Versão original comentada:

```
Program Bhaskara ;
var a, b, c: integer;
    delta, x1, x2: real;
    x: real;

Begin
// Ler os valores das variáveis

  writeln('Insira o valor de a: ' ) ;
  readln( a) ;

  writeln('Insira o valor de b: ' ) ;
  readln( b) ;

  writeln('Insira o valor de c: ' ) ;
  readln( c) ;

//Verifica se o a é diferente de zero
//Se for diferente de zero ele calcula o delta e mostra o valor
//se for igual a zero ele mostra uma mensagem de erro
  if (a <> 0 ) then begin
     delta := (b * b) + (-4 * (a * c));
     writeln('delta = ', delta ) ;

 end
 else begin
     writeln('a não pode ser igual a 0' ) ;

 end;
//Verifica se delta é maior ou igual a zero
//Se for maior ou igual a zero, executa o outro if
//Se for menor que zero ele mostra uma mensagem

if (delta >= 0 ) then begin

//Verifica se delta é igual a zero
//Se o delta for igual a zero, ele calculae mostra o valor de x1 sem somar a raiz de
delta
//Se o delta for diferemte de zero, ele calcula e mostra o valor de x1 e x2
 if (delta = 0 ) then begin
  x1 := -b / (2 * a);
  writeln('x1 = ', x1 ) ;

 end
  else begin
      x1 := (-b + SQRT(delta)) / (2 * a);
  writeln('x1 = ', x1 ) ;

  x2 := (-b - SQRT(delta)) / (2 * a);
  writeln('x2 = ', x2 ) ;

 end;

end
else begin
writeln('a equação não tem valores reais' ) ;

end;

End.
```

Alerta geral:
a indentação está irregular (*ver versões alteradas*).

1) Não é necessário solicitar os valores separadamente;
2) Se a entrada for feita para cada valor separadamente, então é interessante fazer a verificação do valor de "a" logo que ele seja lido, sem forçar o operador a digitar os demais (*ver observação a seguir e versões alteradas*);
3) O uso do "writeln" faz com que o valor seja digitado a partir da linha seguinte; usa-se o "write" quando se deseja que o(s) valor(es) seja(m) digitado(s) na mesma linha do texto de orientação.

1) Desta maneira, quando o "a" for zero, a mensagem será mostrada, mas o programa continuará executando as instruções seguintes e o erro de divisão por zero não será evitado; a continuação só deve ocorrer se "a <> 0";
2) Não é necessário mostrar o valor de "delta"; isto pode ser útil apenas durante a elaboração do programa para ver se está fazendo certo; o fluxograma não tinha isto;
3) Somente os "writeln" necessitam dos "( )", neste trecho.

Cuidado:
o uso excessivo de comentários (e uso de comentários muito longos) pode inverter o seu propósito e acabar dificultando a leitura e a compreensão do programa.
Observações:
- a variável "x" declarada não foi usada;
- os "if" não precisam de parêntesis, em Pascal (basta "*if delta >= 0 then* ..." , por exemplo);
- "begin-end" só é necessário para agrupar uma sequência de comandos; com um só, não é necessário e dificulta a leitura.

Versão alterada (sugestão), mantida a indentação no estilo da versão original:

```
Program Bhaskara ;
var a, b, c: integer;
    delta, x1, x2: real;
    x: real;

Begin
// Ler os valores das variáveis, garantindo que o valor de "a"
// seja diferente de zero para calcular "delta" e prosseguir

  writeln('Insira o valor de a: ' ) ;
  readln( a) ;

  if a <> 0 then begin
    writeln('Insira o valor de b: ' ) ;
    readln( b) ;

    writeln('Insira o valor de c: ' ) ;
    readln( c) ;

    delta := b*b - 4*a*c;

// Verifica "delta" e, sendo maior ou igual a zero,
//   se for igual a zero, calcula e mostra o valor de "x1",
//   se for maior que zero, calcula e mostra o valor de "x1" e "x2";
// Se for menor que zero, mostra uma mensagem

    if delta >= 0 then begin
      if delta = 0 then begin
        x1 := -b / (2 * a);
        writeln('x1 = ', x1 );
      end
      else begin
        x1 := (-b + SQRT(delta)) / (2 * a);
        writeln('x1 = ', x1 ) ;

        x2 := (-b - SQRT(delta)) / (2 * a);
        writeln('x2 = ', x2 ) ;
      end;
    end
    else
      writeln('a equação não tem valores reais' ) ;
  end
  else    //sendo a = 0, mostra uma mensagem de erro
    writeln('a não pode ser igual a 0' ) ;

End.
```

Este par "begin-end" também não é necessário, embora às vezes seu uso seja aconselhável (quando ocorre uma sequência longa de comandos). *>> pergunte em aula, por favor!!!*

Observe a redução expressiva no tamanho do programa.

Versão alterada (sugestão), com a indentação no estilo alternativo recomendado pelo professor:

```
Program Bhaskara ;
var a, b, c: integer;
    delta, x1, x2: real;
    x: real;

Begin
// Ler os valores das variáveis, garantindo que o valor de "a"
// seja diferente de zero para calcular "delta" e prosseguir

  writeln('Insira o valor de a: ' ) ;
  readln( a) ;

  if a <> 0 then
    begin
    writeln('Insira o valor de b: ' ) ;
    readln( b) ;

    writeln('Insira o valor de c: ' ) ;
    readln( c) ;

    delta := b*b - 4*a*c;

// Verifica "delta" e, sendo maior ou igual a zero,
//   se for igual a zero, calcula e mostra o valor de "x1",
//   se for maior que zero, calcula e mostra o valor de "x1" e "x2";
// Se for menor que zero, mostra uma mensagem

    if delta >= 0 then
      if delta = 0 then
        begin
        x1 := -b / (2 * a);
        writeln('x1 = ', x1 );
        end
      else
        begin
        x1 := (-b + SQRT(delta)) / (2 * a);
        writeln('x1 = ', x1 ) ;

        x2 := (-b - SQRT(delta)) / (2 * a);
        writeln('x2 = ', x2 ) ;
        end
    else
      writeln('a equação não tem valores reais' ) ;
    end
  else    //sendo a = 0, mostra uma mensagem de erro
    writeln('a não pode ser igual a 0' ) ;
End.
```

Observe que este comando "if" (mais interno) é o comando (um só, embora estruturado) que faz parte da cláusula "then" do "if" externo (if delta >= 0), cuja cláusula "else" também só tem um comando.