

Capítulo 3

Estruturas de seleção

CONCEITOS

Nossos algoritmos até agora seguiram um mesmo padrão: entrava-se com dados, estes eram processados e alguma informação era mostrada na tela.

Dessa forma, o computador mais parecia uma máquina de calcular. O aprendizado de novos conceitos, como a estrutura de seleção, nos dará uma visão maior da complexidade de tarefas que ele poderá executar.

Vamos refletir sobre a importância dessa estrutura, lendo com atenção as afirmativas a seguir:

1. Distribuição gratuita de cestas básicas.
2. Distribuição gratuita de cestas básicas para famílias com 4 ou mais componentes.
3. Distribuição gratuita de ingressos para o teatro, sendo dois para pessoas do sexo feminino e um para pessoas do sexo masculino.

Se observarmos essas afirmativas podemos concluir que:

- Na primeira, todas as pessoas recebem a cesta básica, o que equivale-ria a um comando seqüencial.
- Na segunda, só recebem as cestas básicas as famílias com pelo menos quatro integrantes.
- Na terceira, dependendo do sexo, recebe-se um ou dois ingressos.

Assim, podemos avaliar a importância do teste nas duas últimas afir-mativas, pois ações diferentes são executadas de acordo com o resultado.

Um exemplo do nosso dia-a-dia: imagine-se diante de um caixa eletrônico e suponha que sua senha seja 1234:

Na tela aparece a mensagem: O cursor (■ ou) fica piscando:	Digite sua senha: ■	
Você digita os algarismos da sua senha	1234	1233
Neste momento, a Unidade Aritmética e Lógica (um dos componentes da CPU) verifica se os números que você digitou são iguais a 1234. Caso tenham sido, aparece na tela a mensagem: VÁLIDA; mas se você digitou algo diferente, aparece na tela a mensagem: INVÁLIDA.	VÁLIDA	INVÁLIDA

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.

SINTAXES

Seleção Simples

```
se ( condição )
{
    comando ; ou
    < seqüência de comandos separados por ; >
}
```

A sintaxe acima representa a afirmativa 2, pois se a família tiver, no mínimo, quatro componentes, recebe a cesta básica; mas se a família tiver menos que quatro componentes, não recebe nada.

Seleção Composta

```
se ( condição )
{
    comando ; ou
    < seqüência de comandos separados por ; >
}
senao
{
    comando ; ou
    < seqüência de comandos separados por ; >
}
```

A sintaxe acima representa a afirmativa 3 onde, dependendo do sexo, recebe-se um ou dois convites.

OBSERVAÇÕES

1. *Podemos constatar que esta estrutura faz parte do nosso cotidiano:*
 - *Se eu não tiver prova, vou à praia; senão vou estudar.*
 - *Se eu tiver aumento, troco de carro; senão espero o 13º salário.*
 - *Se minha média for maior ou igual a sete, passo direto; senão vou à prova final.*
2. *A única coisa diferente é a forma como iremos escrevê-la, pois as chaves { e } são obrigatórias uma vez que delimitam os comandos que pertencem a cada bloco, assim como os parênteses (e) que delimitam a condição.*
3. *Essa é uma estrutura muito importante em algoritmos porque nos dá a possibilidade de verificar o que foi digitado pelo usuário ou qual o conteúdo de uma variável após um processamento etc.*

Vamos analisar cada linha da segunda sintaxe que é a mais completa, pois um ou vários comandos serão executados caso a condição testada seja verdadeira e outros comandos serão executados caso a condição seja falsa.

1ª linha: se (condição)

Condição

- A condição é uma expressão lógica testada pela Unidade Aritmética e Lógica, devolvendo como resposta: verdadeiro ou falso.

Convém aqui recordar os operadores relacionais e os operadores lógicos:

Operadores Relacionais	Usaremos	Operadores lógicos	Usaremos
igual	<code>==</code>	conjunção (e)	<code>&&</code>
diferente	<code>< ></code>	disjunção (ou)	<code> </code>
maior	<code>></code>	negação (não)	<code>!</code>
menor que	<code><</code>		
maior ou igual a	<code>>=</code>		
menor ou igual a	<code><=</code>		

a. A condição pode ser uma simples *expressão relacional* formada de *dois operandos do mesmo tipo* e de *um operador relacional* ($>$, $<$, \geq , \leq , \neq e \neq).

A > B	lê-se: o conteúdo da variável A é maior do que o conteúdo da variável B?
A < 12	lê-se: o conteúdo da variável A é menor do que 12?
resp \neq "S"	lê-se: o conteúdo da variável resp é diferente da letra S?
resp $=\!=$ "BRASIL"	lê-se: o conteúdo da variável resp é igual a BRASIL?

b. A condição pode ser uma *expressão lógica* formada de *pelo menos duas expressões relacionais* que precisarão ser unidas por um dos operadores lógicos ($\&\&$ ou $\|$).

A \geq 1 $\&\&$ A $<$ 9	lê-se: o conteúdo da variável A é maior ou igual a 1 e menor que 9?
resp $=\!=$ "S" $\ $ resp $=\!=$ "s"	lê-se: o conteúdo da variável resp é igual a S ou igual a s?

Você deve estar perguntando: e os parênteses, não são necessários ?

Não, porque as operações aritméticas têm maior prioridade, depois as relacionais e por último as lógicas; mas, *cuidado*, nem todas as linguagens agem dessa maneira.

Você deverá ter uma atenção especial quando unir vários operadores lógicos para fazer seus testes. Veremos isso mais adiante.

2^a linha: {

Indica o início do bloco caso a condição testada seja verdadeira.

3^a linha: comando ; ou < seqüência de comandos separados por ; >

Nesta parte, são relacionados os comandos que serão executados caso a condição seja verdadeira.

4^a linha: }

Indica o fim do bloco caso a condição testada seja verdadeira.

5^a linha: senao

Este comando faz parte da estrutura do se e só deverá ser usado quando pelo menos uma ação tiver de ser executada se a condição testada for falsa. | 63

algoritmo 83

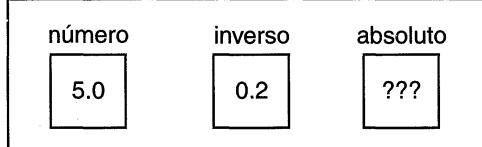
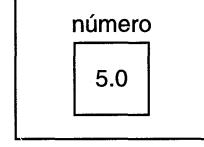
Ler um número e, se ele for positivo, imprimir seu inverso; caso contrário, imprimir o valor absoluto do número.

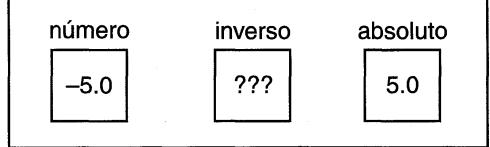
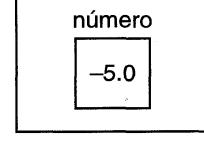
Solução nº 1:

```
prog inversoabsoluto
real numero, inverso, absoluto;
imprima "digite numero:";
leia numero;
se( numero > 0. )
{ inverso <- 1 / numero;
imprima "\nverso: ", inverso;}
senao
{ absoluto <- numero * -1;
# ou absoluto <- abs(numero);
imprima "\nabsoluto: ", absoluto;}
imprima "\n";
fimprog
```

Solução nº 2:

```
prog inversoabsoluto
real numero;
imprima "digite numero:";
leia numero;
se( numero > 0. )
{ imprima "\nverso: ", 1 /
numero; }
senao
{ imprima "absoluto: ", numero *
(-1);
#imprima "absoluto: ", abs(numero);
}
imprima "\n";
fimprog
```

VÍDEO	VÍDEO
Digite numero:5. inverso:0.2	Digite numero:5. inverso:0.2
MP Memória Principal (MP)  A diagram showing the state of memory (MP) for a positive number. It consists of three boxes labeled 'número', 'inverso', and 'absoluto'. The 'número' box contains '5.0', the 'inverso' box contains '0.2', and the 'absoluto' box contains '???'. Below the boxes is the text 'MP Memória Principal (MP)'.	MP Memória Principal (MP)  A diagram showing the state of memory (MP) for a positive number. It consists of one box labeled 'número' containing '5.0'. Below the box is the text 'MP Memória Principal (MP)'.

VÍDEO	VÍDEO
Digite numero:-5. absoluto:5.0	Digite numero:-5. absoluto:5.0
MP Memória Principal (MP)  A diagram showing the state of memory (MP) for a negative number. It consists of three boxes labeled 'número', 'inverso', and 'absoluto'. The 'número' box contains '-5.0', the 'inverso' box contains '???' (with a question mark inside), and the 'absoluto' box contains '5.0'. Below the boxes is the text 'MP Memória Principal (MP)'.	MP Memória Principal (MP)  A diagram showing the state of memory (MP) for a negative number. It consists of one box labeled 'número' containing '-5.0'. Below the box is the text 'MP Memória Principal (MP)'.

algoritmo 84

Ler um número e imprimir se ele é par ou ímpar.

```
prog parimpar
int a;
imprima "\nDigite numero: ";
leia a;
se(a % 2 == 0)
{imprima "\nPAR";}
senao
{imprima "\nIMPAR";}
imprima "\n";
fimprog
```

VÍDEO

Digite numero:24

PAR

Digite numero:25

IMPAR

Se você já entendeu tudo o que foi explicado até aqui, então será capaz de deduzir o que ficará armazenado nas variáveis do trecho do algoritmo a seguir, sabendo-se que elas tanto podem ser do tipo **int**, **real** ou **string**.

DESAFIO

```
•
•
•
se(a > b)
{ aux <- a;
  a <- b;
  b <- aux;
}
se(a > c)
{ aux <- a;
  a <- c;
  c <- aux;
}
```

```

se(b > c)
{ aux <- b;
  b <- c;
  c <- aux;
}
.
.
.
```

Resposta: Na variável *a*, ficará o menor; na variável *b*, o do meio; e, na variável *c*, o maior (se forem números); ou então ficarão as três palavras em ordem alfabética, uma vez que também é uma ordem crescente.

SES ANINHADOS (ENCAIXADOS)

Muitas vezes, em algumas aplicações, sentiremos a necessidade de tomar outras decisões dentro de uma das alternativas da estrutura do **se**; a isso chamamos de **ses aninhados**.

Vejamos um exemplo clássico de algoritmos:

algoritmo 85

```

prog descubra
real a, b, c, max;
imprima "\ndigite 1 numero: ";
leia a;
imprima "\ndigite 2 numero: ";
leia b;
imprima "\ndigite 3 numero: ";
leia c;
se (a > b)
{
  se (a > c)
  { max <- a ; }
  senao
  { max <- c; }
}
senao
{
  se ( b > c)
  { max <- b; }
  senao
  { max <- c ; }
}
imprima "\n",max;
imprima "\n";
fimprog
```

Você descobriu o que faz esse algoritmo?

