

As conversões de português estruturado para Pascal foram feitas pelo aluno Cleber Alves e ainda não foram todas implementadas e testadas.

Algoritmo para calcular a idade média dos alunos

Faça um programa estruturado para determinar:

- (a) a quantidade de alunos do sexo masculino
- (b) a quantidade de alunas do sexo feminino
- (c) a idade média dos alunos
- (d) a idade média das alunas

Português estruturado

Início

```
//Declaração de variáveis
Inteiro: iMasc, iFem, iSomaMasc, iSomaFem, iIdade;
Real: rMediaMasc, rMediaFem, rMedia;
Caracter: cSexo;
// Início das variáveis
iMasc ← 0;
iFem ← 0;
iSomaMasc ← 0;
iSomaFem ← 0;
Leia (iIdade, cSexo);
Enquanto idade > 0 faça
    Se cSexo = 'M' então
        iMasc ← iMasc + 1
        iSomaMasc ← iSomaMasc + iIdade;
    Senão
        iFem ← iFem + 1;
        iSomaFem ← iSomaFem + iIdade;
Fim_se
Leia (iIdade, cSexo)
Fim_enquanto.
// Saída dos resultados
rMediaMasc ← iSomaMasc / iMasc;
rMediaFem ← iSomaFem / iFem;
rMedia ← [(iSomaMasc + iSomaFem) / (iMasc + iFem)];
Escreva (iMasc, rMediaMasc, iFem, rMediaFem, iMasc + iFem,
        rMedia);
```

Fim

Pascal

```
Program nome_do_programa;
Uses crt;
Var
    {Declaração de variáveis}
    iMasc, iFem, iSomaMasc, iSomaFem, iIdade : Integer;
    rMediaMasc, rMediaFem, rMedia : Real;
    cSexo : Char;
Begin
    {Início das variáveis}
    iMasc := 0;
    iFem := 0;
    iSomaMasc := 0;
    iSomaFem := 0;
    Readln (iIdade, cSexo);
    While idade > 0 do
        Begin
            If cSexo = 'M' then
                Begin
                    iMasc := iMasc + 1;
                    iSomaMasc := iSomaMasc + iIdade;
                End
            Else
                Begin
                    iFem := iFem + 1;
                    iSomaFem := iSomaFem + iIdade;
                End;
            Readln (iIdade, cSexo)
        End;
    {Saída dos resultados}
    rMediaMasc := iSomaMasc / iMasc;
    rMediaFem := iSomaFem / iFem;
    rMedia := [(iSomaMasc + iSomaFem) / (iMasc + iFem)];
    Writeln (iMasc, rMediaMasc, iFem, rMediaFem, iMasc + iFem,
            rMedia);
Readkey
End.
```

Programa

Português estruturado

// foi omitida a declaração de variáveis

Início

Leia (expoente);

$B \leftarrow 2;$

$I \leftarrow 1;$

Enquanto $I < \text{expoente}$ *faça*

$B \leftarrow B + B;$

$I \leftarrow I + 1;$

Fim_enquanto

Escreva ('Resultado =', B);

Fim.

Pascal

Program nome_do_programa;

Var

{Foi omitida a declaração de variáveis}

Begin

Readln (expoente);

B := 2;

I := 1;

While I < expoente do

Begin

B := B + B;

I := I + 1;

End;

Writeln ('Resultado =', B);

End.

Programa

Português estruturado

Início

```
// Declaração de variáveis
Inteiro: B, I, Expoente;
Leia (expoente);
B ← 3;
I ← 1;
Enquanto I < expoente faça
    B ← B + B + B;
    I ← I + 1;
Fim_enquanto
Escreva ('Resultado = ', B)
```

Fim

Pascal

```
Program nome_do_programa;
Var
    {Declaração de variáveis}
    B, I, Expoente : Integer;
Begin
    Readln (expoente);
    B := 3;
    I := 1;
    While I < expoente do
        Begin
            B := B + B + B;
            I := I + 1;
        End;
    Writeln ('Resultado = ', B)
End.
```

Programa

Português estruturado

```
//Variáveis declaradas
Início
  Leia (base, expoente);
  B ← base;
  I ← 1;
  Enquanto I < expoente faça
    P ← B;
    J ← 1;
    Enquanto J < base faça
      B ← B + P;
      J ← J + 1;
    Fim_enquanto
    I ← I + 1;
  Fim_enquanto
  Escreva ('Resultado = ', B);
Fim
```

Pascal

```
Program nome_do_programa;
Var
  {Variáveis declaradas}
Begin
  Readln (base, expoente);
  B := base;
  I := 1;
  While I < expoente do
    Begin
      P := B;
      J := 1;
      While J < base do
        Begin
          B := B + P;
          J := J + 1;
        End;
      I := I + 1;
    End;
  Writeln ('Resultado = ', B);
End.
```

Programa

Português estruturado

Início

```
Real; rN1, rN2, rN3, rN4, rMN;  
Escreva ('Digite as notas bimestrais');  
Leia (rN1, rN2, rN3, rN4),  
rMN ←(rN1+rN2+rN3+rN4)/4;  
Escreva ('Média annual = ', rMN);
```

Fim.

