HEWLETT-PACKARD

Linear Programming Pac

Series 80

HEWLETT PACKARD

Series 80

Linear Programming Pac

August 1982

Reorder Number 00085-90623

Printed in U.S.A.

Introduction

The Linear Programming (LP) Pac provides you with the ability to solve a wide variety of optimization problems such as chemical blending, feed mix, production scheduling, investment portfolio selection and marketing media selection.

The pac contains two sets of programs for solving LP problems on the HP-85 with either 16K or 32K bytes of user memory. Each set contains seven programs which chain together automatically, plus a binary program which greatly reduces the time required for convergence to a solution. The manual includes a program description, User Instructions and two example problems.

The pac is primarily designed for the person who has already gained some understanding of the process of formulating an LP problem and interpreting its solution. If you are unacquainted with the LP technique, we recommend that you familiarize yourself with the subject before attempting to use this pac. A particularly clear presentation is contained in *An Introduction to Quantitative Methods for Decision Making* by Richard E. Trueman, published by Holt, Rinehart and Winston, New York, New York, 1977.

The solution procedure begins with the entry of a problem from either the keyboard or tape cartridge. After entry, the problem may be modified, printed or stored before being solved. After solution, the answer is printed and a sensitivity analysis may be performed.

The technique used to optimize the objective function is a modification of the simplex method which incorporates variable bounds. Lower bounds on variables result in tableau modifications before optimizing begins.[1] Upper bounds are incorporated in the algorithm.[2] As a result, upper and lower bounds on variables do not need to be formulated as constraints, which conserves problem space.

You do not need a knowledge of the BASIC programming language to use the Linear Programming Pac; however, you should be familiar with sections 1 through 5 of the Owner's Manual. If you are a programmer, you may want to make use of the pac's Remarks program, which contains program listing comments and variable definitions to aid you in following program flow. Appendix A contains User Instructions for the Remarks program.

---

[1] Llewellyn, Robert W., *Linear Programming,* Holt, Rinehart and Winston, New York, New York, 1964, pp. 133-134.

[2] Luenberger, D. G., *Introduction to Linear and Non-Linear Programming,* Addison-Wesley Publishing Company, Reading, Mass., 1973, pp. 48-53.

Contents

Linear Programming

Program Description

Enter

The program begins with the entry of a problem from the keyboard or tape cartridge. You must observe the maximum problem dimensions (see the *Problem Dimensions* section) and the following guidelines:

1. Names for problems, variables and constraints may be up to 6 alphanumeric characters in length.
2. Constraints must be entered in this order: < =, =, > =.
3. Constraint right-hand side values should be greater than or equal to zero.
4. Data input should not overflow the data output display format: DDDDD.DD.
5. Special function keys may be used when you are prompted. They provide program control by allowing you to chain to program segments or access subroutines. They are not meant to be used as direct interrupts anywhere in the program.
6. If you are entering a problem previously stored on a tape cartridge other than the LP Pac cartridge, follow the User Instructions to insure that the LP Pac cartridge is in the tape transport except when you are loading or storing data.

Modify

The Modify program segment will allow you to: a) correct any incorrect entries made while entering a problem, and b) create a new problem, with a new name, by modifying an old problem.

You may add a maximum of two (2) constraints more than the number in the problem when it was originally entered. You may delete as many constraints as desired as long as at least one (1) constraint remains in the problem. For every constraint deleted, one more may be added. The list of constraints will be displayed before and after each addition or deletion. Modifying can only be done before solving the problem.

Print

This program segment will give you a record of the problem structure. Printing can only be done before solving the problem.

Store

Problems are stored on tape by user-specified name, and all reference to stored problems is by name. The problem will be stored using the current name, unless there is already a problem by that name stored on the tape cartridge. In that case you may use the current name or create a new name. If you use the current name, then the original data stored under that name will be erased.

You may store problems either on the LP Pac cartridge or on a data storage cartridge. If you do the latter, follow the User Instructions to insure that the LP Pac cartridge is in the tape transport except when you are loading or storing data. The data storage cartridge must be initialized before use with the ERASETAPE command.

Storing can only be done before solving the problem.

Solve

Before optimization begins, the tableau is completed by adding the necessary surplus, slack and artificial variables. You then have the option of printing the initial tableau after which the optimization begins.

If the solution is unbounded, or if there is no feasible solution, an appropriate message is printed.

Answer

After optimization is complete, the answer is printed, including the basis, dual variables and the value of the objective function. You then have the option of printing the final tableau.

The dual variable value, or ''shadow price,'' is the amount of change in the value of the objective function for each unit by which the constraint right-hand side (RHS) value is changed.

Sensitivity Analysis

Through this analysis you may obtain more information about the constraints and variables in and out of the solution (basis). The analysis includes constraint right-hand side (RHS) value ranging, basis variable coefficient ranging and non-basis variable coefficient ranging.

The sensitivity analysis calculates the upper and lower limits within which you may change any one constraint RHS value or objective function coefficient without changing the basis. Basis variable values may change and the objective function value may change, but the basis will remain the same.

The non-basis variable coefficient ranging determines the amount of change necessary for a non-basis variable to enter the basis.

Remarks Program

The LP Pac cartridge also contains a Remarks program which will help you if you are examining the program listings. Refer to Appendix A.

Problem Dimensions

Maximum problem size is dependent on available read-write memory (random-access memory or RAM). Two different versions of the LP Pac are contained on the tape cartridge and disc. The first requires 16K bytes of RAM and the second requires 32K bytes. There is a distinction that should be noted between the problem dimensions available with the tape and disc versions of the Linear Programming Pac. The following table gives the problem size restrictions for each version:

		Tape		Disc	
		16K	32K	16K	32K
Tableau Matrix	A (,)	(18,47)	(30,85)	(18,40)	(30,79)
Variables	N	1 to 40	1 to 78	1 to 33	1 to 72
Constraints	M	1 to 13	1 to 25	1 to 13	1 to 25
>= Constraints	G	0 to 13	0 to 25	0 to 13	0 to 25
	N+M+G	2 to 41	2 to 79	2 to 34	2 to 73

For each constraint (M) and each >= constraint (G), the maximum number of variables (N) is reduced by one.
Some extreme examples are:

	Tape				Disc			
Memory	N	M	G	N+M+G	N	M	G	N+M+G
16K	40	1	0	41	33	1	0	34
16K	15	13	13	41	8	13	13	34
32K	78	1	0	79	72	1	0	73
32K	29	25	25	79	23	25	25	73

Note: Problems previously stored on tape or disc using the 16K version of the LP Pac can only be accessed with the 16K version. The same restriction also applies to the 32K version.

Glossary of Linear Programming Terms

Artificial Variable

A positive variable added to the left-hand side of an equality or a >= (greater than or equal to) inequality in order to generate an initial basic feasible solution.

Basic Feasible Solution

A solution which satisfies all constraints and which lies at an extreme point (vertex) of the solution space as defined by the constraint set. The initial basic feasible solution has a value of zero for all problem variables and for the objective function.

Basis

The set of variables with positive values (basis variables), forming a basic feasible solution. A problem with N variables and M constraint equations will have at most M variables in the basis.

Dual Variable Value ("shadow price")

The amount of change in the objective function value resulting from a one unit change in the RHS value of the associated constraint.

Feasible Solution

Any solution of the problem which satisfies all of the constraints.

Infeasible Solution

A solution in which one or more variables has a negative value. This indicates that one or more constraints or variable bounds cannot be satisfied.

Iteration

Stepwise progress toward an optimal solution by generating a new basic feasible solution.

Non-Basis Variable

A variable which has a zero value and which is not included in the basis. A variable which is at its upper or lower bound is included in this category.

Objective Function

The function to be optimized (maximized or minimized).

Optimality

A condition in which no further improvement in the value of the objective function is possible. This is called the optimal solution.

Problem Variable

A variable appearing in the objective function.

Sensitivity Analysis

A post-optimality analysis to determine the upper and lower ranging values on constraint RHS values and objective function coefficients. This analysis determines the range of validity of dual variable values.

Slack Variable

A positive variable added to the left-hand side of a <= (less than or equal to) inequality to create an equality.

Surplus Variable

A positive variable subtracted from the left-hand side of a >= (greater than or equal to) inequality to create an equality.

Tableau

An array of variable coefficients and constraint RHS values created by removing the variable symbols from the set of linear equations in the LP model.

Unbounded Solution

A condition in which the value of the objective function can be made arbitrarily large.

Format of User Instructions

The User Instructions are your guide to operating the programs in this pac.

Certain key words have been used to indicate specific types of operations. You should become familiar with the meanings of these words so that the intent of the User Instructions can then easily be followed.

Key Word	Meaning/Use
INSERT	Put the tape cartridge into the tape transport
PRESS	Push an immediate execute key, e.g., [END LINE] or [RUN]
TYPE	Push a series of keys which form a command, e.g., Type: [REW LOAD] "LP"
ENTER	Push a series of keys as a response to a machine prompt, e.g., Enter: The name of the LP problem [END LINE].
GO TO Step n	Change the flow in the User Instructions.
REPEAT	Designates a repeatable group of instructions
NOTE:	Extra comments concerning instructions for this step

The User Instructions are written in outline form so that you can easily follow the instructions and the flow of operation.

Whenever a special function key is labeled HELP, the program includes a "HELP" section which displays a short description of the function of each special function key. Whenever GUIDE or DISPLAY GUIDELINES? is displayed, summary information from the program description and User Instructions is presented to aid you in solving your LP problems. After solving a few LP problems using the written User Instructions, you should be able to solve problems rapidly, referring only to the program "HELP" and "GUIDELINES" to refresh your memory.

The program flow will often ask for a "YES" or "NO" answer to a question. A "YES" answer requires that you enter Y or YES before pressing [END LINE]. However, you may answer "NO" simply by pressing [END LINE]. Entering N or NO beforehand is optional.

Program Operation Hints

These programs have been designed to execute with a minimum amount of difficulty, but problems may occur which you can easily solve during program operation. There are four different types of errors or warnings that can occur while executing a program: input errors, math errors, tape errors and image format string errors.

The input errors include errors 43, 44, and 45. All of these errors will cause a message to be output followed by a new question mark as a prompt for the input. You should verify your mistake and then enter the corrected input. The programs will not proceed until the input is acceptable. There is a more complete discussion of INPUT in your Owner's Manual.

