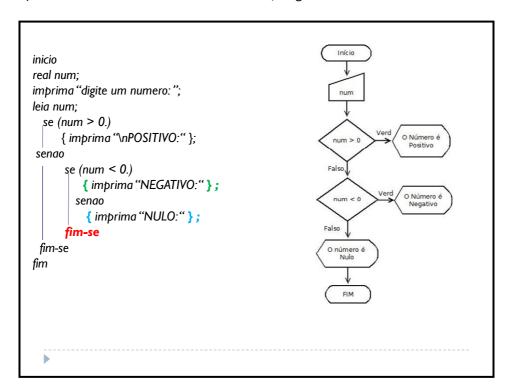
3) Criar um Algoritmo que entre com o valor de x, calcule e imprima o valor de f(x)

$$fx = (x^3 - 2 * x) / \sqrt{x}$$

Note: O denominador não pode ser igual a zero e não existe raiz negativa nos números reais.



- SES Aninhados (Encaixados)
- 4) Ler um número e exibir se ele é Positivo, Negativo ou Nulo.



Lista 3:

- 1. Entrar com um número e exibi-lo caso seja maior que 20
- 2. Construir um pseudocódigo que leia dois valores numéricos inteiros e efetue a adição: caso o resultado seja maior que 10, apresentá-lo
- 3. Entrar com um numero e informar se ele é divisível por 3 e 7
- 4. Entrar com o nome, sexo, idade de uma pessoa. Se a pessoa for do sexo feminino e tiver menos que 25 anos, imprimir nome e mensagem: *ACEITA*. Caso contrário, imprimir nome e a mensagem: *NÃO ACEITA*. (considerar f ou F minúsculo e maiúsculo)
- 5. Entrar com três números e armazenar o maior número na variável de nome maior
- 6. Criar um algoritmo que entre com o valor de x, calcule e exiba o valor de f(x). Representar utilizando fluxograma e pseudolinguagem.

$$f(x) = \frac{5x + 3}{\sqrt{x^2 - 16}}$$

- 7. Construir um algoritmo que leia dois números e efetue a adição. Caso o valor seja maior que 20, este deverá ser apresentado somando-se a ele mais 8; caso o valor somado seja menor ou igual a 20, este deverá ser apresentado subtraindo-se 5
- 8. Entrar com o ano de nascimento de uma pessoa e ano atual. Imprimir a idade da pessoa. Não se esqueça de verificar se o ano da nascimento é válido.
- 9. Dado um par de valores (x, y) que representam as coordenadas de um ponto no plano, determinar o quadrante ao qual pertence o ponto, ou se está sobre um dos eixos cartesianos. Representar o algoritmo na forma de fluxograma.
- 10. Reajustar o salário de acordo com a função. Se for técnico, aumentar o salário 50%, se for gerente, aumentar 30% e se for outro cargo, aumentar 20%. Representar o algoritmos na forma de fluxograma.
- 11. Faça um algoritmo que pergunte em que ano você nasceu e exiba quantos anos você tem. Exiba também se você é mais velho que 18 anos, se está entre 15 e 18 anos, ou se tem menos que 15 anos. Representar o algoritmos na forma de fluxograma.
- 12. Fazer um algoritmo que leia o percurso em quilômetros e o tipo do carro. Exiba na tela o consumo estimado de combustível, sabendo-se que um carro tipo *C* faz 12 Km com um litro de gasolina, um tipo *B* faz 10 Km e o tipo *A*, 8 Km por litro.

Obs:

- Testar a possibilidade de letra maiúscula e minúscula para o tipo do carro.
- Verificar se o usuário digitará outro caractere diferente de A, B ou C para o tipo de carro.

Obs: Nos exercícios acima que não foram especificados a forma de representação adotar como padrão a forma de pseudolinguagem.