

# Apostila de Algoritmos

Algoritmo não é a solução de um problema, pois, se assim fosse, cada problema teria um único algoritmo. Algoritmo é um caminho para a solução de um problema, e em geral, os caminhos que levam a uma solução são muitas.

O aprendizado de algoritmos não se consegue a não ser através de muitos exercícios.

Algoritmos não se aprende:

- Copiando Algoritmos
- Estudando Algoritmos

Algoritmos só se aprendem:

- Construindo Algoritmos
- Testando Algoritmos

## 1. Fases de um Algoritmo

Quando temos um problema e vamos utilizar um computador para resolvê-lo inevitavelmente temos que passar pelas seguintes etapas:

- Definir o problema.
- Realizar um estudo da situação atual e verificar quais a(s) forma(s) de resolver o problema.
- Terminada a fase de estudo, utilizar uma linguagem de programação para escrever o programa que deverá a princípio, resolver o problema.
- Analisar junto aos usuários se o problema foi resolvido. Se a solução não foi encontrada, deverá ser retornado para a fase de estudo para descobrir onde está a falha.

Estas são de forma bem geral, as etapas que um analista passa, desde a apresentação do problema até a sua efetiva solução. Iremos, neste curso, nos ater as etapas de estudo, também chamada de análise, e a etapa de programação. Mas antes vamos definir o seguinte conceito: Programar um computador consiste em elaborar um conjunto finito de instruções, reconhecidas pela máquina, de forma que o computador execute estas instruções. Estas instruções possuem regras e uma Sintaxe própria, como uma linguagem tipo português ou inglês, sendo isto chamadas de linguagem de computador.

No mundo computacional existe uma grande variedade de linguagens Pascal, C, C++, Cobol, Fortran, etc... . Nós iremos focar uma delas, o Pascal.

## 2. Estrutura de Algoritmos

Antes de utilizarmos uma linguagem de computador, é necessário organizar as ações a serem tomadas pela máquina de forma organizada e lógica, sem nos atermos as regras rígidas da Sintaxe de uma linguagem. Para isto utilizaremos uma forma de escrever tais ações, conhecida como algoritmo, ou pseudo-código.

Conceituando: Algoritmo consiste em um conjunto lógico e finito de ações ( instruções ) que resolvem um determinado problema.

Os algoritmos terão a seguinte estrutura:

```
ALGORITMO <Nome do algoritmo>  
<definições>  
INÍCIO  
<Comandos>  
FIM
```

Em Pascal ficaria da seguinte forma:

```
PROGRAM <Nome do programa>;  
<definições>  
BEGIN  
<Comandos>;  
END.
```

### 3. Variáveis

O computador possui uma área de armazenamento conhecida como memória. Todas as informações existentes no computador estão ou na memória primária ( memória RAM ), ou na memória secundária ( discos, fitas, CD-ROM etc ). Nós iremos trabalhar, neste curso, somente com a memória primária, especificamente com as informações armazenadas na RAM ( memória de acesso aleatório ).

A memória do computador pode ser entendida como uma seqüência finita de caixas, que num dado momento, guardam algum tipo de informação, como número, uma letra, uma palavra, uma frase etc, não importa, basta saber que lá sempre existe alguma informação. O computador, para poder trabalhar com alguma destas informações, precisa saber onde, na memória, o dado está localizado. Fisicamente, cada caixa, ou cada posição de memória, possui um endereço, ou seja, um número, que indica onde cada informação está localizada. este número é representado através da notação hexadecimal, tendo o tamanho de quatro, ou mais bytes. Abaixo segue alguns exemplos:

Endereço Físico	Informação
3000:	B712 'João'
2000:	12EC 12345
3000:	0004 'H'

Como pode ser observado, o endereçamento das posições de memória através de números hexadecimais é perfeitamente compreendido pela máquina, mas para nós humanos torna-se uma tarefa complicada. Pensando nisto, as linguagens de computador facilitaram o manuseio, por parte dos usuários, das posições de memória da máquina, permitindo que, ao invés de trabalhar diretamente com os números hexadecimais, fosse possível dar nomes diferentes a cada posição de memória. Tais nomes seriam de livre escolha do usuário. Com este recurso, os usuários ficaram livres dos endereços físicos ( números hexadecimais ) e passaram a trabalhar com endereços lógicos ( nomes dados pelos próprios usuários ). Desta forma, o Exemplo acima, poderia ser alterado para ter o seguinte aspecto:

Endereço Físico	Informação
Nome	'João'
número	12345
letra	'H'

Como tínhamos falado, os endereços lógicos são como caixas, que num dado instante guardam algum tipo de informação. Mas é importante saber que o conteúdo desta caixa não é algo fixo, permanente, na verdade, uma caixa pode conter diversas informações, ou seja, como no Exemplo acima, a caixa ( Endereço Lógico ) rotulada de "Nome" num dado momento contém a informação "João", mas em um outro momento, poderá conter uma outra informação, por Exemplo "Pedro".

Com isto queremos dizer que o conteúdo de uma destas caixas ( endereço lógico ) podem variar, isto é podem sofrer alterações em seu conteúdo. Tendo este conceito em mente, a partir de agora iremos chamar de forma genérica, as caixas ou endereços lógicos, de variáveis.

Desta forma podemos dizer que uma variável é uma posição de memória, representada por um Nome simbólico ( atribuído pelo usuário ), a qual contém, num dado instante, uma informação.

### 3.1 Formação de Variáveis

Uma variável é formado por uma letra ou então por uma letra seguida de letras ou dígitos, em qualquer número. Não é permitido o uso de espaços em branco ou de qualquer outro caractere, que não seja letra ou dígito, na formação de um identificador.

Na formação do nome da variável de um nome significativo, para que se possa ter idéia do seu conteúdo sem abri-lá. Se utilizar palavras para compor o nome da variável utilize o " \_ " underline para separar as palavras.

### 3.2 Conteúdo de uma Variável

Dados - Elementos conhecidos de um problema

Informação - Um conjunto estruturado de dados, transmitido conhecimento

### 3.3 Tipos de Variáveis

Considere a fórmula matemática simples do calculo do volume de uma esfera:

onde se encontram:

1- valores que podem ser classificados como:

a) valores constantes, invariantes em todas as aplicações da fórmula, no caso dos valores 4, 3 e  $\pi$  aos denomina-se constantes;

b) valores a serem substituídos na fórmula, em cada aplicação; a representação destes valores, usualmente é feita através de letras, que recebem o nome de variáveis e tornam a fórmula genérica, possível de ser aplicada para resolver uma certa classe de problemas e não apenas um problema específico.

2 - Operações a serem feitas sobre determinados operandos (Valores), para a obtenção da solução do problema.

## 4. Tipos de Dados

O Pascal exige que no momento que formos trabalhar com variáveis, indiquemos o tipo de informação que uma variável pode conter, isto é, se uma dada posição de memória armazenará um número ou uma letra etc. Para isto, a linguagem Pascal possui definido alguns tipos que deverão ser usados quando formos trabalhar com variáveis. Alguns tipos são:

Algoritmo Pascal Descrição

a) Inteiro a) INTEGER: Representa números entre -32768 até +32767. ocupa 2 bytes na memória.

b) Real b) REAL: Representa os números entre  $2.9 \times 10^{-39}$  até  $1.7 \times 10^{38}$  . Ocupa 6 bytes na memória.

e) Carácter e) CHAR: Representa um dos caracteres, da tabela ASCII. Ocupa 1 byte na memória.

d) Cadeia f) STRING: Conjunto de caracteres ( CHAR ). Ocupa de 1 a 255 bytes na memória.

g) Lógica g) BOOLEAN: Valor lógico. Assuma somente dois valores: TRUE(Verdade) ou FALSE(Falso). ocupa 1 byte na memória.

c) WORD: Números de 0 até 65535. Ocupa 2 bytes na memória.

d) BYTE: Números de 0 até 255. Ocupa 1 byte na memória.

h) Short Int Representa os números entre -128 até 128 . Ocupa 1 bytes na memória.

i) LongInt Representa os números entre - 2.147.483.648 até 2.147.483.648 . Ocupa 4 bytes na memória.

- j) Single Representa os números entre  $1.5 \times 10^{-45}$  até  $3.4 \times 10^{38}$  . Ocupa 4 bytes na memória.
- l) Double Representa os números entre  $5 \times 10^{-324}$  até  $1.7 \times 10^{308}$  . Ocupa 8 bytes na memória.

Iremos gradativamente trabalhar com cada um destes tipos e mostrar as suas características.

## 5. Definição de Variáveis

Para definir uma ou mais variáveis, devemos utilizar, em Pascal, a palavra VAR. Para manter uma compatibilidade, utilizaremos a mesma nomenclatura em nossos algoritmos. Abaixo segue Exemplo de algumas definições de variáveis:

```
a) ALGORITMO Teste PROGRAM Teste;
VARIÁVEIS VAR
Palavra : CADEIA Palavra: String;
INICIO BEGIN
<comandos> <comandos>;
FIM END.
```

```
b) ALGORITMO Teste PROGRAM Teste;
VARIÁVEIS VAR
Letra, Carácter: CHARACTER Letra, Carácter: CHAR;
INICIO BEGIN
<comandos> <comandos>;
FIM END.
```

```
c) ALGORITMO Teste PROGRAM Teste;
VARIÁVEIS VAR
Letra, Carácter: CHARACTER Letra, Carácter: CHAR;
Número: INTEIRO Número: INTEGER;
INICIO BEGIN
<comandos> <comandos>;
FIM END.
```

Obs.: Os nomes dados as variáveis não podem ser os mesmos nomes de palavras reservadas do Pascal, tais como PROGRAM, BEGIN, END, VER, etc...

### 5.1 Comando de atribuição

Quando definimos uma variável é natural atribuímos a ela uma informação. Uma das formas de colocar um valor dentro de uma variável, conseqüentemente colocado este dado na memória do computador, é através da atribuição direta, do valor desejado que a variável armazena. Para isto utilizaremos o símbolo ( := (Pascal) , (Algoritmo) ), que significa: recebe, ou seja, a posição, de memória que uma variável representa, receberá uma informação, a qual será armazenada no interior desta variável.

Exemplo:

```
ALGORITMO Teste PROGRAM Teste;
VARIÁVEIS VAR
Número: INTEIRO Número: INTEGER;
INICIO BEGIN
Número 10 Número: =10;
FIM END.
```

O Exemplo acima nos informa que:

a) Foi definido uma variável, a qual demos o Nome de "Número", e informamos que esta variável, ou posição de memória, só poderá aceitar dados, que sejam numéricos e que estejam entre - 32768 a +32767 ( tipo INTEGER ).

b) Atribuimos à variável "Número" o valor 10

A memória se comportaria da seguinte forma, de acordo com os itens acima:

a) Variável Conteúdo  
Número indefinido

b) Variável Conteúdo  
Número 10

## 5.2 Variáveis do tipo CHAR(Character) e STRING(Cadeia)

As definições de variáveis como sendo do tipo CHAR e STRING, possuem algumas curiosidades que merecem um cuidado especial por parte do usuário.

### 5.2.1 Uso das aspas ( ' )

Quando estivermos fazendo a atribuição de um valor para uma variável do tipo CHAR (Character) ou STRING (Cadeia), temos que ter o cuidado de colocar o valor ( dado ) entre aspas ( ' ), pois esta é a forma de informar que a informação é character.

