```
$oninline
/*
 GAMS program used to estimate input oriented technical efficiency.

 Author:  John B. Walden
          NMFS/NEFSC
          166 Water St.
          Woods Hole, MA  02543
          (508) 495-2355
          John.Walden@Noaa.Gov

This version will estimate technical efficiency for
each DMU, based on an input oriented variable returns
to scale DEA model, with global weak disposability of inputs.

References:Fare, R, S. Grosskopf and C.A. Knox Lovell. 1994.
Production Frontiers. Cambridge university Press.

Olesen, O.B. and N.C. Petersen. 1996. A Presentation
of GAMS for DEA. Computers and Operations Research,
Vol. 23, N0. 4, pp 323-339.

*/

/* The following line turns off listing of
some elements in the GAMS listing file*/

$OFFSYMLIST OFFSYMXREF OFFUELLIST OFFUELXREF

OPTION SYSOUT=OFF, LIMROW=0, LIMCOL=0;

/*NEXT DEFINE INPUTS AND OUTPUTS*/


SET INOUT /y1*y3, x1*x5/
   OUTPUT(INOUT) /y1*y3/
   INPUT(INOUT) /x1*x5/


   OBS /1*200/
   SUBOBS(OBS) /1*50/
   ACTOBS(OBS)

/*We have allocated enough memory for 200 observations, but our
data set only contains 50 observations (subobs)*/
;

/*Next, define an alias for the set SUBOBS */

alias (subobs, subobs1)

/* The include statement below reads in an external data
file which contains a table of observations, inputs
and outputs. The files is in comma separted value
format (CSV) which can be written from, and read
into spreadsheet programs
```

```
The offlisting command means that the data won't
be included in the listing file.

*/

$OFFLISTING

TABLE ACT(OBS,INOUT)  INPUT OUTPUT TABLE
$ondelim
$INCLUDE "data_test.csv"
$offdelim

$ONLISTING

VARIABLES

 lambda      efficiency score
 weight(obs)  weights;

 POSITIVE Variable weight;

 EQUATIONS
   CONSTR1(OUTPUT, OBS)  DEA constraint for each output
   CONSTR2(INPUT, OBS)  DEA constraint for each input
   CONSTR3  DEA constraint for imposing VRS;

   CONSTR1(OUTPUT, ACTOBS).. SUM(SUBOBS, WEIGHT(SUBOBS)*ACT(SUBOBS,OUTPUT)) =G=
ACT(ACTOBS,OUTPUT);

   CONSTR2(INPUT, ACTOBS).. SUM(SUBOBS, WEIGHT(SUBOBS)*ACT(SUBOBS,INPUT)) =E=
LAMBDA*ACT(ACTOBS,INPUT);

   CONSTR3.. SUM(SUBOBS, WEIGHT(SUBOBS)) =E= 1;

/*Define a parameter to hold results for each pass through
the loop*/


 PARAMETER

   score1(obs)  efficiency scores
;

/*Define an external file to hold results*/

file primal /teinp_res.txt/;

/*The model defined below consists of three equations.
CONSTR1, CONSTR2, CONSTR3. We use the term ALL in the model statement
rather than listing the equations separately as we did in the
CRS example */

 MODEL TEDEA /ALL/;

tedea.solprint=2;         /*Turn off writing results to solution file*/
tedea.solvelink=2;      /*Keep model in memory. Improves solution time*/

 LOOP(SUBOBS1,

    Option solslack=1;
    ACTOBS(OBS) = NO;
    ACTOBS(SUBOBS1) = YES;
    SOLVE TEDEA minimizing Lambda USING LP;


    score1(SUBOBS1) = lambda.l;

/*the put statement below writes out the solver status and model
status to an external file. If the model for an individual observation
does not solve, examination of the file will allow one to determine
which observation did not solve*/

    put primal;

     if ((tedea.modelstat eq 1 and tedea.solvestat eq 1),

         put @1, subobs1.tl, @10, "optimal", @25, "normal completion" /

     else

        put @1, subobs1.tl, @10, tedea.modelstat:>2:0,
               @25, tedea.solvestat:>2:0/
      );
    );


/*The next file is to output results to a .csv file.
Results could also be printed to the listing file with
the use of the display command*/

file res /te_inp_vrs_wd.csv/;

res.pc=5;
res.pw=100;


put res;

put 'obs', 'score' /;
loop (subobs1,

   put subobs1.tl, score1(subobs1)/
   );

putclose;
```