The second type of error that might occur is a math error (errors 1 thru 13). With DEFAULT ON, the first eight errors listed in Appendix E of your Owner's Manual cause a warning message to be output, but program execution will not be halted. The cause of these errors can usually be attributed to specific characteristics of your data and the type of calculations being performed. In most cases, there is no cause for alarm, but you should direct your attention to a possible problem. An example of such a case is found in the Standard Pac when the curve fitting program computes a curve fit to your data which has a value of 1 for the coefficient of determination, r^2. The computation of the F ratio results in a divide by zero, Warning 8.

The third type of error, tape errors (60 thru 75), may be due to several different problems. Some of the most likely causes are the tape being write-protected, the wrong cartridge (or no cartridge) being inserted, a bad tape cartridge, or wrong data file name specification during program execution. Appendix E of your Owner's Manual should be consulted for a complete listing.

The fourth type of error is due to generalizing the output to anticipated data ranges. In many cases, the output has assumed ranges which may or may not be appropriate with your data. Adjusting the image format string for your data will solve this type of problem. You may also want to change the image string if you require more digits to the right of the decimal point.

Whenever a running program is interrupted from the keyboard by inadvertantly pressing a key, the system beeps. To continue program execution, press (CONT).

Most programs assume that the printer is 2 and the CRT is 1 and use PRINT and DISP statements accordingly. If you want to ensure that the peripherals are defined as the programs assume, press (RESET) before running a program. The currently defined key labels are obtainable at any time while a program is running by pressing (KEY LABEL). Remember to press (CLEAR -LINE) before pressing (END LINE) if the key labels are in the input line. All files on the tape cartridge have been secured using a security code of HP and a security type of 2. To store a changed version of a program, you must first unsecure the file using HP as the security code and 2 as the security type.

These are the more common problems which may occur during program operation. Your Owner's Manual should be consulted if you need more assistance.

User Instructions

1. Insert the LP Pac tape cartridge into the tape transport.
2. To load the program:
   a. Type: (REW LOAD) "LP" (END LINE) to load the 16K RAM version.

   OR:

   a. Type: (REW LOAD) "LP32" (END LINE) to load the 32K RAM version.

   Note: The 32K RAM version will not run on an HP-85 with only 16K RAM.
3. To start the program:
   a. Press: (RUN)
4. When the keys are labeled:

   ```
   HELP      GUIDE
   ENTER
   ```

   a. Press: KEY #1 (ENTER) to enter a problem.
   b. Go to step 5.

   OR:

   a. Press: KEY #5 (HELP) to display key functions.
   b. Go to step 4.

   OR:

   a. Press: KEY #6 (GUIDE) to display guidelines.
   b. Go to step 4.
5. When NAME OF PROBLEM(6 CHAR. MAX.)? is displayed:
   a. Enter: The problem name (END LINE), 6 alphanumeric characters max.
6. When ENTER PROBLEM FROM KEYBOARD OR TAPE(K/T)? is displayed:
   a. Enter: K (END LINE) if problem will be entered from the keyboard.
   b. Go to step 9.

   OR:

   a. Enter: T (END LINE) if problem will be entered from a tape cartridge.
   b. Go to step 7.

   Note: If problem data is stored on another tape cartridge, insert the cartridge before you press (END LINE).
7. When PROBLEM (name) ENTERED is displayed:
   a. Go to step 23.
8. If (name) NOT FOUND ON TAPE SELECT ANOTHER PROBLEM NAME/TAPE NAME OF PROBLEM(6 CHAR. MAX.)? is displayed:
   a. Enter: The new problem name (END LINE), 6 alphanumeric characters max.
   b. Go to step 6.

   OR:

   a. Insert: The correct tape cartridge in the tape transport.
   b. Enter: The same problem name (END LINE), 6 alphanumeric characters max.
   c. Go to step 6.
9. When MAXIMIZE OR MINIMIZE (MAX/MIN)? is displayed:
   a. Enter: MAX (END LINE) if entering a maximize problem.

   OR:

   a. Enter: MIN (END LINE) if entering a minimize problem.
10. When # OF VARIABLES? is displayed:
    a. Enter: The number of variables (END LINE).

    Note: Refer to the *Problem Dimensions* section.
11. When # OF CONSTRAINTS? is displayed:
    a. Enter: The number of constraints (END LINE).

12. When # OF <= CONSTRAINTS? is displayed:
    a. Enter: The number of "less than or equal to" constraints [END LINE].
13. When # OF = CONSTRAINTS? is displayed:
    a. Enter: The number of "equal to" constraints [END LINE].
14. When # OF >= CONSTRAINTS? is displayed:
    a. Enter: The number of "greater than or equal to" constraints [END LINE].
15. If INCONSISTENT DATA is displayed:
    a. One or more of the following happened:
        1. An entry in step 10 or 11 is zero or negative.
        2. An entry in step 12, 13 or 14 is negative.
        3. The sum of entries in steps 12, 13 and 14 does not equal the entry in step 11.
        4. Problem dimensions are beyond the limits specified.
    b. Go to step 10.
16. When NAME FOR VARIABLE # (1 to N) (6 CHAR. MAX.)? is displayed:
    a. Enter: The variable name [END LINE], 6 alphanumeric characters max.
    b. Repeat step 16 for each variable.
17. When NAME FOR CONSTRAINT # (1 to M) (6 CHAR. MAX.)? is displayed:
    a. Enter: The constraint name [END LINE], 6 alphanumeric characters max.
    b. Repeat step 17 for each constraint.

    Constraints are entered in this order: <=, =, >=.

18. When COEFFICIENT FOR (constraint name), (variable name)? is displayed:
    a. Enter: The coefficient [END LINE] for that constraint and variable combination.
    b. Repeat step 18 for each variable.

    Note: Data should be entered to conform with the program output display format DDDDD.DD.

19. When RHS VALUE OF CONSTRAINT (constraint name)? is displayed:
    a. Enter: The right-hand side (RHS) value of that constraint [END LINE].
    b. Repeat steps 18 and 19 for each constraint.

    Note: RHS values should be >= zero.

20. When OBJ FUNC COEFF FOR (variable name)? is displayed:
    a. Enter: The objective function coefficient for that variable [END LINE].
    b. Repeat step 20 for each variable.
21. When UPPER BOUND ON (variable name) (-1=UNBND)? is displayed:
    a. Enter: The upper bound value for that variable [END LINE].
    b. Repeat step 21 for each variable.

    OR:

    a. Enter: -1 [END LINE] if the variable is unbounded.
    b. Repeat step 21 for each variable.
22. When LOWER BOUND ON (variable name)? is displayed:
    a. Enter: The lower bound for that variable [END LINE].
    b. Repeat step 22 for each variable.

    OR:

    a. Enter: 0 [END LINE] if the lower bound for that variable is 0.
    b. Repeat step 22 for each variable.
23. When the keys are labeled:

```
-----------------------------------
HELP     GUIDE    STORE
ENTER    MODIFY   PRINT    SOLVE
```

    a. Press: KEY # 1 (ENTER), to enter another problem.
    b. Go to step 5.

    OR:

a. Press: KEY #2 (MODIFY) to modify this problem.
b. Go to step 24.

OR:

a. Press: KEY #3 (PRINT), to print this problem.
b. Go to step 23.

OR:

a. Press: KEY #4 (SOLVE), to solve this problem.
b. Go to step 61.

OR:

a. Press: KEY #5 (HELP), to display key functions.
b. Go to step 23.

OR:

a. Press: KEY #6 (GUIDE), to display guidelines for the appropriate program segment.
b. Go to step 23.

OR:

a. Press: KEY #7 (STORE), to store this problem on tape cartridge.
b. Go to step 56.

Note: The LP Pac tape cartridge should be in the tape transport before you press any special function keys.

24. When DISPLAY GUIDELINES? is displayed:
    a. Enter: Y (END LINE) to display the guidelines for problem modification.

    OR:

    a. Enter: N (END LINE) if guidelines are not needed.
25. When RENAME (problem name)? is displayed:
    a. Enter: Y (END LINE) to rename the current problem.
    b. Go to step 26.

    OR:

    a. Enter: N (END LINE) to continue.
    b. Go to step 27.
26. When NEW NAME? is displayed:
    a. Enter: The new problem name (END LINE), 6 alphanumeric characters max.
27. When (0 to M + 1) CONSTRAINT(S) MAY BE ADDED is displayed:
    a. If one or more constraints may be added, go to step 28.

    OR:

    a. If 0 constraints may be added, go to step 33.
28. When ADD A CONSTRAINT? is displayed:
    a. Enter: Y (END LINE) to add a constraint.
    b. Go to step 29.

    OR:

    a. Enter: N (END LINE) to continue.
    b. Go to step 33.
29. When CONSTRAINT TYPE? ENTER <, =, > is displayed:
    a. Enter: < (END LINE) for a "less than or equal to" constraint.

    OR:

    a. Enter: = (END LINE) for an "equal to" constraint.

    OR:

    a. Enter: > (END LINE) for a "greater than or equal to" constraint.
30. When NAME OF NEW CONSTRAINT? is displayed:
    a. Enter: The constraint name (END LINE), 6 alphanumeric characters max.
31. When COEFF FOR (variable name)? is displayed:
    a. Enter: The coefficient for that variable (END LINE).
    b. Repeat step 31 for each variable.
32. When RHS VALUE OF NEW CONSTRAINT? is displayed:
    a. Enter: The constraint RHS value (END LINE).
    b. Go to step 27.
33. When DELETE A CONSTRAINT? is displayed:
    a. Enter: Y (END LINE) to delete a constraint.
    b. Go to step 35.

OR:

a. Enter: N (END LINE) to continue.

b. Go to step 36.

34. If NO CONSTRAINTS MAY BE DELETED is displayed:

a. Only one constraint remains.

b. Go to step 36.

35. When CONSTRAINT #? is displayed:

a. Enter: The number of the constraint to be deleted (END LINE).

b. Go to step 33.

36. When CHANGE CONSTRAINT COEFF? is displayed:

a. Enter: Y (END LINE) to change a constraint coefficient.

b. Go to step 37.

OR:

a. Enter: N (END LINE) to continue.

b. Go to step 40.

37. When CONSTRAINT #, VARIABLE #? is displayed:

a. Enter: The constraint and variable numbers of the coefficient to be changed (END LINE). (Example: For constraint #3 and variable #2, enter 3,2 (END LINE)).

38. When OLD COEFF FOR (constraint name, variable name) = (old coefficient) ; NEW COEFF = ? is displayed:

a. Enter: The new coefficient (END LINE).

39. When MORE COEFF TO CHANGE? is displayed:

a. Enter: Y (END LINE) to change more coefficients.

b. Go to step 37.

OR:

a. Enter: N (END LINE) to continue.

40. When CHANGE CONSTRAINT RHS VALUES? is displayed:

a. Enter: Y (END LINE) to change constraint RHS values.

b. Go to step 41.