Exemplo:

```
ALGORITMO Teste
VARIÁVEIS
Letra : CHARACTER
Nome : CADEIA
INICIO
Letra 'A'
Nome 'João'
FIM
```

### 5.2.2 Manipulação de caracteres individuais em STRING's (Cadeias)

Muitas vezes é necessário manipular caracteres individuais em uma STRING (Cadeia) . O Pascal possui uma forma toda especial de permitir tal operação, através do uso de colchetes( [ ] ) logo após o Nome da variável do tipo STRING (Cadeia) , e o número do caracter que se deseja manipular entre os colchetes.

Exemplo: Atribuir o primeiro caracter de uma STRING a uma variável do tipo CHAR.

```
ALGORITMO AtribuiString;
VARIÁVEIS
letra : CHARACTER
Nome : CADEIA
INICIO
Nome 'Joao'
letra Nome[1]
FIM
```

Quando definimos uma variável como sendo do tipo STRING não estamos alocando 1 posição de memória apenas ( uma caixa, pela analogia inicial ), mas na verdade, estamos alocando até 255

caixas, uma para cada caracter da STRING ( lembre-se que uma STRING pode ter no máximo 255 caracteres ). Ao utilizarmos o símbolo de colchete, estamos na verdade indicando qual o caracter ( qual a caixa ) desejamos manipular.

De acordo com o Exemplo acima, teríamos na memória a seguinte situação:

a) Alocamos 255 bytes ( caracteres ) na memória. A estas posições de memória é dado o Nome de "Nome". Inicialmente estas posições de memória ( Caixas ) possuem o conteúdo indefinido.

b) Alocamos 1 byte ( caracter ) na memória. A este caracter é dado o Nome de "Letra". Inicialmente esta posição de memória possui o conteúdo indefinido.

c) Na memória temos a seguinte situação:

Nome  
1 2 3 ... 255

Letra

d) Atribuímos a variável "Nome" o valor "João", obtendo na memória a seguinte configuração

Nome  
1 2 3 4 ... 255  
'J' 'o' 'a' 'o'

e) Atribuímos a variável "Letra" o primeiro caracter da variável "Nome", ou seja, o conteúdo da primeira posição de memória ( caixa ). Na memória teremos a seguinte configuração:

Nome  
1 2 3 4 ... 255  
'J' 'o' 'a' 'o'

Letra

'J'

Obs.: É possível definir variáveis do tipo STRING(Cadeia) com menos de 255 caracteres. Para isto, basta colocar, após a palavra STRING(Cadeia), o número de caracteres desejados entre colchetes ( []).

Exemplo:

```
ALGORITMO Define
VARIÁVEIS
Nome: CADEIA[80]
INICIO
FIM
```

Desta forma, o espaço ocupado por uma variável STRING(Cadeia) passa de 255 bytes para apenas 80 bytes, na memória.

### 5.2.3 Exercícios 0:

1) Dar o tipo de cada uma das constantes

- a) 613
- b) 613,0
- c) -613
- d) "613"
- e)  $-3,012 * 10^{15}$
- f)  $17 * 10^{12}$
- g)  $-28,3 * 10^{-23}$
- h) "Fim de Questão"

2) Faça um PROGRAMA para atribuir a variáveis as seguintes informações:

- a) 12345
- b) 123456
- c) -1122
- d) 10
- e) VERDADE
- f) 12345605

3) No seguinte PROGRAMA existe algum erro? Onde?

```
ALGORITMO Teste
VARIÁVEIS
Maria : CADEIA
idade : INTEIRO
letra : CHARACTER
Maria : REAL
INICIO
idade 23
idade 678
idade letra
letra ABC
letra A
letra 2
FIM
```

4) Qual a diferença existente nas seguintes atribuições?

a) Letra 'A'  
Nome 'João'

b) Letra A  
Nome João

5) É correto definir uma variável como sendo Character e atribuímos a ela o valor: 'PEDRO'? E se a variável fosse definida como CADEIA, a mesma poderia receber um valor do tipo CHARACTER?

## 6. Operadores

### 6.1 Operadores matemáticos

Os operadores matemáticos são os seguintes:

- a) Adição : +
- b) Subtração : -
- c) Multiplicação : \*
- d) Divisão : /
- e) Divisão inteira : DIV (QUOCIENTE)
- f) Resto da divisão : MOD (RESTO)
- g) Exponenciação : \*\*
- h) Diferença : <>
- i) Menor Igual : <=
- j) Maior Igual : >=

## 6.2 Operadores Lógicos

Os operadores lógicos, realizam as operações da álgebra booleana. Os operadores são os seguintes:

- a) AND (E)
- b) OR (OU)
- c) NOT (NÃO)
- d) XOR (NÃO OU)

Exemplo:

- a) Operador AND:

TRUE AND TRUE => TRUE  
TRUE AND FALSE => FALSE  
FALSE AND TRUE => FALSE  
FALSE AND FALSE => FALSE

- b) Operador OR:

TRUE OR TRUE => TRUE  
TRUE OR FALSE => TRUE  
FALSE OR TRUE => TRUE  
FALSE OR FALSE => FALSE

- c) Operador NOT:

NOT TRUE => FALSE  
NOT FALSE => TRUE

- d) Operador XOR:

TRUE XOR TRUE => FALSE  
TRUE XOR FALSE => TRUE  
FALSE XOR TRUE => TRUE  
FALSE XOR FALSE => FALSE

### 6.2.1 Exercícios 1 :



1) indique qual o resultado será obtido das seguintes expressões:

- a)  $1 / 2$
- b)  $1 \text{ DIV } 2$
- c)  $1 \text{ MOD } 2$
- d)  $(200 \text{ DIV } 10) \text{ MOD } 4$
- e)  $5^{**}2 + 3$
- f)  $6 + 19 - 23$
- g)  $3,0 * 5,0 + 1$
- h)  $1/4 + 2$
- i)  $29,0/7 + 4$
- j)  $3/6,0 - 7$

2) Indique o resultado das seguintes expressões:

- a)  $2 > 3$
- b)  $(6 < 8) \text{ OR } (3 > 7)$
- c)  $(( (10 \text{ DIV } 2) \text{ MOD } 6) > 5) \text{ XOR } (3 < (2 \text{ MOD } 2))$
- d)  $\text{NOT } (2 < 3)$

3) Escreva o comando de atribuição e resolva a expressão das seguintes formulas matemáticas.

a) onde  $A= 2, B= 6, C = 3, D=4, E=8, F=4$

b) onde  $X = 2$

4) Construa o algoritmo que calcule as seguintes expressões:

- a)  $2 + 3 * \{ 23 - 12 + [ \{ (123 / 34) + 10 \} / 2 - 1 * (45 - 12) ] / 3 \}$
- b)  $[ (2 * 3)^2 - 1 ]^4$
- c)  $(2 + [ 2 * 3 - (4 / 20)^2 ] ) / (10 * \{ [(7 * 3) - 2]^3 \})$

Obs.: O operador "+" caso seja usado entre variáveis do tipo CHAR ou STRING, causará uma ação conhecida por concatenação, ou seja, juntar os caracteres ou STRING's usados na operação em uma só STRING.

Exemplo:

```
ALGORITMO Concatena
VARIÁVEIS
Letra1, Letra2 : CHARACTER
Nome1, Nome2, Nome : CADEIA
INICIO
Letra1 'D'
Letra2 'a'
Nome1 'Joao'
Nome2 'Silva'
Nome Nome1 + Letra1 + Letra2 + Nome2
FIMmplo:
```

As instruções acima resultarão no armazenamento do Nome 'João Da Silva' na variável rotulada de "Nome".

## 7. Comandos de Entrada e Saída

O computador não é uma máquina isolada, pois ele precisa se comunicar com o mundo exterior com vídeo, impressora, teclado, discos, fitas etc. Para realizar esta comunicação existem comandos que permitem que informações sejam exibidas, por Exemplo, no vídeo, como também existem comandos que permitem que informações sejam colocadas na memória do computador através do teclado do PC.

Os comandos que iremos ver são os comandos LEIA e ESCRIVA, respectivamente, comando de entrada e de Saída ( em Pascal escreve-se READ e WRITE ).

Exemplo 1: Escrever um algoritmo para ler um valor numérico do teclado e atribuí-lo a uma variável do tipo numérica.

```
ALGORITMO LeNúmero
VARIÁVEIS
Num : INTEIRO
INICIO
LEIA(Num)
FIM
```

O algoritmo acima, executará os seguintes comandos:

a) Define uma variável rotulada "Num", a qual só poderá armazenar dados numéricos, que estejam compreendidos entre -32768 a +32767 ( INTEIRO ), sendo que inicialmente o conteúdo desta variável está indefinido.

b) interrompe o processamento até que uma informação seja digitada, seguida do pressionamento da tecla ENTER. Caso a informação seja digitada seja compatível com o tipo INTEGER (INTEIRO), este valor será armazenado dentro da variável "Num".

Exemplo 2: Fazer um algoritmo para escrever no vídeo do PC uma mensagem qualquer.

```
ALGORITMO EscreveMsg
INICIO
ESCREVA ('Alo Mundo')
FIM
```

Obs.: A mensagem está entre aspas porque representa uma cadeia de caracteres.

É perfeitamente possível mandar escrever o conteúdo de variáveis no vídeo. Desta forma, o Exemplo acima poderia ser escrito do seguinte modo:

```
ALGORITMO EscreveMsg
VARIÁVEIS
Msg : CADEIA
INICIO
Msg 'Alo Mundo'
ESCREVA (Msg)
FIM
```

Obs.: Em Pascal, o Exemplo ficaria como é mostrado abaixo

```
PROGRAM EscreveMsg;
VAR
Msg : STRING;
BEGIN
Msg := 'Alo Mundo';
WRITE (Msg);
END.
```

Obs.: Em Pascal, quando usamos os comandos READ e WRITE, o cursor continuará posicionado na mesma linha. Para evitar tal situação, devemos pular para nova linha. Isto é obtido acrescentando as letras LN ( de LINE NEW, ou seja, linha nova ) no final dos comandos READ e WRITE. Desta forma os comandos terão a seguinte grafia: READLN e WRITELN.

#### 7.1.1 Exercícios 2:

1) Faça um algoritmo para ler as seguintes informações de uma pessoa: Nome, Idade, Sexo, Peso, Altura, Profissão, Rua, Bairro, Cidade, Estado, CEP, Telefone.

2) Dado as seguintes informações de um funcionário: Nome, idade cargo e o seu salário bruto considere:

a) O salário bruto teve um reajuste de 38%.

b) O funcionário receberá uma gratificação de 20% do salário bruto.

c) O Salário total é descontado em 15%

Faça um algoritmo e um algoritmo para:

· Imprimir Nome, idade e cargo.

· Imprimir o salário bruto.

· Imprimir o salário líquido.

3) Faça um algoritmo para ler a base e a altura de um triângulo. Em seguida, escreva a área do mesmo.

Obs.:  $\text{Área} = (\text{Base} * \text{Altura}) / 2$

4) Faça um algoritmo para:

a) Ler uma CADEIA em forma de data. Exemplo : '26/09/95'

b) escrever o dia , mês e ano desta data.

5) Uma empresa tem para um determinado funcionário uma ficha contendo o nome, número de horas trabalhadas e o n0 de dependentes de um funcionário.

Considerando que:

a) A empresa paga 12 reais por hora e 40 reais por dependentes.

b) Sobre o salário são feito descontos de 8,5% para o INSS e 5% para IR.

