```
! General Equilibrium Model of an economy,  (GenEqlbx);
! Data based on Kehoe, Math Prog, Study 23(1985), page 253;
! Find clearing prices for commodities/goods and
  equilibrium production levels for processes in
  an economy.
    This model has multiple equilibria. Three of them are:
      PRICE(1)  PRICE(2)   PRICE(3)  PRICE(4)     LEVEL(1)  LEVEL(2)
  1)    1         1        1           1          52          69
  2)  0.637688    1        0.154606   2.207706    42.70128   81.19803
  3)  1.100547    1        1.234609   0.664845    53.18013   65.14815

Ref: Kehoe, T. J.(1985), "A numerical investigation of multiplicity of equilibria,"
   Mathematical Programming Study 23, pp. 240-258.  ;
! Keywords:  Complementarity constraints,  Equilibrium, General equilibrium,
     Kehoe, LINGO, Multiple solutions;
  SETS:
   GOOD: PRICE, H;
   SECTOR;
   GXS( GOOD, SECTOR): ALPHA, W;
   PROCESS: LEVEL, RC;
   GXP( GOOD, PROCESS): MAKE;
  ENDSETS
  DATA:
 ! There are 4 goods, 4 sectors. In general need not be same number;
    GOOD = 1..4;
  SECTOR = 1..4;
  ! Demand curve parameter for each good i & sector j;
   ALPHA =
     .5200  .8600  .5000  .0600
     .4000  .1     .2     .25
     .04    .02    .2975  .0025
     .04    .02    .0025  .6875;
  ! Initial wealth of good i by for sector j;
    W =
     50     0      0      0
      0    50      0      0
      0     0    400      0
      0     0      0    400;
  PROCESS= 1   2;  ! There are two processes to make goods;
  !Amount produced of good i per unit of process j;
    MAKE =
          6   -1
         -1    3
         -4   -1
         -1   -1;
  ! Weights for price normalization constraint;
    H = .25 .25 .25 .25;
  ENDDATA
  !-----------------------;
  ! Variables:
     LEVEL(p) = level or amount at which we operate
                process p.
        RC(p) = reduced cost of process p,
              = cost of inputs to process p - revenues from outputs
                of process p,  per unit.
     PRICE(g) = equilibrium price for good g;

  ! Constraints;
  !  For each good G, we must have supply = demand, i.e.,
     initial amount + production = sum over sectors, S,
     of amount demanded of G at given prices;
   @FOR( GOOD( G):
     @SUM( SECTOR( M): W( G, M))
     + @SUM( PROCESS( P): MAKE( G, P) * LEVEL( P))
     = @SUM( SECTOR( S):
             ALPHA( G, S) *
       @SUM( GOOD( I): PRICE( I) * W( I, S))/ PRICE( G));
       );
```

```
   ! Each process at best breaks even. RC is constrained >= 0;
    @FOR( PROCESS( P):
      RC( P) = @SUM( GOOD( G): - MAKE( G, P) * PRICE( G));
   ! Complementarity constraints.
      If process p has a positive opportunity cost (RC > 0),
       then do not use it (LEVEL = 0).
     Also, if use it ( LEVEL > 0),
      then must not have a positive opportunity cost (RC = 0);
      RC( P)*LEVEL( P) = 0;
        );

   ! Price normalization constraint. Choose prices scale to 1.
     Without a scaling constraint there can be an infinity of solutions;
     @SUM( GOOD( G): H( G) * PRICE( G)) = 1;

   !   You can arbitrarily limit to a particular solution by
     bounding the prices.
       Be sure to turn on the global solver by clicking:
    LINGO -> Options -> Global;
!   @BND( 0, PRICE( 1), .65);
!   @BND( 0.65, PRICE( 1), 1.05);
   @BND( 1.05, PRICE( 1), 9999);
```