Resposta: Armazena na variável **max** o maior número entre 3, imprimindo-o.

Desafio: Normalmente, o uso de **ses** aninhados melhora a performance do algoritmo. Será que nesse caso aconteceu isso? Tente usar o trecho do desafio anterior e melhore esta solução. A resposta estará mais adiante.

Outros exemplos:

algoritmo 86

Ler um número e imprimir se ele é positivo, negativo ou nulo.

```
prog pnn
    real num;
    imprima "\nDigite numero: ";
    leia num;
    se(num > 0.)
    {imprima "\nPOSITIVO";}
    senao
    { se(num < 0.)
    { imprima "\nNEGATIVO";}
    senao
    { imprima "\nNULO";}
    }
    imprima "\n";
fimprog
```

VÍDEO

Digite numero:34.

POSITIVO

Digite numero:-12.

NEGATIVO

Digite numero:0.

NULO

Você deve estar fazendo algumas perguntas:

1. Por que não se perguntou se o número era igual a zero?

Resposta: Quando temos a possibilidade de três respostas, só precisamos fazer | 69

duas perguntas, pois a segunda pergunta nos dá as duas últimas respostas. Veja bem: se descartarmos a possibilidade de o número ser maior do que 0, ficamos com duas possibilidades: o número ser igual a 0 ou menor do que 0; dessa forma, uma pergunta é satisfatória.

2. Mas se eu fizer, estarei errado(a)?

Resposta: Não, mas não é necessário.

3. Por que a solução não poderia ser como a seguir?

```
prog se
    real num;
    imprima "\nDigite numero: ";
    leia num;
    se(num > 0.)
    { imprima "\nPOSITIVO";}
    se(num < 0.)
    { imprima "\nNEGATIVO";}
    se(num == 0.)
    { imprima "\nNULO";}
    imprima "\n";
fimprog
```

Resposta: Esta solução, embora você consiga atingir os objetivos do algoritmo, apresenta um grande inconveniente: sempre serão executados três testes, mesmo quando já tivermos classificado o número. Entretanto, na 1^a solução, outro teste só será executado se ainda não tivermos chegado a uma conclusão sobre o número.

Esta estrutura precisa de vários ciclos para ser executada; portanto, evite usá-la desnecessariamente.

algoritmo 87

Criar um algoritmo que permita ao aluno responder qual a capital do Brasil. Todas as possibilidades deverão ser pensadas.

```
prog geo
    string resp;
    imprima "\nQual a capital do Brasil? ";
    leia resp;
    se(resp == "BRASÍLIA" || resp == "Brasília")
    { imprima "\nPARABÉNS!"}
    senao
    { se(resp=="brasília" || resp=="BRASÍLIA"|| resp=="Brazília" || resp=="brazília")
    { imprima "\nCERTO! Mas atenção para grafia: Brasília ou BRASÍLIA "}
    senao
    { imprima "\nERRADO! ESTUDE MAIS!"}
```

```
}
```

```
imprima "\n";
```

```
fimprog
```

↳ *Digite no ambiente Linux por causa da acentuação.*

VÍDEO

Qual a capital do Brasil? brasília
CERTO! Mas atenção para grafia: Brasília ou BRASÍLIA

Qual a capital do Brasil? BRASÍLIA
PARABÉNS!

Qual a capital do Brasil? Brasília
PARABÉNS!

Qual a capital do Brasil? Brazília
CERTO! Mas atenção para grafia: Brasília ou BRASÍLIA

Qual a capital do Brasil? BRAZÍLIA
CERTO! Mas atenção para grafia: Brasília ou BRASÍLIA

Qual a capital do Brasil? brazília
CERTO! Mas atenção para grafia: Brasília ou BRASÍLIA

Qual a capital do Brasil? Vitória
ERRADO! ESTUDE MAIS!

algoritmo 88

Algoritmo Calculadora

```
prog calculadora
  string resp;
  real a,b;
  imprima "\n\n\t\t\t*****\n";
  imprima "\t\t\tCALCULADORA*\n";
  imprima "\t\t\t*****\n";
  imprima "\n\t\t\t para somar\n";
  imprima "\n\t\t\t- para subtrair\n";
  imprima "\n\t\t\t* para multiplicar\n";
  imprima "\n\t\t\t/ para dividir\n";
  imprima "\n\n\t\t\tDigite opcao: ";
  leia resp;
```

```

se(resp=="+")
{
    imprima "\nDigite 1 numero com ponto: ";
    leia a;
    imprima "\nDigite 2 numero com ponto: ";
    leia b;
    imprima "\nSOMA: ",a + b;
}
senao
{
    se(resp=="-")
    {
        imprima "\nDigite 1 numero com ponto: ";
        leia a;
        imprima "\nDigite 2 numero com ponto: ";
        leia b;
        imprima "\nSUBTRACAO: ",a - b;
    }
    senao
    {
        se(resp=="*")
        {
            imprima "\nDigite 1 numero com ponto: ";
            leia a;
            imprima "\nDigite 2 numero com ponto: ";
            leia b;
            imprima "\nMULTIPLICACAO: ",a * b;
        }
        senao
        {
            se(resp=="/")
            {
                imprima "\nDigite 1 numero com ponto: ";
                leia a;
                imprima "\nDigite 2 numero com ponto: ";
                leia b;
                imprima "\nDIVISAO: ",a / b;
            }
            senao
            {
                imprima "\nOPÇÃO NAO DISPONIVEL!";
            }
        }
    }
}
imprima "\n";
fimprog

```

VÍDEO

```
*****
*CALCULADORA*
*****
+ para somar
- para subtrair
* para multiplicar
/ para dividir
```

Digite opção:*

Digite 1 numero com ponto:23.

Digite 2 numero com ponto:12.

```
*****
*CALCULADORA*
*****
+ para somar
- para subtrair
* para multiplicar
/ para dividir
```

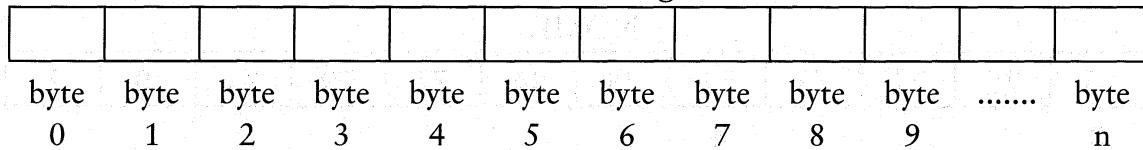
Digite opção:&

OPCAO NAO DISPONIVEL!

UM POUCO MAIS SOBRE VARIÁVEL STRING

Já vimos que a variável simples string é armazenada na memória como se fosse um vetor onde cada caractere é armazenado em uma posição conforme o esquema a seguir:

variável string



Analisando sob esse prisma, podemos dizer que a variável simples string é um conjunto de n variáveis de um único *caractere* e todas com o mesmo nome.

Quando se compararam duas variáveis strings, na verdade, estamos comparando os códigos dos caracteres de cada string, baseado em seus va-

lores no código ASCII (Apêndice II). Dessa forma, não é difícil entender o uso dos sinais de **>**, **<** ou de **=**.

Usar o sinal de **>** significa perguntar se *vem depois na ordem alfabética* (ou **<** vem antes), uma vez que o código ASCII está em ordem alfabética, isto é, a letra A tem um código menor do que a letra B e assim sucessivamente. Veja o trecho da tabela a seguir:

CARACTERE	Código ASCII em decimal
A	65
B	66
C	67
...	...
L	76
...	...
Z	90
...	...
a	97
b	98
c	99
...	...
l	208
...	...
z	122

Observando o trecho da tabela acima, você poderá compreender por que uma palavra escrita em letras maiúsculas (ALGORITMOS) não é considerada igual a uma palavra escrita em letras minúsculas (algoritmos):

NOME1												
65	76	71	79	82	73	84	77	79	83	\0	?	
0	1	2	3	4	5	6	7	8	9			

NOME2												
97	108	103	111	114	105	116	109	111	115	\0	?	
0	1	2	3	4	5	6	7	8	9			

ALGUMAS QUESTÕES

1. Algumas linguagens permitem que se monte uma expressão relacional com os nomes das variáveis:

COMPARAÇÃO
se (NOME1 > NOME2)

Outras linguagens não permitem o uso dos operadores $>$, $<$ ou $=$ e oferecem funções que viabilizam esta operação:

se (strcmp(NOME1, NOME2) == "igual")	Essa expressão verifica se os conteúdos das variáveis são iguais.
se (strcmp (NOME1, NOME2) == "maior")	Essa expressão verifica se o conteúdo da 1 ^a variável, na ordem alfabética, vem depois do conteúdo da 2 ^a variável.
se (strcmp (NOME1, NOME2) == "menor")	Essa expressão verifica se o conteúdo da 1 ^a variável, na ordem alfabética, vem antes do conteúdo da 2 ^a variável.

↳ No interpretador que sugerimos, você poderá fazer das duas maneiras.

2. Mas, independente do exposto acima, como a Unidade Aritmética e Lógica compara esses dados? Vejamos o exemplo a seguir, sabendo-se que a variável NOME1 armazena a palavra *caso* e a variável NOME2 armazena a palavra *casa*:

NOME1

99	97	115	111	\0	?
0	1	2	3	4	5

NOME2

99	97	115	97	\0	?
0	1	2	3	4	5

Se usássemos a expressão Se (NOME1 < NOME2), a Unidade Aritmética e Lógica faria subtrações, considerando como minuendo o 1º código de NOME1 e, como subtraendo, o 1º código de NOME2 e, assim sucessivamente, até que o resultado fosse diferente de 0.

1º passo:

$$99 - 99 = 0$$

continuo porque o resultado foi 0 (“igual”).

2º passo:

$$97 - 97 = 0$$

continuo porque o resultado foi 0 (“igual”).

3º passo:

$$115 - 115 = 0$$

continuo porque o resultado foi 0 (“igual”).

4º passo:

$$111 - 97 = 14$$

paro e vou verificar o que foi pedido

Como foi perguntado se NOME1 > NOME2 e tenho 14 (“maior”) como resposta, vou dizer que é VERDADE. (*Esse seria o “raciocínio” da UAL.*)

3. Algumas linguagens permitem que se atribua uma variável caracter a outra variável caracter enquanto outras oferecem uma função para produzir o mesmo efeito:

ATRIBUIÇÃO	FUNÇÃO
AUX <- NOME1;	copia(AUX, NOME1);

4. Algumas linguagens têm o operador // ou + (coloca o conteúdo de uma após o da outra):

CONCATENAÇÃO e ATRIBUIÇÃO	
Em algumas linguagens	Em nosso estudo
NOME <- NOME1 // NOME2;	NOME <- strconcat(NOME1,NOME2);

Outras linguagens não permitem nenhuma das formas acima e oferecem funções que viabilizam estas operações:

CONCATENAÇÃO e ATRIBUIÇÃO
copia(NOME, NOME1);
concatena(NOME, NOME2);

ALTERNATIVA DE MÚLTIPLAS ESCOLHAS

É uma alternativa para os ses aninhados, deixando o algoritmo com uma estrutura melhor.

