```
    1: Script started on Thu Mar 15 13:32:29 2012
    2: bash-3.2$ cat -n guessverify.pl
    3:      1  % $Id: guessverify.pl,v 1.4 2012-03-15 13:31:34-07 - - $
    4:      2
    5:      3  %
    6:      4  % Illustrate guess and verify style.
    7:      5  % We have a set of numbers and we want to find all pairs
    8:      6  % where the first number is greater than the second.
    9:      7  %
   10:      8
   11:      9  a_number( 1.41421356237309504880).
   12:     10  a_number( 2.5).
   13:     11  a_number( 2.7182818284590452354).
   14:     12  a_number( 3.14159265358979323846).
   15:     13  a_number( 6.02e23).
   16:     14  a_number( 8).
   17:     15
   18:     16  guess( X, Y) :- a_number( X), a_number( Y).
   19:     17
   20:     18  verify( X, Y) :- X > Y.
   21:     19
   22:     20  getpair( X, Y) :- guess( X, Y), verify( X, Y).
   23:     21
   24:     22  % TEST: getpair( X, Y).
   25:     23  % TEST: ;
   26:     24  % TEST: ;
   27:     25  % TEST: ;
   28:     26  % TEST: ;
   29:     27  % TEST: ;
   30:     28  % TEST: ;
   31:     29  % TEST: ;
   32:     30  % TEST: ;
   33:     31  % TEST: ;
   34:     32  % TEST: ;
   35:     33  % TEST: ;
   36: bash-3.2$ gprolog
   37: GNU Prolog 1.3.1
   38: By Daniel Diaz
   39: Copyright (C) 1999-2009 Daniel Diaz
   40: | ?- [guessverify].
   41: compiling /afs/cats.ucsc.edu/courses/cmps112-wm/Languages/prolog/Examples/guessv
erify.pl for byte code...
   42: /afs/cats.ucsc.edu/courses/cmps112-wm/Languages/prolog/Examples/guessverify.pl c
ompiled, 33 lines read - 1454 bytes written, 10 ms
   43:
   44: yes
   45: | ?- getpair( A, B).
   46:
   47: A = 2.5
   48: B = 1.4142135623730951 ? ;
   49:
   50: A = 2.7182818284590451
   51: B = 1.4142135623730951 ? ;
   52:
   53: A = 2.7182818284590451
   54: B = 2.5 ? ;
   55:
   56: A = 3.1415926535897931
   57: B = 1.4142135623730951 ? ;
   58:
   59: A = 3.1415926535897931
   60: B = 2.5 ? ;
   61:
   62: A = 3.1415926535897931
```

```
  63: B = 2.7182818284590451 ? ;
  64:
  65: A = 6.02e+23
  66: B = 1.4142135623730951 ? ;
  67:
  68: A = 6.02e+23
  69: B = 2.5 ? ;
  70:
  71: A = 6.02e+23
  72: B = 2.7182818284590451 ? ;
  73:
  74: A = 6.02e+23
  75: B = 3.1415926535897931 ? ;
  76:
  77: A = 6.02e+23
  78: B = 8 ? ;
  79:
  80: A = 8
  81: B = 1.4142135623730951 ? ;
  82:
  83: A = 8
  84: B = 2.5 ? ;
  85:
  86: A = 8
  87: B = 2.7182818284590451 ? ;
  88:
  89: A = 8
  90: B = 3.1415926535897931 ? ;
  91:
  92: no
  93: | ?- ;;
  94: .
  95: uncaught exception: error(syntax_error('user_input:3 (char:17) previous operator
needs brackets'),read_term/3)
  96: | ?-
  97:
  98: bash-3.2$ exit
  99: exit
 100:
 101: Script done on Thu Mar 15 13:33:27 2012
```