Faça um algoritmo para ler o Nome, número de horas trabalhadas e número de dependentes de um funcionário. Após a leitura, escreva qual o Nome, salário bruto, os valores descontados para cada tipo de imposto e finalmente qual o salário líquido do funcionário.

6) Faça um algoritmo que calcule a média aritmética de 4 valores inteiros.

7) O preço de um automóvel é calculado pela soma do preço de fábrica com o preço dos impostos (45% do preço de fábrica) e a percentagem do revendedor (28% do preço de fábrica). Faça um algoritmo que leia o nome do automóvel e o preço de fábrica e imprima o nome do automóvel e o preço final.

## 8. Processos de Repetição e Seleção

### 8.1 Estrutura de Repetição ENQUANTO-FAÇA (WHILE DO)

a) ALGORITMO

ALGORITMO ExEnquanto

ENQUANTO <Condição for verdadeira> FAÇA

<Comandos>  
FIM ENQUANTO

## b) PASCAL

```
PROGRAM ExEnquanto;  
WHILE <Condição for verdadeira> DO  
BEGIN  
<Comandos>;  
END.
```

Exemplo : Faça um algoritmo para ler e escrever o Nome de 20 pessoas.

```
ALGORITMO LeEscreve  
VARIÁVEIS  
Nome : CADEIA  
Total : INTEIRO  
INICIO  
Total 0  
ENQUANTO Total<20 FACA  
LEIA(Nome)  
ESCREVA ('Nome=', Nome)  
Total Total + 1  
FIM ENQUANTO  
END.
```

Em pascal:

```
PROGRAM LeEscreve;  
VAR  
Nome : STRING;  
Total : INTEGER;  
BEGIN  
Total := 0;  
WHILE Total<20 DO  
BEGIN  
READ(Nome);  
WRITE ('Nome=', Nome);  
Total := Total + 1;  
END;  
END.
```

### 8.1.1 Exercícios 3:

- 1) Faça um algoritmo para ler base e altura de 50 triângulos e imprimir a sua área.
- 2) Faça um algoritmo para calcular N!.
- 3) Faça um algoritmo para calcular um valor A elevado a um expoente B. Os valores A e B deverão ser lidos. Não usar  $A^{**} B$ .
- 4) Faça um algoritmo para:
  - a) Ler um valor x qualquer
  - b) Calcular  $Y = (x+1)+(x+2)+(x+3)+(x+4)+(x+5)+\dots+(x+100)$ .
- 5) Faça um algoritmo para somar os números pares < 1000 e ao final imprimir o resultado.
- 6) Faça um algoritmo para gerar os termos da seguinte P.G.: 3,9,27,81,...,calcule o 100º termo.

- 7) Ampliando o exercício acima, faça um algoritmo para calcular a soma dos termos da P.G..
- 8) Faça um algoritmo para calcular a série Fibonacci até o N-ésimo termo. A série tem a seguinte forma: 1,1,2,3,5,8,13,21,34,...
- 9) Faça um algoritmo para ler um valor X e calcular  $Y = X+2X+3X+4X+5X+\dots+20X$
- 10) Faça um algoritmo para ler e escrever o Nome, idade e sexo de um número indeterminado de alunos. Ao final escreva o total de alunos lidos.
- 11) Para cada nota de compra, tem-se o Nome do produto comprado, o valor e o imposto. Faça um algoritmo que escreva o valor total bruto, o imposto total cobrado e o valor total líquido de todas as notas. Considere 500 notas
- 12) O valor aproximado de PI pode ser calculado usando-se a série  
sendo  
fazer um algoritmo para calcular e escrever o valor de PI com 51 termos
- 13) Faça um algoritmo para ler um valor X e um valor n. Após, calcule a seguinte expressão:
- 14) Faça um algoritmo para somar os restos da divisão por 3 de 200 números.
- 15) Faça um algoritmo que calcule a hipotenusa de 10 triângulos.  
 $hipotenusa^2 = cateto^2 + cateto^2$
- 16) Faça um algoritmo para calcular a área de N quadriláteros. Fórmula: Área = Lado \* Lado
- 17) Faça um algoritmo para calcular a área de N Círculos . Fórmula : Área =  $p \text{ raio}^2$   $p = 3,141592$ .
- 18) Fazer um algoritmo que calcule escreva a soma dos 50 primeiros termos das seguinte série:

## 8.2 Estrutura de repetição REPITA-ATÉ ( Repeat Until)

```
ALGORITMO
REPITA
<Comandos>
ATE <Condição for verdadeira>
```

```
PASCAL
REPEAT
<Comandos>
UNTIL <Condição for verdadeira>
```

Exemplo: Faça um ALGORITMO para ler e escrever o Nome de 20 pessoas.

```
ALGORITMO LeEscreve
VARIÁVEIS
Nome : CADEIA
Total : INTEIRO
INICIO
Total 0
REPITA
```

```

LEIA(Nome)
ESCREVA('Nome=',Nome)
Total Total + 1
ATÉ Total >=20
FIM
Em pascal:
PROGRAM LeEscreve;
VAR
Nome : STRING;
Total : INTEIRO;
BEGIN
Total := 0;
REPEAT
READ(Nome);
WRITE('Nome=',Nome);
Total := Total + 1;
UNTIL Total >=20;
END.

```

### 8.2.1 Exercícios 4:

- 1) Leia 20 valores reais e escreva o seu somatório.
- 2) Dado uma frase de exatamente 80 caracteres, escreva a frase de trás para frente, um caracter por linha.
- 3) Dado um número indeterminado de funcionários, onde é fornecido o Nome, número de dependentes e o número de horas trabalhadas. Pede-se que seja impresso, para cada funcionário, o seu Nome, salário bruto, salário líquido e o valor descontado. A empresa, paga R\$ 2.250,00 por hora de trabalho, e R\$ 125,55 por dependente, e ainda faz um desconto de 12% sobre o salário bruto. Pede-se ainda que seja impresso o total de funcionários da empresa, o total gasto com salários, e o total descontado.
- 4) Faça um ALGORITMO para:
  - a) Leia um valor X e um valor N.
  - b) Calcular:  $Y = X - 2X + 4X - 6X + 8X - 10X + \dots NX$ .
- 5) Faça um ALGORITMO para ler um valor X e um Valor N. Após, calcule a seguinte expressão:
- 6) Dado uma quantidade indeterminada de palavras, sendo que as mesmas sempre terminam com o caracter "\$", escreva o tamanho de cada uma das mesmas ( o sinal de "\$" não deve ser considerado na contagem ).
- 7) Dado uma palavra terminada com o sinal "\$", faça um ALGORITMO para criar uma nova STRING, a ser formada com os caracteres da primeira palavra invertidos.
- 8) Faça um algoritmo que calcule a hipotenusa de 10 triângulos.  
 $hipotenusa^2 = cateto^2 + cateto^2$
- 9) Faça um algoritmo para calcular a área de N quadriláteros. Fórmula:  $Área = Lado * Lado$
- 10) Faça um algoritmo para calcular a área de N Círculos . Fórmula :  $Área = p \text{ raio}^2$   $p = 3,141592$ .
- 11) Fazer um algoritmo que calcule escreva o valor de S onde:

12) Fazer um algoritmo que calcule e escreva o valor de S onde:

### 8.3 Estrutura de repetição PARA-FAÇA (FOR ... TO... DO)

#### ALGORITMO

```
PARA <Variável> DE <INICIO> Até <FIM>, PASSO <N> FAÇA  
<Comandos>  
FIMPARA
```

#### PASCAL

```
a) FOR <Variável > : = <INICIO> TO <FIM> DO  
<Comando>;
```

```
b) FOR <Variável > : = <INICIO> TO <FIM> DO  
BEGIN  
<Comando>;  
END;
```

Exemplo: Faça um algoritmo para ler e escrever o Nome de 20 pessoas.

```
ALGORITMO LeEscreve  
VARIÁVEIS  
Nome : CADEIA  
Cont : INTEIRO  
INICIO  
PARA Cont DE 1 ATE 20, PASSO 1, FAÇA  
LEIA(Nome)  
ESCREVA(Nome)  
FIMPARA  
FIM
```

Em pascal:

```
PROGRAM LeEscreve;  
VAR  
Nome : STRING;  
Cont : INTEIRO;  
BEGIN  
FOR Cont : = 1 TO 20 DO  
BEGIN  
READ(Nome);  
WRITE(Nome);  
END;  
END.
```

Obs.:

a) A variável de controle, no Exemplo acima é a variável "Cont", é automaticamente incrementada em um unidade, a partir do valor inicial ( 1 pelo Exemplo acima ), até que seja ultrapassado o limite final definido ( 20 no Exemplo acima ).

B) Quando existe a necessidade de interromper o processamento antes de ser alcançado o limite final definido para a estrutura, não deverá ser usado a estrutura PARA-FAÇA, pela simples razão que a variável de controle não deve ser alterada propositadamente pelo usuário. A alteração do conteúdo das variáveis de controle do PARA-FAÇA são de inteira responsabilidade da própria estrutura.

#### 8.3.1 Exercícios 5:

1) Uma empresa possui 100 vendedores que ganham por comissão sobre cada produto vendido. Cada vendedor em um determinado mês vendeu X produtos, onde cada produto pode ou não ter preços iguais. A empresa deseja obter um relatório com o Nome, o total de vendas e o valor a ser pago a cada vendedor. A comissão paga pela empresa é de 30% sobre o valor de cada produto vendido.

2) Dado uma relação de 1000 números em graus Célcus, faça um PROGRAMA que imprima o seguinte relatório:

Graus Fahrenheit Graus Célcus

xxxxxx xxxxxx

Obs.:

3) Dado um número indeterminado de palavras, sendo que cada palavra possui um tamanho indeterminado, o qual deverá ser lido pelo teclado, faça um PROGRAMA que imprima um relatório como é mostrado no Exemplo abaixo: Suponha que a palavra digitada foi "MARIA".

M A

A I

R R

I A

A M

4) Escrever um programa para gerar e escrever uma tabela com os valores de seno de um Angulo A em radianos, utilizando a série de Mac-Laurin Truncada apresentada a seguir:

Condições: os valores dos ângulos devem variar de 0.0 a 6.3, inclusive, de 0.1 em 0.1

Obs.: Em pascal, caso seja necessário que o "Loop" PARA-FAÇA ao invés de incrementar a variável de controle decrementa-a, a Sintaxe a ser utilizada será a seguinte

a) FOR <Variável > : = <FIM> DOWNTO <INICIO> DO  
<Comando>

b) FOR <Variável > : = <FIM> DOWNTO <INICIO> DO  
BEGIN  
<Comando>;  
END;

Em algoritmo:

a) PARA <Variável > DE <FIM> ATÉ <INICIO> PASSO (-1) FAÇA  
<Comando>  
FIM PARA

## 9. Estrutura de seleção

### 9.1 Estrutura SE...ENTÃO...SENÃO (IF...THEN...ELSE)

ALGORITMO



```
SE <Condição FOR verdade> ENTÃO
<Comandos>
[SENÃO
<Comandos>] Colchete indica que o comando é opcional
FIMSE
```

## PASCAL

```
a) IF <Condição for verdade> THEN
<Comandos>
[ELSE
<Comandos>.]
```

```
b) IF <Condição for verdade> THEN
BEGIN
<Comandos>
END
[ELSE
BEGIN
<Comandos>;
END;]
```

Exemplo: Dado dois valores A e B quaisquer, faça um algoritmo que imprima se  $A > B$ , ou  $A < B$ , ou  $A = B$

```
ALGORITMO Maior
VARIÁVEIS
A,B : INTEIRO
INICIO
ESCREVA('Digite os valores A e B');
SE A > B ENTÃO
ESCREVA('A é maior que B')
SENÃO
SE A < B ENTÃO
ESCREVA('A é menor que B')
SENÃO
ESCREVA('A é igual a B')
FIM
```

### Em pascal

```
PROGRAM Maior;
VAR
A,B : BYTE;
BEGIN
WRITE('Digite os valores A e B');
IF A > B THEN
WRITE('A é maior que B')
ELSE
IF A < B THEN
WRITE('A é menor que B')
ELSE
WRITE('A é igual a B');
END.
```

### 9.1.1 Exercícios 6:

- 1) Faça um algoritmo que leia os valores A, B, C e diga se a soma de  $A + B$  é menor que C.
- 2) Faça um algoritmo que leia dois valores inteiros A e B se os valores forem iguais deverá se somar os dois, caso contrário multiplique A por B ao final do calculo atribuir o valor para uma variável C.