Pascal

```
Program nome_do_programa;  
Uses crt;  
Var  
    rN1, rN2, rN3, rN4, rMN : Real;  
Begin  
    Writeln ('Digite as notas bimestrais');  
    Readln (rN1, rN2, rN3, rN4),  
    rMN :=(rN1+rN2+rN3+rN4)/4;  
    Writeln ('Média anual = ', rMN);  
Readkey;  
End.
```

Construa o algoritmo:

Quantidade de latas necessárias e o custo para pintar tanques cilíndricos de combustível sabendo a altura e o raio do cilindro, ainda sabe-se que o valor de cada lata de tinta é de R\$50,00, cada lata tem 5 litros, cada litro pinta 3m².

Observação: área do cilindro = $2\pi (r^2 + rh)$.

Entradas	rAltura rRaio
Saídas	rQuantLatas rCustoPintura
Constantes	rPi := 3,14 rLitrosPorLata := 5 rPreçoLata := 50,00 rRendPorLitro := 3
Fórmulas auxiliares	rAreaTotal := rAreaLat+2*rAreaBase rAreaLat := 2*rPi*rRaio*rAltura rAreaBase := rPi*rRaio*rRaio

Português estruturado

Início

```
//Declaração de variáveis
real: rAltura, rRaio; //Entrada
real: rAreaTotal, rAreaLat, rAreaBase; //Auxiliar
inteiro: iQuantLatas;
real: rQuantLitros, rCusto;
//Declaração das constantes
constante: (rPi ← 3,14), (rLitrosPorlata ← 5),
            (rPreçoLata ← 50,00), (rRendPorLitro ← 3);
//Leitura da altura e raio
Escreva ('Digite a altura e raio (m)');
Leia (rAltura, rRaio);
//Cálculos
rAreaLat ← 2*rPi*rRaio*rAltura;
rAreaBase ← rPi*rRaio*rRaio;
rAreaTotal ← rAreaLat+2*rAreaBase;
rQuantLitros ← rAreaTotal/rRendPorLitro;
iQuantLatas ← DIV(rQuantLitros, rLitrosPorLata);
Se MOD(rQuantLitros, rLitrosPorLata) > 0 então
    iQuantLatas ← iQuantlatas+1;
Fim_se
rCusto ← iQuantLatas*rPreçoLata
Escreva ('Número de latas = ', iQuantLatas, 'Custo = ',
rCusto);
```

Fim

Pascal

```
Program nome_do_programa;  
Uses crt;  
Var  
    {Declaração de variáveis}  
    rAltura, rRaio : real; {Entrada}  
    rAreaTotal, rAreaLat, rAreaBase : real; {Auxiliar}  
    iQuantLatas : Integer;  
    rQuantLitros, rCusto : real;  
Const  
    {Declaração das constantes}  
    rPi := 3.14;  
    rLitrosPorLata := 5;  
    rPreçoLata := 50.00;  
    rRendPorLitro := 3;  
Begin  
    {Leitura da altura e raio}  
    Writeln ('Digite a altura e raio (m)');  
    Readln (rAltura, rRaio);  
    {Cálculos}  
    rAreaLat := 2*rPi*rRaio*rAltura;  
    rAreaBase := rPi*rRaio*rRaio;  
    rAreaTotal := rAreaLat+2*rAreaBase;  
    rQuantLitros := rAreaTotal/rRendPorLitro;  
    iQuantLatas := rQuantLitros DIV rLitrosPorLata;  
    If rQuantLitros MOD rLitrosPorLata > 0 then  
        Begin  
            iQuantLatas := iQuantlatas+1;  
        End;  
    rCusto := iQuantLatas*rPreçoLata  
    Writeln ('Número de latas = ', iQuantLatas, 'Custo = ',  
        rCusto);  
Readkey;  
End.
```

Construa um algoritmo para calcular as raízes de uma equação do 2º grau:

$ax^2 + bx + c = 0$

Dados de entrada: a, b e c

Dados de saída: x1 e x2

Português estruturado

Início

//Declaração de variáveis

real: rA, rB, rC, rX1, rX2, rDelta;

//Leitura dos coeficientes

Escreva ('Digite a, b e c');

Leia (rA, rB, rC);

//Cálculo do discriminante

rDelta ← (rB*rB-4*rA*rC);

Se rDelta < 0 então

 Escreva ('Não existem raízes no campo dos reais');

 Saia;

Fim_se

rDelta ← RAD(rDelta);

rX1 ← (-rB+rDelta/2*rA;

rX2 ← (-rB-rDelta/2*rA;

Escreva ('x₁ = ' rX1, 'x₂ = ', rX2)

Fim

Pascal

Program eq_do_segundo_grau;

Uses crt;

Var

 {Declaração de variáveis}

 rA, rB, rC, rX1, rX2, rDelta : real;

Begin

 {Leitura dos coeficientes}

 Writeln ('Digite a, b e c');

 Readln (rA, rB, rC);

 {Cálculo do discriminante

 rDelta := (rB*rB-4*rA*rC);

 If rDelta < 0 then

 Begin

 Writeln ('Não existem raízes no campo dos
 reais');

 Exit;

 End;

 rDelta := SQRT(rDelta);

 rX1 := (-rB+rDelta)/2*rA;

 rX2 := (-rB-rDelta)/2*rA;

 Writeln ('x1 = ' rX1, 'x2 = ', rX2)

Readkey;

End.