OR:

a. Enter: N (END LINE) to continue.

b. Go to step 44.

41. When CONSTRAINT #? is displayed:

a. Enter: The constraint number (END LINE).

42. When OLD RHS VALUE, CONSTRAINT (constraint name) = (old value) ; NEW VALUE = ? is displayed:

a. Enter: The new constraint RHS value (END LINE).

43. When MORE CONSTRAINT RHS VALUES TO CHANGE? is displayed:

a. Enter: Y (END LINE) to change more values.

b. Go to step 41.

OR:

a. Enter: N (END LINE) to continue.

44. When CHANGE OBJ FUNC COEFF? is displayed:

a. Enter: Y (END LINE) to change the objective function coefficients.

b. Go to step 45.

OR:

a. Enter: N (END LINE) to continue.

b. Go to step 48.

45. When VARIABLE #? is displayed:

a. Enter: The variable number (END LINE).

46. When OLD OBJ FUNC COEFF FOR (variable name) = (old coefficient value) ; NEW = ? is displayed:

a. Enter: The new coefficient (END LINE).

47. When MORE OBJ FUNC COEFF TO CHANGE? is displayed:

a. Enter: Y (END LINE) to change more coefficients.

b. Go to step 45.

OR:

a. Enter: N (END LINE) to continue.

48. When CHANGE UPPER BOUNDS? is displayed:

a. Enter: Y (END LINE) to change upper bounds.

b. Go to step 49.

OR:

a. Enter: N (END LINE) to continue.

b. Go to step 52.

49. When VARIABLE #? is displayed:
   a. Enter: The variable number (END LINE).
50. When OLD UPPER BOUND ON (variable name) = (old value) ;(-1=UNBND); NEW = ? is displayed:
   a. Enter: The new upper bound value (END LINE).
   OR:
   a. Enter: -1 (END LINE) if variable is unbounded.
51. When MORE UPPER BOUNDS TO CHANGE? is displayed:
   a. Enter: Y (END LINE) to change more upper bounds.
   b. Go to step 49.
   OR:
   a. Enter: N (END LINE) to continue.
52. When CHANGE LOWER BOUNDS? is displayed:
   a. Enter: Y (END LINE) to change lower bounds.
   b. Go to step 53.
   OR:
   a. Enter: N (END LINE) to continue.
   b. Go to step 23.
53. When VARIABLE #? is displayed:
   a. Enter: The variable name (END LINE).
54. When OLD LOWER BOUND ON (variable name) = (old value) ; NEW = ? is displayed:
   a. Enter: The new lower bound value (END LINE).
   OR:
   a. Enter: 0 (END LINE) if there is no lower bound.
55. When MORE LOWER BOUNDS TO CHANGE? is displayed:
   a. Enter: Y (END LINE) to change more lower bounds.
   b. Go to step 53.
   OR:
   a. Enter: N (END LINE) to continue.
   b. Go to step 23.
56. To store the problem on a second cartridge:
   a. Remove the LP Pac cartridge.
   b. Insert a second cartridge.

Note: If not done previously, you may initialize the second cartridge with the ERASE TAPE command while the HP-85 is in PAUSE status.

57. Press (CONT) to continue.
58. When (problem name) STORED ON TAPE is displayed:
   a. Reinsert the LP Pac cartridge, if necessary, before pressing any special function keys.
   b. Go to step 23.
59. If (problem name) ALREADY EXISTS ON TAPE
   USE (problem name) AND ERASE ORIGINAL PROBLEM? is displayed:
   a. Enter: Y (END LINE) to use the current problem name and erase the problem on tape which uses the same name.
   b. Go to step 58.
   OR:
   a. Enter: N (END LINE) to use another name for the current problem.
60. When NEW NAME FOR PROBLEM? is displayed:
   a. Enter: The new problem name (END LINE), 6 alphanumeric characters max.
   b. Go to step 58.
61. The problem name will be printed. Then the number of variables of each type (problem, surplus, slack and artificial) will be printed.
62. When PRINT INITIAL TABLEAU? is displayed:
   a. Enter: Y (END LINE) to have the tableau printed before optimization.
   OR:
   a. Enter: N (END LINE) to continue.
63. If SOLUTION UNBOUNDED and PRINT FINAL TABLEAU? are displayed:
   a. Enter: Y (END LINE) to have final tableau printed.
   b. Go to step 70.

OR:

a. Enter: N (END LINE) to continue.

b. Go to step 70.

64. If NO FEASIBLE SOLUTION is displayed:

a. One or more constraints or variable bounds cannot be met. Go to step 70.

65. The answer will now be printed.

66. When PRINT FINAL TABLEAU? is displayed:

a. Enter: Y (END LINE) to have the final tableau printed.

OR:

a. Enter: N (END LINE) to continue.

67. When DISPLAY ANSWER GUIDELINES? is displayed:

a. Enter: Y (END LINE) to display answer guidelines.

OR:

a. Enter: N (END LINE) to continue.

68. When SENSITIVITY ANALYSIS DESIRED? is displayed:

a. Enter: Y (END LINE) to have the sensitivity analysis performed.

b. Go to step 69.

OR:

a. Enter: N (END LINE) to continue.

b. Go to step 70.

69. When DISPLAY GUIDELINES? is displayed:

a. Enter: Y (END LINE) to display the sensitivity analysis guidelines.

OR:

a. Enter: N (END LINE) to continue.

70. When ENTER A NEW PROBLEM? is displayed:

a. Enter: Y (END LINE) to enter a new problem.

b. Go to step 4.

OR:

a. Enter: N (END LINE) to end.

First Example

Statement of the Problem:

Our first example is a profit maximization problem. XYZ Machine and Foundry Company supplies automobile engine parts to the replacement parts market. They want to develop a production plan for the next month which will maximize the contribution to fixed costs and profit.

XYZ manufactures five products: cast aluminum pistons and cast aluminum intake manifolds, forged aluminum pistons, forged steel valves and forged steel connecting rods.

The company has a casting foundry, a forge and a machine shop. The machine shop has four types of machines: lathe, grinder, milling machine and borer or drill. All operations to manufacture the five products are performed at XYZ, so the company's manufacturing capacity is controlled by its production rates for the different operations. Table I shows the number of units which each operation can complete per day, if exclusively assigned to that product. A blank indicates that the product does not require that operation.

Table I

Daily Production Rates
(if operation exclusively assigned)

	PRODUCT				
OPERATION	**Cast Alum. Pistons**	**Forged Alum. Pistons**	**Cast Intake Manifolds**	**Forged Valves**	**Forged Conn. Rods**
Casting Foundry	500	—	150	—	—
Forge	—	225	—	600	375
Lathe	225	225	—	450	—
Grinder	480	480	—	480	160
Milling	—	—	200	—	133
Boring & Drilling	—	—	160	—	480

To convert daily to monthly rates, XYZ will use a 20-day month. Due to market demand, XYZ can only sell limited quantities of some of the products, and must make at least a minimum number of other products. The maximum and minimum production volumes for the next month, together with the unit dollar contribution to fixed costs and profit for each product, are presented in Table II.

Table II

Production Requirements Per Month
And Profit Contribution Per Unit, $

	Cast Alum. Pistons	**Forged Alum. Pistons**	**Cast Intake Manifolds**	**Forged Valves**	**Forged Conn. Rods**
Maximum Prod.	NONE	2000	3000	NONE	NONE
Minimum Prod.	0	1000	0	4000	1000
Profit Contr., $	3	4	12	2.5	5

The problem for XYZ is to determine (a) the optimal production plan to maximize contribution to fixed costs and profit, and (b) the profit contribution this optimal production plan will realize.

Setting up the Problem:

The first step is to convert Table I to a set of constraint equations. The casting foundry constraint can be used as an example. The foundry can cast 500 pistons *or* 150 manifolds per day. Each manifold produced uses 500/150 or 3.33 times as much available capacity as each piston produced. Restated, the constraint is:

$$1 \text{ (cast pistons)} + 3.33 \text{ (intake manifolds)} <= 500/\text{day or } 10000/\text{month}$$

The other five constraint equations are set up in a similar manner. A copy of the data input form, located in the Appendix, is used to present the complete problem prepared for entry into your HP-85. Follow the User Instructions to enter the problem from the keyboard or tape cartridge (stored as 16K data file MCHINE).

Problem Name: MCHINE

Maximize or Minimize: MAX

No. Variables: 5

No. Constraints: 6

No. <= 6 No. = 0 No. >= 0

Variable Name / Constraint Name	CSTPST	FRGPST	MANFLD	VALVE	CONROD	Constraint Type (<=, =, >=)	Constraint RHS Value
CASTNG	1	0	3.33	0	0	<=	10000
FORGE	0	2.67	0	1	1.6	<=	12000
LATHE	2	2	0	1	0	<=	9000
GRINDR	1	1	0	1	3	<=	9600
MILLNG	0	0	2	0	3	<=	8000
BORING	0	0	3	0	1	<=	9600
Objective Function Coeff	3	4	12	2.5	5		
Upper Bound	-1	2000	3000	-1	-1		
Lower Bound	0	1000	0	4000	1000		

NOTE: Constraints should be entered in the following order <=, =, >=. Constraint RHS values should be >= 0.

After entering the problem you have the option to print it out.

Printout of the Problem

```
              MCHINE
VARIABLE    #    1 = CSTPST
VARIABLE    #    2 = FRGPST
VARIABLE    #    3 = MANFLD
VARIABLE    #    4 = VALVE
VARIABLE    #    5 = CONROD

CONSTRAINT #    1 = CASTNG
CONSTRAINT #    2 = FORGE
CONSTRAINT #    3 = LATHE
CONSTRAINT #    4 = GRINDR
CONSTRAINT #    5 = MILLNG
CONSTRAINT #    6 = BORING


CONSTRAINT CASTNG
    +1.00 CSTPST    +3.33 MANFLD
 <=     10000.00


CONSTRAINT FORGE
    +2.67 FRGPST    +1.00 VALVE
    +1.60 CONROD <=     12000.00


CONSTRAINT LATHE
    +2.00 CSTPST    +2.00 FRGPST
    +1.00 VALVE  <=      9000.00


CONSTRAINT GRINDR
    +1.00 CSTPST    +1.00 FRGPST
    +1.00 VALVE     +3.00 CONROD
 <=      9600.00


CONSTRAINT MILLNG
    +2.00 MANFLD    +3.00 CONROD
 <=      8000.00


CONSTRAINT BORING
    +3.00 MANFLD    +1.00 CONROD
 <=      9600.00


OBJECTIVE FUNCTION
MAXIMIZE
    -3.00 CSTPST    -4.00 FRGPST
   -12.00 MANFLD    -2.50 VALVE
    -5.00 CONROD
```

The printout consists of four parts:

1) list of variable and constraint names.