- 3) Faça um algoritmo que leia o nome e a três notas de uma disciplina de uma aluno e ao final escreva o nome do aluno, sua média e se ele foi aprovado a média é 8.
- 4) Faça um algoritmo que leia 3 números inteiros e imprima o menor deles.
- 5) Dado um conjunto de n registros cada registro contendo um valor real, faça um algoritmo que calcule a média dos valores maiores que 4.
- 6) Dado uma relação de 100 carros imprima quantos são da cor azul. Sendo para cada carro tem-se uma ficha contendo o nome e a cor.
- 7) Dados três valores distintos, fazer um algoritmo que, após a leitura destes dados coloque-os em ordem crescente.
- 8) Dado uma série de 20 valores reais, faça um algoritmo que calcule e escreva a média aritmética destes valores, entretanto se a média obtida for maior que 8 deverá ser atribuída 10 para a média.
- 9) Faça um algoritmo para somar os números pares < 1000 e ao final imprimir o resultado.
- 10) Dado um conjunto de 20 valores reais, faça um algoritmo que:
  - a) Imprima os valores que não são negativos.
  - b) Calcule e imprima a média dos valores < 0.
- 11) Suponha que para cada aluno de sua sala exista uma ficha contendo o nome e a idade do aluno. Supondo que exista 50 alunos, faça um algoritmo que determine quantos alunos tem idade maior que 30.
- 12) Dado modelo, ano de fabricação, cor e placa de 1000 carros, faça um algoritmo que:
  - a) Imprima quantos são, da cor verde e o percentual em relação ao total.
  - b) Imprima quantos foram fabricados antes de 1990 e o percentual em relação ao total.
  - c) Imprima quantos são FIAT UNO e o percentual em relação ao total.
  - d) Imprima quantos carros tem na placa o dígito 5 e o percentual em relação ao total. Considere placas com seguinte formato "AANNNN".
- 13) Dado Nome e notas ( total de 6 ) de n alunos, faça um algoritmo que:
  - a) Imprima Nome e média dos alunos aprovados. Média  $\geq 7.0$ .
  - b) Imprima Nome e média dos alunos em recuperação.  $5.0 \geq$  Média  $< 7.0$ .
  - c) Imprima Nome e média dos alunos reprovados. Média  $< 5.0$ .
- 14) Faça um algoritmo que simule um cronometro com horas, minutos e segundos. A simulação deverá mostrar no vídeo o cronometro no seguinte formato HH: MM: SS. Inicialmente o cronometro será zerado ( 00: 00: 00 ), sendo que os segundos começarão a ser incrementados, depois os minutos e depois as horas. Lembre-se que a cada 60 segundos os minutos deverão ser incrementados, depois os minutos voltarão a zero. Quando as horas estiverem prestes a chegar em 25, o cronometro deverá voltar a zero ( 00: 00: 00 ).
- 15) Dado Nome e salário de um número indeterminado de funcionários, imprima o Nome e o salário de funcionário que tem o maior salário, bem como o Nome e o salário que possui o menor salário.
- 16) Dado uma relação de 100 STRING's no formato de data ( Exemplo '10/12/95' ), imprima aquelas que correspondem ao mês '09'.

17) Dado uma relação de 100 STRING's no formato de data ( Exemplo '10/12/95' ), imprima um relatório no seguinte formato:

Data Por Extenso

26/09/95 26 de setembro de 1995

18) Dados uma relação de n STRING's, todas terminadas com o caracter "\$", faça um algoritmo que escreva a STRING e o seu tamanho.

19) Dado uma relação de n STRING's, todas terminadas com o caracter "\$", faça um algoritmo que imprima aquelas que tem tamanho par.

20) Dado uma relação de n STRING's, todas terminadas com o caracter "\$", faça um algoritmo que escreva as STRING's ímpares cujo caracter central é uma vogal.

21) Dado três valores X, Y, Z, verificar se eles podem ser os comprimentos dos lados de um triângulo, e se forem, verificar se é um triângulo equilátero, isósceles ou escaleno. Se eles não formarem um triângulo, escrever uma mensagem.

Antes da elaboração do algoritmo, torna-se necessária a revisão de algumas propriedades e definições.

Propriedade - O comprimento de cada lado de um triângulo é menor do que a soma dos comprimentos dos outros dois lados.

Definição 1 - Chama-se triângulo equilátero os que tem os comprimentos dos três lados iguais,

Definição 2 - Chama-se triângulo isósceles ao triângulo que tem os comprimentos de dois lados iguais.

Definição 3 - Chama-se triângulo escaleno ao triângulo que tem os comprimentos dos três lados diferentes.

22) Um hotel cobra R\$ 300,00 por diária e mais uma taxa adicional de serviços. Se a diária for menor que 15 a taxa é de R\$ 20,00. Se o número de diárias for igual a 15 a taxa é de R\$ 14,00 e se o número for maior que 15 a taxa é de R\$ 12,00. Considerando-se que se tenha 200 pessoas e para cada pessoa tenha-se um registro contendo seu nome e o número de diárias.

Faça um algoritmo que imprima o nome e o total a pagar de cada pessoa e imprimir também o total ganho pelo hotel e o número total de diárias.

23) Dado que cada pessoa tenha o seu nome e a sua idade e o seu peso em uma ficha, faça um algoritmo que:

a) Imprimir o nome da pessoa cujo a idade esta na faixa de 20 a 30 anos inclusive os extremos;

b) Calcule a idade média das pessoas com peso maior que 80 Kg, considere que exista N pessoas.

24) Faça um algoritmo que calcule o salário e o imposto de renda de um grupo de pessoas, considerando:

a) Para cada pessoa tem-se: Nome, Número de Dependentes, Renda Bruta Anual;

b) O imposto é calculado segundo a tabela abaixo:

Renda Líquida %

até R\$ 10.000,00 isento

> R\$ 10.000,00 até R\$ 30.000,00 5%

> R\$ 30.000,00 até R\$ 60.000,00 10%

> R\$ 60.000,00 15%

c) Há um desconto de R\$ 600,00 para cada dependente.

d) Sendo que a renda líquida é igual a renda bruta menos o abatimento de cada dependente.

25) O Departamento de transito do estado anotou dados de acidentes de transito no ultimo ano. Para cada motorista envolvido no acidente, tem-se as seguintes informações:

- Ano de nascimento;
- Sexo (M - Masculino, F - Feminino)
- Procedência ( 0 - Capital, 1 - Interior, 2 - Outro estado);

Faça um algoritmo que :

- a) Calcule a porcentagem de motoristas com menos de 21 anos;
  - b) Calcule quantas mulheres são da capital;
  - c) Calcule quantos motoristas do interior do estado tem idade maior que 60 anos;
  - d) Verifique se existe alguma mulher com idade maior que 60 anos;
- R.

26) Suponha que para cada tipo de computador fabricado, a SEI, tem os seguintes informações:

- Nome do computador;
- O nome do fabricante;
- e a sua capacidade de armazenamento;

Faça um algoritmo que:

- a) Determine qual o valor da maior capacidade de memória;
- b) Determine quantos computadores diferentes a IBM fabrica;
- c) Verifique se tem algum computador chamado PS/1. Se tiver qual o seu fabricante;

## 9.2 Estrutura CASE

### ALGORITMO

```
ESCOLHA <Valor>  
<Opções> : <Comandos>  
.  
.  
.  
<Opções> : <Comandos>  
[ Senão  
<Comandos>]  
FIMESCOLHA
```

### PASCAL

```
a) CASE <valor> OF  
<opções> : <Comandos>;  
.  
.  
.  
<opções> : <Comandos>;  
[Else  
<Comandos>;  
END;
```

```
b) CASE <valor> OF  
<opções> : BEGIN  
<Comandos>;  
END;  
.  
.  
.  
<opções> : <Comandos>;  
[ Else  
BEGIN  
<Comandos>;
```

```
END;  
END;
```

O comando ESCOLHA ( CASE ), corresponde ao comando SE-ENTÃO mas de uma forma mais compacta nas operações de seleção. Por Exemplo, seja a seguinte estrutura escrita com o comando SE-ENTÃO:

```
SE ( IDADE >= 0 ) E ( IDADE <= 3 ) ENTÃO  
ESCREVA( 'BEBÊ'  
SENÃO  
SE ( IDADE >= 4 ) E ( IDADE <= 10 ) ENTÃO  
ESCREVA( 'CRIANÇA'  
SENÃO  
SE ( IDADE >= 11 ) E ( IDADE <= 18 ) ENTÃO  
ESCREVA( 'ADOLESCENTE'  
SENÃO  
ESCREVA( 'ADULTO'  
FIM SE  
FIM SE  
FIM SE
```

Em pascal

```
IF ( IDADE >= 0 ) AND ( IDADE <= 3 ) THEN  
WRITE( 'BEBÊ'  
ELSE  
IF ( IDADE >= 4 ) AND ( IDADE <= 10 ) THEN  
WRITE( 'CRIANÇA'  
ELSE  
IF ( IDADE >= 11 ) AND ( IDADE <= 18 ) THEN  
WRITE( 'ADOLESCENTE'  
ELSE  
WRITE( 'ADULTO');
```

O Exemplo mostrado poderia ser escrito da seguinte forma usando o comando ESCOLHA:

```
ESCOLHA Idade  
0..3 : ESCREVA('BEBÊ')  
4..10 : ESCREVA('CRIANÇA')  
11..18 : ESCREVA('ADOLESCENTE')  
SENÃO  
ESCREVA('ADULTO')  
FIM ESCOLHA
```

```
Em pascal:  
CASE Idade OF  
0..3 : WRITE('BEBÊ');  
4..10 : WRITE('CRIANÇA');  
11..18 : WRITE('ADOLESCENTE');  
ELSE  
WRITE('ADULTO');  
END;
```

Abaixo seguem mais exemplos de construções com o comando ESCOLHA:

```
a) ESCOLHA Letra  
'A','E','I','O','U' : ESCREVA('Vogais')  
FIMESCOLHA
```

```
b) ESCOLHA VALOR  
1..5,9,13..255 : <Comandos>  
FIMESCOLHA
```

```
Em pascal  
a) CASE Letra OF
```

```
'A','E','I','O','U' : WRITE('Vogais');  
END;
```

```
b) CASE valor OF  
1..5,9,13..255 : <Comandos>;  
END;
```

Obs.: O comando Escolha não aceita valores do tipo REAL e STRING.