À partir dos lados de um triângulo, verifique se o mesmo é equilátero, isósceles ou escaleno.

Entradas: Lado1, Lado2, Lado3

Saídas: Frases

Português estruturado

Início

```
//Declaração de variáveis
real: rLado1, rLado2, rLado3;
Escreva ('Digite o valor dos lados');
Leia (rLado1, rLado2, rLado3);
Se rLado1 = rLado2 então
    Se rLado 2 = rLado3 então
        Escreva ('O triângulo é equilátero');
    Senão
        Escreva ( 'O triângulo é isósceles');
Fim_se
Senão
    Se rLado 2 = rLado3 então
        Escreva ('O triângulo é isósceles');
    Senão
        Escreva ('O triângulo é escaleno')
Fim_se
Fim_se
```

Fim

Pascal

```
Program lado_do_triag;
Uses crt;
Var
    {Declaração de variáveis}
    rLado1, rLado2, rLado3 : real;
Begin
    Writeln ('Digite o valor dos lados');
    Readln (rLado1, rLado2, rLado3);
    If rLado1 = rLado2 then
        Begin
            If rLado2 = rLado3 then
                Begin
                    Writeln ('O triângulo é equilátero');
                End
            Else
                Begin
                    Writeln ( 'O triângulo é isósceles');
                End;
        End
    Else
        Begin
            If rLado2 = rLado3 then
                Begin
                    Writeln ('O triângulo é isósceles');
                End
            Else
                Begin
                    Writeln ('O triângulo é escaleno')
                End;
        End;
    End;
Readkey;
End.
```

Construa um algoritmo que calcule a média aritmética entre quatro notas fornecidas pelo usuário. Verifique ainda se o aluno foi aprovado (média ≥ 7) ou reprovado.

Dados de entrada: Nota 1, Nota2, Nota3, Nota4

Dados de saída: Frase

Português estruturado

Início

```
//Declaração de variáveis
real: rNota1, rNota2, rNota3, rNota4, rMedia;
//Leitura das notas
Escreva ('Digite as 4 notas');
Leia(rNota1, rNota2, rNota3, rNota4);
//Cálculo da média
rMedia ← (rNota1+rNota2+rNota3+rNota4)/4;
//Seleção da frase
Se rMedia >= 7 então
    Escreva ('Aluno aprovado');
Senão
    Escreva ('Aluno reprovado');
Fim_se
```

Fim

Pascal

```
Program media_notas;
Uses crt;
Var
    {Declaração de variáveis}
    rNota1, rNota2, rNota3, rNota4, rMedia : real;
Begin
    {Leitura das notas}
    Writeln ('Digite as 4 notas');
    Readln (rNota1, rNota2, rNota3, rNota4);
    {Cálculo da média}
    rMedia := (rNota1+rNota2+rNota3+rNota4)/4;
    {Seleção da frase}
    If rMedia >= 7 then
        Begin
            Writeln ('Aluno aprovado');
        End
    Else
        Begin
            Writeln ('Aluno reprovado');
        End;
    Readkey;
End.
```

Contando as faltas

Dados de entrada: Nota 1, Nota2, Nota3, Nota4

Dados de saída: Frase

Português estruturado

Início

```
//Declaração de variáveis
inteiro: iFaltas
real: rNota 1, rNota2, rNota3, rNota4, rMedia;
//Leitura das notas
Escreva ('Digite as 4 notas');
Leia(rNota 1, rNota2, rNota3, rNota4);
Escreva ('Digite o número de faltas');
Leia (iFaltas)
//Cálculo da média
rMedia ← (rNota 1+rNota2+rNota3+rNota4)/4;
//Seleção da frase
Se rMedia < 7 então
    Escreva ('Aluno reprovado');
Senão
    Se iFaltas > 18 então
        Escreva('Aluno reprovado por faltas')
    Senão
        Escreva ('Aluno aprovado')
    Fim_se
Fim_se
```

Fim

Pascal

```
Program media_com_faltas;
Uses crt;
Var
    {Declaração de variáveis}
    iFaltas : Integer;
    rNota1, rNota2, rNota3, rNota4, rMedia : real;
Begin
    {Leitura das notas}
    Writeln ('Digite as 4 notas');
    Readln (rNota1, rNota2, rNota3, rNota4);
    Writeln ('Digite o número de faltas');
    Readln (iFaltas);
    {Cálculo da média}
    rMedia := (rNota1+rNota2+rNota3+rNota4)/4;
    {Seleção da frase}
    If rMedia < 7 then
        Begin
            Writeln ('Aluno reprovado');
        End
    Else
        Begin
            If iFaltas > 18 then
                Begin
                    Writeln('Aluno reprovado por faltas');
                End
            Else
                Begin
                    Writeln ('Aluno aprovado');
                End;
        End;
    Readkey; .
End.
```

Sabendo que uma sala de aula tem 50 alunos determine a média aritmética da primeira prova dos alunos.