Sintaxe:

```
escolha (expressão)
{
    caso <rótulo 1> : comando1;
                           comando2;
                           pare;
    caso <rótulo 2> : comando1;
                           comando2;
                           pare;
    caso <rótulo n> : comando1;
                           comando2;
                           pare;
    senao comando;
}
```

Considerações:

1. A expressão é avaliada e o valor será comparado com um dos rótulos.
2. A opção *senao* é opcional.
3. O rótulo será aqui definido como uma *constante caracter* (*de um caracter*) ou uma *constante numérica inteira*, embora em algumas linguagens possam ser usadas constantes caracter com mais de um caracter.
4. A estrutura é muito usada em algoritmos com menus, tornando-os mais claros do que quando usamos ses aninhados.
5. O interpretador que sugerimos nessa versão não apresenta essa estrutura.

Exemplo:

Escrever um algoritmo que leia um peso na Terra e o número de um planeta e imprima o valor do seu peso neste planeta. A relação de planetas é dada a seguir juntamente com o valor das gravidades relativas à Terra:

#	gravidade relativa	Planeta
1	0,37	Mercúrio
2	0,88	Vênus
3	0,38	Marte
4	2,64	Júpiter
5	1,15	Saturno
6	1,17	Urano

Para calcular o peso no planeta use a fórmula:

$$P_{\text{planeta}} \cdot \frac{P_{\text{terra}}}{10} \cdot \text{gravidade}$$

algoritmo 89

```
prog pesodoplaneta
    int op;
    real pterra;
    imprima " planetas que podem ser analisados: ";
    imprima " 1 mercurio ";
    imprima " 2 venus ";
    imprima " 3 marte ";
    imprima " 4 jupiter ";
    imprima " 5 saturno ";
    imprima " 6 urano ";
    imprima " escolha o planeta a ser analisado ";
    leia op ;
    imprima " entre com um peso na terra ";
    leia pterra;
    escolha ( op)
{
    caso 1: imprima "seu peso no planeta terra é: ", (pterra/10)*0.37 ;pare;
    caso 2: imprima "seu peso no planeta terra é: ", (pterra/10)*0.88;pare;
    caso 3: imprima "seu peso no planeta terra é: ", (pterra/10)*0.38;pare;
    caso 4: imprima "seu peso no planeta terra é: ", (pterra/10)*2.64;pare;
    caso 5: imprima "seu peso no planeta terra é: ", (pterra/10)*1.15;pare;
    caso 6: imprima "seu peso no planeta terra é: ", (pterra/10)*1.17;pare;
    senao: imprima "este planeta não pode ser analisado";
}
imprima "\n";
fimprog
```

EXERCÍCIOS – LISTA 2

ESTRUTURA DO SE

algoritmo 90

Entrar com um número e imprimi-lo caso seja maior que 20.

```
prog sel
    real numero;
    imprima "\n\ndigite numero: ";
    leia numero;
    se ( numero > 20. )
    {
        imprima numero;
    }
    imprima "\n";
fimprog
```

algoritmo 91

Construir um algoritmo que leia dois valores numéricos inteiros e efetue a adição; caso o resultado seja maior que 10, apresentá-lo.

```
prog se2
    int num1, num2, soma;
    imprima "\n\nDigite 1 numero: ";
    leia num1;
    imprima "\n\nDigite 2 numero: ";
    leia num2;
    soma <- num1 + num2;
    se ( soma > 10 )
    {   imprima "\nsoma: ", soma; }
    imprima "\n";
fimprog
```

algoritmo 92

Construir um algoritmo que leia dois números e efetue a adição. Caso o valor somado 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.

```
prog se3
    real num1, num2, soma;
    imprima "\nDigite 1 numero: ";
    leia num1;
    imprima "\nDigite 2 numero: ";
    leia num2;
    soma <- num1 + num2;
    se ( soma > 20. )
    {   imprima "\nsoma: ", soma + 8; }
    senao
    {   imprima "\nsoma: ", soma - 5; }
    imprima "\n";
fimprog
```

algoritmo 93

Entrar com um número e imprimir a raiz quadrada do número caso ele seja positivo e o quadrado do número caso ele seja negativo.

```
prog se4
    real numero;
    imprima "\nDigite numero: ";
    leia numero;
    se ( numero > 0. )
    {   imprima "\nraiz: ", raiz(numero); }
```

```
senao
{
    se ( numero < 0. )
        { imprima "\nquadrado: ", numero ** 2; }
    }
    imprima "\n";
fimprog
```

algoritmo 94

Entrar com um número e imprimir uma das mensagens: é multiplo de 3 ou não é multiplo de 3.

```
prog se5
int numero;
imprima "\ndigite numero: ";
leia numero;
se ( numero % 3 == 0 )
{   imprima "\nmultiplo de 3 ";}
senao
{   imprima "\nnao e multiplo de 3"; }
imprima "\n";
fimprog
```

algoritmo 95

Entrar com um número e informar se ele é ou não divisível por 5.

```
prog se6
int numero;
imprima "\ndigite numero: ";
leia numero;
se ( numero % 5 == 0 )
{   imprima "\ne divisivel por 5 ";}
senao
{   imprima "\nnao e divisivel por 5"; }
imprima "\n";
fimprog
```

algoritmo 96

Entrar com um número e informar se ele é divisível por 3 e por 7.

```
prog se7
int numero;
imprima "\ndigite numero: ";
leia numero;
se (numero %3==0 && numero %7==0 ) /*poderia testar (numero %21 == 0) */
{   imprima "\ndivisivel por 3 e por 7"; }
senao
```

```
{ imprima "\nnão é divisível por 3 e por 7"; }
imprima "\n";
fimprog
```

algoritmo 97

Entrar com um número e informar se ele é divisível por 10, por 5, por 2 ou se não é divisível por nenhum destes.

```
prog se8
int numero;
imprima "\ndigite numero: ";
leia numero;
se ( numero % 10 == 0 )
{ imprima "\nmúltiplo de 10"; }
senao
{
  se ( numero % 2 == 0 )
  { imprima "\nmúltiplo de 2"; }
  senao
  {
    se ( numero % 5 == 0 )
    { imprima "\nmúltiplo de 5"; }
    senao
    { imprima "\nnão é múltiplo de 2 nem de 5"; }
  }
}
imprima "\n";
fimprog
```

algoritmo 98

A prefeitura do Rio de Janeiro abriu uma linha de crédito para os funcionários estatutários. O valor máximo da prestação não poderá ultrapassar 30% do salário bruto. Fazer um algoritmo que permita entrar com o salário bruto e o valor da prestação e informar se o empréstimo pode ou não ser concedido.

```
prog se9
real sb, vp;
imprima "\ndigite o salario: ";
leia sb;
imprima "\ndigite o valor da prestacao: ";
leia vp;
se( vp <= 0.3 * sb )
{ imprima "\nemprestimo concedido ";}
senao
{ imprima "\nemprestimo negado ";}
imprima "\n";
fimprog
```

algoritmo 99

Ler um número inteiro de 3 casas decimais e imprimir se o algarismo da casa das centenas é par ou ímpar.

```
prog se10
    int num, c;
    imprima "\nnumero de 3 algarismos: ";
    leia num;
    c <- num div 100;
    se( c % 2 == 0 )
    {imprima "\no algarismo das centenas e par: ",c;}
    senao
    {imprima "\no algarismo das centenas e impar: ",c;}
    imprima "\n";
fimprog
```

algoritmo 100

Ler um número inteiro de 4 casas e imprimir se é ou não múltiplo de quatro o número formado pelos algarismos que estão nas casas das unidades de milhar e das centenas.

```
prog se11
    int num, c;
    imprima "\nnumero de 4 algarismos: ";
    leia num;
    c <- num div 100;
    se( c % 4 == 0 )
    {imprima "\no numero e multiplo de 4: ",c;}
    senao
    {imprima "\no numero nao e multiplo de 4: ",c;}
    imprima "\n";
fimprog
```

algoritmo 101

Construir um algoritmo que indique se o número digitado está compreendido entre 20 e 90 ou não.

```
prog se12
    real num;
    imprima "\ndigite numero: ";
    leia num;
    se ( num > 20. && num < 90. )
    { imprima "\no numero esta na faixa de 20 a 90, exclusive"; }
    senao
    { imprima "\no numero esta fora da faixa de 20 a 90"; }
    imprima "\n";
fimprog
```

algoritmo 102

Entrar com um número e imprimir uma das mensagens: maior do que 20, igual a 20 ou menor do que 20.

```
prog sel3
    real numero;
    imprima "\ndigite numero: ";
    leia numero;
    se ( numero > 20. )
    { imprima "\nmaior que 20"; }
    senao
    {
        se ( numero < 20. )
        { imprima "\nmenor que 20"; }
        senao
        { imprima "\nigual a 20"; }
    }
    imprima "\n";
fimprog
```

algoritmo 103

Entrar com o ano de nascimento de uma pessoa e o ano atual. Imprimir a idade da pessoa. Não se esqueça de verificar se o ano de nascimento é um ano válido.

```
prog sel4
    int anon, anoa;
    imprima "\nEnter com ano atual: ";
    leia anoa;
    imprima "\nEnter com ano de nascimento: ";
    leia anon;
    se ( anon > anoa )
    { imprima "\nAno de Nascimento Invalido"; }
    senao
    { imprima "\nIdade: " , anoa - anon; }
    imprima "\n";
fimprog
```

algoritmo 104

Entrar com nome, sexo e idade de uma pessoa. Se a pessoa for do sexo feminino e tiver menos que 25 anos, imprimir nome e a mensagem: ACEITA. Caso contrário, imprimir nome e a mensagem: NÃO ACEITA. (Considerar f ou F.)

```
prog sel5
    int idade;
    string nome, sexo;
    imprima "\ndigite nome: ";
```

```

leia nome;
imprima "\ndigite sexo: ";
leia sexo;
imprima "\ndigite idade: ";
leia idade;
se( (sexo == "F" || sexo=="f") && idade<25)
{ imprima "\n", nome, " ACEITA"; }
senao
{ imprima "\n", nome, " NAO ACEITA"; }
imprima "\n";
fimprog

```

algoritmo 105

Entrar com a sigla do estado de uma pessoa e imprimir uma das mensagens:

- *carioca*
- *paulista*
- *mineiro*
- *outros estados*

```

prog se16
string sigla;
imprima "\ndigite sigla: ";
leia sigla;
se ( sigla == "RJ" || sigla == "rj")
{ imprima "\ncarioca"; }
senao
{ se (sigla == "SP" || sigla == "sp")
{ imprima "\npaulista"; }
senao
{ se ( sigla == "MG" || sigla == "mg")
{ imprima "\nmineiro"; }
senao
{ imprima "\noutros estados"; }
}
}
imprima "\n";
fimprog