### 9.2.1 Exercícios 7:

1) Faça um algoritmo que leia um número que represente um determinado mês do ano. Após a leitura escreva por extenso qual o mês lido. Caso o número digitado não esteja na faixa de 1..12 escreva uma mensagem informando o usuário do erro da digitação.

2) Faça um algoritmo que leia um número qualquer. Caso o número seja par menor que 10, escreva 'Número par menor que Dez', caso o número digitado seja ímpar menor que 10 escreva 'Número Ímpar menor que Dez', caso contrário Escreva 'Número fora do Intervalo'.

3) Uma empresa irá dar um aumento de salário aos seus funcionários de acordo com a categoria de cada empregado. O aumento seguirá a seguinte regra:

- a) Funcionários das categorias A, C, F, e H ganharão 10% de aumento sobre o salário
- b) Funcionários das categorias B, D, E, I, J e T ganharão 15% de aumento sobre o salário
- c) Funcionários das categorias K e R ganharão 25% de aumento sobre o salário
- d) Funcionários das categorias L, M, N, O, P, Q, R e S ganharão 35% de aumento sobre o salário
- e) Funcionários das categorias U, V, X, Y, W e Z ganharão 50% de aumento sobre o salário

4) Faça um algoritmo que Escreva Nome, categoria e salário reajustado de cada Empregado. Considere 1000 funcionários na empresa.

5) Dado uma STRING terminada com o caracter "\$" faça um algoritmo que imprima quantas vogais existem nesta STRING.

6) Dado uma STRING terminada com o caracter "\$" faça um algoritmo que imprima quantas Ocorrências de cada vogal em separado existe nesta STRING.

## 10. Variáveis Compostas Homogêneas

Vimos, no início deste curso, ser possível dar um Nome para uma posição de memória, sendo que a esta será associado um valor qualquer. Pois bem, acontece que, muitas vezes, esta forma de definição, ou melhor dizendo, de alocação de memória, não é suficiente para resolver certos problemas computacionais. Imagine por Exemplo, como faríamos para construir um algoritmo, para ler o Nome de N Pessoas e que imprimisse um relatório destes mesmos nomes, mas ordenados alfabeticamente? Não seria uma tarefa simples, haja visto não ser possível determinar quantos nomes seriam lidos, mesmo que soubéssemos o número de pessoas, digamos 1.000 pessoas, teríamos que definir 1.000 variáveis do tipo STRING, como é mostrado abaixo:

```
ALGORITMO Loucura;  
VARIÁVEIS  
Nome1,  
Nome2,  
Nome3,
```

```
.  
. .  
Nome999,  
Nome1000: CADEIA  
INICIO  
<Comandos>  
FIM
```

Considere o tamanho do algoritmo, e o trabalho braçal necessário para construí-lo. Isto só com 1.000 Nome, imagine agora 1.000.000 de pessoas. A construção deste algoritmo começaria a ficar inviável na prática. Para resolver problemas como este, e outros, foi criado um novo conceito para alocação de memória sendo, desta forma, também criada uma nova maneira de definir variáveis, a qual foi denominada de variável indexada.

Uma variável indexada corresponde a uma seqüência de posições de memória, a qual daremos único Nome, sendo que cada uma destas pode ser acessada através do que conhecemos por índice. O índice corresponde a um valor numérico ( exceto REAL ), ou a um valor caracter ( exceto STRING ). Cada uma das posições de memória de uma variável indexada pode receber valores no decorrer do algoritmo como se fosse uma variável comum, a única diferença reside na Sintaxe de utilização desta variável.

### 10.1 Variáveis Indexadas Uni Dimensionais(Vetores)

Também conhecida por "Vetor". Uma variável uni-dimencional, como o próprio Nome já indica, possui apenas uma dimensão, sendo ser possível definir variáveis com quaisquer tipo de dados validos do Pascal.

#### a) Definição:

```
ALGORITMO Define  
VARIÁVEIS  
<Nome>: VETOR [INICIOV : FIMV] DE <tipo>  
INICIO  
<Comandos>  
FIM  
Em pascal:  
PROGRAM Define;  
VAR  
<Nome>: ARRAY [INICIO..FIM] OF <tipo>;  
BEGIN  
<Comandos>;  
END.
```

Obs.:

- a) "ARRAY" é uma palavra reservada do Pascal
- b) Os valores "INICIOV" e "FIMV" correspondem aos índices inicial e final
- c) Uma variável indexada pode ser apenas de um tipo de dado

Exemplo Definir uma variável indexada como sendo do tipo REAL, sendo que a mesma deverá corresponder a 10 posições de memória.

```
ALGORITMO Exemplo  
VARIÁVEIS  
Vet : VETOR [1 : 10] OF REAL  
INICIO  
<Comandos>  
FIM  
Em pascal:  
PROGRAM Exemplo;  
VAR
```

```
Vet: ARRAY [1..10] OF REAL;  
BEGIN  
<Comandos>;  
END.
```

No Exemplo acima, após a definição da variável, a memória estará como mostrado no esquema abaixo:

Vet

1 2 3 4 5 6 7 8 9 10

Os Valores numéricos apresentados acima correspondem aos índices da variável.

B) Atribuição  
<Nome>[<Índice>] Valor;  
LEIA (<Nome>[<Índice>]);

Em pascal:

```
<Nome>[<Índice>] :=Valor;  
READ (<Nome>[<Índice>]);
```

Exemplo:  
ALGORITMO Atribui  
VARIÁVEIS  
Nomes : VETOR [1 : 20] DE CADEIA  
i : INTEIRO  
BEGIN  
Nomes[1] := 'João da Silva'  
PARA i = 2 ATE 20, PASSO 1, FAÇA  
LEIA(Nomes[i])  
FIM PARA  
FIM  
Em pascal:  
PROGRAM Atribui;  
VAR  
Nomes : ARRAY[1..20]OF STRING;  
i : INTEGER;  
BEGIN  
Nomes[1] := 'João da Silva';  
FOR i := 2 TO 20 DO  
READ(Nomes[i]);  
END.

### 10.1.1 Exercícios 8:

- 1) Faça um algoritmo que leia, via teclado, 200 valores do tipo inteiro e os guarde na memória.
- 2) Amplie o exercício anterior emitindo um relatório com todos os números pares que você leu.
- 3) Faça um algoritmo que determine qual o menor valor existente no vetor do exercício número 1.
- 4) Faça um algoritmo que determine qual o maior valor existente no vetor do exercício número 1.
- 5) Faça um algoritmo que imprima quantos dos valores do exercício número 1 são maiores que a média desses valores.
- 6) Faça um algoritmo que leia 10 nomes e os guarde na memória.



- 7) Amplie o exercício 6 imprimindo o menor Nome do vetor
- 8) Amplie o exercício 6 imprimindo o maior Nome do vetor
- 9) Faça um algoritmo que leia 20 palavras, e após a leitura, realiza um processo qualquer que inverta os caracteres de cada uma das palavras.
- 10) Faça um algoritmo que leia 10 nomes e os guarde na memória. Após a leitura, emita um relatório com todos os nomes que são palíndromes. Uma palavra palíndrome é aquela que a sua leitura é a mesma da esquerda para a direita e vice versa. Exemplo: ARARA, ANA, etc.
- 11) Faça um algoritmo que leia, Nome idade e sexo de N pessoas. Após a leitura faça:
- Imprima o Nome, idade e sexo das pessoas cuja idade seja maior que a idade da primeira pessoa.
  - Imprima o Nome e idade de todas as mulheres.
  - Imprima o Nome dos homens menores de 21 anos.
- 12) Faça um algoritmo para ler 20 valores e que imprima os que são maiores que a média dos valores.
- 13) Faça um algoritmo que leia Nome e notas ( total de 12 ) de N alunos de um colégio. Após a leitura faça:
- Imprima o Nome e a média dos alunos aprovados ( Média  $\geq 7.0$  ).
  - Imprima o Nome e a média dos alunos em Recuperação (  $5.0 \geq$  Média  $< 7.0$  ).
  - Imprima o Nome e a média dos alunos reprovados ( Média  $< 5.0$  ).
  - Imprima o percentual de alunos aprovados.
  - Imprima o percentual de alunos reprovados.
- 14) Dado Nome e salário de 10 funcionários faça:
- Imprima o Nome e o salário dos funcionários que ganham mais de R\$ 500,00
  - Para os funcionários que ganham menos de R\$ 200,00 conceda um aumento de 20%. Imprima Nome e o novo salário destes funcionários aumentados
- 15) Faça um algoritmo que crie uma tabela no vídeo do computador com todos os caracteres ASCII, sendo que deverá ser impresso o caracter, bem como o seu valor decimal.
- 16) Um teste composto por 10 questões foi proposto numa classe de n alunos. Cada questão admite as alternativas identificadas pelas letras A, B, C, D e E. Para cada aluno tem-se o registro contendo o seu nome e as suas respostas. Sendo dado o gabarito das questões faça um algoritmo que:
- Escreva uma relação contendo o nome e o número de acertos de cada aluno;
  - Determine quantos alunos acertaram as questões 1 e 2 e tiveram 7 acertos;
  - Escreva o nome do aluno, que obteve a melhor nota e pior nota;
- 17) Faça um algoritmo para ler 50 valores inteiros. Após imprima tais valores ordenados crescentemente.
- 18) Continuando o exercício anterior, emita um relatório com os valores ordenados decrescentemente.
- 19) Faça um algoritmo que leia 10 valores numéricos inteiros. Após a leitura emita um relatório com cada valor diferente e o número de vezes que o mesmo apareceu no vetor.

20) Dado uma relação de N nomes, faça um algoritmo que verifique se uma determinada pessoa está neste vetor. O Nome da pessoa a ser pesquisada deverá ser lido, bem como os nomes a serem colocados no vetor.

21) Dado uma frase, faça um algoritmo que determine qual a consoante mais utilizada.

22) Faça um algoritmo que leia N letras e armazene cada uma em uma posição de memória. Coloque no vetor, após a última letra lida, o caracter "\$", indicando final dos caracteres

23) Faça um algoritmo que:

a) Leia um vetor A com N elementos e um vetor B com N elementos do tipo CADEIA.

b) Intercale estes vetores A e B, formando um outro vetor C da seguinte forma.

C[1] A[1]

C[2] B[1]

C[3] A[2]

C[4] B[2]

c) Caso um vetor ( A ou B ) termine antes do outro, o vetor C deverá ser preenchido com os elementos do vetor que ainda possui informações.

24) Faça um algoritmo que:

a) Leia um vetor A com N elementos já ordenados e um vetor B com M elementos também já ordenados.

b) Intercale os dois vetores A e B, formando um vetor C, sendo que ao final do processo de intercalação, o vetor C continue ordenado. Nenhum outro processo de ordenação poderá ser utilizado além da intercalação dos vetores A e B.

c) Caso um vetor ( A ou B ) termine antes do outro, o vetor C deverá ser preenchido com os elementos do vetor que ainda possui informações.

25) Faça um algoritmo que:

a) Leia um vetor com N elementos formado por valores do tipo inteiro.

b) Após a leitura, modifique o vetor de forma que o mesmo contenha na parte superior somente valores pares, e na parte inferior os valores ímpares.

c) Ordene crescentemente os números pares, e decrescentemente os números ímpares.

