LPLAB - A COMPUTER-ASSISTED INSTRUCTION PROGRAM
FOR SELECTED LINEAR PROGRAMMING METHODS

by

William Cotter Lovejoy III

A Thesis Submitted to the Faculty of the
DEPARTMENT OF SYSTEMS AND INDUSTRIAL ENGINEERING
In Partial Fulfillment of the Requirements
For the Degree of
MASTER OF SCIENCE
WITH A MAJOR IN SYSTEMS ENGINEERING
In the Graduate College
THE UNIVERSITY OF ARIZONA

1972
STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: William

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

ROBERT L. BAKER
Associate Professor of Systems and Industrial Engineering
ACKNOWLEDGMENTS

The author wishes to thank Dr. Robert L. Baker for his guidance, assistance, and patience throughout this work and Dr. A. Wayne Wymore and Dr. Duane L. Dietrich for their guidance and support in many areas during the past twenty months. He also wishes to thank Captain Russell Mayers for permission to use his SIMPLEX algorithm as a portion of the control program of LPLAB.
<table>
<thead>
<tr>
<th>TABLE OF CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
</tr>
<tr>
<td>CHAPTER</td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
</tr>
<tr>
<td>2 COMPUTER-ASSISTED INSTRUCTION</td>
</tr>
<tr>
<td>3 THE LINEAR PROGRAMMING LABORATORY</td>
</tr>
<tr>
<td>4 TYPICAL LPLAB-USER DIALOGUE</td>
</tr>
<tr>
<td>5 DETAILED USER INSTRUCTIONS</td>
</tr>
<tr>
<td>Preliminary Preparation</td>
</tr>
<tr>
<td>Activation and Deactivation of the Laboratory</td>
</tr>
<tr>
<td>Control of LPLAB</td>
</tr>
<tr>
<td>Use of the SIMPLEX Algorithm Section of the LPLAB</td>
</tr>
<tr>
<td>Use of the Transportation Algorithm Section of the LPLAB</td>
</tr>
<tr>
<td>6 RECOMMENDATIONS AND CONCLUSIONS</td>
</tr>
<tr>
<td>APPENDIX A: LPLAB CONTROL PROGRAM</td>
</tr>
<tr>
<td>APPENDIX B: LPLAB TEXT FILES</td>
</tr>
<tr>
<td>SELECTED BIBLIOGRAPHY</td>
</tr>
</tbody>
</table>
ABSTRACT

This thesis presents the development of a computer-assisted instruction program called the Linear Programming Laboratory (LPLAB). LPLAB uses a time-sharing computer system such that the user may interact with the laboratory to drill, exercise, and experiment with selected linear programming topics. The user communicates with the computer using a remote teletype terminal. The user does not need knowledge of any programming language to effectively use the laboratory. The laboratory is designed such that it can be useful for a beginning student, a student enrolled in a formal linear programming course of instruction, or a student desiring to refresh his linear programming skills. The laboratory has considerable flexibility to allow each category of user to pursue his particular goals.

The thesis contains a review of selected efforts in computer-assisted instruction, the purpose and organization of the laboratory, a representative LPLAB-user dialogue, and detailed user instructions. The complete FORTRAN IV program and its files are included.
CHAPTER 1

INTRODUCTION

This thesis reports the development of a computer-assisted instruction program called the Linear Programming Laboratory (LPLAB). The LPLAB is designed to provide the user with a computer-controlled environment for drill, experimentation, self-examination, and review of selected linear programming methods. Six areas of linear programming are reviewed in detail: general linear programming, graphical programming, the SIMPLEX algorithm, sensitivity or post-optimality analysis, the transportation problem algorithm, and the optimal assignment problem algorithm. Three of these areas are implemented for computer solution and student experimentation: the SIMPLEX algorithm, the transportation problem algorithm, and sensitivity analysis. There is room provided for a review section and an experimentation section concerning the Dual SIMPLEX algorithm which is scheduled for inclusion into the LPLAB in Fall 1972 as an honors thesis. The LPLAB is programmed in FORTRAN IV and has been implemented on the Control Data Corporation (CDC) 6400 computer using the CDC INTERCOM 2 time-sharing system at The University of Arizona, Tucson.