```

algoritmo 106

Entrar com um nome e imprimi-lo se o primeiro caractere for a letra A (considerar letra minúscula ou maiúscula).

```

prog se17
string nome, letra;
imprima "\ndigite nome: ";

```

```
leia nome;
letra <- strprim(nome);
se ( letra == "A" || letra == "a" )
{ imprima "\n", nome; }
imprima "\n";
fimprog
```

algoritmo 107

Entrar com o nome de uma pessoa e só imprimi-lo se o prenome for JOSÉ.

```
prog se18
string nome;
imprima "\ndigite nome: ";
leia nome;
se (strnprim(nome, 5) == "JOSÉb" || nome=="JOSÉ" )
{ imprima "\n", nome; }
imprima "\n";
fimprog
```

⇨ b significa pressionar a barra de espaço

⇨ Há duas possibilidades, pois pode ser que a pessoa só digite JOSÉ e não o nome completo.

algoritmo 108

Idem ao anterior, porém considerar: JOSÉ, José ou josé.

```
prog se19
string nome, prenome;
imprima "\ndigite nome: ";
leia nome;
se( prenome == "JOSÉb" || prenome == "Joséb" || prenome == "joséb"
|| nome=="josé" || nome == "José" || nome=="JOSÉ" ) #mesma linha
{ imprima "\n", nome; }
imprima "\n";
fimprog
```

algoritmo 109

Criar um algoritmo que entre com dois nomes e imprimi-los em ordem alfabética.

```
prog se20a
    string nome1,nome2;
    imprima "\ndigite 1º nome: ";
    leia nome1;
    imprima "\ndigite 2º nome: ";
    leia nome2;
    se (nome1< nome2)
    { imprima "\n", nome1," ", nome2 ; }
    senao
    { imprima "\n", nome2," ", nome1; }
    imprima "\n";
fimprog
```

```
prog se20b
    string nome1,nome2;
    imprima "\ndigite 1º nome: ";
    leia nome1;
    imprima "\ndigite 2º nome: ";
    leia nome2;
    se (strcmp(nome1, nome2) ==
"menor")
    { imprima "\n", nome1," ", nome2; }
    senao
    { imprima "\n", nome2," ", nome1; }
    imprima "\n";
fimprog
```

algoritmo 110

Criar um algoritmo que leia dois números e imprimir uma mensagem dizendo se são iguais ou diferentes.

```
prog se21
    real a, b;
    imprima "\ndigite 1º numero: ";
    leia a;
    imprima "\ndigite 2º numero: ";
    leia b;
    se (a == b)
    { imprima "\niguais"; }
    senao
    { imprima "\ndiferentes"; }
    imprima "\n";
fimprog
```

algoritmo 111

Entrar com dois números e imprimir o maior número (suponha números diferentes).

```
prog se22
    real a, b;
    imprima "\ndigite 1º numero: ";
    leia a;
    imprima "\ndigite 2º numero: ";
```

```
leia b;  
se (a > b)  
{ imprima "\nmaior: ", a; }  
senao  
{ imprima "\nmaior: ", b; }  
imprima "\n";  
fimprog
```

algoritmo 112

Entrar com dois números e imprimir o menor número (suponha números diferentes).

```
prog se23  
    real a, b;  
    imprima "\ndigite 1º numero: ";  
    leia a;  
    imprima "\ndigite 2º numero: ";  
    leia b;  
    se (a < b)  
    { imprima "\nmenor: ", a; }  
    senao  
    { imprima "\nmenor: ", b; }  
    imprima "\n";  
fimprog
```

algoritmo 113

Entrar com dois números e imprimi-los em ordem crescente (suponha números diferentes).

```
prog se24  
    real a, b;  
    imprima "\ndigite 1º numero: ";  
    leia a;  
    imprima "\ndigite 2º numero: ";  
    leia b;  
    se (a < b)  
    { imprima "\n",a , " ",b; }  
    senao  
    { imprima "\n",b , " ",a; }  
    imprima "\n";  
fimprog
```

algoritmo 114

Entrar com dois números e imprimi-los em ordem decrescente (suponha números diferentes).

```
prog se25  
    real a, b;
```

```

imprima "\ndigite 1º numero: ";
leia a;
imprima "\ndigite 2º numero: ";
leia b;
se (a > b)
{ imprima "\n",a , " ",b; }
senao
{ imprima "\n",b , " ",a; }
imprima "\n";
fimprog

```

algoritmo 115

Criar o algoritmo que deixe entrar com dois números e imprimir o quadrado do menor número e a raiz quadrada do maior número, se for possível.

```

prog se26
real x, y;
imprima "\ndigite valor de x: ";
leia x;
imprima "\ndigite valor de y: ";
leia y;
se(x > y)
{imprima "\n", raiz(x), "\n", y **2;}
senao
{ se(y > x)
{imprima "\n", raiz(y), "\n", x **2;}
senao
{ imprima "\nnumeros iguais";}
}
imprima "\n";
fimprog

```

algoritmo 116

Entrar com três números e imprimir o maior número (suponha números diferentes).

```

prog se27solucao1
real a, b, c;
imprima "\ndigite 1 numero: ";
leia a;
imprima "\ndigite 2 numero: ";
leia b;
imprima "\ndigite 3 numero: ";
leia c;
se (a > b)
{
  se (a > c)
  { imprima "\nmaior: ", a ; }
}

```

```

senao
{ imprima "\nmaior: ", c; }
}
senao
{
  se ( b > c )
  { imprima "\nmaior: ", b; }
  senao
  { imprima "\nmaior: ", c ; }
}
imprima "\n";
fimprog

prog se27solucao2
real a, b, c;
imprima "\ndigite 1 numero: ";
leia a;
imprima "\ndigite 2 numero: ";
leia b;
imprima "\ndigite 3 numero: ";
leia c;
se ( a > b && a > c )
{ imprima "\nmaior: ", a ; }
senao
{
  se ( b > c )
  { imprima "\nmaior: ", b ; }
  senao
  { imprima "\nmaior: ", c ; }
}
imprima "\n";
fimprog

```

algoritmo 117

Entrar com três números e armazenar o maior número na variável de nome maior (suponha números diferentes).

⇨ Esta é uma solução alternativa e melhor para o algoritmo anterior.

```

prog se28solucao1
real a, b, c, maior;
imprima "\ndigite 1 numero: ";
leia a;
imprima "\ndigite 2 numero: ";
leia b;
imprima "\ndigite 3 numero: ";
leia c;

```

```

se (a > b)
{ maior <- a; }
senao
{ maior <- b ; }
se (c > maior)
{ maior <- c; }
imprima "\nmaior: ", maior;
imprima "\n";
fimprog

prog se28solucao2
real a, b, c, maior;
imprima "\ndigite 1 numero: ";
leia a;
imprima "\ndigite 2 numero: ";
leia b;
imprima "\ndigite 3 numero: ";
leia c;
maior <- a;
se (b > maior)
{ maior <- b ; }
se (c > maior)
{ maior <- c; }
imprima "\nmaior: ", maior;
imprima "\n";
fimprog

```

algoritmo 118

Entrar com três números e imprimi-los em ordem crescente (suponha números diferentes).

↳ Várias são as soluções para colocar três números em ordem crescente, decrescente, maior e menor ou intermediário. Acreditamos que esta solução seja bem simples e tem como objetivo armazenar na variável *a* o menor valor; na variável *b*, o segundo valor; e na variável *c*, o maior valor.

```

prog se29
real a, b, c, aux;
imprima "\ndigite 1 numero: ";
leia a;
imprima "\ndigite 2 numero: ";
leia b;
imprima "\ndigite 3 numero: ";
leia c;
se ( a > b )
{ aux <- a; a <- b; b <- aux; }

```

```

se ( a > c )
{ aux <- a; a <- c; c <- aux; }
se ( b > c )
{ aux <- b; b <- c; c <- aux; }
imprima "\nOrdem Cresecnte: ", a, " ", b, " ", c ;
imprima "\n";
fimprog

```

algoritmo 119

Entrar com três números e imprimi-los em ordem decrescente (suponha números diferentes).

```

prog se30
real a, b, c, aux;
imprima "\ndigite 1 numero: ";
leia a;
imprima "\ndigite 2 numero: ";
leia b;
imprima "\ndigite 3 numero: ";
leia c;
se ( a > b )
{ aux <- a; a <- b; b <- aux; }
se ( a > c )
{ aux <- a; a <- c; c <- aux; }
se ( b > c )
{ aux <- b; b <- c; c <- aux; }
imprima "\nOrdem Decresecente: ", c, " ", b, " ", a ;
imprima "\n";
fimprog

```

algoritmo 120

Entrar com três números e armazená-los em três variáveis com os seguintes nomes: maior, intermediário e menor (suponha números diferentes).

```

prog se31a
real a, b, c, maior, intermediario, menor;
imprima "\ndigite 1 numero: ";
leia a;
imprima "\ndigite 2 numero: ";
leia b;
imprima "\ndigite 3 numero: ";
leia c;
se ( a > b)
{ se ( c > a)
{ maior <- c;
intermediario <- a ;
menor <- b;
}
}

```

```

    }
senao
{ se ( c > b)
{ maior <- a;
intermediario <- c;
menor <- b;
}
senao
{ maior <- a;
intermediario <- b;
menor <- c ;
}
}
}
senao
{ se ( c >b)
{ maior <- c;
intermediario <- b;
menor <- a;
}
senao
{ se ( c > a)
{ maior <- b;
intermediario <- c;
menor <- a;
}
senao
{ maior <- b;
intermediario <- a;
menor <-c;
}
}
}
imprima "\nmaior: ", maior;
imprima "\nintermediario: ", intermediario;
imprima "\nmenor: ", menor;
imprima "\n";
fimprog

prog se31b
real a, b, c, aux, maior, intermediario,menor;
imprima "\ndigite 1 numero: ";
leia a;
imprima "\ndigite 2 numero: ";
leia b;
imprima "\ndigite 3 numero: ";
leia c;
se ( a > b )
{ aux <- a;      a <- b;      b <- aux; }

```

```

se ( a > c )
{   aux <- a;      a <- c;      c <- aux; }
se ( b > c )
{   aux <- b;      b <- c;      c <- aux; }
maior <- c;
intermediario <- b;
menor <- a;
imprima "\nmaior: ", maior;
imprima "\nintermediario: ", intermediario;
imprima "\nmenor: ", menor;
imprima "\n";
fimprog