26) Dado um conjunto de n registros, cada registro contendo o nome e a idade dessa pessoa faça um algoritmo que:

a) Guarde todos os dados na memória;

b) Escreva o nome das pessoas que tem idade maior que a idade da primeira;

c) Listar os nomes das pessoas que tem a idade maior que a média;

d) Listar o nome da pessoa de maior e menor idade

27) Dado que para cada aluno de uma turma de "N" alunos se tenha, o seu nome, e as notas das 8 avaliações. Faça um algoritmo que:

Imprima o nome a média de cada aluno;

a) Calcule a Percentagem de alunos cujo o nome comece com a letra "A";

b) Para cada aluno imprima uma mensagem dizendo se o aluno tem ou não notas repetidas;

c) Determine quantos alunos tem pelo menos duas notas acima de 7;

28) Dado um vetor X de n elementos faça um algoritmo que:

a) Crie outro vetor Y contendo os elementos de x que estão na faixa entre 10 e 40;

b) Crie outro vetor W contendo os números que estão nas posições pares;

c) Pesquise a existência de um determinado elemento Y no vetor X;

d) Escreva o menor e maior elemento do vetor X;

29) Em uma cidade do interior, sabe-se que, de janeiro a abril de 1976 (121 dias), não ocorreu temperatura inferior a 15o C nem superior a 40o C. As temperatura verificadas em cada dia estão disponíveis em uma unidade de entrada de dados.

Fazer um algoritmo que calcule e escreva:

- a) a menor temperatura ocorrida;
- b) a maior temperatura ocorrida;
- c) a temperatura média
- d) o número de dias nos quais a temperatura foi inferior a média à temperatura média.

30) Fazer um algoritmo que:

- a) leia o valor de n ( $n \leq 1000$ ) e os n valores de um variável indexada A de valores inteiros ordenados de forma crescente;
- b) determine e escreva, para cada número que se repete no conjunto, a quantidade de vezes em que ele aparece repetido;
- c) elimine os elementos repetidos, formando um novo conjunto;
- d) escreva o conjunto obtido no item c;

31) Um armazém trabalha com 100 mercadorias diferentes identificadas pelos números inteiros de 1 a 100. O dono do armazém anota a quantidade de cada mercadoria vendida durante o mês. Ele tem uma tabela que indica, para cada mercadoria, o preço de venda. Escreva um algoritmo para calcular o faturamento mensal do armazém. A tabela de preços é fornecida seguida pelos números das mercadorias e as quantidades vendidas. Quando uma mercadoria não tiver nenhuma venda, é informado o valor zero no lugar da quantidade.

32) Uma grande firma deseja saber quais os três empregados mais recentes. Fazer um algoritmo para ler um número indeterminado de informações (máximo de 300) contendo o número do empregado e o número de meses de trabalho deste empregado e escrever os três mais recentes. Observações: A última informação contém os dois números iguais a zero. Não existem dois empregados admitidos no mesmo mês.

Exemplo:

EMPREGADOS

224 1731 2210 4631 ... 526

1 2 3 4 300

MESES

17 3 9 2 ... 10

1 2 3 4 300

Empregado mais recente: 4631.

33) Fazer um algoritmo que:

- a) leia uma variável indexada A com 30 valores reais;
- b) leia uma outra variável indexada B com 30 valores reais;
- c) leia o valor de uma variável X;
- d) verifique qual o elemento de A é igual a X;
- e) escreva o elemento de B de posição correspondente à do elemento A igual a X;

34) Fazer um programa que:

- a) leia o valor de M ( $M \leq 30$ ) e os valores de uma variável indexada A;
- b) leia o valor de N ( $N \leq 20$ ) e os valores de uma variável indexada B;
- c) determine o conjunto  $C = A \cup B$  (união de A com B), onde C não deverá conter elementos repetidos (A e B não contém elementos repetidos);
- d) escreva os elementos contidos em A, B e C;

35) Intercalação é o processo utilizado para construir uma tabela ordenada, de tamanho  $n + m$ , a partir de duas tabelas já ordenadas de tamanhos  $n$  e  $m$ . Por exemplo, a partir das tabelas:

A= 1 3 6 7

e

B= 2 4 5

construímos a tabela

C= 1 2 3 4 5 6 7

Fazer um algoritmo que:

- leia  $NA$ , número de elementos do conjunto A ( $NA \leq 100$ );
- leia, em seguida, os elementos do conjunto A;
- leia, logo após o valor de  $NB$ , número de elementos do conjunto B ( $NB \leq 100$ );
- leia, finalmente, os elementos do conjunto B;
- crie e imprima um conjunto C, ordenado, de tamanho  $NA + NB$ , a partir dos conjuntos originais A e B.

Observação:

- Considerar os elementos de A e B como inteiros.

36) Para cada nota de compra, tem-se o Nome do produto comprado, o valor e o imposto. Faça um algoritmo que escreva Nome, valor e imposto de cada produto. Bem como o valor total bruto, o imposto total cobrado e o valor total líquido de todas as notas. Considere 500 notas

37) Uma empresa deseja fazer um levantamento do seu estoque. Para isto possui para cada tipo de produto, o seu Nome, o valor unitário e a sua quantidade em estoque. A empresa deseja emitir um relatório contendo, o Nome, quantidade, valor, unitário e o saldo (valor total) em estoque de cada tipo de produto. Ao final deseja-se saber quantos tipos de produtos existem em estoque e o saldo de todos os produtos.

## 10.2 Variáveis Indexadas Bidimensionais (Matrizes)

Também conhecida por "Matriz". Uma variável Bidimensional, como o próprio Nome já indica, possui duas dimensões, sendo ser possível definir variáveis com quaisquer tipo de dados válidos do Pascal.

a11 a12

A = a21 a22

a31 a33

Matriz A 3x2 (Linha x Coluna)

a) Definição:

ALGORITMO Define

VARIÁVEIS

<Nome>: MATRIZ [I : J, L:M] DE <tipo>

INICIO

<Comandos>

FIM

Em pascal:

PROGRAM Define;

VAR

<Nome>: ARRAY [1..J, 1..M] OF <tipo>;

BEGIN

<Comandos>;  
END.

Exemplo Definir uma variável indexada bidimensional para armazenar os dados de uma matriz 4 por 4 de números do tipo REAL, sendo que a mesma deverá corresponder no total a 16 posições de memória.

ALGORITMO Exemplo;  
VARIÁVEIS  
MAT : MATRIZ [1:4,1:4] DE REAL  
INICIO  
<Comandos>  
FIM  
Em pascal:  
PROGRAM Exemplo;  
VAR  
MAT: ARRAY [1..4,1..4] OF REAL;  
BEGIN  
<Comandos>;  
END.

No Exemplo acima, após a definição da variável, a memória estará como mostrado no esquema abaixo:

MAT

1,1 1,2 1,3 1,4  
2,1 2,2 2,3 2,4  
3,1 3,2 3,3 3,4  
4,1 4,2 4,3 4,4

Os Valores numéricos apresentados acima correspondem aos índices da variável.

B) Atribuição

<Nome>[<Índice>,<Índice>] Valor;  
LEIA (<Nome>[<Índice>,<Índice>]);

Em pascal:

<Nome>[<Índice>,<Índice>] :=Valor;  
READ (<Nome>[<Índice>,<Índice>]);

Exemplo:

ALGORITMO Atribui  
VARIÁVEIS  
Nomes : VETOR [1:4,1:4] DE CADEIA  
I,J : INTEIRO  
BEGIN  
PARA I 1 ATE 4, PASSO 1, FAÇA  
PARA J 1 ATE 4, PASSO 1, FAÇA  
LEIA (Nomes [ I , J ] )  
FIM PARA  
FIM PARA  
FIM  
Em pascal:  
PROGRAM Atribui;  
VAR

```

Nomes : ARRAY[1..4,1..4] OF STRING;
I,J : INTEGER;
BEGIN
FOR I: = 1 TO 4 DO
FOR := J := 1 TO 4 DO
READ (Nomes[ I,J ]);
END.

```

### 10.2.1 Exercícios 9:

- 1) Faça um algoritmo para ler e imprimir uma matriz 2x4 de números inteiros.
- 2) Dado uma matriz de ordem 3x3 faça um algoritmo que:
  - a) Calcule a soma dos elementos da primeira coluna;
  - b) Calcule o produto dos elementos da primeira linha;
  - c) Calcule a soma de todos os elementos da matriz;
  - d) Calcule a soma do diagonal principal;
- 3) Dado uma matriz de ordem NxN faça um algoritmo que verifique se a matriz é simétrica ( $a_{ij}=a_{ji}$ ).
- 4) Dado uma matriz NxM de valores reais faça um algoritmo que faça a leitura destes valores e ao final da leitura de todos, imprimir o seguintes relatório:
  - a) Qual a Soma dos valores de cada coluna da matriz;
  - b) Listar os valores que são menores que a média dos valores;
  - c) Qual a soma dos elementos da diagonal secundária;
- 5) Dado uma matriz NxM de valores inteiros faça um algoritmo que faça a leitura destes valores e ao final coloque os elementos ordenados primeiro pela linha e depois pela coluna.
- 6) Dado duas matrizes A e B de ordem NxN faça um algoritmo que some as duas e gere a matriz C. Os elementos da matriz C são a soma dos respectivos elementos de A e B.
- 7) Dado uma matriz NxM de valores inteiros determine a sua matriz transposta e imprima.
- 8) Fazer um algoritmo que efetue um produto matricial. Seja A(m x n) e B (n x m) as matrizes fatores, sendo  $m \leq 40$  e  $n \leq 70$ . Deverão ser impressas as matrizes A, B e a matriz-produto obtida.

### 11. Funções pré-definidas do Pascal

O Pascal, possui muitas funções/Rotinas pré-definidas, sendo que iremos estudar algumas das principais. Caso haja necessidade, o manual da linguagem Pascal pode, e deve, ser usado como fonte de pesquisa de outras funções.

Dentre as Rotinas que iremos abordar, algumas necessitam que um comando especial seja colocado no início do PROGRAMA. Tal comando é mostrado abaixo:

```

PROGRAM Teste;
USES CRT;
BEGIN
<Comandos>;
END.

```

Obs.: As funções que necessitarem o comando especial mostrado acima terão colocados ao lado a palavra CRT.

11.1 Descrição das Funções/Rotinas:

Rotina : ABS()

Função : Retorna o valor absoluto de um valor numérico.

Sintaxe : Resultado: =ABS(Valor)

Exemplo:

```
PROGRAM Teste;
VAR
X1 : REAL;
X2 : INTEGER;
BEGIN
X1: =ABS( -2.3 ); ( 2.3 )
X2: =ABS( -157 ); ( 157 )
END.
```

Rotina : CHR()

Função : Retorna um caracter da tabela ASCII de acordo com um determinado valor numérico

Sintaxe : Resultado: =CHR(Valor)

Exemplo:

```
PROGRAM Teste;
VAR
X1 : CHAR;
BEGIN
X1: =CHR( 65 ); ( 'A' )
END.
```

Rotina : CLRSCR

Função : Limpa a tela de vídeo

Sintaxe : CLRSCR

Exemplo:

```
PROGRAM Teste;
USES CRT;
BEGIN
CLRSCR;
END.
```

Rotina : CONCAT()

Função : Concatena ( Junta ) uma seqüência de STRING's

Sintaxe : Resultado: =CONCAT( s1,s2,...,sn)

Exemplo:

```
PROGRAM Teste;
VAR
s1,s2 : STRING;
BEGIN
s1: =CONCAT('João','Mineiro'); ('João Mineiro' )
s2: = CONCAT('ABC','DEFG','HIJ'); ('ABCDEFHIJ')
END.
```

Rotina : COPY()

Função : Copia n caracteres de uma STRING a partir de uma posição específica

Sintaxe : Resultado: =COPY(s1, posição , quantidade)

Exemplo:

```
PROGRAM Teste;
VAR
s1 :STRING;
BEGIN
s1: =COPY('ABCDEFGH',2,3); ('BCD')
END.
```

Rotina : COS()

Função : Retorna o cosseno de um valor numérico

Sintaxe : Resultado: = COS(Valor)

Exemplo:

```
PROGRAM Teste;
VAR
x : REAL;
BEGIN
x: =COS(10);
END.
```

Rotina : DEC()  
Função : Decrementa uma variável numérica  
Sintaxe : DEC(Valor)  
Exemplo:  
PROGRAM Teste;  
VAR  
x : INTEGER;  
BEGIN  
x: =10;  
DEC(x) ; ( 9 )  
END.