Português estruturado

Início

```
//Declaração de variáveis
real: rNota, rMedia;
inteiro: i;
//Leitura das notas
rMedia ← 0;
Para i ← 1 até 50 faça
    Escreva ('Nota do aluno = ', i);
    Leia (rNota);
Fim_para
rMedia ← rMedia/50;
Escreva ('Média = ', rMedia);
```

Fim

Pascal

```
Program media_50_alunos;
Uses crt;
Var
    {Declaração de variáveis}
    rNota, rMedia : real;
    i : Integer;
Begin
    {Leitura das notas}
    rMedia := 0;
    For i := 1 to 50 do
        Begin
            Writeln ('Nota do aluno = ', i);
            Readln (rNota);
        End;
    rMedia := rMedia/50;
    Writeln ('Média = ', rMedia);
    Readkey;
End.
```

Agora calcule a média de n alunos

Português estruturado

```
Início
    //Declaração de variáveis
    real: rNota, rMedia;
    inteiro: i, iNumAlunos;
    Leia (iNumAlunos);
    //Leitura das notas
    rMedia ← 0;
    Para i ← 1 até iNumAlunos faça
        Escreva ('Nota do aluno = ', i);
        Leia (rNota);
        rMedia ← rMedia + rNota;
    Fim_para
    rMedia ← rMedia/iNumAlunos;
    Escreva ('Média = ', rMedia);
Fim
```

Observação 1:

```
Leia (rNota);
Enquanto rNota < 0 ou rNota > 10 faça
    Leia (rNota)
Fim_enquanto
```

Observação 2:

```
Faça
    Leia (rNota);
    Enquanto rNota < 0 ou rNota > 10;
```

Pascal

```
Program media_n_alunos;
Uses crt;
Var
    {Declaração de variáveis}
    rNota, rMedia : Real;
    i, iNumAlunos : Integer;
Begin
    Readln (iNumAlunos);
    {Leitura das notas}
    rMedia := 0;
    For i := 1 to iNumAlunos do
        Begin
            Writeln ('Nota do aluno = ', i);
            Readln (rNota);
            rMedia := rMedia + rNota;
        End;
    rMedia := rMedia/iNumAlunos;
    Writeln ('Média = ', rMedia);
    Readkey;
End.
```

Observação 1:

```
Readln (rNota);
While rNota < 0 or rNota > 10 do
    Readln (rNota)
End.
```

Observação 2:

```
Do
    Readln (rNota);
    While rNota < 0 or rNota > 10;
```

Construa um algoritmo que, dado um conjunto de valores inteiros e positivos, determine qual o menor e o maior valor do conjunto. O final do conjunto de valores é conhecido através do -1 e não deve ser considerado.

Português estruturado

Início

```
//Declaração de variáveis
Inteiro: iValor, iMenor, iMaior;
//Preparo
Escreva ('Digite um valor inteiro ou -1 para sair');
Leia (iValor);
iMenor ← iValor;
iMaior ← iValor;
Enquanto iValor > -1 faça
    Se iValor > iMaior então
        iMaior ← iValor;
    Fim_se
    Se iValor < iMenor então
        iMenor ← iValor;
    Fim_se
    Escreva ('Digite um valor inteiro ou -1 para sair');
    Leia (iValor);
Fim_Enquanto
Escreva ('O menor valor é = ', iMenor,
        'O maior valor é = ', iMaior);
```

Fim

Pascal

```
Program nome_do_programa;
Uses crt;
Var
    {Declaração de variáveis}
    iValor, iMenor, iMaior : Integer
Begin
    {Preparo}
    Writeln ('Digite um valor inteiro ou -1 para sair');
    Readln (iValor);
    iMenor := iValor;
    iMaior := iValor;
    While iValor > -1 do
        Begin
            If iValor > iMaior then
                Begin
                    iMaior := iValor;
                End;
            If iValor < iMenor then
                Begin
                    iMenor := iValor;
                End;
            Writeln ('Digite um valor inteiro ou -1 para sair');
            Readln (iValor);
        End;
    Writeln ('O menor valor é = ', iMenor);
    Writeln ('O maior valor é = ', iMaior);
    Readkey;
End.
```

A conversão de graus Fahrenheit para centígrados é conhecida pela fórmula $C = \frac{5(F - 32)}{9}$

9. Escreva um algoritmo que calcule e escreva uma tabela de graus centígrados em graus Fahrenheit que variem de 50 a 150 de 1 em 1.

Português estruturado

Início

```
//Declaração de variáveis  
Real: rF, rC;  
Escreva ('F ... C');  
Para rF ← 50 até 150  
    rC ← (rF-32)*5/9;  
    Escreva (rF, '...', rC);  
Fim_Para
```

Fim

Pascal

```
Program converte_temperatura;  
Uses crt;  
Var  
    {Declaração de variáveis}  
    rF, rC : Real;  
Begin  
    Writeln ('F .. C');  
    For rF := 50 to 150 do  
        Begin  
            rC := (rF-32)*5/9;  
            Writeln (rF, '..', rC);  
        End;  
    Readkey;  
End.
```

O mesmo algoritmo anterior para um intervalo qualquer de temperatura e variações quaisquer.