```

algoritmo 121

Efetuar a leitura de cinco números inteiros diferentes e identificar o maior e o menor valor.

```

prog se32
int n1, n2, n3, n4, n5, maior, menor;
imprima "\ndigite 1 numero: ";
leia n1;
imprima "\ndigite 2 numero: ";
leia n2;
imprima "\ndigite 3 numero: ";
leia n3;
imprima "\ndigite 4 numero: ";
leia n4;
imprima "\ndigite 5 numero: ";
leia n5;
se (n1< >n2 && n1< >n3 && n1< >n4 && n1< >n5 && n2< >n3 && n2< >n4
&& n2< >n5 && n3< >n4 && n3< >n5 && n4< >n5)
{
/* Teste necessário porque o exercício fala sobre os números serem
diferentes. Veja como é importante saber COMBINAÇÃO */
se (n1 > n2)
{   maior <- n1;   menor <- n2; }
senao
{   maior <- n2;   menor <- n1; }
se (n3 > maior)
{   maior <- n3; }
senao
{
    se (n3 < menor)
    {   menor <- n3; }
}
se (n4 > maior)
{   maior <- n4; }

```

```

senao
{
    se (n4 < menor)
    { menor <- n4; }
}
se (n5 > maior)
{ maior <- n5; }
senao
{
    se (n5 < menor)
    { menor <- n5; }
}
imprima "\nMaior= ", maior;
imprima "\nMenor= ", menor;
}
senao
{ imprima "\nOs valores devem ser diferentes ";}
imprima "\n";
fimprog

```

↳ Um algoritmo para achar o menor ou o maior com vários números fica extremamente longo. Quando você aprender as estruturas de repetição, verá que esse algoritmo poderá ser melhorado.

algoritmo 122

Ler três números e imprimir se eles podem ou não ser lados de um triângulo.

```

prog se33
    real a, b, c;
    imprima "\ndigite 1 lado: ";
    leia a;
    imprima "\ndigite 2 lado: ";
    leia b;
    imprima "\ndigite 3 lado: ";
    leia c;
    se ( a < b + c && b < a + c && c < a + b )
    { imprima "\nPodem ser lados de um triangulo"; }
    senao
    { imprima "\nNão podem ser lados de um triangulo"; }
    imprima "\n";
fimprog

```

algoritmo 123

Ler três números, os possíveis lados de um triângulo, e imprimir a classificação segundo os lados.

```

prog se34
real a, b, c;
imprima "\ndigite 1 lado: ";
leia a;
imprima "\ndigite 2 lado: ";
leia b;
imprima "\ndigite 3 lado: ";
leia c;
se ( a < b + c && b < a + c && c < a + b )
{
    se ( a == b && a == c )
    { imprima "\nTriangulo equilatero"; }
    senao
    {
        se ( a == b || a == c || b == c )
        { imprima "\nTriangulo isosceles"; }
        senao
        { imprima "\nTriangulo escaleno"; }
    }
}
senao
{ imprima "\nAs medidas não formam um triangulo"; }
imprima "\n";
fimprog

```

algoritmo 124

Ler três números, os possíveis lados de um triângulo, e imprimir a classificação segundo os ângulos.

```

prog se35
real a, b, c, maior, lados;
imprima "\ndigite 1 lado: ";
leia a;
imprima "\ndigite 2 lado: ";
leia b;
imprima "\ndigite 3 lado: ";
leia c;
se ( a < b + c && b < a + c && c < a + b )
{
    se ( a > b && a > c )
    { maior <- a ** 2; lados <- b ** 2 + c ** 2; }
    senao
    { se ( b > c )
        { maior <- b ** 2; lados <- a ** 2 + c ** 2; }
        senao
        { maior <- c ** 2; lados <- a ** 2 + b ** 2; }
    }
}

```

```

    se ( maior == lados )
    { imprima "\nTriangulo Retangulo"; }
    senao
    { se ( maior > lados )
        { imprima "\nTriangulo Obtusangulo"; }
        senao
        { imprima "\nTriangulo Acutangulo"; }
    }
}
senao
{ imprima "\nas medidas não formam um triangulo"; }
imprima "\n";
fimprog

```

algoritmo 125

Entrar com a idade de uma pessoa e informar:

- *se é maior de idade*
- *se é menor de idade*
- *se é maior de 65 anos*

```

prog se36
int idade;
imprima "\ndigite sua idade: ";
leia idade;
se( idade >= 65 )
{ imprima "\nmaior de 65"; }
senao
{ se ( idade >= 18 )
    { imprima "\nmaior de idade"; }
    senao
    { imprima "\nmenor de idade"; }
}
imprima "\n";
fimprog

```

algoritmo 126

Ler um número e imprimir se ele é igual a 5, a 200, a 400, se está no intervalo entre 500 e 1000, inclusive, ou se ele está fora dos escopos anteriores.

```

prog se37
real x;
imprima "\ndigite valor de X: ";
leia x ;
se ( x == 5. )
{ imprima "\n0 numero é o 5"; }

```

```

senao
{ se ( x ==200. )
{ imprima "\n0 numero é o 200"; }
senao
{ se ( x == 400. )
{ imprima "0 numero é o 400"; }
senao
{ se ( x >= 500. && x <= 1000.)
{ imprima "\nIntervalo 500-1000"; }
senao
{ imprima "\nFora do escopo"; }
}
}
imprima "\n";
fimprog

```

algoritmo 127

Entrar com nome, nota da PR1 e nota da PR2 de um aluno. Imprimir nome, nota da PR1, nota da PR2, média e uma das mensagens: Aprovado, Reprovado ou em Prova Final (a média é 7 para aprovação, menor que 3 para reprovação e as demais em prova final).

```

prog se38
real nota1, nota2, media;
string nome;
imprima "\ndigite nome: ";
leia nome;
imprima "\ndigite 1ª nota: ";
leia nota1;
imprima "\ndigite 2ª nota: ";
leia nota2;
media <- (nota1 + nota2)/ 2;
se ( media >=7. )
{ imprima nome," ", media, " AP"; }
senao
{ se ( media < 3. )
{ imprima nome," ", media, " RP"; }
senao
{ imprima nome," ", media, " PF"; }
}
imprima "\n";
fimprog

```

algoritmo 128

Entrar com um verbo no infinitivo e imprimir uma das mensagens:

- verbo não está no infinitivo
- verbo da 1^a conjugação
- verbo da 2^a conjugação
- verbo da 3^a conjugação

```
prog se39
    string verbo, letra;
    int n;
    imprima "\ndigite verbo: ";
    leia verbo;
    letra <- strlft(verbo);
    se ( letra == "R" || letra == "r")
    {
        n <- strtam(verbo) - 2;
        letra <- strelem(verbo,n);
        se ( letra == "A" || letra == "a" )
        { imprima "\n1a conjugacao"; }
        senao
        {
            se ( letra == "E" || letra == "e" || letra == "o" || letra == "O" )
            { imprima "\n2a conjugacao"; }
            senao
            {
                se ( letra == "I" || letra == "i" )
                { imprima "\n3a conjugacao"; }
                senao
                { imprima "\nnao existe verbo com terminacao ur"; }
            }
        }
    }
    senao
    { imprima "\nnao e um verbo no infinitivo"; }
    imprima "\n";
fimprog
```

algoritmo 129

Entrar com o salário de uma pessoa e imprimir o desconto do INSS segundo a tabela a seguir:

menor ou igual a R\$ 600,00	isento
maior que R\$ 600,00 e menor ou igual a R\$ 1200,00	20%
maior que R\$ 1200,00 e menor ou igual a R\$ 2000,00	25%
maior que R\$ 2000,00	30%

```

prog se40
  real salario, desconto;
  imprima "\ndigite salario: ";
  leia salario;
  se ( salario <= 600. )
  { desconto <- 0.;}
  senao
  { se ( salario <= 1200. )
  { desconto <- salario * 0.2; }
  senao
  { se ( salario <= 2000. )
  { desconto <- salario * 0.25; }
  senao
  { desconto <- salario * 0.30; }
  }
  }
  imprima "\ndesconto: ",desconto;
  imprima "\n";
fimprog

```

algoritmo 130

Um comerciante comprou um produto e quer vendê-lo com um lucro de 45% se o valor da compra for menor que R\$ 20,00; caso contrário, o lucro será de 30%. Entrar com o valor do produto e imprimir o valor da venda.

```

prog se41
  real valor;
  imprima "\ndigite valor do produto: ";
  leia valor;
  se( valor < 20. )
  { imprima "\nValor de venda: ", valor * 1.45 ; }
  senao
  { imprima "\nValor de venda: " , valor *1.3; }
  imprima "\n";
fimprog

```

algoritmo 131

A turma de Programação I, por ter muitos alunos, será dividida em dias de provas. Após um estudo feito pelo coordenador, decidiu-se dividi-la em três grupos. Fazer um algoritmo que leia o nome do aluno e indicar a sala em que ele deverá fazer as provas, tendo em vista a tabela a seguir e sabendo-se que todas as salas se encontram no bloco F:

A – K : sala 101

L – N : sala 102

O – Z : sala 103

```

prog se42
  string nome,L;
  imprima "\nDigite nome: ";
  leia nome;
  L<-strprim(nome);
  se ((L>="A" && L<="K") || (L>="a" && L<="k"))
  { imprima "\n", nome, " sala: 101"; }
  senao
  {
    se ((L>="L" && L<="N") || (L>="l" && L<="n"))
    { imprima "\n", nome, " sala: 102"; }
    senao
    {
      se ((L>="O" && L<="Z") || (L>="o" && L<="z"))
      { imprima "\n", nome, " sala: 103"; }
      senao
      { imprima "\n NOME INVALIDO"; }
    }
  }
  imprima "\n";
fimprog

```

algoritmo 132

Fazer um algoritmo que possa converter uma determinada quantia dada em reais para uma das seguintes moedas:

- *f – franco suíço*
- *l – libra esterlina*
- *d – dólar*
- *m – marco alemão*

```

prog se43
  real r,l,d,f,m;
  string moeda;
  imprima "\ndigite valor em reais a ser convertido: ";
  leia r;
  imprima "\ndigite f - franco suico l - libra esterlina d - dolar m -
marco alemão: ";
  leia moeda;
  se( moeda == "f" || moeda == "F")
  { imprima "\ndigite valor de um franco suico em reais: ";
    leia f;
    imprima "\nTotal de francos suicos: ", r /f;}
  senao
  { se( moeda == "l" || moeda == "L")
    { imprima "\ndigite valor de uma libra esterlina em reais: ";
      leia l;
    }
  }

```

```

imprima "\nTotal de libras esterlinas: ", r /l;
senao
{ se( moeda == "d" || moeda == "D")
{ imprima "\ndigite valor de um dolar em reais: ";
leia d;
imprima "\nTotal de dolares: ", r /d;}
senao
{ se( moeda == "m" || moeda == "M")
{ imprima "\ndigite valor de um marco alemão em reais: ";
leia m;
imprima "\nTotal de marcos alemães: ", r /m;}
senao
{ imprima "\nmoeda desconhecida";}
}
}
imprima "\n";
fimprog