Rotina : DELAY() ( CRT )  
Função : Interrompe o processamento por um número especificado de milissegundos  
Sintaxe : DELAY(Tempo)  
Exemplo:  
PROGRAM Teste;  
USES CRT;  
BEGIN  
DELAY(200);  
END.

Rotina : DELETE()  
Função : Deleta n caracteres de uma STRING, a partir de uma posição inicial  
Sintaxe : DELETE(s ,posição, quantidade)  
Exemplo:  
PROGRAM Teste;  
VAR  
s : STRING;  
BEGIN  
s: ='João da Silva';  
DELETE (s,5,3); ('João Silva')  
END.

Rotina : EXP()  
Função : Retorna "e" elevado a um determinado valor numérico  
Sintaxe : Resultado: =EXP(Valor)  
Exemplo:  
PROGRAM Teste;  
VAR  
x : REAL;  
BEGIN  
x: : =EXP(10);  
END.

Rotina : FRAC()  
Função : Retorna a parte fracionária de um valor numérico  
Sintaxe : Resultado: =FRAC(Valor)  
Exemplo:  
PROGRAM Teste  
VAR  
x: REAL  
BEGIN  
x: =FRAC(2.345) ( 345 )  
END

Rotina : GOTOXY() ( CRT )  
Função : Posiciona o cursor no vídeo em uma determinada coluna ( x ) e linha ( y )  
Sintaxe : GOTOXY( coluna, linha )  
Exemplo:  
PROGRAM Teste;  
USES CRT;  
BEGIN  
GOTOXY(10,20);  
END.

Rotina : INC()  
Função : Incrementa uma variável numérica



Sintaxe : INC(Valor)

Exemplo:

```
PROGRAM Teste;
```

```
VAR
```

```
x : INTEGER;
```

```
BEGIN
```

```
x: =10;
```

```
INC(x); (11)
```

```
END.
```

Rotina : INSERT()

Função : Insere uma STRING dentro de outra STRING a partir de uma determinada posição

Sintaxe : INSERT(STRING\_Fonte, STRING\_Destino, posição)

Exemplo:

```
PROGRAM Teste;
```

```
VAR
```

```
s : STRING;
```

```
BEGIN
```

```
s: ='João Silva';
```

```
INSERT('Da ',s,6); ('João Da Silva')
```

```
END.
```

Rotina : INT()

Função : Retorna a parte inteira de um valor numérico

Sintaxe : REsultado: =INT(Valor)

Exemplo:

```
PROGRAM Teste;
```

```
VAR
```

```
x : REAL;
```

```
BEGIN
```

```
x: =INT(2.345); (2)
```

```
END.
```

Rotina : KEYPRESSED ( CRT )

Função : Retorna TRUE se uma tecla foi pressionada, FALSE case contrário

Sintaxe : KEYPRESSED

Exemplo:

```
PROGRAM Teste;
```

```
USES CRT;
```

```
BEGIN
```

```
REPEAT UNTIL KEYPRESSED;
```

```
END.
```

Rotina : LENGTH()

Função : Retorna o número de caracteres de uma STRING

Sintaxe : Resultado: = LENGTH(s)

Exemplo:

```
PROGRAM Teste;
```

```
VAR
```

```
x : INTEGER;
```

```
BEGIN
```

```
x: = LENGTH('João'); ( 4 )
```

```
END.
```

Rotina : LN()

Função : Retorna o logaritmo natural de um determinado valor numérico

Sintaxe : Resultado: =LN(Valor)

Exemplo:

```
PROGRAM Teste;
```

```
VAR
```

```
x : REAL;
```

```
BEGIN
```

```
x: =EXP(LN(2)*3); (* 2**3 ( 8 ) *)
```

```
END.
```

Rotina : ODD()

Função : Retorna TRUE se uma determinado valor numérico FOR impar, FALSE caso contrário

Sintaxe :

Exemplo:  
PROGRAM Teste;  
VAR  
x : Boolean;  
BEGIN  
c: =odd(65); ( TRUE )  
END.

Rotina : ORD()  
Função : Retorna qual o número na tabela ASCII de um determinado caracter  
Sintaxe : Resultado: =ORD(Caracter)  
Exemplo:  
PROGRAM Teste;  
VAR  
x : INTEGER;  
BEGIN  
x: =ORD('A'); ( 65 )  
END.

Rotina : PI  
Função : Retorna o valor de PI  
Sintaxe : Resultado: =PI  
Exemplo:  
PROGRAM Teste;  
VAR  
x : REAL;  
BEGIN  
x: =PI; ( 4.1415926535... )  
END.

Rotina : POS()  
Função : Retorna a posição da primeira ocorrência dos caracteres de uma STRING dentro de outra STRING  
Sintaxe : Resultado: =POS(Caracteres,STRING\_A\_SER\_PROCURADA)  
Exemplo:  
PROGRAM Teste;  
VAR  
x : INTEGER;  
BEGIN  
x: =POS('ABC','DEABCFG'); ( 3 )  
END.

Rotina : READKEY ( CRT )  
Função : Faz a leitura de um caracter do teclado, não sendo necessário pressionar ENTER  
Sintaxe : Resultado: =READKEY  
Exemplo:  
PROGRAM Teste;  
USES CRT;  
VAR  
tecla: CHAR;  
BEGIN  
tecla: =READKEY;  
END.

Rotina : SIN()  
Função : Retorna o seno de um valor numérico  
Sintaxe : Resultado: =SIN(Valor)  
Exemplo:  
PROGRAM Teste;  
VAR  
x : REAL;  
BEGIN  
x : = SIN(10);  
END.

Rotina : SQR()  
Função : Retorna um valor numérico elevado ao quadrado

Sintaxe : Resultado: =SQR(Valor)

Exemplo:

```
PROGRAM Teste;
VAR
x : INTEGER;
BEGIN
x: =SQR(3); ( 9 )
END.
```

Rotina : SQRT()

Função : Retorna a raiz quadrada de um valor numérico

Sintaxe : Resultado: =str(Valor)

Exemplo:

```
PROGRAM Teste
VAR
x: INTEGER
BEGIN
x: =SQRT(9) ( 3 )
END
```

Rotina : STR()

Função : Converte um valor numérico para a sua representação em STRING

Sintaxe : STR(Valor, STRING\_Resultante)

Exemplo:

```
PROGRAM Teste;
VAR
s : STRING;
BEGIN
STR(2.345,s);
END.
```

Rotina : TEXTBACKGROUND() ( CRT )

Função : Altera a cor de fundo nas operações de E/S

Sintaxe : TEXTBACKGROUND(Cor)

Exemplo:

```
PROGRAM Teste;
USES CRT;
BEGIN
TEXTBACKGROUND(0); (Preto)
TEXTBACKGROUND(1); (Azul)
TEXTBACKGROUND(2); (Verde)
TEXTBACKGROUND(3); (Ciano)
TEXTBACKGROUND(4); (Vermelho)
TEXTBACKGROUND(5); (Mangenta)
TEXTBACKGROUND(6); (Marrom)
TEXTBACKGROUND(7); (Cinza)
END.
```

Rotina : TEXTCOLOR() ( CRT )

Função : Altera a cor das letras nas operações de E/S

Sintaxe : TEXTCOLOR(Cor)

Exemplo:

```
PROGRAM Teste;
USES CRT;
BEGIN
TEXTCOLOR (0); (Preto)
TEXTCOLOR(1); (Azul)
TEXTCOLOR (2); (Verde)
TEXTCOLOR(3); (Ciano)
TEXTCOLOR(4); (Vermelho)
TEXTCOLOR(5); (Mangenta)
TEXTCOLOR(6); (Marrom)
TEXTCOLOR(7); (Cinza)
END.
```

Rotina : TRUNC()

Função : Trunca um valor REAL para um valor Inteiro

Sintaxe : Resultado: =TRUNC(Valor)

Exemplo:

```
PROGRAM Teste;
VAR
x : INTEGER;
BEGIN
x: =TRUNC(2.345); ( 2 )
END
```

Rotina : UPCASE()  
Função : Converte um caracter minúsculo para maiúsculo  
Sintaxe : Resultado: =UPCASE(Caracter)  
Exemplo:  
PROGRAM Teste;  
VAR  
x : CHAR;  
BEGIN  
x: =UPCASE('a'); ( 'A' )  
END.

Rotina : VAL()  
Função : Converte uma STRING para a sua representação numérica  
Sintaxe : VAL(String\_ORIGEM,VALOR\_RESPOSTA,CODIGO\_ERRO)  
Exemplo:  
PROGRAM Teste;  
VAR  
x, e : INTEGER;  
BEGIN  
VAL('12345',x,e); ( 12345 )  
END.

## 12. Problemas Resolvidos em Algoritmos, Pascal e C

Alguns Exemplos de problemas resolvidos em Algoritmos, Pascal e C.

### 12.1 Algoritmo do Triangulo

Faça um algoritmo para ler a base e a altura de um triângulo. Em seguida, escreva a área do mesmo.

Obs: Área = ( Base \* Altura ) / 2

Em Algoritmo:

```
Algoritmo Triangulo
Variaveis:
base, altura, area : real;
Inicio
Leia(Base)
Leia(Altura)
Area (Base * Altura)/2
Escreva (Area)
Fim
```

Em Pascal:

```
program triangulo;
var
area, base, altura: real;
```

```

begin
{ Entrada }
write ('Digite a base: ');
readln (base);
write ('Digite a altura: ');
readln (altura);

{ Calculos }
area:= (base*altura)/2;

{ Saida}
writeln ('A area do triangulo e: ',area:10:2);
end.

```

Em C:

```

#include <stdio.h>

main ()
{
/* declaracao de variaveis */
float area,base,altura;
/* Entrada de dados */
printf ("Digite base: ");
scanf ("%f", &base) ;
printf ("Digite altura: ");
scanf ("%f", &altura);
/* Calculos */
area = (base*altura)/2;

/* Resultados */
printf ("A area do triangulo e: %10.2f\n", area);
}

```

## 12.2 Algoritmo Preço Automóvel

O preço de um automóvel é calculado pela soma do preço de fábrica com o preço dos impostos (45% do preço de fábrica) e a percentagem do revendedor (28% do preço de fábrica). Faça um algoritmo que leia o nome do automóvel e o preço de fábrica e imprima o nome do automóvel e o preço final.