2) list of constraint equations with constraint coefficients and RHS values.

3) objective function coefficients.

4) variable upper and lower limits.

```
VARIABLE LIMITS
 1000.00 <=  FRGPST  <=  2000.00
    0.00 <=  MANFLD  <=  3000.00
 4000.00 <=  VALVE   <=    UNBND
 1000.00 <=  CONROD  <=    UNBND
```

Solution of the Problem:

The first step in optimization is to define all variables which will be used in solving the problem. In this example the variables in columns 1 through 5 of the tableau are the problem variables.

Surplus variables are positive variables subtracted from the left-hand side of > = (greater than or equal to) inequalities in order to make them equalities. This problem includes no > = inequalities, so there are no surplus variables.

Slack variables are positive variables added to the left-hand side of <= (less than or equal to) inequalities in order to make them equalities. Since all six constraints are <= inequalities, variables 6 through 11 are slack variables.

Artificial variables are positive variables added to the left-hand side of equalities and >= inequalities. They are used to generate an initial basic feasible solution, from which the iterative optimization process begins. No artificial variables are needed.

A printout is made showing the problem variables and any surplus, slack and artificial variables, as necessary.

```
                 MCHINE
VARIABLES            FROM   THROUGH

PROBLEM               1        5
SLACK                 6       11
```

The next step is to form the initial tableau. This is an array of coefficients used in optimizing an LP problem by the simplex method. The initial tableau may be printed prior to the beginning of optimization. For this example, the initial tableau is as follows:

```
TABLEAU AFTER    0 ITERATIONS
       1.00      0.00       3.33
       0.00      0.00       1.00
       0.00      0.00       0.00
       0.00      0.00   10000.00

       0.00      2.67       0.00
       1.00      1.60       0.00
       1.00      0.00       0.00
       0.00      0.00    3730.00
```

```
 2.00      2.00       0.00
 1.00      0.00       0.00
 0.00      1.00       0.00
 0.00      0.00    3000.00

 1.00      1.00       0.00
 1.00      3.00       0.00
 0.00      0.00       1.00
 0.00      0.00    1600.00

 0.00      0.00       2.00
 0.00      3.00       0.00
 0.00      0.00       0.00
 1.00      0.00    5000.00

 0.00      0.00       3.00
 0.00      1.00       0.00
 0.00      0.00       0.00
 0.00      1.00    8600.00

-3.00     -4.00     -12.00
-2.50     -5.00       0.00
 0.00      0.00       0.00
 0.00      0.00       0.00
```

The difficulty in recognizing the coefficients in the tableau is due to print format restrictions. If the tableau could be printed without limits on paper width, it would appear as in Table III.

Table III

Initial Tableau

	CSTPST	FRGPST	MANFLD	VALVE	CONROD	SLACK 1	SLACK 2	SLACK 3	SLACK 4	SLACK 5	SLACK 6	RHS
SLACK 1	1	0	3.33	0	0	1	0	0	0	0	0	10000.00
SLACK 2	0	2.67	0	1	1.6	0	1	0	0	0	0	3730.00
SLACK 3	2	2	0	1	0	0	0	1	0	0	0	3000.00
SLACK 4	1	1	0	1	3	0	0	0	1	0	0	1600.00
SLACK 5	0	0	2	0	3	0	0	0	0	1	0	5000.00
SLACK 6	0	0	3	0	1	0	0	0	0	0	1	8600.00
OBJ FNC	−3	−4	−12	−2.5	−5	0	0	0	0	0	0	0

With the initial tableau in this format it is easier to see that the initial basis includes the six slack variables (columns 6 through 11). The signs are reversed on the objective function coefficients (row 7) because the algorithm is a minimization algorithm and the example problem is a maximization problem. The constraint RHS values (column 12) are modified by the presence of the variable lower bounds. We can see this by examining constraint #5 (MILLNG):

$$\text{Tableau RHS value} = \text{original RHS value} - \sum_{\text{all variables}} \left[(\text{variable coeff})\,(\text{lower limit})\right]$$

$$5000.00 = 8000.00 - (2.00)\,(0) - (3.00)\,(1000)$$

Note that variable upper bounds are incorporated in the algorithm and that neither upper nor lower bounds are formal constraints.

Finally, the objective function value for the initial tableau is zero (row 7, column 12).

After the tableau is printed, the optimization is performed. The HP-85 LP Pac includes a binary language program which performs each optimization iteration approximation six times faster than would be the case if the routine had been written in BASIC.

After optimization the results are printed.

```
    OPTIMAL SOLUTION: MCHINE

  BASIS AFTER     4 ITERATIONS

 VARIABLE                    VALUE
 SLACK     1              1275.000
 SLACK     2               860.000
 CSTPST                    400.000
 VALVE                    4200.000
 MANFLD                   2500.000
 SLACK     6              1100.000
 FRGPST AT UPR BND        2000.000
 CONROD AT LWR BND        1000.000
```

The final basis contains all variables which are in the solution (all variable with values > zero). Any variables which are at upper or lower bounds are also displayed although these variables are not in the final basis. In this example, optimization was reached after 4 iterations.

The optimal production plan for XYZ for the next month is:

400	cast aluminum pistons
2000	forged aluminum pistons
2500	cast intake manifolds
4200	forged valves
1000	forged connecting rods

The value for the objective function is then printed.

```
OBJ FUNC VALUE =        54700.000
```

This plan for XYZ's production next month will generate $54,700.00 contribution to fixed costs and profit.

Three slack variables are in the basis indicating that the corresponding constraints are not binding. The amount of each constraint used equals the RHS value less the slack value. These amounts are presented in Table IV.

Table IV

Non-Binding Constraints

Slack Variable	Constraint Name	RHS Value	–	Slack Value	=	Amount Used
1	CASTNG	10000.00	–	1275.00	=	8725.00
2	FORGE	12000.00	–	860.00	=	11140.00
6	BORING	9600.00	–	1100.00	=	8500.00

Dual variable values ("shadow prices") give significant information about the economic value of the constraints. The value of the dual variable is the amount of increase in the objective function value for each unit by which the constraint RHS is relaxed. (Relaxed means increased for <= constraints in this example.)

```
          DUAL VARIABLES

COLUMN      CONSTRAINT      VALUE
   6          CASTNG         0.000
   7          FORGE          0.000
   8          LATHE           .500
   9          GRINDR         2.000
  10          MILLNG         6.000
  11          BORING         0.000
```

In this example another "unit" of grinder capacity (increasing the GRINDR constraint #4 RHS value from 9600 to 9601) would increase the objective function value by $2.00.

Dual variable values hold only so long as the solution basis does not change. Sensitivity analysis is performed to determine over what range the dual variable values will remain valid.

Constraints which have zero values for the dual variables, such as CASTNG and FORGE are already in excess. Increasing the constraint RHS value would only increase the value of the associated slack variable and would not improve the solution.

The final tableau may then be printed.

```
TABLEAU AFTER     4 ITERATIONS
        0.00       1.00        0.00
        0.00      -2.00        1.00
        0.00      -1.00        1.00
       -1.67       0.00     1275.00

        0.00      -2.67        0.00
        0.00      -4.40        0.00
        1.00       1.00       -2.00
        0.00       0.00      860.00

        1.00      -1.00        0.00
        0.00      -3.00        0.00
        0.00       1.00       -1.00
        0.00       0.00      400.00

        0.00       0.00        0.00
        1.00       6.00        0.00
        0.00      -1.00        2.00
        0.00       0.00      200.00

        0.00       0.00        1.00
        0.00       1.50        0.00
        0.00       0.00        0.00
         .50       0.00     2500.00

        0.00       0.00        0.00
        0.00      -3.50        0.00
        0.00       0.00        0.00
       -1.50       1.00     1100.00

        0.00       1.00        0.00
        0.00      19.00        0.00
        0.00        .50        2.00
        6.00       0.00    35700.00
```

The final tableau, displayed in full width, is shown in Table V.

Table V

Final Tableau

	CSTPST	FRGPST	MANFLD	VALVE	CONROD	SLACK 1	SLACK 2	SLACK 3	SLACK 4	SLACK 5	SLACK 6	RHS
SLACK 1	0	1	0	0	−2	1	0	−1	1	−1.67	0	1275.00
SLACK 2	0	−2.67	0	0	−4.4	0	1	1	−2	0	0	860.00
CSTPST	1	−1	0	0	−3	0	0	1	−1	0	0	400.00
VALVE	0	0	0	1	6	0	0	−1	2	0	0	200.00
MANFLD	0	0	1	0	1.5	0	0	0	0	.5	0	2500.00
SLACK 6	0	0	0	0	−3.5	0	0	0	0	−1.5	1	1100.00
OBJ FNC	0	1	0	0	19.0	0	0	.5	2	6	0	35700.00

Every basis variable has row coefficients of zero in all rows except its own, in which it has a coefficient of one. Non-basis variables have objective function row coefficients which may be called "shadow prices" as they reflect an "opportunity cost". In this example, connecting rods (CONROD) are at their lower bound of 1000 units Each unit which is required to be made reduces the objective function value by $19. If we could *relax* the lower bound so only 999 needed to be made then the objective function would improve by $19.

Forged pistons (FRGPST) are at their upper bound of 2000. If we could relax this upper bound so one *more* unit could be made then the objective function would improve by $1.

The "shadow prices" for slacks 3, 4 and 5 are the dual variable values discussed earlier.

The final tableau RHS column contains basis variable values and the objective function value. Both have been modified by the variable upper and lower bounds.

For maximization problems the basis variable value equals the final tableau RHS value plus lower bound (or plus zero if no lower bound).

For minimization problems if an upper bound is present, the basis variable value equals the upper bound minus the final tableau RHS value. If there is no upper bound the calculation is the same as for maximization problems.

The objective function value is calculated as follows:

$$\text{Obj. Func. Value} = \text{final tableau obj. func. value} + \sum_{\text{all variables}} \left[(\text{obj. func. coeff.})(\text{lower bound})\right]$$

$$54700 = 35700 + (4/\text{FRGPST})\,(1000) + (2.5/\text{VALVE})\,(4000) + (5/\text{CONROD})\,(1000)$$

Tableau elements in columns corresponding to non-basis variables represent substitution rates for basis variables. They are used in the sensitivity analysis which follows.

Sensitivity Analysis

The sensitivity analysis gives more information about the variables and constraints in and out of the basis.