```

algoritmo 133

Segundo uma tabela médica, o peso ideal está relacionado com a altura e o sexo. Fazer um algoritmo que receba a altura e o sexo de uma pessoa, calcular e imprimir o seu peso ideal, utilizando as seguintes fórmulas:

- para homens: $(72.7 * H) - 58$
- para mulheres: $(62.1 * H) - 44.7$

```

prog se44
string sexo;
real h, peso;
imprima "\nentre com a altura: ";
leia h;
imprima "\nentre com o sexo M / F: ";
leia sexo;
se( sexo == "M" || sexo == "m")
{peso <- 72.7 * h - 58;}
senao
{peso <- 62.1 * h - 44.7;}
imprima "\nseu peso ideal : ", peso;
imprima "\n";
fimprog

```

algoritmo 134

A confederação brasileira de natação irá promover eliminatórias para o próximo mundial. Fazer um algoritmo que receba a idade de um nadador e imprimir a sua categoria segundo a tabela a seguir:

<i>Categoria</i>	<i>Idade</i>
<i>Infantil A</i>	<i>5 – 7 anos</i>
<i>Infantil B</i>	<i>8 – 10 anos</i>
<i>Juvenil A</i>	<i>11 – 13 anos</i>
<i>Juvenil B</i>	<i>14 – 17 anos</i>
<i>Sênior</i>	<i>maiores de 18 anos</i>

```

prog se45
    int idade;
    imprima "\nentre com a idade: ";
    leia idade;
    se( idade < 5 )
    { imprima "\nnao existe categoria para essa idade" ;}
    senao
    { se( idade <= 7 )
        { imprima "\ncategoria Infantil A" ;}
        senao
        { se( idade <= 10 )
            { imprima "\ncategoria Infantil B";}
            senao
            { se( idade <= 13 )
                { imprima "\ncategoria Juvenil A"; }
                senao
                { se( idade <= 17 )
                    { imprima "\ncategoria Juvenil B";}
                    senao
                    { imprima "\ncategoria Senior";}
                }
            }
        }
    }
    imprima "\n";
fimprog

```

algoritmo 135

Criar um algoritmo que leia a idade de uma pessoa e informar a sua classe eleitoral:

- *não-eleitor (abaixo de 16 anos)*
- *eleitor obrigatório (entre 18 e 65 anos)*
- *eleitor facultativo (entre 16 e 18 anos e maior de 65 anos)*

```

prog se46
    int idade;
    imprima "\ndigite idade: ";

```

```

leia idade;
se ( idade < 16 )
{ imprima "\nNao eleitor"; }
senao
{
    se ( idade > 65 )
    { imprima "\nEleitor facultativo"; }
    senao
    { imprima "\nEleitor obrigatorio"; }
}
imprima "\n";
fimprog

```

algoritmo 136

Depois da liberação do governo para as mensalidades dos planos de saúde, as pessoas começaram a fazer pesquisas para descobrir um bom plano, não muito caro. Um vendedor de um plano de saúde apresentou a tabela a seguir. Criar um algoritmo que entre com o nome e a idade de uma pessoa e imprimir o nome e o valor que ela deverá pagar.

- até 10 anos – R\$ 30,00
- acima de 10 até 29 anos - R\$ 60,00
- acima de 29 até 45 anos - R\$ 120,00
- acima de 45 até 59 anos - R\$ 150,00
- acima de 59 até 65 anos - R\$ 250,00
- maior que 65 anos – R\$ 400,00

```

prog se47
int idade;
string nome;
imprima "\nEnter com nome: ";
leia nome;
imprima "\nEnter com a idade: ";
leia idade;
se ( idade <= 10 )
{ imprima "\n", nome, " pagara R$ 30,00"; }
senao
{
    se ( idade <= 29 )
    { imprima "\n", nome, " pagara R$ 60,00"; }
    senao
    {
        se ( idade <= 45 )
        { imprima "\n", nome, " pagara R$ 120,00"; }
        senao
    }
}

```

```

{
    se ( idade <= 59 )
    { imprima "\n", nome, " pagara R$ 150,00"; }
    senao
    {
        se ( idade <= 65 )
        { imprima "\n", nome, " pagara R$ 250,00"; }
        senao
        { imprima "\n", nome, " pagara R$ 400,00"; }
    }
}
imprima "\n";
fimprog

```

algoritmo 137

Ler três valores inteiros (variáveis a, b e c) e efetuar o cálculo da equação de segundo grau, apresentando: as duas raízes, se para os valores informados for possível fazer o cálculo (delta positivo ou zero); a mensagem “Não há raízes reais”, se não for possível fazer o cálculo (delta negativo); e a mensagem “Não é equação do segundo grau”, se o valor de a for igual a zero.

```

prog se48
real a, b, c, d, x1, x2;
imprima "\nEntre com valor de a: ";
leia a;
imprima "\nEntre com valor de b: ";
leia b;
imprima "\nEntre com valor de c: ";
leia c;
se (a == 0.)
{ imprima "\n Nao e equacao do 2 grau";}
senao
{
d<- b**2 - 4 * a * c;
se (d >=0.)
{
    d<- raiz(d);
    x1<- (-b + d)/ (2 *a);
    x2 <-(-b - d)/ (2 *a);
    imprima "\n X1= ", x1, "\t\tX2= ", x2;
}
senao
{ imprima "\nNao ha raizes reais"; }
}
imprima "\n";
fimprog

```

algoritmo 138

Ler um número inteiro entre 1 e 12 e escrever o mês correspondente. Caso o usuário digite um número fora desse intervalo, deverá aparecer uma mensagem informando que não existe mês com este número.

```
prog se49
    int num;
    imprima "\ndigite vumnumero de 1 - 12: ";
    leia num ;
    se (num==1)
    { imprima "\njaneiro"; }
    senao
    { se (num==2)
        { imprima "\nfevereiro"; }
        senao
        { se (num==3)
            { imprima "\nmarco"; }
            senao
            { se (num==4)
                { imprima "\nabril"; }
                senao
                { se (num==5)
                    { imprima "\nmaio"; }
                    senao
                    { se (num==6)
                        { imprima "\njunho"; }
                        senao
                        { se (num==7)
                            { imprima "\n julho"; }
                            senao
                            { se (num==8)
                                { imprima "\nagosto"; }
                                senao
                                { se (num==9)
                                    { imprima "\nsetembro"; }
                                    senao
                                    { se (num==10)
                                        { imprima "\noutubro"; }
                                        senao
                                        { se (num==11)
                                            { imprima "\nnovembro"; }
                                            senao
                                            { se (num==12)
                                                { imprima "\ndezenbro"; }
                                                senao
                                                { imprima "\nnao existe mes correspondente"; }
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```

algoritmo 139

Sabendo que somente os municípios que possuem mais de 20.000 eleitores aptos têm segundo turno nas eleições para prefeito caso o primeiro colocado não tenha mais do que 50% dos votos, fazer um algoritmo que leia o nome do município, a quantidade de eleitores aptos, o número de votos do candidato mais votado e informar se ele terá ou não segundo turno em sua eleição municipal.

```
prog se50  
    int ne, votos;
```

```

string nome;
imprima "\nDigite nome do Municipio: ";
leia nome;
imprima "\nnumero de eleitores aptos: ";
leia ne;
imprima "\nnumero de votos do candidato mais votado: ";
leia votos;
se (ne >20000 && votos <= ne div 2)
{ imprima "\n", nome, " tera segundo turno"; }
senao
{ imprima "\n", nome, " nao terá segundo turno"; }
imprima "\n";
fimprog

```

algoritmo 140

Um restaurante faz uma promoção semanal de descontos para clientes de acordo com as iniciais do nome da pessoa. Criar um algoritmo que leia o primeiro nome do cliente, o valor de sua conta e se o nome iniciar com as letras A, D, M ou S, dar um desconto de 30%. Para o cliente cujo nome não se inicia por nenhuma dessas letras, exibir a mensagem “Que pena. Nesta semana o desconto não é para seu nome; mas continue nos prestigiando que sua vez chegará”.

```

prog se51
string nome,L;
real vc, vcd;
imprima "\nDigite nome: ";
leia nome;
imprima "\nDigite valor da conta: ";
leia vc;
L->strprim(nome);
se ( L=="A" || L=="a" || L=="D" || L=="d" || L=="M" || L=="m" || L=="S"
|| L=="s" )
{ imprima "\n", nome, " valor da conta com desconto: R$ ", formatar(vc *
0.7, 2); }
senao
{ imprima "\n Que pena. Nesta semana o desconto não é para seu nome; mas
continue nos prestigiando que sua vez chegará"; }
imprima "\n";
fimprog

```

algoritmo 141

Criar um algoritmo que leia o nome e o total de pontos de três finalistas de um campeonato de pingue-pongue e exibir a colocação da seguinte forma:

Vencedor: _____	XXXX ptos
Segundo colocado: _____	XXXX ptos
Terceiro colocado: _____	XXXX ptos

```

prog se52
    int p1, p2, p3, aux;
    string n1,n2,n3,auxn;
    imprima "\ndigite 1 nome: ";
    leia n1;
    imprima "\ndigite pontos: ";
    leia p1;
    imprima "\ndigite 2 nome: ";
    leia n2;
    imprima "\ndigite pontos: ";
    leia p2;
    imprima "\ndigite 3 nome: ";
    leia n3;
    imprima "\ndigite pontos: ";
    leia p3;
    se ( p1 < p2 )
    { aux <- p1; p1 <- p2; p2 <- aux; auxn <- n1; n1 <- n2; n2 <- auxn; }
    se ( p1 < p3 )
    { aux <- p1; p1 <- p3; p3 <- aux; auxn <- n1; n1 <- n3; n3 <- auxn; }
    se ( p2 < p3 )
    { aux <- p2; p2 <- p3; p3 <- aux; auxn <- n2; n2 <- n3; n3 <- auxn; }
    imprima "\nVENCEDOR : ", n1, " ",p1, " pontos";
    imprima "\nSEGUNDO COLOCADO : ", n2, " ",p2, " pontos";
    imprima "\nTERCEIRO COLOCADO: ", n3, " ",p3, " pontos";
    imprima "\n";
fimprog

```

algoritmo 142

Em um campeonato nacional de arco-e-flecha, tem-se equipes de três jogadores para cada estado. Sabendo-se que os arqueiros de uma equipe não obtiveram o mesmo número de pontos, criar um algoritmo que informe se uma equipe foi classificada, de acordo com a seguinte especificação:

- *ler os pontos obtidos por cada jogador da equipe;*
- *mostrar esses valores em ordem decrescente;*
- *se a soma dos pontos for maior do que 100, imprimir a média aritmética entre eles; senão, imprimir a mensagem "Equipe desclassificada".*

```

prog se53
    int p1, p2, p3, aux, soma;
    imprima "\n Digite os pontos do primeiro jogador: ";
    leia p1;
    imprima "\n Digite os pontos do segundo jogador: ";
    leia p2;

```

```

imprima "\n Digite os pontos do terceiro jogador: ";
leia p3;
se(p1 > p2)
{ aux<- p1 ; p1<- p2; p2<- aux; }
se(p1 > p3)
{ aux<- p1 ; p1<- p3; p3<- aux; }
se(p2 > p3)
{ aux<- p2 ; p2<- p3; p3<- aux; }
imprima "\n p1=", p1, "\n p2=", p2, "\n p3=", p3;
soma <- p1 + p2 + p3;
se(soma > 100)
{ imprima "\nMedia=", formatar(soma/3,2);}
senao
{ imprima "\nEquipe desclassificada!";}
imprima "\n";
fimprog

```

algoritmo 143

Criar um algoritmo que verifique a(s) letra(s) central(is) de uma palavra. Se o número de caracteres for ímpar, ele verifica se a letra central é uma vogal; caso contrário, verifica se é um dos dígrafos rr ou ss (só precisa testar letras minúsculas).