Em Algoritmo:

```

Algoritmo automovel
Variaveis
Nome : cadeia
Precofabrica, Precofinal, Imposto : real
Inicio
Leia (Nome, Precofabrica)
Imposto Precofabrica * (0,45 + 0,28)
Precofinal Precofabrica + Imposto
Escreva (Nome, Precofinal)
Fim

```

Em Pascal:

```

program preco;
var
nome: string;
Precofabrica, PrecoFinal, imposto : real;
begin
write ('Digite nome: ');
readln (nome);

```

```

write ('Digite preco de fabrica: ');
readln (Precofabrica);
Imposto := Precofabrica * (0.45 + 0.28);
PrecoFinal := PrecoFabrica + imposto;
writeln ('Automovel:', nome);
writeln ('Preco de venda:', PrecoFinal:10:2);
end.

```

Em C:

```

#include <stdio.h>

main ()
{
/* declaracao de variaveis */
char nome[60];
float precofabrica, precofinal, imposto;

/* Entrada de dados */
printf ("Digite nome: ");
scanf ("%s", &nome);
printf ("Digite preco de fabrica: ");
scanf ("%f", &precofabrica);

/* Calculos */
imposto = precofabrica *(0.45 + 0.28);
precofinal = precofabrica + imposto;

/* Resultados */
printf ("Automovel: %s\n", nome);
printf ("Preco de venda: %10.2f\n", precofinal);
}

```

### 12.3 Algoritmo Media Valor

Dado uma série de 20 valores reais, faça um algoritmo que calcule e escreva a média aritmética destes valores, entretanto se a média obtida for maior que 8 deverá ser atribuída 10 para a média.

```

R.
Algoritmo Serie
Variaveis
Contador: Inteiro
N,Soma, Media :Real
Inicio
Contador 0
Soma 0
Enquanto Contador < 20 faca
Leia (N)
Soma Soma + N
Contador Contador + 1
Fim Enquanto
Media Soma / 20
Se Media > 8 Entao
Media 10
Fim Se
Escreva (Media)
Fim

```

Em Pascal:

```

Program MEDIA_20;
Uses crt;
var
conta:integer;

```

```

media,num,soma:real;
Begin
Clrscr;
conta := 0;
soma := 0;
Writeln ('Digite 20 numeros');
While conta < 20 do
Begin
read (num);
soma := soma + num;
conta := conta + 1;
End;
media := (soma / 20);
if media > 8 then
Writeln ('media =10');
else
Writeln ('A media ,');
Writeln (media);
End.

```

Em C:

```

#include<stdio.h>
#include<conio.h>
main()
{
float media,num,soma;
int cont;
soma=0;
cont=0;
clrscr();
printf("\nDigite 20 numeros\n");
while (cont<20)
{
scanf("%f",&num);
soma=soma+num;
num=soma;
cont++;
}
media=num/20;
if (media > 8)
{
printf("Media , igual a 10");
}
else
{
printf("Media , igual a: %2.2f\n",media);
}
}
return(0);
}

```

## 12.4 Algoritmo 3 Menores

Faça um algoritmo que leia 3 números inteiros e imprima o menor deles.

Em Algoritmo:

```

Algoritmo Menor
Variaveis
A, B, C, Menor : Inteiro
Inicio
Leia ( A,B,C)
Se (A < B) E (A <C) Entao
Menor A
Senao

```

```
Se B < C Entao
Menor B
Senao
Menor C
Fim se
Fim se
Escreva (Menor)
Fim
```

Em Pascal:

```
Program MENOR_N;
Uses crt;
var
n1,n2,n3,menor:integer;
Begin
Clrscr;
Writeln ('Digite 3 valores');
Read (N1,N2,N3);
If (N1<N2) And (N1<N3) Then
menor:=N1
Else
If (N2<N1) And (N2<N3) Then
menor:=N2
Else
menor:=N3
Writeln ('O menor valor ,');
Writeln (menor);
End.
```

Em C:

```
#include<stdio.h>
#include<conio.h>
main()
{
int n1,n2,n3;
clrscr();
printf("\nDigite 3 valores\n");
scanf("%d",&n1);
scanf("%d",&n2);
scanf("%d",&n3);
if ((n1<n2) && (n1<n3))
{
printf("\nO menor numero , :%d\n",n1);
}
else
{
if ((n2<n1) && (n2<n3))
{
printf("\nO menor numero , :%d\n",n2);
}
else
{
printf("\nO menor numero , :%d\n",n3);
}
}
return(0);
}
```

## 12.5 Algoritmo Media Maior que 4

Dado um conjunto de n registros cada registro contendo um valor real, faça um algoritmo que calcule a média dos valores maiores que 4.



## Em Algoritmo:

```
Algoritmo Media
Variaveis
N, Conta, Maior4 : Inteiro
Soma, Media, Numero :Real
Inicio
Conta 1
Maior4 0
Soma 0
Leia (N)
Enquanto Conta < N Faça
Leia (Numero)
Se Numero > 4 Entao
Soma Soma + Numero
Maior4 Maior4 + 1
Fim Se
Conta Conta + 1
Fim Enquanto
Media Soma / Maior4
Escreva (Media)
Fim
```

## Em Pascal:

```
program maior4;
uses
CRT;
var
n, aux: integer;
soma, media, numero: real;
begin
Clrscr;
write ('Digite valores diferentes de 999');
writeln;
aux := 0;
soma := 0;
read (numero);
while numero <> 999 do
begin
if numero > 4 then
begin
soma:= soma + numero;
aux := aux + 1;
end;
read (numero);
end;
media := (soma/aux);
write ('MEDIA=');
write (media);
end.
```

## Em C:

```
#include <iostream.h>
#include <conio.h>
main ()
{
clrscr();
int n;
float soma,media,numero,aux;
cout <<"digite valores diferentes de 999 -> ";
cin >> numero;
aux =1;
soma =0;
while (numero != 999)
{
if (numero > 4)
{
soma = soma + numero;
aux =aux+1; }
}
```

```

media = (soma/aux);
cout << "\nEntre com o proximo ->";
cin >> numero;
}
cout << media;
return 0;
}

```

## 12.6 Algoritmo Salário

Uma empresa tem para um determinado funcionário uma ficha contendo o nome, número de horas trabalhadas e o nº de dependentes de um funcionário.

Considerando que:

a) A empresa paga 12 reais por hora e 40 reais por dependentes.

Sobre o salário são feito descontos de 8,5% para o INSS e 5% para IR.

Faça um algoritmo para ler o Nome, número de horas trabalhadas e número de dependentes de um funcionário. Após a leitura, escreva qual o Nome, salário bruto, os valores descontados para cada tipo de imposto e finalmente qual o salário líquido do funcionário.

Em Algoritmo:

```

Algoritmo salario
variaveis
nome : cadeia
dependente : inteiro
horas, salariol, salariob, Descontol, DescontoR : real;
Inicio
Leia(nome,horas, dependente)
salariob (12 * horas) + (40 * dependente)
Descontol (salariob*0,085)
DescontoR (salariob*0,05)
salariol (salariob - Descontol, DescontoR)
Escreva(nome,salarioB, Descontol, DescontoR, salariol)
Fim

```

Em Pascal:

```

Program Salario;
Uses
Crt;
Var
Nome:String;
Numhora, Salbruto, Salliq:Real;
Numdep:Integer;
Begin
Clrscr;
Write ('Digite O Nome Do Funcionario:');
Read (Nome);
Writeln;
Write ('Numero De Horas Trabalhadas:');
Read (Numhora);
Writeln;
Write ('E Numero De Dependentes:');
Read (Numdep);
Writeln;
Salbruto:=(12*Numhora)+(40*Numdep);
Salliq:=Salbruto-((Salbruto*0.085)+(Salbruto*0.05));
Write ('Nome Do Funcionario:');
Write (Nome);
Writeln;
Write ('Salario Liquido:');
Write (Salliq);
End.

```

Em C:

```
#include <iostream.h>
#include <conio.h>
#include <stdio.h>

main ()
{
clrscr();
int numdep;
float numhora,salbruto,salliq;

char nome[20];
cout << "Digite o nome do funcionario -> ";
scanf ("%s", &nome);
cout << "Numero de horas trabalhadas -> ";
scanf ("%f", &numhora);
cout << "Numero de dependentes -> ";
scanf ("%f", &numdep);

salbruto = (12*numhora)+(40*numdep);
salliq = salbruto -((salbruto*0.085)+(salbruto*0.05));

cout << "\nNome do Funcionario: "<< nome;
cout << "\nSalario Liquido: " << salliq;
return 0;
}
```

## 12.7 Algoritmo 50 Triangulos

1) Faça um algoritmo para ler base e altura de 50 triângulos e imprimir a sua área.

Em Algoritmo:

```
Algoritmo Triangulo
Variaveis
Contador: Inteiro
Base, Haltura, Area :Real
Inicio
Contador 1
Enquanto Contador < 51 faca
Leia (Base
Leia (Haltura)
Area (Base * Haltura)/2
Escreva( Area)
Contador Contador + 1
Fim Enquanto
Fim
```

Em Pascal:

```
program triangulo;
uses crt;
var
base,altura,area:real;
contador:integer;
begin
clrscr;
contador:=1;
while contador < 51 do
```

```

begin
writeln('Digite a Base');
read(base);
writeln('Digite a Altura');
read(altura);
area:=(base*altura)/2;
write('Esta e a area do triangulo ');
writeln (area);
contador:=contador+1;
end;
repeat until keypressed;
end.

```

Em C:

```

/*program triangulo*/
#include <stdio.h>

main ()
{
float base,altura,area;
int contador = 1;

while (contador < 51) {
printf ("\nDigite a Base\n");
scanf ("%f",&base);
printf ("\nDigite a Altura\n");
scanf ("%f",&altura);
area=(base*altura)/2;
printf ("\nA area do triangulo de numero %d eh %f",area);
contador++;
}
return (0);
}

```

## 12.8 Algoritmo Media Menores

Dado um conjunto de 20 valores reais, faça um algoritmo que:

- Imprima os valores que não são negativos.
- Calcule e imprima a média dos valores < 0.

Em Algoritmo:

```

Algoritmo Conjunto20
Variaveis
N, SomaMenor, MediaMenor : Real
ContaMenor, Contador : Inteiro
Inicio
Contador 0
ContaMenor 0
SomaMenor 0
Enquanto Contador < 20 faca
Leia(N)
Se N >= 0 Entao
Escreva (N)
Senao
SomaMenor SomaMenor + N
ContaMenor ContaMenor + 1
Fim Se
Contador Contador + 1
Fim Enquanto
MediaMenor SomaMenor / ContaMenor
Fim

```

Em Pascal:

```

program numeros;
uses crt;

```

```

var
valor,media,soma:real;
cont,contpos:integer;
begin
clrscr;
cont:=1;
while cont < 21 do
begin
writeln('Digite um valor real');
read(valor);
if valor < 0 then
begin
media:=media+valor;
contpos:=contpos+1;
end
else
begin
write (valor);
writeln (' e um valor maior que Zero');
end;
cont:=cont+1;
end;
media:=media/contpos;
write ('O numero de valores menores que Zero sao ');
writeln (contpos);
write ('A media do valores menores que Zero e ');
writeln (media);
repeat until keypressed;
end.

```

Em C:

```

/* numeros */
#include <stdio.h>

main()
{
float valor ,media;
int cont = 1,
contpos= 0;

while (cont < 21) {
printf ("\nDigite um valor real\n");
scanf ("%f", &valor);
if (valor < 0) {
media=media+valor;
contpos++;
}
else
{
printf ("\n %f",valor);
printf (" e um valor maior que Zero");
}
cont++;
}
media = media / contpos;
printf ("\nO numero de valores menores que Zero eh %d",contpos);
printf ("\nA media dos valores menores que Zero e %f",media);
return (0);
}

```