SENSITIVITY ANALYSIS

CONSTRAINT RHS VALUE RANGING

CON	LOWER LIMIT	RHS VALUE	UPPER LIMIT
CASTNG	8725.00	10000.00	UNBND
FORGE	11140.00	12000.00	UNBND
LATHE	8600.00	9000.00	9200.00
GRINDR	9500.00	9600.00	10000.00
MILLNG	3000.00	8000.00	8733.33
BORING	8500.00	9600.00	UNBND

```
        OBJ FUNC COEFF RANGING
            BASIS VARIABLES

   VAR       LOWER     OBJ FNC     UPPER
             LIMIT      VALUE      LIMIT
CSTPST        2.50       3.00       4.00
VALVE         1.50       2.50       3.00
MANFLD         .00      12.00      UNBND

        OBJ FUNC COEFF RANGING
          NON-BASIS VARIABLES

   VAR       LOWER     OBJ FNC     UPPER
             LIMIT      VALUE      LIMIT
FRGPST       UNBND       4.00       5.00
CONROD       UNBND       5.00      24.00
```

Within the upper and lower limits you may change any one constraint RHS value or objective function coefficient without changing the basis (the variables which are in the solution). Their values may change and the objective function value may change but the basis will remain the same. If a variable or constraint coefficient is changed beyond the limits, or if more than one at a time is changed, the solution or basis variables may leave the solution.

Sensitivity analysis determines the range of validity of dual variable values. Earlier we saw that increasing the GRINDR constraint from 9600 to 9601 units would increase the objective function value by $2.00. Now we see that the range of applicability of this $2.00 per unit change is only from 9500 to 10,000 units. Beyond these limits, the basis will change as will the dual variable values. When UNBND is printed, there is no limit (upper or lower) for that RHS value or variable coefficient.

Variables which are at their upper or lower bounds are included in the category of non-basis variables. Non-basis variable ranging values are the same as "shadow prices." In this example, connecting rods (CONROD) have a "shadow price" or "opportunity cost" of $19 per unit. This is the difference between the $5 objective function coefficient and the $24 ranging upper limit. If the unit profit contribution for connecting rods were increased by more than $19 we would expect this product to enter the basis.

Second Example

Statement of the Problem:

Our second example is a cost minimization problem. A farmer wants to develop a feeding formula for broiler starters (chickens 0 to 4 weeks old). The feed mix must meet certain nutritional requirements, regardless of cost.

There are ten different ingredients which may be used in the chicken feed. Each ingredient provides different percentages of the nutrients. These ingredients and their nutrient percentages are shown in Table I.

Table I

Ingredient	Nutrient % By Weight: Protein	Calcium	Available Phosphorus	Lysine	Methionine	Methionine & Cystine	Energy Kcal/Kg	Price c/Kg
Corn	8.8	.03	.27	.2	.17	.26	3450	10.0
Alfalfa	15.2	1.23	.22	.6	.2	.37	1600	13.0
Fishmeal	61.3	5.5	2.81	5.3	1.8	2.74	2900	43.5
Meat & Bone Meal	50.6	10.57	5.07	3.5	.7	1.3	2000	25.0
Rice Bran	13.0	.6	1.8	.5	.2	.5	2100	10.1
Soybean	45.8	.32	.67	2.9	.6	1.27	2250	19.0
Defluorinated Phosphate	0	32.0	18.0	0	0	0	0	22.0
Ground Limestone	0	35.8	0	0	0	0	0	1.1
Cottonseed Meal	49.0	.28	.9	1.7	.8	2.0	1900	11.0
Fat	0	0	0	0	0	0	8000	40.0

Certain nutrients are required to achieve the desired growth rate. The nutrient requirements for the chicken feed are given in Table II.

Table II

Nutrient Requirements

Nutrient or Energy	Minimum Requirement (lower or >= constraint) %	Maximum Allowable (upper or <= constraint) %
Protein	25	NONE
Calcium	0.8	1.2
Avail. Phosphorus	0.5	0.7
Lysine	1.3	NONE
Methionine	0.5	NONE
Methionine & Cystine	0.9	NONE
Energy	3200 Kcal/Kg	NONE
Weight	1 Kg	1 Kg

The chicken feed can only contain certain percentages of some of the ingredients, either due to limited availability of the ingredient or because a certain ingredient must be included. Upper and lower bounds on ingredients are given in Table III.

Table III

Ingredient Requirements

Ingredient	Lower Bound, Kg	Upper Bound, Kg
Corn	0	.4
Alfalfa	.02	.08
Fishmeal	0	.10
Meat & Bone Meal	0	.10
Rice Bran	0	NONE
Soybean	0	NONE
Defluorinated Phosphate	0	NONE
Ground Limestone	0	NONE
Cottonseed Meal	0	.25
Fat	0	.10

The farmer's problem is to determine (a) the optimal mix of ingredients to minimize cost per kilogram of feed, and (b) to calculate the minimum cost per kilogram of this mix.

Setting up the Problem:

The first step is to convert Tables I and II to a set of constraint equations. Refer to the data input form to see how this was done. The nutrient-ingredient relationships are expressed in percent in both tables so this unit of measure can be used in the constraint equations. This is also true of the energy-ingredient relationships in constraint #10. Constraint #3, (WEIGHT) is included to insure that the mix will be exactly one kilogram in weight.

The objective function coefficients are expressed in cents/Kg rather than dollars/Kg to avoid losing information due to display format limitations.

The upper and lower bounds on ingredients in Table III are expressed in kg to be consistent with the set of constraint equations.

Follow the User Instructions to enter the problem from the keyboard or tape cartridge (stored as 16K data file CHICK).

Problem Name: CHICK

Maximize or Minimize MIN **No. Variables** 10

No. Constraints 10 **No. <=** 2 **No. =** 1 **No. >=** 7

Variable Name / Constraint Name	CORN	ALFLFA	FISHML	BONEML	RICEBR	SOY BN	PHOSPH	LIMEST	CTTNSD	FAT	Constraint Type (<=, =, >=)	Constraint RHS Value
CALCUP	.03	1.23	5.5	10.57	.6	.32	32.	35.8	.28	0	<=	1.2
PHOSUP	.27	.22	2.81	5.07	1.8	.67	18.	0	.9	0	<=	.7
WEIGHT	1	1	1	1	1	1	1	1	1	1	=	1.
PRTNLO	8.8	15.2	61.3	50.6	13.0	45.8	0	0	49.0	0	>=	25.
CALCLO	.03	1.23	5.5	10.57	.6	.32	32.	35.8	.28	0	>=	.8
PHOSLO	.27	.22	2.81	5.07	1.8	.67	18.	0	.9	0	>=	.5
LYSNLO	.2	.6	5.3	3.5	.5	2.9	0	0	1.7	0	>=	1.3
METHLO	.17	.2	1.8	.7	.2	.6	0	0	.8	0	>=	.5
MECYLO	.26	.37	2.14	1.3	.5	1.27	0	0	2.	0	>=	.9
ENRGLO	3450	1600	2900	2000	2100	2250	0	0	1900	8000	>=	3200
Objective Function Coeff	10	13	43.5	25	10.1	19	22	1.1	11	40		
Upper Bound	.4	.08	.1	.1	-1	-1	-1	-1	.25	.1		
Lower Bound	0	.02	0	0	0	0	0	0	0	0		

NOTE: Constraints should be entered in the following order <=, =, >=. Constraint RHS values should be >= 0.

After entering the problem you have the option to print it out.

Printout of the Problem

```
                 CHICK

VARIABLE    #    1  =  CORN
VARIABLE    #    2  =  ALFLFA
VARIABLE    #    3  =  FISHML
VARIABLE    #    4  =  BONEML
VARIABLE    #    5  =  RICEBR
VARIABLE    #    6  =  SOYBN
VARIABLE    #    7  =  PHOSPH
VARIABLE    #    8  =  LIMEST
VARIABLE    #    9  =  CTTNSD
VARIABLE    #   10  =  FAT

CONSTRAINT  #    1  =  CALCUP
CONSTRAINT  #    2  =  PHOSUP
CONSTRAINT  #    3  =  WEIGHT
CONSTRAINT  #    4  =  PRTNLO
CONSTRAINT  #    5  =  CALCLO
CONSTRAINT  #    6  =  PHOSLO
CONSTRAINT  #    7  =  LYSNLO
CONSTRAINT  #    8  =  METHLO
CONSTRAINT  #    9  =  MECYLO
CONSTRAINT  #   10  =  ENRGLO

CONSTRAINT CALCUP
     +.03 CORN        +1.23 ALFLFA
    +5.50 FISHML     +10.57 BONEML
     +.60 RICEBR       +.32 SOYBN
   +32.00 PHOSPH     +35.80 LIMEST
     +.28 CTTNSD <=          1.20

CONSTRAINT PHOSUP
     +.27 CORN         +.22 ALFLFA
    +2.81 FISHML      +5.07 BONEML
    +1.80 RICEBR       +.67 SOYBN
   +18.00 PHOSPH       +.90 CTTNSD
 <=           .70

CONSTRAINT WEIGHT
    +1.00 CORN        +1.00 ALFLFA
    +1.00 FISHML      +1.00 BONEML
    +1.00 RICEBR      +1.00 SOYBN
    +1.00 PHOSPH      +1.00 LIMEST
    +1.00 CTTNSD      +1.00 FAT
  =          1.00

CONSTRAINT PRTNLO
    +8.80 CORN       +15.20 ALFLFA
   +61.30 FISHML     +50.60 BONEML
   +13.00 RICEBR     +45.80 SOYBN
   +49.00 CTTNSD >=         25.00
```

```
CONSTRAINT CALCLO
     +.03 CORN      +1.23 ALFLFA
    +5.50 FISHML   +10.57 BONEML
     +.60 RICEBR     +.32 SOYBN
   +32.00 PHOSPH   +35.80 LIMEST
     +.28 CTTNSD >=         .80


CONSTRAINT PHOSLO
     +.27 CORN       +.22 ALFLFA
    +2.81 FISHML    +5.07 BONEML
    +1.80 RICEBR     +.67 SOYBN
   +18.00 PHOSPH     +.90 CTTNSD
 >=          .50


CONSTRAINT LYSNLO
     +.20 CORN       +.60 ALFLFA
    +5.30 FISHML    +3.50 BONEML
     +.50 RICEBR    +2.90 SOYBN
    +1.70 CTTNSD >=        1.30


CONSTRAINT METHLO
     +.17 CORN       +.20 ALFLFA
    +1.80 FISHML     +.70 BONEML
     +.20 RICEBR     +.60 SOYBN
     +.80 CTTNSD >=         .50


CONSTRAINT MECYLO
     +.26 CORN       +.37 ALFLFA
    +2.74 FISHML    +1.30 BONEML
     +.50 RICEBR    +1.27 SOYBN
    +2.00 CTTNSD >=         .90


CONSTRAINT ENRGLO
 +3450.00 CORN   +1600.00 ALFLFA
 +2900.00 FISHML +2000.00 BONEML
 +2100.00 RICEBR +2250.00 SOYBN
 +1900.00 CTTNSD +8000.00 FAT
 >=      3200.00


OBJECTIVE FUNCTION
MINIMIZE
   +10.00 CORN     +13.00 ALFLFA
   +43.50 FISHML   +25.00 BONEML
   +10.10 RICEBR   +19.00 SOYBN
   +22.00 PHOSPH    +1.10 LIMEST
   +11.00 CTTNSD   +40.00 FAT
```

```
VARIABLE LIMITS
     0.00 <=  CORN     <=      .40
      .02 <=  ALFLFA   <=      .08
     0.00 <=  FISHML   <=      .10
     0.00 <=  BONEML   <=      .10
     0.00 <=  CTTNSD   <=      .25
     0.00 <=  FAT      <=      .10
```