Português estruturado

Início

```
//Declaração de variáveis
Real: rF, rC, rFI, rFF, rVar;
//Entrada de dados
Escreva ('Digite o valor inicial em °F');
Leia (rFI);
Escreva ('Digite o valor final em °F');
Leia (rFF);
Escreva('A variação da tabela');
Leia(rVar);
Para rF ← rFI até rFF de rVar em rVr faça
    rC ← (rF - 32)*5/9;
    Escreva (rFL, ... , LrC);
Fim_Para
```

Fim

Pascal

```
Program nome_do_programa;
Uses crt;
Var
    {Declaração de variáveis}
    rF, rC, rFI, rFF, rVar : Real;
Begin
    {Entrada de dados}
    Writeln ('Digite o valor inicial em °F');
    Readln (rFI);
    Writeln ('Digite o valor final em °F');
    Readln (rFF);
    Writeln('A variação da tabela');
    Readln(rVar);
    For rF := rFI to rFF de rVar em rVr do
        Begin
            rC := (rF - 32)*5/9;
            Writeln (rFL, .. , LrC);
        End;
    Readkey;
End.
```

Obs não sei como fazer o comando **de <cond> em**

Construa um algoritmo que leia um conjunto de dados contendo altura e sexo (masculino e feminino) de 50 pessoas e depois calcule e escreva:

A maior e a menor altura do grupo;

A média de altura das mulheres;

número de homens e a diferença percentual entre estes e as mulheres.

Atenção: este exercício não foi resolvido pelo professor, a resolução abaixo foi feita pelo aluno Cleber Alves

Português estruturado

Início

```
//Declaração de variáveis
Real: rAltura, rAlturaMulher, rMaior, rMenor rMediaMulher,
      rPercHomem, rDifPerc;
Caracter: cSexo;
Inteiro: iMulher, i, iHomem;
//Entrada de dados
Escreva ('Digite a altura');
Leia (rAltura);
Escreva ('Digite o sexo');
Leia (cSexo);
//Inicio das variáveis
rAlturaMulher ← 0;
rMaior ← rAltura;
rMenor ← rAltura;
iMulher ← 0;
//Processamento
Para i ← 1 até 50 faça
  Se rAltura > rMaior então
    rMaior ← rAltura;
  Fim_Se
  Se rAltura < rMenor então
    rMenor ← rAltura;
  Fim_Se
  Se cSexo = "F" então
    rAlturaMulher ← rAlturaMulher + rAltura;
    iMulher ← iMulher + 1
  Fim_Se
Fim_Para
Se iMulher = 0 então
  rMediaMulher ← 0;
Senão
  rMediaMulher ← rAlturaMulher/iMulher;
Fim_Se
iHomem ← 50 - iMulher;
rPercHomem ← 2 * iHomem;
rDifPerc ← 100 - rPercHomem;
Escreva ('A maior altura do grupo é =', rMaior, 'A menor altura
do grupo é =', rMenor, 'A média das alturas das
mulheres é =', rMediaMulher, 'A diferença percentual
é =', rDifPerc);
```

Fim

Pascal

```
Program alt_sexo;
Uses crt;
Var
  {Declaração de variáveis}
  rAltura, rAlturaMulher, rMaior, rMenor: Real;
  rMediaMulher, rPercHomem, rDifPerc : Real;
  cSexo : Char;
  iMulher, i, iHomem : Integer;
Begin
  {Entrada de dados}
  Writeln ('Digite a altura');
  Readln (rAltura);
  Writeln ('Digite o sexo');
  Readln (cSexo);
  {Inicio das variáveis}
  rAlturaMulher := 0;
  rMaior := rAltura;
  rMenor := rAltura;
  iMulher := 0;
  {Processamento}
  For i := 1 to 50 do
    Begin
      If rAltura > rMaior then
        Begin
          rMaior := rAltura;
        End;
      If rAltura < rMenor then
        Begin
          rMenor := rAltura;
        End;
      If cSexo = "F" then
        Begin
          rAlturaMulher := rAlturaMulher + rAltura;
          iMulher := iMulher + 1
        End;
    End;
  If iMulher = 0 then
    Begin
      rMediaMulher := 0;
    End
  Else
    Begin
      rMediaMulher := rAlturaMulher/iMulher;
    End;
  iHomem := 50 - iMulher;
  rPercHomem := 2 * iHomem;
  rDifPerc := 100 - rPercHomem;
  Writeln ('A maior altura do grupo é = ' , rMaior);
  Writeln ('A menor altura do grupo é = ' , rMenor);
  Writeln ('A média das alturas das mulheres é = ' , rMediaMulher);
  Writeln ('A diferença percentual é = ' , rDifPerc);
  Readkey;
End.
```

Dados três lados de um triângulo, classifique-o quanto aos lados.

Português estruturado

Início

```
//Declaração de variáveis  
Caracter: cTipo;  
Real: rLado1, rLado2, rLado3;  
Escreva ('Digite os lados do triângulo');  
Leia (rLado1, rLado2, rLado3);  
Se rLado1 = rLado2 então  
    Se rLado1 = rLado3 então  
        cTipo ← "Equilátero";  
    Senão  
        cTipo ← "Isósceles";  
Fim_Se  
Senão  
    Se rLado1 = rLado3 então  
        cTipo ← "Isósceles";  
    Senão  
        Se rLado 2 = rLado3;  
            cTipo ← "Isósceles";  
        Senão  
            cTipo ← "Escaleno";  
    Fim_Se  
Fim_Se  
Escreva ('O triângulo é' , cTipo)  
Fim
```

