```
 1: % $Id: euclid.pl,v 1.2 2015-11-16 12:54:31-08 - - $
 2:
 3: % Euclid's algorithm for greatest common divisor.
 4: % The C version:
 5: % int gcd (int x, int y) {
 6: %    while (x != y) if (x > y) x -= y; else y -= x;
 7: %    return x;
 8: % }
 9:
10: gcd( X, Y, Z ) :- X > Y, T is X - Y, gcd( T, Y, Z ).
11: gcd( X, Y, Z ) :- X < Y, T is Y - X, gcd( X, T, Z ).
12: gcd( X, X, X ).
13:
14: trace( gcd ).
15:
16: % TEST: gcd( 111, 259, Z ).
```

```
    1: bash-1$ gprolog
    2: GNU Prolog 1.4.4 (64 bits)
    3: Compiled Nov  6 2014, 18:04:28 with gcc
    4: By Daniel Diaz
    5: Copyright (C) 1999-2013 Daniel Diaz
    6: | ?- [euclid].
    7: compiling /afs/cats.ucsc.edu/courses/cmps112-wm/Languages/prolog/Example
s/euclid.pl for byte code...
    8: /afs/cats.ucsc.edu/courses/cmps112-wm/Languages/prolog/Examples/euclid.p
l compiled, 16 lines read - 1402 bytes written, 27 ms
    9:
   10: (1 ms) yes
   11: | ?- trace.
   12: The debugger will first creep -- showing everything (trace)
   13:
   14: yes
   15: {trace}
   16: | ?- gcd( 111, 259, G ).
   17:       1    1  Call: gcd(111,259,_23) ?
   18:       2    2  Call: 111>259 ?
   19:       2    2  Fail: 111>259 ?
   20:       2    2  Call: 111<259 ?
   21:       2    2  Exit: 111<259 ?
   22:       3    2  Call: _121 is 259-111 ?
   23:       3    2  Exit: 148 is 259-111 ?
   24:       4    2  Call: gcd(111,148,_23) ?
   25:       5    3  Call: 111>148 ?
   26:       5    3  Fail: 111>148 ?
   27:       5    3  Call: 111<148 ?
   28:       5    3  Exit: 111<148 ?
   29:       6    3  Call: _199 is 148-111 ?
   30:       6    3  Exit: 37 is 148-111 ?
   31:       7    3  Call: gcd(111,37,_23) ?
   32:       8    4  Call: 111>37 ?
   33:       8    4  Exit: 111>37 ?
   34:       9    4  Call: _277 is 111-37 ?
   35:       9    4  Exit: 74 is 111-37 ?
   36:      10    4  Call: gcd(74,37,_23) ?
   37:      11    5  Call: 74>37 ?
   38:      11    5  Exit: 74>37 ?
   39:      12    5  Call: _355 is 74-37 ?
   40:      12    5  Exit: 37 is 74-37 ?
   41:      13    5  Call: gcd(37,37,_23) ?
   42:      14    6  Call: 37>37 ?
   43:      14    6  Fail: 37>37 ?
   44:      14    6  Call: 37<37 ?
   45:      14    6  Fail: 37<37 ?
   46:      13    5  Exit: gcd(37,37,37) ?
   47:      10    4  Exit: gcd(74,37,37) ?
   48:       7    3  Exit: gcd(111,37,37) ?
   49:       4    2  Exit: gcd(111,148,37) ?
   50:       1    1  Exit: gcd(111,259,37) ?
   51:
   52: G = 37 ?
   53:
   54: (1 ms) yes
   55: {trace}
   56: | ?-
```