Solution of the Problem:

Refer to the first example for discussion of surplus, slack and artificial variables, and the initial tableau.

```
                  CHICK

VARIABLES             FROM   THROUGH

PROBLEM                 1       10
SURPLUS                11       17
SLACK                  18       19
ARTIFICIAL             20       27

TABLEAU AFTER    0 ITERATIONS
          .03      1.23        5.50
        10.57       .60         .32
        32.00     35.80         .28
         0.00      0.00        0.00
         0.00      0.00        0.00
         0.00      0.00        1.00
         0.00      0.00        0.00
         0.00      0.00        0.00
         0.00      0.00        0.00
         1.18

          .27       .22        2.81
         5.07      1.80         .67
        18.00      0.00         .90
         0.00      0.00        0.00
         0.00      0.00        0.00
         0.00      0.00        0.00
         1.00      0.00        0.00
         0.00      0.00        0.00
         0.00      0.00        0.00
          .70

         1.00      1.00        1.00
         1.00      1.00        1.00
         1.00      1.00        1.00
         1.00      0.00        0.00
         0.00      0.00        0.00
         0.00      0.00        0.00
         0.00      1.00        0.00
         0.00      0.00        0.00
         0.00      0.00        0.00
          .98
```

8.80	15.20	61.30
50.60	13.00	45.80
0.00	0.00	49.00
0.00	-1.00	0.00
0.00	0.00	0.00
0.00	0.00	0.00
0.00	0.00	1.00
0.00	0.00	0.00
0.00	0.00	0.00
24.70		

.03	1.23	5.50
10.57	.60	.32
32.00	35.80	.28
0.00	0.00	-1.00
0.00	0.00	0.00
0.00	0.00	0.00
0.00	0.00	0.00
1.00	0.00	0.00
0.00	0.00	0.00
.78		

.27	.22	2.81
5.07	1.80	.67
18.00	0.00	.90
0.00	0.00	0.00
-1.00	0.00	0.00
0.00	0.00	0.00
0.00	0.00	0.00
0.00	1.00	0.00
0.00	0.00	0.00
.50		

.20	.60	5.30
3.50	.50	2.90
0.00	0.00	1.70
0.00	0.00	0.00
0.00	-1.00	0.00
0.00	0.00	0.00
0.00	0.00	0.00
0.00	0.00	1.00
0.00	0.00	0.00
1.29		

.17	.20	1.80
.70	.20	.60
0.00	0.00	.80
0.00	0.00	0.00
0.00	0.00	-1.00
0.00	0.00	0.00
0.00	0.00	0.00
0.00	0.00	0.00
1.00	0.00	0.00
.50		

```
      .26        .37       2.74
     1.30        .50       1.27
     0.00       0.00       2.00
     0.00       0.00       0.00
     0.00       0.00       0.00
    -1.00       0.00       0.00
     0.00       0.00       0.00
     0.00       0.00       0.00
     0.00       1.00       0.00
      .89

  3450.00    1600.00    2900.00
  2000.00    2100.00    2250.00
     0.00       0.00    1900.00
  8000.00       0.00       0.00
     0.00       0.00       0.00
     0.00      -1.00       0.00
     0.00       0.00       0.00
     0.00       0.00       0.00
     0.00       0.00       1.00
  3168.00

    10.00      13.00      43.50
    25.00      10.10      19.00
    22.00       1.10      11.00
    40.00       0.00       0.00
     0.00       0.00       0.00
     0.00       0.00       0.00
     0.00       0.00       0.00
     0.00       0.00       0.00
     0.00       0.00       0.00
     0.00
```

After the initial tableau is printed out the optimization is performed and the results are printed.

```
   OPTIMAL SOLUTION: CHICK

 BASIS AFTER   22 ITERATIONS

VARIABLE                      VALUE
SLACK     1                    .400
SURPLUS   9                    .108
SURPLUS   7                    .025
SURPLUS   4                   2.125
LIMEST                         .005
SOYBN                          .123
FAT                            .093
RICEBR                         .027
FISHML                         .083
SURPLUS   6                    .200
CORN    AT UPR BND             .400
CTTNSD  AT UPR BND             .250
ALFLFA  AT LWR BND             .020
```

The ingredients and amounts used are:

CORN	.400 kg
ALFALFA	.020 kg
FISHMEAL	.083 kg
RICE BRAN	.027 kg
SOYBEAN	.123 kg
GROUND LIMESTONE	.005 kg
COTTONSEED MEAL	.250 kg
FAT	.093 kg
TOTAL	1.001 kg

Truncation error of .001 occurs from using only three decimal places for accuracy in the output.

```
OBJ FUNC VALUE =          16.922
```

The minimum cost per kilogram of feed mix is $.169.

Of the ten constraints, five are non-binding, indicated by the presence of their associated slack or surplus variables in the basis. Non-binding constraints are presented in Table IV.

Table IV

Non-Binding Constraints

Slack/Surplus	Constraint Name	RHS Value	±	Slack/Surplus Value	=	Solution Value
SLACK 1	CALCUP	1.2%	−	.400%	=	.800%
SURPLUS 4	PRTNLO	25.0%	+	2.125%	=	27.125%
SURPLUS 6	PHOSLO	.5%	+	.200%	=	.700%
SURPLUS 7	LYSNLO	1.3%	+	.025%	=	1.325%
SURPLUS 9	MECYLO	.9%	+	.108%	=	1.008%

The other five constraints are binding, and by examining the dual variable values (''shadow prices'') we can see how the solution may be improved by changing the constraint RHS values.

```
             DUAL VARIABLES

COLUMN       CONSTRAINT          VALUE
  11            PRTNLO           0.000
  12            CALCLO            .136
  13            PHOSLO           0.000
  14            LYSNLO           0.000
  15            METHLO          18.192
  16            MECYLO           0.000
  17            ENRGLO            .005
  18            CALCUP           0.000
  19            PHOSUP            .744
  20            WEIGHT           3.771
```

Constraint #5 (CALCLO) requires the percentage of calcium to be >= .8%. Relaxing this constraint by one unit (1%) would improve the objective function value by $.136. Such a large relaxation is, of course, not possible, since the original RHS value is only .8%. However, a 0.15% relaxation (from 0.8% to 0.65%) is possible and this would improve the objective function value by (.15) (.136) or .020, from 16.922 cents to 16.902 cents per kilogram.

Constraint #8 (METHLO) requires the percentage of Methionine to be >= .50%. The effect of a one unit (1%) change is $18.192. Increasing the constraint RHS value to .52% would increase the objective function by (.02) (18.192) or .364, from 16.922 to 17.286 cents per kilogram.

The final tableau may be printed out.

```
TABLEAU AFTER     22 ITERATIONS
         0.00         0.00         0.00
         0.00         0.00         0.00
         0.00         0.00         0.00
         0.00         0.00         1.00
         0.00         0.00         0.00
         0.00         0.00         1.00
         0.00         0.00         0.00
        -1.00         0.00         0.00
         0.00         0.00         0.00
          .40

         -.30          .36         0.00
         -.60         0.00         0.00
        -2.30         0.00          .42
         0.00         0.00          .02
         0.00         0.00        -1.57
         1.00          .00         0.00
         -.13          .59         0.00
         -.02         0.00         0.00
         1.57        -1.00         -.00
          .11
```

-1.17	1.25	0.00
-4.21	0.00	0.00
-14.39	0.00	-1.84
0.00	0.00	.05
0.00	1.00	-3.76
0.00	.00	0.00
-.81	1.67	0.00
-.05	0.00	-1.00
3.76	0.00	-.00
.03		

-17.62	21.08	0.00
-70.31	0.00	0.00
-240.71	0.00	-4.38
0.00	1.00	1.11
0.00	0.00	-44.64
0.00	.01	0.00
-13.61	39.63	-1.00
-1.11	0.00	0.00
44.64	0.00	-.01
2.13		

-.02	.06	0.00
.14	0.00	0.00
.39	1.00	.02
0.00	0.00	-.03
0.00	0.00	.06
0.00	.00	0.00
-.03	.06	0.00
.03	0.00	0.00
-.06	0.00	-.00
.01		

-.78	1.13	0.00
-2.27	0.00	1.00
-11.92	0.00	-1.04
0.00	0.00	.04
0.00	0.00	-.65
0.00	.00	0.00
-.67	1.50	0.00
-.04	0.00	0.00
.65	0.00	-.00
.12		

.24	.05	0.00
.03	0.00	0.00
.30	0.00	-.08
1.00	0.00	.01
0.00	0.00	-.10
0.00	.00	0.00
.01	.30	0.00
-.01	0.00	0.00
.10	0.00	-.00
.01		

-.14	.14	0.00
2.27	1.00	0.00
9.96	0.00	.05
0.00	0.00	.01
0.00	0.00	.93
0.00	.00	0.00
.55	.27	0.00
-.01	0.00	0.00
-.93	0.00	-.00
.03		

-.18	.28	1.00
-.89	0.00	0.00
-2.87	0.00	.10
0.00	0.00	.01
0.00	0.00	.44
0.00	.00	0.00
-.16	.53	0.00
-.01	0.00	0.00
-.44	0.00	-.00
.02		

0.00	0.00	0.00
0.00	0.00	0.00
0.00	0.00	0.00
0.00	0.00	0.00
1.00	0.00	0.00
0.00	0.00	0.00
1.00	0.00	0.00
0.00	-1.00	0.00
0.00	0.00	0.00
.20		