Pascal

```
Program clas_triangu;
Uses crt;
Var
    {Declaração de variáveis}
    cTipo : Char;
    rLado1, rLado2, rLado3 : Real;
Begin
    Writeln ('Digite os lados do triângulo');
    Readln (rLado1, rLado2, rLado3);
    If rLado1 = rLado2 then
        Begin
            If rLado1 = rLado3 then
                Begin
                    cTipo := "Equilátero";
                End
            Else
                Begin
                    cTipo := "Isósceles";
                End;
        End
    Else
        Begin
            If rLado1 = rLado3 then
                Begin
                    cTipo := "Isósceles";
                End
            Else
                Begin
                    If rLado 2 = rLado3;
                    Begin
                        cTipo := "Isósceles";
                    End
                    Else
                        Begin
                            cTipo := "Escaleno";
                        End;
                End;
        End;
    Writeln ('O triângulo é' , cTipo)
    Readkey;
End
```

Calcular o MDC entre dois números dados

Algoritmo de Euclides

MDC (126, 20)

Quociente		6	3	3
Dividendo/ divisor	126	20	6	2 é o MDC
Resto	6	2	0	

Português estruturado

```
//MDC 25-03-2000 - LPT I - UNiCEUB
```

```
Início
```

```
  //Declaração de variáveis
```

```
  Inteiro: iNum1, iNum2, iDividendo, iDivisor, iResto;
```

```
  Escreva ('Digite dois números inteiros maiores que zero');
```

```
  Leia (iNum1, iNum2);
```

```
  Se iNum1 > iNum2 então
```

```
    iDividendo ← iNum1;
```

```
    iDivisor ← iNum2;
```

```
  Senão
```

```
    iDividendo ← iNum2;
```

```
    iDivisor ← iNum1;
```

```
  Fim_Se
```

```
  Faça
```

```
    iResto ← MOD (iDividendo, iDivisor);
```

```
    iDividendo ← iDivisor;
```

```
    iDivisor ← iResto;
```

```
  Enquanto iResto <>0;
```

```
  Escreva ('MDC (' , iNum1, ', ', iNum2, ') = ', iDividendo);
```

```
Fim
```

Pascal

```
{MDC 25-03-2000 - LPT I - UniCEUB}
Program mdc;
Uses crt;
Var
    {Declaração de variáveis}
    iNum1, iNum2, iDividendo, iDivisor, iResto : Integer;
Begin
    Writeln ('Digite dois números inteiros maiores que zero');
    Readln (iNum1, iNum2);
    If iNum1 > iNum2 then
        Begin
            iDividendo := iNum1;
            iDivisor := iNum2;
        End
    Else
        Begin
            iDividendo := iNum2;
            iDivisor := iNum1;
        End;
    Do
        iResto := iDividendo MOD iDivisor;
        iDividendo := iDivisor;
        iDivisor := iResto;
    Until iResto <>0;
    Writeln ('MDC (', iNum1, ', ', iNum2, ') = ', iDividendo);
    Readkey;
End.
```

Calcule o MMC entre dois números dados.

$$\text{Observação: MMC (N1, N2)} = \frac{N1 * N2}{\text{MDC}(N1, N2)}$$

Português estruturado

//MMC - UniCEUB - 25-03-2000

Início

//Declaração de variáveis

Inteiro: iNum1, iNum2, iDividendo, iDivisor, iResto, iMDC, iMMC;

Escreva ('Digite dois números inteiros positivos');

Leia (iNum1, iNum2);

//Cálculo do MDC

iDividendo ← iNum1;

iDivisor ← iNum2;

Faça

 iResto ← MOD (iDividendo, iDivisor);

 iDividendo ← iDivisor;

 iDivisor ← iResto;

Enquanto iResto <> 0;

iMDC ← iDivisor

iMMC ← (iNum1 * iNum2)/iMDC;

Escreva ('MMC (' , iNum1, ', ' iNum2, ') = ', iMMC);

Fim

Pascal

{MMC - UniCEUB - 25-03-2000}

Program mmc;

Uses crt;

Var

 {Declaração de variáveis}

 iNum1, iNum2, iDividendo, iDivisor : Integer;

 iResto, iMDC, iMMC : Integer;

Begin

 Writeln ('Digite dois números inteiros positivos');

 Readln (iNum1, iNum2);

 {Cálculo do MDC}

 iDividendo := iNum1;

 iDivisor := iNum2;

 Do

 iResto := iDividendo MOD iDivisor;

 iDividendo := iDivisor;

 iDivisor := iResto;

 Until iResto <> 0;

 iMDC := iDivisor

 iMMC := (iNum1 * iNum2)/iMDC;

 Writeln ('MMC (' , iNum1, ', ' iNum2, ') = ', iMMC);

 Readkey;

End.

Dado um número inteiro, determine se ele é primo.