This thesis contains six chapters: Introduction, Computer-assisted Instruction, The Linear Programming Laboratory, Typical LPLAB-user Dialogue, Detailed User Instructions, and Recommendations and
Conclusions. There are two appendices: LPLAB Control Program and LPLAB Text Files.

Chapter 2 is a discussion of certain computer-based instruction techniques upon which the LPLAB is patterned. Chapter 3 explains what the LPLAB is and how it is generally used. Chapter 4 presents a typical dialogue between the laboratory and a user. Chapter 5 is a complete description of how to use the LPLAB, written in such a manner that it may be extracted from this thesis and employed as a user's manual. Chapter 6 is a summary of the thesis and recommendations for extensions of the laboratory.

Appendix A is the control program in the FORTRAN IV programming language. Appendix B provides the three files of information from which the drill and examination dialogue is extracted by the control program.
CHAPTER 2

COMPUTER-ASSISTED INSTRUCTION

The computer is well-known for its scientific and business applications, whereby the user accomplishes in seconds that which would manually require man-years. In the past few years, however, educators have begun to realize the computer can release teachers from tedious classroom chores so they can spend more time analyzing student progress and providing individual attention to students.

Bitzer and Johnson (1971, p. 961) state that the present status of education and the needs it must meet make it mandatory that educators soon find means with which to increase the quantity, quality, and scope of their product. They feel that those who have turned to the high speed computer "recognize the unique value of the computer-based system in adapting the selection and presentation of instructional material to the pace and style of the individual" student.

Suppes, Jerman, and Brian (1968) feel that a "revolution" has already begun in the use of computer-based devices in the classroom. They devised a system using teletype terminals located in fourth, fifth, and sixth grade classrooms in Stanford, California, connected to a central computer at Stanford University. They were attempting to demonstrate that "the use of computer-based instructional devices does offer the real promise of a technique that can be used in ordinary schools to meet the problems of individual difference at a much deeper
level and in a more scientific way than has yet been possible" (p. 3). Their tests were spread over the period 1964-1968 and from them they concluded that "a single computer can handle several hundred students simultaneously, and yet give each student the impression that he is receiving individualized instruction" (p. 4). These efforts were primarily in the field of elementary mathematics where student response could easily be measured as to correctness and time required to attain an answer. The long-term progress of a student was provided to the teacher in such a format that progress of the student and his class might be analyzed against various statistical norms.

In follow-up control group experiments, Suppes and Morningstar (1969) concluded that students taught by using their computer-based devices generally learned at a faster rate, enjoyed the learning process, and desired that the program be extended to other areas.

Iverson (1968, p. 15) states that there are three aspects of computer-assisted instruction: (1) exercise and experimentation, (2) examination, and (3) drill. "Exercises can be used simply to enhance facility in some process already well-understood, as in drill exercises in multiplication. However, when used to elicit or elucidate new ideas, assigned exercises are essentially guided experimentation—a well-designed set of exercises guides the student through a sequence of experiments."

Iverson's examinations were typically multiple choice or true-false requiring single answer responses. He stated that "giving the student a terminal to use in an examination makes it practicable to
pose more concrete problems" (p. 42). Later systems are not so restrictive as is described later in this chapter.

Iverson's main contention was that the computer can remove the repetitive, noncreative work from the teacher.

In administering drill, the computer has two important advantages—it is tireless and virtually infallible. The complexity of drill can range from simple checking and correction of responses, through the compilation of statistics on the timing and correctness of responses, to the use of such statistics to diagnose conceptual difficulties indicated by the responses (p. 44).