```

prog se54
string pal, p1, p2, p3;
int n1, n2;
imprima "\ndigite uma palavra: ";
leia pal;
se(strtam(pal) % 2 == 0)
{ n1<-strtam(pal) div 2;
  n2<- n1-1;
  p1<-strelem(pal,n1);
  p2<-strelem(pal,n2);
  p3<-strconcat(p2,p1);
  se( p3 == "rr" || p3 == "ss" )
  {imprima "\nE rr OU ss";}
  senao
  {imprima "\nNAO E rr OU ss";}
}
senao
{ n1<-strtam(pal) div 2;
  p1<-strelem(pal,n1);
  se( p1 == "a" || p1 == "e"|| p1 == "i" || p1 == "o" || p1 == "u")
  {imprima "\nE VOGAL";}
  senao
  {imprima "\nNAO E VOGAL";}
}
imprima "\n";
fimprog

```

algoritmo 144

O banco XXX concederá um crédito especial com juros de 2% aos seus clientes de acordo com o saldo médio no último ano. Fazer um algoritmo que leia o saldo médio de um cliente e calcule o valor do crédito de acordo com a tabela a seguir. Imprimir uma mensagem informando o saldo médio e o valor do crédito.

SALDO MÉDIO	PERCENTUAL
de 0 a 500	nenhum crédito
de 501 a 1000	30% do valor do saldo médio
de 1001 a 3000	40% do valor do saldo médio
acima de 3001	50% do valor do saldo médio

```
prog se55
real saldomedio, credito;
imprima "\ndigite saldo medio: ";
leia saldomedio;
se( saldomedio < 501. )
{ credito <-0.; }
senao
{ se(saldomedio < 1001.)
{ credito <- saldomedio * 0.3;}
senao
{ se(saldomedio < 3001.)
{ credito <- saldomedio * 0.4;}
senao
{ credito <- saldomedio * 0.5;}
}
}
se(credito < > 0.)
{ imprima "\nComo seu saldo era de:",saldomedio,"seu credito sera
de:",credito;}# continuacao da linha anterior
senao
{ imprima "\nComo seu saldo era de:", saldomedio,",voce nao tera
nenhum credito";}# continuacao da linha anterior
imprima "\n";
fimprog
```

algoritmo 145

A biblioteca de uma universidade deseja fazer um algoritmo que leia o nome do livro que será emprestado, o tipo de usuário (professor ou aluno) e possa imprimir um recibo conforme mostrado a seguir. Considerar que o professor tem dez dias para devolver o livro e o aluno só três dias.

Nome do livro

Tipo de usuário:

110 | Total de Dias:

```

prog se56
  string livro, tipo;
  int dia;
  imprima "\ndigite nome do livro: ";
  leia livro;
  imprima "\ndigite: professor / aluno: ";
  leia tipo;
  se( tipo == "professor" || tipo == "Professor" || tipo == "PROFESSOR")
  { dia <-10 ;}
  senao
  { se( tipo == "aluno" || tipo == "Aluno" || tipo == "ALUNO")
    { dia <-3 ;}
    senao
    { dia <- 0;}
  }
  se( dia == 0)
  { imprima "\n Tipo de usuario desconhecido";}
  senao
  { imprima "\nnome do livro: ", livro;
    imprima "\nnome tipo de usuário: ", tipo;
    imprima "\ntotal de dias: ", dia;
  }
  imprima "\n";
fimprog

```

algoritmo 146

Fazer um algoritmo que leia o percurso em quilômetros, o tipo do carro e informe o consumo estimado de combustível, sabendo-se que um carro tipo C faz 12 km com um litro de gasolina, um tipo B faz 9 km e o tipo C, 8 km por litro.

```

prog se57
  real percurso, consumo;
  string tipo;
  imprima "\ndigite tipo de carro( A / B / C): ";
  leia tipo;
  imprima "\ndigite percurso: ";
  leia percurso;
  se ( tipo == "C" || tipo == "c")
  { consumo <- percurso / 12;}
  senao
  {
    se ( tipo == "B" || tipo == "b")
    { consumo <- percurso / 10; }
    senao
    {
      se ( tipo == "A" || tipo == "a")

```

```

{ consumo <- percurso / 8; }
senao
{ consumo <- 0.; }
}
}
se ( consumo < > 0.)
{ imprima "\nConsumo estimado em litros: ", formatar(consumo,2);}
senao
{ imprima "\nModelo inexistente";}
imprima "\n";
fimprog

```

algoritmo 147

Criar um algoritmo que informe a quantidade total de calorias de uma refeição a partir da escolha do usuário que deverá informar o prato, a sobremesa e bebida (veja a tabela a seguir).

PRATO	SOBREMESA	BEBIDA
Vegetariano 180cal	Abacaxi 75cal	Chá 20cal
Peixe 230cal	Sorvete diet 110cal	Suco de laranja 70cal
Frango 250cal	Mousse diet 170cal	Suco de melão 100cal
Carne 350cal	Mousse chocolate 200cal	Refrigerante diet 65cal

```

prog se58
int op, os, ob, calorias;
calorias <- 0;
imprima "\nDigite 1- Vegetariano 2- Peixe 3- Frango 4- Carne: ";
leia op;
se(op==1)
{calorias <- calorias +180;}
senao
{ se(op==2)
{calorias <- calorias +230;}
senao
{se(op==3)
{calorias <- calorias +250;}
senao
{se(op==4)
{calorias <- calorias +350;}
}
}
}
imprima "\nDigite 1- Abacaxi 2- Sorvete diet 3- Suco melao 4-
refrigerante diet: ";

```

```

se(op==1)
{calorias <- calorias +75;}
senao
{ se(op==2)
{calorias <- calorias +110;}
senao
{se(op==3)
{calorias <- calorias +170;}
senao
{se(op==4)
{calorias <- calorias +200;}
}
}
}
imprima "\nDigite 1- Cha 2- Suco de laranja 3- Mousse diet 4- Mousse
de chocolate: ";
leia op;
se(op==1)
{calorias <- calorias +20;}
senao
{ se(op==2)
{calorias <- calorias +70;}
senao
{se(op==3)
{calorias <- calorias +100;}
senao
{se(op==4)
{calorias <- calorias +65;}
}
}
}
imprima "\n Total de calorias: ", calorias;
imprima "\n";
fimprog

```

algoritmo 148

Criar um algoritmo que leia o destino do passageiro, se a viagem inclui retorno (ida e volta) e informar o preço da passagem conforme a tabela a seguir.

DESTINO	IDA	IDA E VOLTA
Região Norte	R\$500,00	R\$900,00
Região Nordeste	R\$350,00	R\$650,00
Região Centro-Oeste	R\$350,00	R\$600,00
Região Sul	R\$300,00	R\$550,00

```

prog se59
    int op, iv;
    imprima "\nDigite 1- Regiao Norte 2- Regiao Nordeste 3- Regiao Centro-Oeste 4-
Regiao Sul: ";
    leia op;
    imprima "\nDigite 1- Ida 2- Ida e Volta: ";
    leia iv;
    se(op == 1)
    { se(iv == 1)
        { imprima "\nO valor da passagem de ida para R.Norte R$500.00";}
        senao
        { imprima "\nO valor da passagem de ida-volta para R.Norte: R$950.00";}
    }
    senao
    { se(op == 2)
        { se(iv == 1)
            { imprima "\nO valor da passagem de ida para R.Nordeste: R$350.00";}
            senao
            { imprima "\nO valor da passagem de ida-volta para R.Nordeste: R$650.00";}
        }
        senao
        { se(op == 3)
            { se(iv == 1)
                { imprima "\nO valor da passagem de ida para R.C.Oeste: R$350.00";}
                senao
                { imprima "\nO valor da passagem de ida-volta para R.C.Oeste: R$600.00";}
            }
            senao
            { se(op == 4)
                { se(iv == 1)
                    { imprima "\nO valor da passagem de ida para R.Sul: R$300.00";}
                    senao
                    { imprima "\nO valor da passagem de ida-volta para R.Sul: R$550.00";}
                }
                senao
                { imprima "\nOpcao Inexistente";}
            }
        }
    }
    imprima "\n";
fimprog

```

algoritmo 149

Um comerciante calcula o valor da venda, tendo em vista a tabela a seguir:

VALOR DA COMPRA	VALOR DA VENDA
<i>Valor < R\$ 10,00</i>	<i>lucro de 70%</i>
<i>R\$10,00 ≤ valor < R\$ 30,00</i>	<i>lucro de 50%</i>
<i>R\$30,00 ≤ valor < R\$ 50,00</i>	<i>lucro de 40%</i>
<i>Valor ≥ R\$50,00</i>	<i>lucro de 30%</i>

Criar o algoritmo que possa entrar com nome do produto e valor da compra e imprimir o nome do produto e o valor da venda.

```
prog se60
    real vcompra, vvenda;
    string nomep;
    imprima "\ndigite nome do produto: ";
    leia nomep;
    imprima "\ndigite valor da compra: ";
    leia vcompra;
    se ( vcompra < 10. )
    { vvenda <- vcompra * 1.7; }
    senao
    {
        se ( vcompra < 30. )
        { vvenda <- vcompra * 1.5; }
        senao
        {
            se ( vcompra < 50. )
            { vvenda <- vcompra * 1.4; }
            senao
            { vvenda <- vcompra * 1.3; }
        }
    }
    imprima "\n", nomep, "será vendido por R$\"", formatar(vvenda +
0.001,2);
    imprima "\n";
fimprog
```

algoritmo 150

Criar um algoritmo que leia um ângulo em graus e apresente:

- o seno do ângulo, se o ângulo pertencer a um quadrante par;
- o co-seno do ângulo, se o ângulo pertencer a um quadrante ímpar.