Português estruturado

```
//Número primo (n>2)
Início
  //Declaração de variáveis
  Lógico: lPrimo;
  Inteiro: iNum, iResto, iDivisor, iLimSup;
  Escreva ('Digite um número inteiro > 2');
  Leia (iNum);
  //Verificar se é par
  iResto ← MOD (iNum, 2);
  Se iResto <> 0 então
    lPrimo ← Verdade;
    iDivisor ← 3
    iLimSup ← RAD(iNum) +1;
  Senão
    lPrimo ← Falso
  Fim_Se
  Enquanto lPrimo e iDivisor ≤ iLimSup
    iResto ← MOD (iNum, iDivisor);
    Se iResto = 0 então
      lPrimo ← Falso;
    Senão
      iDivisor ← iDivisor + 2;
    Fim_Se
  Fim_enquanto
  Se lPrimo então
    Escreva (iNum, ' = Primo' );
  Senão
    Escreva (iNum, ' = não é Primo')
  Fim_Se
Fim
```

Pascal

```
{Número primo (n>2)}
Program primo;
Uses crt;
Var
    {Declaração de variáveis}
    lPrimo : boolean;
    iNum, iResto, iDivisor, iLimSup : Integer;
Begin
    Writeln ('Digite um número interio > 2');
    Readln (iNum);
    {Verificar se é par}
    iResto := iNum MOD 2;
    If iResto <> 0 then
        Begin
            lPrimo := Verdade;
            iDivisor := 3
            iLimSup := RAD(iNum) +1;
        End
    Else
        Begin
            lPrimo := Falso
        End;
    While lPrimo and iDivisor ≤ iLimSup
        Begin
            iResto := iNum MOD iDivisor;
            If iResto = 0 then
                Begin
                    lPrimo := Falso;
                End
            Else
                Begin
                    iDivisor := iDivisor + 2;
                End;
        End;
    If lPrimo then
        Begin
            Writeln (iNum, ' = é primo' );
        End
    Else
        Begin
            Writeln (iNum, ' = não é primo')
        End;
    Readkey;
End.
```

Programa
Português estruturado

Início

```
//Declaração de variáveis
real: rMedia;
inteiro:iAltura[10], i;
//Leitura das alturas
Para i ← 1 até 10 faça
    Escreva ('Altura do aluno', i);
    Leia (iAltura[i]);
Fim_Para
rMedia ← 0;
Para i ← 1 até 10 faça
    rMedia ← rMedia + iAltura[i];
Fim_Para
rMedia ← rMedia/10;
```

Fim.

Pascal

```
Program nome_do_programa;
Uses crt;
Var
    {Declaração de variáveis}
    rMedia : real;
    iAltura[10], i : Integer;
Begin
    {Leitura das alturas}
    For i := 1 to 10 do
        Begin
            Writeln ('Altura do aluno', i);
            Readln (iAltura[i]);
        End;
    rMedia := 0;
    For i := 1 to 10 do
        Begin
            rMedia := rMedia + iAltura[i];
        End;
    rMedia := rMedia/10;
    Readkey;
End.
```

Desenvolver um programa que leia uma cadeia de caracteres e inverta a posição dos mesmos:

Português estruturado

Início

```
//Declaração de variáveis
caractere: cFrase[80], cFraseInv[80];
inteiro: iTam;
Escreva ('Digite uma frase');
Leia (cFrase);
iTam ← tamanho_cadeia(cFrase);
Para i ← 1 até iTam faça
    cFraseInv[iTam - i + 1] ← cFrase[i];
Fim_Para
Escreva (cFraseInv);
```

Fim

Pascal

```
Program inverte_cadeia;
Uses crt;
Var
    {Declaração de variáveis}
    cFrase[80], cFraseInv[80] : char;
    iTam : Integer;
Begin
    Writeln ('Digite uma frase');
    Readln (cFrase);
    iTam := lenght(cFrase);
    For i := 1 to iTam do
        Begin
            cFraseInv[iTam - i + 1] := cFrase[i];
        End;
    Writeln (cFraseInv);
    Readkey;
End.
```

**Ler um conjunto de números inteiros positivos até que um valor não positivo seja lido.
Levantar o índice do menor e do maior valor do conjunto.
Exibir os índices e os valores correspondentes ao menor e ao maior número.
Observação: utilizar um vetor para armazenar os valores do conjunto de números
inteiro positivos.**

Português estruturado

Início

```
//Declaração de variáveis
inteiro: iNum[100], iValorMaior, iIndMaior, iValorMenor,
        iIndMenor, iQuantNum, i;
//Leitura do conjunto
iQuantNum ← 0;
Escreva ('Digite um número inteiro positivo');
Leia (iNum[i]);
i ← 1;
iValorMaior ← iNum[i];
iValorMenor ← iNum[i];
iIndMaior ← 1;
iIndMenor ← 1;
Enquanto i < 100 e iNum[i] > 0 faça
    i ← i + 1;
    Escreva ('Digite um número inteiro positivo');
    Leia (iNum[i]);
    Se iNum[i] < iValorMenor e iNum[i] > 0 então
        iValorMenor ← iNum[i];
        iIndMenor ← i;
    Fim_Se
    Se iNum[i] > iValorMaior e iNum[i] > 0 então
        iValorMaior ← iNum[i];
        iIndMaior ← i;
    Fim_Se
Fim_Enquanto
Se iNum[i] < 0 então
    iQuantNum ← i - 1;
Senão
    iQuantNum ← i;
Fim_Se
//Exibição dos resultados
Escreva ('Menor número_índice', iIndMenor, 'Valor = ',
        iNum[iIndMenor]), 'Maior número_índice ',
        iIndMaior, 'Valor = ', iNum[iIndMaior]);
Fim
```