Among the most versatile and sophisticated of the computer-based systems is PLATO (Bitzer and Johnson 1971) developed at the University of Illinois. The PLATO system is implemented on a CDC 6400 series computer which controls cathode ray tube displays with full alpha-numeric and graphical capabilities. In its current version, it makes full use of movie projectors, audio tapes, micro-film, and is designed to be compatible with nearly any display device desired by the user. It is designed such that the author of a teaching routine need not know a programming language, but the "sophisticated software system permits an experienced author to develop extremely complex teaching strategies and lessons" (p. 961).

To illustrate PLATO's unique capabilities, Bitzer and Johnson describe a test where students were examined on their ability to prove mathematical theorems. Each theorem could not be proven by the program authors in the required number of steps. The program was not set up to compare sets of proper proofs against student input but to determine the mathematical correctness of each step presented. One precocious
student did solve the long theorem in exactly twelve steps. Using other computer-based scoring methods he would have been marked wrong, but the PLATO system graded him correctly.

Use of the PLATO system, now in its fourth generation, has shown that "typically, students acquire the same level of achievement in one-third to one-half the time required in the ordinary classroom" (Bitzer and Johnson 1971, p. 964).

The current problem with PLATO and other systems is cost. Even though a single computer with 64 to 128 thousand storage capacity can easily handle 4,000 students, the cost per student-hour is more than the 30 to 50 cents per student-hour needed to make the system practical.
CHAPTER 3

THE LINEAR PROGRAMMING LABORATORY

The Linear Programming Laboratory (LPLAB) was designed to allow the user to gain a more thorough understanding of Linear Programming through drill, examination, and experimentation. The program was written in the FORTRAN IV programming language for the CDC 6400 computer at The University of Arizona. The system was implemented using a remote teletype console connected to the computer via the CDC INTERCOM 2 timesharing system.

Chapter 5 contains a complete description of the program and instructions on how to use it. The student sits at a remote terminal and gains access to the computer and to the LPLAB by following the procedure detailed in the User's Manual. The computer communicates with the student by printing English messages and text material on the student's terminal. The student responds by typing single-word or number replies on the keyboard and depressing the return key which then causes the transmission of the reply to the computer.

The LPLAB is divided into eleven sections: (1) User's introductory material, (2) Review of General Linear Programming, (3) Graphical Programming Examples, (4) Review of the SIMPLEX Algorithm, (5) Exercising the SIMPLEX Algorithm to include Sensitivity Analysis, (6) Review of the Dual SIMPLEX Algorithm, (7) Exercising the Dual SIMPLEX

Sections 1 and 3 of the LPLAB are a frame-by-frame presentation of material that aid the user in other portions of LPLAB. Sections 2, 4, 6, 8, 9, and 11 permit the user to drill in the disciplines covered and to take periodic examinations to test his proficiency in those areas. Sections 5, 7, and 10 permit the user to experiment in the areas covered and to drill in the iterative solution technique of the SIMPLEX Algorithm.

The LPLAB has met the three goals for which it was designed. Those goals are (1) to provide a comprehensive review of linear programming and several of its extensions for a user who is several semesters or years removed from formal instruction in the subjects, (2) to allow a user to experiment with and solve actual linear programming problems, and (3) to provide a computer-controlled extension to academic courses currently being taught within the curriculum of the Department of Systems and Industrial Engineering.

The SIMPLEX applied portion of the LPLAB allows the user to enter a linear programming problem and solve it. The program permits the user to check the solution iteration-by-iteration or to go directly to the final optimum solution. After the problem is solved, the user may perform a sensitivity analysis on the solved problem to determine what effect the varying of any portion of the problem will have on the
final solution. This particular technique is most valuable in solving practical linear programming problems.

The Transportation Problem Applied section of the LPLAB presents only the final solution of an entered problem. If a user desires to accomplish sensitivity analysis on the transportation problem, he must convert the problem to one solvable by the SIMPLEX Algorithm and use the SIMPLEX Algorithm applied portion of the laboratory. The transformation of the transportation problem to a standard linear programming problem is discussed in detail in the Transportation Problem Review Section of the LPLAB.