```
prog se61
    real ang, rang;
    imprima "\ndigite angulo em graus: ";
    leia ang;
    rang <- ang * pi / 180 ;
    se((rang > pi/2 && rang <= pi) || (rang > 3*pi/2 && rang <= 2*pi))
    { imprima "\nseno: ", sen(rang); }
    senao
    { imprima "\nco-seno: ", cos(rang); }
    imprima "\n";
fimprog
```

algoritmo 151

Um endocrinologista deseja controlar a saúde de seus pacientes e, para isso, se utiliza do Índice de Massa Corporal (IMC). Sabendo-se que o IMC é calculado através da seguinte fórmula:

$$IMC = \frac{\text{peso}}{\text{altura}^2}$$

Onde:

- peso é dado em Kg
- altura é dada em metros

Criar um algoritmo que apresente o nome do paciente e sua faixa de risco, baseando-se na seguinte tabela:

IMC	FAIXA DE RISCO
abaixo de 20	abaixo do peso
a partir de 20 até 25	normal
acima de 25 até 30	excesso de peso
acima de 30 até 35	obesidade
acima de 35	obesidade mórbida

```
prog se62
    real peso, altura, imc;
    imprima "\ndigite peso: ";
    leia peso;
    imprima "\ndigite altura: ";
    leia altura;
    imc <- peso / altura **2;
    se ( imc < 20.)
    {   imprima "\nabaixo do peso";}
    senao
    {
        se ( imc <=25. )
        {   imprima "\nnormal"; }
        senao
        {
            se ( imc <=30. )
            {   imprima "\nexcesso de peso "; }
            senao
            {
                se ( imc <=35. )
                {   imprima "\nobesidade "; }
                senao
                {   imprima "\no obesidade mórbida"; }
            }
        }
    }
    imprima "\n";
fimprog
```

algoritmo 152

Criar um algoritmo que a partir da idade e peso do paciente calcule a dosagem de determinado medicamento e imprima a receita informando quantas gotas do medicamento o paciente deve tomar por dose. Considere que o medicamento em questão possui 500 mg por ml, e que cada ml corresponde a 20 gotas.

■ Adultos ou adolescentes desde 12 anos, inclusive, se tiverem peso igual ou acima de 60 quilos devem tomar 1000 mg; com peso abaixo de 60 quilos devem tomar 875 mg.

■ Para crianças e adolescentes abaixo de 12 anos é dosagem é calculada pelo peso corpóreo conforme a tabela a seguir:

5 kg a 9 kg = 125 mg	24.1 kg a 30 kg = 500 mg
9.1 kg a 16 kg = 250 mg	acima de 30 kg = 750 mg
16.1 kg a 24 kg = 375 mg	

prog se63

```
int idade;
real peso,gotas;
imprima "\ndigite peso: ";
leia peso;
imprima "\ndigite idade: ";
leia idade;
gotas <- 500 / 20; # calculo do número de mg por gotas
se (peso <5.)
{ imprima "\nnao pode tomar o remedio porque nao tem peso. Consulte
medico." }
senao
{
  se ( idade >= 12)
  {
    se ( peso >= 60. )
    { imprima "\nTomar ",1000 / gotas, " gotas"; }
    senao
    { imprima "\nTomar ", 875 /gotas, " gotas"; }
  }
  senao
  {
    se ( peso <= 9.)
    { imprima "\nTomar ", 125 / gotas, " gotas"; }
    senao
    {
      se ( peso <= 16. )
      { imprima "\nTomar ", 250 / gotas, " gotas"; }
      senao
      {
```

```

    se ( peso <= 24. )
    { imprima "\nTomar ", 375 / gotas, " gotas"; }
    senao
    {
        se ( peso <=30. )
        { imprima "\nTomar ", 500 / gotas, " gotas"; }
        senao
        { imprima "\nTomar ", 750 / gotas, " gotas"; }
    }
}
}
imprima "\n";
fimprog

```

algoritmo 153

O prefeito do Rio de Janeiro contratou uma firma especializada para manter os níveis de poluição considerados ideiais para um país do 1º mundo. As indústrias, maiores responsáveis pela poluição, foram classificadas em três grupos. Sabendo-se que a escala utilizada varia de 0,05 e que o índice de poluição aceitável é até 0,25, fazer um algoritmo que possa imprimir intimações de acordo com o índice e a tabela a seguir:

<i>índice</i>	<i>indústrias que receberão intimação</i>
0,3	1º grupo
0,4	1º e 2º grupos
0,5	1º, 2º e 3º grupos

```

prog se64
real indice;
imprima "\ndigite indice de poluicao: ";
leia indice;
se ( indice >= 0.5 )
{ imprima "\nsuspender atividades as industrias dos grupos 1, 2 e 3";}
senao
{
    se ( indice >= 0.4 )
    { imprima "\nsuspender atividades as industrias dos grupos 1 e 2 ";}
    senao
    {
        se ( indice >= 0.3 )
        { imprima "\nsuspender atividades as industrias dos grupos 1 ";}
        senao
        { imprima "\nindice de poluicao aceitavel para todos os grupos ";}
    }
}
imprima "\n";
fimprog

```

algoritmo 154

A polícia rodoviária resolveu fazer cumprir a lei e cobrar dos motoristas o DUT. Sabendo-se que o mês em que o emplacamento do carro deve ser renovado é determinado pelo último número da placa do mesmo, criar um algoritmo que, a partir da leitura da placa do carro, informe o mês em que o emplacamento deve ser renovado.

↳ Leia a placa do carro em uma variável caracter.

```
prog se65
    string placa, p;
    imprima "\ndigite placa: ";
    leia placa ;
    p <- strlft(placa);
    se ( p == "1" )
    { imprima "\njaneiro"; }
    senao
    { se ( p == "2" )
        { imprima "\nfevereiro"; }
        senao
        { se ( p == "3" )
            { imprima "\nmarco"; }
            senao
            { se ( p == "4" )
                { imprima "\nabril"; }
                senao
                { se ( p == "5" )
                    { imprima "\nmaio"; }
                    senao
                    { se ( p == "6" )
                        { imprima "\njunho"; }
                        senao
                        { se ( p == "7" )
                            { imprima "\n julho"; }
                            senao
                            { se ( p == "8" )
                                { imprima "\nagosto"; }
                                senao
                                { se ( p == "9" )
                                    { imprima "\nsetembro"; }
                                    senao
                                    { se ( p == "0" )
                                        { imprima "\noutubro"; }
                                        senao
                                        { imprima "\nopcao invalida"; }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```

```

        }
    }
}
}

imprima "\n";
fimprog

```

algoritmo 155

Ler uma palavra e, se ela começar pela letra L ou D (também deve ser considerado l ou d), formar uma nova palavra que terá os dois primeiros caracteres e o último, caso contrário a nova palavra será formada por todos os caracteres menos o primeiro.

```

prog se66
string pal, pall, pal2, letra;
int tam;
imprima "\ndigite palavra: ";
leia pal;
letra <- strprim(pal);
se( letra == "l" || letra == "L" || letra == "d" || letra == "D" )
{ pall <- strnprim(pal,2);
  pal2 <- strult(pal);
  pall <- strconcat(pall,pal2);}
senao
{pall <- strresto(pal);}
imprima "\npalavra: ", pall;
imprima "\n";
fimprog

```

algoritmo 156

Criar um algoritmo que leia uma data (dia, mês e ano em separado) e imprima se a data é válida ou não.

```

prog se67
int ANO, MES, DIA, VD;
imprima "\ndigite dia: ";
leia DIA;
imprima "\ndigite mes: ";
leia MES;
imprima "\ndigite ano: ";
leia ANO;
se(ANO>=1)

```

```

{ VD <- 1;
se( MES < 1 || MES > 12 || DIA < 1 || DIA > 31)
{ VD <- 0; }
senao
{
  se( (MES == 4 || MES == 6 || MES == 9 || MES == 11) && DIA > 30)
  {VD <- 0;}
  senao
  { se(MES == 2)
  {
    se((ANO % 4==0 && ANO % 100 <> 0) || ANO % 400==0)
    {
      se(DIA > 29)
      { VD <- 0;}
    }
    senao
    {
      se( DIA > 28)
      { VD <- 0;}
    }
  }
}
}
senao
{ VD <- 0;}
se( VD == 0 )
{ imprima "\nDATA INVALIDA";}
senao
{ imprima "\nDATA VALIDA";}
imprima "\n";
fimprog

```

algoritmo 157

Criar um algoritmo que leia uma data (no formato ddmmaaaa) e imprimir se a data é válida ou não.

```

prog se68
int DATA, ANO, MES, DIA, VD;
imprima "\ndata no formato ddmmaaaa: ";
leia DATA;
DIA <- DATA div 1000000;
MES <- DATA % 1000000 div 10000;
ANO <- DATA % 10000;
se(ANO >=1)
{ VD <- 1;
  se(MES > 12 || DIA < 1 || DIA > 31)
  { VD <- 0; }
}

```

```

senao
{
  se((MES == 4 || MES == 6 || MES == 9 || MES == 11) && DIA > 30)
  { VD <- 0; }

  senao
  {
    se(MES == 2)
    {
      se((ANO % 4 == 0 && ANO % 100 < > 0) || ANO % 400 == 0)
      {
        se(DIA > 29)
        { VD <- 0; }
      }

      senao
      { se( DIA > 28)
        { VD <- 0; }
      }
    }
  }
}

senao
{ VD <- 0; }

se( VD == 0 )
{ imprima "\nDATA INVALIDA"; }

senao
{ imprima "\nDATA VALIDA"; }

imprima "\n";
fimprog

```

algoritmo 158

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

$$f(x) = \frac{8}{2-x}$$

```

prog se69
  real x, fx;
  imprima "\ndigite valor de x: ";
  leia x;
  se(x < > 2. )
  {fx <- 8 / (2 - x);
  imprima "\nf(x)= ",fx;
  }
  senao
  {imprima "\nNAO PODE SER FEITA";}
  imprima "\n";
fimprog

```

algoritmo 159

Criar um algoritmo que entre com o valor de x , calcule e imprima o valor de $f(x)$.

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

```
prog se70
real x, fx;
imprima "\ndigite valor de x: ";
leia x;
se(x > 4. || x < (-4.))
{fx <- (5 * x + 3) / raiz(x**2 - 16);
imprima "\nf(x)= ",fx;
}
senao
{imprima "\nNAO PODE SER FEITA";}
imprima "\n";
fimprog
```

algoritmo 160

Entrar com o valor de x e imprimir y :

$$Y = f(x) \begin{cases} 1, & \text{se } x \leq 1 \\ 2, & \text{se } 1 < x \leq 2 \\ x^2, & \text{se } 2 < x \leq 3 \\ x^3, & \text{se } x > 3 \end{cases}$$

```
prog se71
real x, y;
imprima "\ndigite valor de x: ";
leia x;
se(x <= 1.)
{y <- 1.;}
senao
{ se(x <= 2.)
{y <- 2.;}
senao
{ se(x <= 3.)
{y <- x **2;}
senao
{y <- x **3;}
}
}
imprima "\nvalor de f(x): ",y;
imprima "\n";
fimprog
```