8.00	4.37	0.00
7.43	0.00	0.00
34.81	0.00	9.55
0.00	0.00	.14
0.00	0.00	18.19
0.00	.01	0.00
.74	3.77	0.00
-.14	0.00	0.00
-18.19	0.00	-.01
-16.66		

Sensitivity Analysis

SENSITIVITY ANALYSIS

CONSTRAINT RHS VALUE RANGING

CON	LOWER LIMIT	RHS VALUE	UPPER LIMIT
CALCUP	.80	1.20	UNBND
PHOSUP	.65	.70	.73
WEIGHT	.98	1.00	UNBND
PRTNLO	UNBND	25.00	27.13
CALCLO	.60	.80	1.20
PHOSLO	UNBND	.50	.70
LYSNLO	UNBND	1.30	1.33
METHLO	.49	.50	.53
MECYLO	UNBND	.90	1.01
ENRGLO	UNBND	3200.00	3245.84

OBJ FUNC COEFF RANGING
BASIS VARIABLES

VAR	LOWER LIMIT	OBJ FNC VALUE	UPPER LIMIT
LIMEST	UNBND	1.10	54.05
SOYBN	17.89	19.00	21.52
FAT	27.42	40.00	159.14
RICEBR	UNBND	10.10	11.45
FISHML	36.37	43.50	48.08

OBJ FUNC COEFF RANGING
NON-BASIS VARIABLES

VAR	LOWER LIMIT	OBJ FNC VALUE	UPPER LIMIT
CORN	2.00	10.00	UNBND
ALFLFA	8.63	13.00	UNBND
BONEML	17.57	25.00	UNBND
PHOSPH	UNBND	22.00	UNBND
CTTNSD	1.45	11.00	UNBND

Please refer to the first example for an explanation of the final tableau and sensitivity analysis.

In this second example, the sensitivity analysis was used to determine the range of validity of the dual variable values (''shadow prices'') discussed earlier. Constraint #5 (CALCLO) has a range from .6% to 1.2%, and constraint #8 (METHLO) has a range from .49% to .53%. Within these ranges the dual variable values for these constraints are valid and can be used to estimate a change in the objective function value resulting from a change in a constraint RHS value.

The narrow range for METHLO means that the solution (basis) is particularly sensitive to changes in the RHS value for this constraint.

Appendix A
Remarks Program

To help you understand the flow of the programs contained in the Linear Programming Pac, abbreviated remarks for each of the programs in the pac, as well as definitions of variables used, are contained in a program named "REMARK". When using this program you may want to refer to Appendix C for an explanation of the LP tableau structure.

User Instructions

1. To load the program:
   a. Insert the LP Pac cartridge into the tape transport.
   b. Type: (REW LOAD) "REMARK" (END LINE)
2. When the program has been loaded:
   a. Press: (RUN)
3. When PRINT OR DISPLAY OUTPUT(P/D)? is displayed:
   a. Enter: P (END LINE) to print the output.

   OR:

   a. Enter: D (END LINE) to display the output.

   Note: Contents of the display screen may be output to the printer at any time by pressing (SHIFT) (COPY).
4. When the keys are labeled:

```
-----------------------------------------
SOLVE     ANSWER     SENSAN     VARBLE
  LP      MODIFY     PRINT      STORE
```

   a. Press: KEY #1 (LP) for LP (problem entry) remarks.

   OR:

   a. Press: KEY #2 (MODIFY) for Modify remarks.

   OR:

   a. Press: KEY #3 (PRINT) for Print remarks.

   OR:

   a. Press: KEY #4 (STORE) for Store remarks.

   OR:

   a. Press: KEY #5 (SOLVE) for Solve remarks.

   OR:

   a. Press: KEY #6 (ANSWER) for Answer remarks.

   OR:

   a. Press: KEY #7 (SENSAN) for Sensitivity Analysis remarks.

   OR:

   a. Press: KEY #8 (VARBLE) for program variable definitions.

Appendix B

To obtain a catalog of the programs and data files stored on tape:

a. Insert the LP Pac cartridge into the tape transport.

b. Type: CAT [END LINE]

To purge a problem (data file) from tape:

a. Insert the LP Pac cartridge into the tape transport.

b. Type: PURGE " (problem name) " [END LINE]

To display a listing of one of the LP Pac programs:

a. Insert the LP Pac cartridge into the tape transport.

b. Type: [REW LOAD] " (program name) " [END LINE]

c. Press: [P LST LIST]

To print a listing of one of the LP Pac programs:

a. Insert the LP Pac cartridge into the tape transport.

b. Type: [REW LOAD] " (program name) " [END LINE]

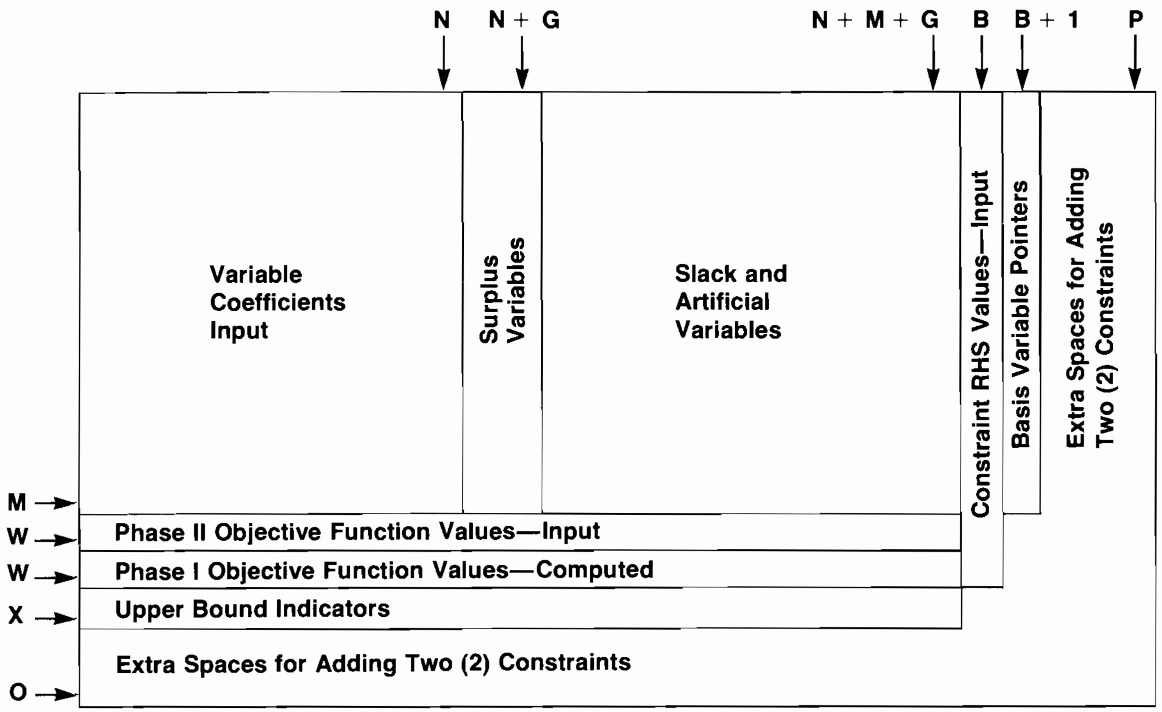
c. Press: [SHIFT] [P LST LIST]

Appendix C

Tableau Structure

Note: Understanding the tableau structure is not necessary for formulating the LP problem and interpreting the solution. It is helpful when using the Remarks program (Appendix A) to follow program flow.

The matrix A(,) contains the variable coefficients, constraint RHS values, objective function values, basis variable pointers and indicators for whether variables are at their upper bounds. The structure is shown below:



In the tableau A(,) the following variables are used:

B is the pointer to the constraint RHS value column in A(,); B = N + M + G + 1

G is the number of greater than or equal to (>=) type constraints.

M is the number of constraints.

N is the number of variables.

O is the number of rows in A(,); O = M + 5

P is the number of columns in A(,); P = N + M + G + 6

W is the phase objective function pointer in A(,).

X is the upper bound indicator pointer in A(,).

Appendix D
Using Your Linear Programming Pac With an Eighty-Column Display

Note: The actual error messages, prompt messages, and report formats may be different than those listed in the manual due to the larger display on your computer.

The eighty-column display Linear Programming (LP) Pac consists of the following programs: LP, MODIFY, PRINT, SOLVE, ANSWER, SENSAN, REMARK and CONVERT which will quickly and efficiently solve linear programming problems of up to 281 combined variables and constraints. Although these programs have the same names as those in the HP-85 LP Pac, they are not the same programs and cannot be interchanged. However, LP problems created and stored on disc using the HP-85 can be retrieved and utilized by the eighty-column display LP Pac by using the CONVERT program to convert the problem to the proper format.

Problem Dimensions

Maximum problem size is dependent on available read-write memory (random-access memory or RAM). The programs in the LP Pac automatically calculate the maximum problem size based on the amount of RAM available according to the table below. When entering a problem, KEY #2 (GUIDE) will list the current size limits that have been established for that problem.

		32K	64K	96K	160K	288K
Tableau Matrix*	A (,)	(21, 56)	(42, 117)	(57, 154)	(79, 211)	(105, 287)
Variables	N	1 to 49	1 to 110	1 to 147	1 to 204	1 to 280
Constraints	M	1 to 16	1 to 37	1 to 52	1 to 74	1 to 100
>= Constraints	G	0 to 16	0 to 37	0 to 52	0 to 74	0 to 100
	N + M + G	2 to 50	2 to 111	2 to 148	2 to 205	2 to 281

*See Appendix C

For each constraint (M) and each >= constraint (G), the maximum number of variables (N) is reduced by one. For example, if your computer has 96K bytes of read-write memory available and M = 50 and G = 30, then maximum N = 68.

Problems previously stored can be accessed if the amount of RAM presently available is greater than or equal to the amount of RAM available when the problem was stored. For example, a problem created with 96K bytes may be accessed with 96K of RAM or larger. If there is not enough read-write memory available for the problem, an error message will be displayed.

Note: Due to the large size of problems created with 288K bytes of RAM, they must be stored alone on a separate data disc if N + M + G is larger than 230.

Using HP-85 Files

Linear programming problems that were stored on disc using the HP-85 can be used by the eighty-column display LP Pac if they are converted to the proper format with the CONVERT program. Problems may have been created using either the disc or tape version of the HP-85 LP Pac; however, problems stored with the tape version must be copied to a disc before conversion. To convert a file, follow the steps below.