The LPLAB has reserved space for sections containing a review of the Dual SIMPLEX Algorithm and application of the algorithm to problems input by the user. These are scheduled for insertion into the LPLAB during the Fall 1972 session as part of an honors thesis.
CHAPTER 4

TYPICAL LPLAB-USER DIALOGUE

This chapter presents a typical dialogue between the laboratory and a user. User responses are underlined.

Initially the laboratory types an introductory paragraph summarizing the options available.

Sections 1 and 2 illustrate the text presentation capabilities of the LPLAB. Section 1 is a frame-by-frame presentation of text concerning general use of the laboratory. Section 2 illustrates the administration of a five-question, true-false examination (the user may elect to skip the examination), the laboratory's response to an illegal entry by the user, and how the user may skip portions of text he does not desire to investigate. Section 2 is terminated using the SKIP command.

Section 5 permits the user to experiment with the SIMPLEX algorithm and perform sensitivity or post-optimality analysis on the resultant solution. A simple problem is entered, solved, and sensitivity analysis is performed on the final solution.

Section 10 permits the user to enter and solve a Transportation problem. The illustrative problem demonstrates the laboratory's capability to add a dummy source when required.

The following sample dialogue illustrates only the major capabilities of the LPLAB. The complete text is contained in Appendix B.
WELCOME TO LP LAB.

THIS IS A COMPUTER ASSISTED LABORATORY IN LINEAR PROGRAMMING. THERE ARE ELEVEN TOPICS YOU MAY SELECT.

WHEN RESPONSES ARE REQUESTED, TYPE THEM EXACTLY AS ASKED FOR.

THE TOPICS ARE AS FOLLOWS:
1. INTRODUCTORY EXPLANATION OF LP LAB.
2. LINEAR PROGRAMMING GENERAL.
3. GRAPHICAL PROGRAMMING EXAMPLES.
4. REVIEW OF SIMPLEX ALGORITHM.
5. SIMPLEX ALGORITHM/SENSITIVITY ANALYSIS APPLIED.
6. REVIEW OF DUAL SIMPLEX ALGORITHM.
7. DUAL SIMPLEX APPLIED.
8. REVIEW OF TRANSPORTATION PROBLEM ALGORITHM.
9. SENSITIVITY ANALYSIS REVIEW.
10. TRANSPORTATION ALGORITHM APPLIED.
11. OPTIMAL ASSIGNMENT PROBLEM.
12. TERMINATE THE LABORATORY.

NOW CHOOSE A MAIN AREA TO INVESTIGATE.
TYPE (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, OR 12)

THIS IS AN INTRODUCTORY EXPLANATION OF THIS LAB.

THE FOLLOWING GENERAL COMMENTS APPLY TO YOUR USE
OF LP LAB.

1. ALWAYS BEGIN TYPING YOUR RESPONSE FROM THE
POSITION WHERE THE CARRIAGE HAS STopped.

2. TO CORRECT AN ERROR YOU CAN BACK SPACE USING
UPPER CASE O AND THEN UPTYPE OVER THE ERROR
OR YOU CAN VOID AN ENTIRE LINE BY DEPRESSING THE
CTRL KEY AND THE X SIMULTANEOUSLY. THIS WILL MOVE
THE CARRIAGE TO THE LEFT MARGIN OF A NEW LINE.
YOU WILL BEGIN TYPING AT THAT POSITION.

3. YOU MUST TYPE ALL RESPONSES EXACTLY AS SHOWN
WHEN SELECTING A REPLY FROM A LIST OF OPTIONS.
TYPE (GO OR SKIP).

4. YOU WILL BE USING TWO TYPES OF NUMBERS (INTEGRAL AND
DECIMAL). INTEGERS MAY BE ONE OR TWO DIGITS AND MUST NOT
HAVE A DECIMAL POINT. IF A TWO DIGIT INTEGRAL IS REQUESTED,
IT MUST BE ENTERED IN THE FORM (03). DECIMAL NUMBERS
CAN BE EITHER STRAIGHT DECIMAL OR F FORMAT.
FOR EXAMPLE: +136.470 IS EQUIVALENT TO +1.36475E+02.
The STRAIGHT DECIMAL NUMBER MUST CONTAIN A DECIMAL
POINT AND MAY CONTAIN UP TO 11 CHARACTERS
COUNTING THE • AND THE SIGN IF SIGN OMITTED.
NUMBERS MUST HAVE EXACTLY THE FORMAT SHOWN ABOVE.
THE SIGN MUST BE PRESENT.
TYPE (GO OR SKIP).

