```
! A blending model in LINGO with a quadratic approximation
  to the blending of octane in gasoline (Motor Octane Number);
! Note: To complete this example, you should also download BlendOctaneQPdata1.xlsx .
  That spreadsheet contains input data;

! Keywords: @OLE(), Blending, Excel, gasoline, LINGO, MON, Octane, Petroleum, Quadractic,
Refinery;

SETS:
 ! The table of Raw Materials;
  RawM: mon, COST, avail, RawMused, fract;
  rxr( RawM, RawM) : qlt;  ! Pairs of RawM and quality effect;
ENDSETS

SUBMODEL BlendGas:
! Variables:
    batch = quantity blended in total,
    RawMused(j) = amount of RawM(j) in blend,
    fract(j) = fraction of RawM(j) in blend;

! Maximize Revenue minus cost;
  max = SPrice*batch - @sum( RawM( j): COST( j)* RawMused( j));

! Amount blended;
     batch = @sum( RawM(j): RawMused( j));

   @for( RawM( j):
!    Blend fractions. This is nonlinear;
     fract(j) = RawMused(j)/batch;
!    Raw material availabilities;
      RawMused(j) <= AVAIL( j);
       );

! The MON constraint. This is nonlinear;
    @sum( RawM(j): MON(j)*fract(j)) + @sum( rxr(i,j): qlt(i,j)*fract(i)*fract(j)) >=
MONtarg;
ENDSUBMODEL
CALC:
! Get data for today's problem;
!  Get Motor Octane Number (MON) target, from range name MONTARG ;
   MONtarg = @OLE( "C:\Temp\BlendOctaneQPdata1.xlsx");
!  Get current selling price, from range name SPRICE ;
   SPrice  = @OLE( "C:\Temp\BlendOctaneQPdata1.xlsx");
!  Get Raw Material names ;
   RawM    = @OLE( "C:\Temp\BlendOctaneQPdata1.xlsx");
!  Get Raw Material costs ;
   COST    = @OLE( "C:\Temp\BlendOctaneQPdata1.xlsx");
!  Get Raw Material availabilities ;
   AVAIL   = @OLE( "C:\Temp\BlendOctaneQPdata1.xlsx");
 ! Stand-alone MON (Motor Octane Number);
   MON     = @OLE( "C:\Temp\BlendOctaneQPdata1.xlsx");


!  Get RawM interaction pairs;
   RxR     = @OLE( "C:\Temp\BlendOctaneQPdata1.xlsx");
!  Get quadratic interactions for pairs;
   QLT     = @OLE( "C:\Temp\BlendOctaneQPdata1.xlsx");

! You can set various parameter with @SET;
  @SET( 'TERSEO',1); ! Output level (0:verbose, 1:terse, 2:only errors, 3:none);
  @SET( 'IPTOLR', 0.0001);! Set IP ending relative optimality tolerance(Should be >0);
!  @SET( 'GLOBAL', 1); ! 0:Do not use Global solver, 1:Use the Globasolver;
!  @SET( 'xsolvr', 14,'c:_20.dll');! Specify an external solver, Highs developmental
solver;

  @SOLVE( BlendGas); ! Solve a specific subproblem;

 ! Send results back to a spreadsheet;
```

```
  @OLE( "C:\Temp\BlendOctaneQPdata1.xlsx")= RawMused;
ENDCALC
```