User Instructions

1. Insert the Linear Programming Pac disc into DRIVE 0 of the disc drive.
2. To load the program:
   a. Type: LOAD "CONVERT".
   Note: KEY #6 may be used here.
   b. Press: (END LINE).
3. To start the program:
   a. Press: (RUN).
4. When Enter the name of the problem to be converted: (10 CHARACTER MAXIMUM)? is displayed:
   a. Type in: the name of the HP-85 LP problem that is to be converted for use with the eighty-column display LP Pac.
   b. Press: (END LINE).
   OR:
   a. Press: (END LINE) to terminate the CONVERT program.
   b. Go to step 7.
5. You will then be asked Was (file name) stored by the TAPE or the DISC version of the HP-85 LP Pac?.
   a. Press: KEY #6 (TAPE) if the problem was created by the tape version of the HP-85 LP Pac and later copied to a disc.
   OR:
   a. Press: KEY #7 (DISC) if the problem was created and stored directly on the disc with the disc version.
6. After Enter the .VOLUME LABEL or :MSUS of the disc? appears:
   a. Enter: the .volume label of the disc containing the problem.
   b. Press: (END LINE).
   OR:
   a. Enter: the :msus of the disc drive that holds the disc with the LP problem file.
   b. Press: (END LINE).

Note: All volume labels must be preceded by a period, and all msus designators must begin with a colon. If they do not, the program will go to step 6.

Note: If FILE CONTAINS ILLEGAL DATA is displayed, it means that the CONVERT program cannot read this file, probably because it was not created using an HP-85 LP Pac. The program will return to step 4 to let you try a new problem.

7. After the LP problem is read in and converted, the keys will be labeled:
   a. Press: KEY #1 (HELP) to display the key functions.
   b. Go to step 7.

   OR:

   a. Press: KEY #8 (Exit Pac) to terminate the LP Pac.
   b. Go to step 8.

   OR:

   a. Press: KEY #2 (CONVERT) to convert another LP problem from an HP-85 file.
   b. Go to step 4.

   OR:

   Note: Refer to the appropriate section in the front of this manual for the instructions for ENTER, MODIFY, and SOLVE.

   a. Press: KEY #3 (ENTER) to enter an LP problem from the keyboard or a disc file.

   OR:

   a. Press: KEY #4 (MODIFY) to modify the converted LP problem.

   OR:

   a. Press: KEY #11 (RENAME) to rename the converted problem only.
   b. Go to step 13.

   OR:

   a. Press: KEY #5 (PRINT) to print out the current problem on the display screen or external printer.

   OR:

   a. Press: KEY #6 (STORE) to store the converted problem in a disc file.
   b. Go to step 10.

   Note: The user should use KEY #11 (RENAME) to rename the problem if the converted problem is to be stored on the same disc as the HP-85 problem without destroying the HP-85 file.

   OR:

   a. Press: KEY #7 (SOLVE) to solve the converted problem.

8. When Enter 'Y' [END LINE] to confirm is displayed:
   a. Enter: Y to terminate the CONVERT program.
   b. Press: (END LINE).
   c. Go to step 9.

   OR:

   a. Enter: N if you do not want to terminate the CONVERT program.
   b. Press: (END LINE).
   c. Go to step 7.

9. The program will beep and display PROGRAM ENDED! to confirm that the CONVERT program has stopped.

10. The program will display the current name of the converted problem and ask you to Enter the .VOLUME LABEL or :MSUS of the disc?
    a. Enter: the .volume label of the disc to receive the converted file.
    b. Press: (END LINE).

    OR:

    a. Enter: the :msus of the disc drive that holds the disc to receive the converted file.
    b. Press: (END LINE).

11. If `File` (file name) `already exists. Select option: (O/R/C)` is displayed:
    a. Enter: O to erase the contents of (file name) on the disc and store the converted problem in its place.
    b. Press: (END LINE).
    c. Go to step 7.

    OR:
    a. Enter: R to rename the current problem.
    b. Press: (END LINE).
    c. Go to step 13.

    OR:
    a. Enter: C to cancel the storing option.
    b. Press: (END LINE).
    c. Go to step 7.
12. Go to step 7.
13. The current name of the converted problem will be displayed along with the prompt

    `New name for the converted LP problem: (10 CHARACTER MAXIMUM)?`

    a. Enter: a new name for the converted problem.
    b. Press: (END LINE).
    c. Go to step 7.

    OR:
    a. Press: (END LINE) to keep the current name.
    b. Go to step 7.

Appendix E

Linear Programming Example

The Lemonade Stand

To get an idea of how LP might be used, let's take a relatively simple example. Suppose it's a hot summer day and we want to set up a lemonade stand. We plan to sell lemonade and iced tea, and feel that we can sell all we can make of either drink. A cup of lemonade uses 3 teaspoons (tsp) lemon powder and 2 tsp sugar. A cup of iced tea uses 1 tsp lemon powder, 2 tsp tea powder and 1 tsp sugar. Lemon powder costs \$.03/tsp and we have 70 tsp. Tea powder costs \$.04/tsp and we have 50 tsp. Sugar costs \$.01/tsp and we have 60 tsp. Paper cups cost \$.02 each and we have 50 cups. We have water and ice cubes, free, in unlimited quantities. Based on the recipes for lemonade and iced tea and on the costs of the materials, lemonade costs \$.13/cup and iced tea \$.14/cup. We plan to sell a cup of either drink for \$.25. Our objective is to make as much profit as we can. We must decide how much of each drink to make and then compute how much profit we'll make.

Our next step is to convert our descriptive model to a mathematical model. Let's take the objective function i.e., making the most profit, first. This consists of a series of problem variables, each variable having a coefficient. We have two variables: lemonade (L) and iced tea (T), each measured in cups. Since our objective is to maximize profit, the coefficient for each variable is its profit contribution per cup. Profit is sales price less cost, on a per cup basis. For lemonade it is \$.25 − \$.13 = \$.12, and for iced tea it is \$.25 − \$.14 = \$11. Therefore, our objective function is:

$$\text{Maximize } Z = .12 \times L + .11 \times T$$

The variable Z is simply the value of the objective function that will be computed for us. If, for example, we sell ten cups of lemonade and ten cups of iced tea, the value, Z, of the objective function is $.12 \times 10 + .11 \times 10 = \$2.30$.

The number of cups of lemonade and iced tea we can sell is limited or constrained by the resources available. There are four of them: lemon powder, tea powder, sugar, and paper cups. For each limiting resource we can describe a constraint mathematically.

For example, lemon powder is used at the rate of 3 tsp/cup of lemonade and 1 tsp/cup of iced tea; also we have a maximum of 70 tsp lemon powder. This statement is converted to the following constraint inequality:

$$3 \times L + 1 \times T \leqslant 70$$

The above is called a constraint because it states a limit on the size of the objective function based on the availability of the lemon powder resource. It's called an inequality because of the sign. By examining this constraint you can see that if we make only iced tea, we could make up to 70 cups. If we make only lemonade, we could make up to 70/3, or 23.3 cups. And we can make any combination of the two drinks as long as the constraint of 70 tsp is not exceeded. The other three resources can be evaluated in the same manner. The LP problem, stated in full, then appears as follows:

Maximize $Z = .12 \times L + .11 \times T$ subject to:

$3 \times L + 1 \times T \leqslant 70$	lemon powder
$2 \times T \leqslant 50$	tea powder
$2 \times L + 1 \times T \leqslant 60$	sugar
$1 \times L + 1 \times T \leqslant 50$	paper cups

After we have formulated the problem in the above manner, we can solve it using our HP Series 80 Personal Computers and the Linear Programming Pac. The time required for solving the problem is only a few seconds, whereas to solve it manually might require an hour or longer.

The LP solution tells us a number of interesting things. For maximum profit we should make 15 cups of lemonade and 25 cups of iced tea. This combination will earn us a profit of $4.55. Any other combination will be less profitable than this one, as you can determine simply by substituting other values for L and T in the objective function and constraint inequalities. In making the optimal combination of drinks, we will use all of the lemon powder and tea powder, while we will have 5 tsp sugar and 10 paper cups left unused. The lemon powder and tea powder are "scarce" resources. This means that we could make more drinks, and thus more profit, if we had more of these scarce resources. For example, within limits, each additional teaspoon of lemon powder would improve our profitability by $.04 above the cost of the powder itself. Knowing the true economic value of this scarce resource tells us how much we'd be willing to pay for it in the market place.

Maximize or Minimize __________ No. Variables __________

Problem Name: __________ No. Constraints __________ No. <= ______ No. = ______ No. >= ______

Variable Name																					Constraint Type (<=, =, >=)	Constraint RHS Value
Constraint Name																						
Objective Function Coeff																						
Upper Bound																						
Lower Bound																						

NOTE: Constraints should be entered in the following order <=, =, >=. Constraint RHS values should be >= 0.

Maximize or Minimize ________ No. Variables ________

Problem Name: ________ No. Constraints ________ No. <= ____ No. = ____ No. >= ____

Variable Name																					Constraint Type (<=, =, >=)	Constraint RHS Value
Constraint Name																						
Objective Function Coeff																						
Upper Bound																						
Lower Bound																						

NOTE: Constraints should be entered in the following order <=, =, >=. Constraint RHS values should be >= 0.

Maximize or Minimize ________ No. Variables ________

Problem Name: ________ No. Constraints ________ No. <= ____ No. = ____ No. >= ____

Variable Name																					Constraint Type (<=, =, >=)	Constraint RHS Value
Constraint Name																						
Objective Function Coeff																						
Upper Bound																						
Lower Bound																						

NOTE: Constraints should be entered in the following order <=, =, >=. Constraint RHS values should be >= 0.

Maximize or Minimize ________ **No. Variables** ________

Problem Name: ________ **No. Constraints** ________ **No. <=** ____ **No. =** ____ **No. >=** ____

Variable Name																					Constraint Type (<=, =, >=)	Constraint RHS Value
Constraint Name																						
Objective Function Coeff																						
Upper Bound																						
Lower Bound																						

NOTE: Constraints should be entered in the following order <=, =, >=. Constraint RHS values should be >= 0.

Maximize or Minimize ______ No. Variables ______

Problem Name: ______ No. Constraints ______ No. <= ______ No. = ______ No. >= ______

Variable Name																					Constraint Type (<=, =, >=)	Constraint RHS Value
Constraint Name																						
Objective Function Coeff																						
Upper Bound																						
Lower Bound																						

NOTE: Constraints should be entered in the following order <=, =, >=. Constraint RHS values should be >= 0.

HEWLETT PACKARD

Personal Computer Division
1010 N.E. Circle Blvd., Corvallis, OR 97330

Reorder Number
00085-90623

Printed in U.S.A.
00085-90835