Pascal

```
Program maior_menor_indice;
Uses crt;
Var
    {Declaração de variáveis}
    iNum[100], iValorMaior, iIndMaior : Integer;
    iValorMenor, iIndMenor, iQuantNum, i : Integer;
Begin
    {Leitura do conjunto}
    iQuantNum := 0;
    Writeln ('Digite um número inteiro positivo');
    Readln (iNum[i]);
    i := 1;
    iValorMaior := iNum[i];
    iValorMenor:= iNum[i];
    iIndMaior := 1;
    iIndMenor := 1;
    While i < 100 and iNum[i] > 0 do
        Begin
            i := i + 1;
            Writeln ('Digite um número inteiro positivo');
            Readln (iNum[i]);
            If iNum[i] < iValorMenor and iNum[i] > 0 then
                Begin
                    iValorMenor := iNum[i];
                    iIndMenor := i;
                End;
            If iNum[i] > iValorMaior and iNum[i] > 0 then
                Begin
                    iValorMaior := iNum[i];
                    iIndMaior := i;
                End;
            End;
        End;
    If iNum[i] < 0 then
        Begin
            iQuantNum := i - 1;
        End
    Else
        Begin
            iQuantNum := i;
        End;
    {Exibição dos resultados}
    Writeln ('Menor número_índice', iIndMenor);
    Writeln ('Menor Valor = ', iNum[iIndMenor]);
    Writeln ('Maior número_índice ', iIndMaior);
    Writeln ('Maior Valor = ', iNum[iIndMaior]);
    Readkey;
End.
```

O mesmo programa valorizando EPS

Português estruturado

Início

```
//Declaração de variáveis
inteiro: iNum[100], iValorMaior, iIndMaior, iValorMenor,
         iIndMenor, iQuantNum, i;
//Leitura do conjunto
i ← 1;
Escreva ('Digite um número inteiro positivo');
Leia (iNum[i]);
Enquanto i < 100 e iNum[i] > 0 faça
    i ← i + 1
    Escreva ('Digite um número inteiro positivo');
    Leia (iNum[i]);
Fim_Enquanto
Se iNum[i] < 0 então
    iQuantNum ← i - 1;
Senão
    iQuantNum ← i;
Fim_Se
//Determinação do menor e do maior valor
iIndMenor ← iIndMaior ← 1;
Para i ← 2 até iQuantNum faça
    Se iNum[i] < iNum[iIndManor] então
        iIndMenor ← i;
    Fim_Se
    Se iNum[i] > iNum[iIndMaior] então
        iIndMaior ← i;
    Fim_Se
Fim_Para
//Exibição dos resultados
Escreva ('Menor número_índice', iIndMenor, 'Valor = ',
iNum[iIndMenor]), 'Maior número_índice', iIndMaior, 'Valor
= ', iNum [iIndMaior]);
```

Fim

Pascal

```
Program maior_menor_indice;
Uses crt;
Var
    {Declaração de variáveis}
    iNum[100], iValorMaior, iIndMaior : Integer;
    iValorMenor, iIndMenor, iQuantNum, i : Integer;
Begin
    {Leitura do conjunto}
    i := 1;
    Writeln ('Digite um número inteiro positivo');
    Readln (iNum[i]);
    While i < 100 and iNum[i] > 0 do
        Begin
            i := i + 1;
            Writeln ('Digite um número inteiro positivo');
            Readln (iNum[i]);
        End;
    If iNum[i] < 0 then
        Begin
            iQuantNum := i - 1;
        End
    Else
        Begin
            iQuantNum := i;
        End;
    {Determinação do menor e do maior valor}
    iIndMenor := iIndMaior := 1;
    For i := 2 to iQuantNum do
        Begin
            If iNum[i] < iNum[iIndManor] then
                Begin
                    iIndMenor := i;
                End;
            If iNum[i] > iNum[iIndMaior] then
                Begin
                    iIndMaior := i;
                End;
        End;
    {Exibição dos resultados}
    Write ('Menor número_índice', iIndMenor);
    Write ('Menor Valor = ', iNum[iIndMenor]);
    Write ('Maior número_índice', iIndMaior);
    Write ('Maior Valor = ', iNum [iIndMaior]);
    Readkey;
End.
```

Programa Português estruturado

Início

```
//Declaração de variáveis
real: rNota[100][3];
inteiro: iFaltas[100];
Para i ← 1 até 100 faça
    Escreva ('Digite as faltas do aluno', i);
    Leia (iFaltas[i]);
    Se iFalta[i] < 0 então
        Quebra;
    Fim_Se
    Para j ← 1 até 3 faça
        Escreva ('Nota', j, 'do aluno', i);
        Leia (rNota[i][j]);
        Se rNotas[i][j] < 0 então
            Quebra;
        Fim_Se
    Fim_Para
Fim_Para
```

Fim

Pascal

```
Program nome_do_programa;
Uses crt;
Var
    {Declaração de variáveis
    rNota[100][3] : real;
    iFaltas[100] : Integer;
Begin
    For i := 1 to 100 do
        Begin
            Writeln ('Digite as faltas do aluno', i);
            Readln (iFaltas[i]);
            If iFalta[i] < 0 then
                Begin
                    Break;
                End;
            For j := 1 to 3 do
                Begin
                    Writeln ('Nota', j, 'do aluno', i);
                    Readln (rNota[i][j]);
                    If rNotas[i][j] < 0 then
                        Begin
                            Break;
                        End;
                End;
            End;
        End;
        Readkey;
End.
```