5. AT ANY TIME DURING PROBLEM SOLVING YOU CAN
TERMINATE LP LAB BY TYPING STOP WHENEVER LP LAB
IS EXPECTING A WORD COMMAND. LP LAB WILL NOT RECOGNIZE
STOP IF IT IS EXPECTING A NUMBER.

6. LP LAB CAN BE ABORTED ANY TIME BY DEPRESSING THE
CTRL KEY AND THE Z SIMULTANEOUSLY AND THEN THE A.
7. IF LP LAW IS INACTIVE FOR MORE THAN ABOUT 90 SEC.
EXCEPT WHEN WAITING FOR YOUR RESPONSE IT MAY BE MALFUNCTIONING. YOU MAY ABORT IF YOU DESIRE.
BUT PRIOR TO ABORTING, YOU CAN TRY ONCE TO CORRECT THE PROBLEM BY TYPING GO.

NOW CHOOSE A MAIN AREA TO INVESTIGATE.
TYPE (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, OR 12) 2

THIS CONTAINS COMMENTS ON GENERAL LINEAR PROGRAMMING.

LINEAR PROGRAMMING (LP) DEALS WITH THE PROBLEM OF ALLOCATING LIMITED RESOURCES AMONG COMPETING ACTIVITIES IN AN OPTIMAL MANNER. ALL MATHEMATICAL FUNCTIONS IN LP MUST BE LINEAR FUNCTIONS. WOULD YOU LIKE TO TRY A TEST TO SEE IF YOU UNDERSTAND LINEAR AND NON-LINEAR FUNCTIONS?
TYPE (YES, NO, OR SKIP)

YES.
AFTER EACH OF THE FOLLOWING, TYPE T FOR TRUE IF IT IS A LINEAR FUNCTION, F FOR FALSE IF IT IS NOT.
X1 + X2 + X3 = 14 TYPE (T OR F) T
CORRECT

(X1)CURED + X3 = 7 TYPE (T OR F) F
INCORRECT

SQUARE ROOT(X1) + X2 = 7 TYPE (T OR F) T
INCORRECT

X1 + X2 + X3 = (? SQUARED TYPE (T OR F) T
CORRECT

14X1 + 22.3X3 - 72.39X4 <= 4.123 TYPE (T OR F) T
CORRECT

YOU MISSED 1. PRETTY GOOD.

THE MATHEMATICAL STATEMENT OF A GENERAL LP PROBLEM IS AS FOLLOWS:
FIND VALUES OF X1, X2, ..., XN WHICH MAXIMIZE (OR MINIMIZE) THE LINEAR FUNCTION Z = C1X1 + C2X2 + ... + CNXN
SUBJECT TO THE RESTRICTIONS:
A11X1 + A12X2 + ... + A1NXN = B1
A21X1 + A22X2 + ... + A2NXN <= B2
.  .  .  .
AM1X1 + AM2X2 + ... + AMNXN = BM
TYPE (GO OR SKIP)

RE-ENTER YOUR ANSWER.
TYPE (GO OR SKIP)

WHERE A11, B1, AND CJ ARE GIVEN CONSTANTS, ALL XI=0, AND THE RELATIONAL OPERATORS IN THE CONSTRAINT EQUATIONS MAY BE >=, =<, OR =. THE FUNCTION Z THAT IS TO BE OPTIMIZED IS THE OBJECTIVE FUNCTION. WOULD YOU LIKE TO SEE THE
FORMULATION OF A SIMPLE LP PROBLEM? TYPE (YES, NO, OR SKIP)

YES

THE DIET PROBLEM

OBJECTIVE: DETERMINE THE LEAST COST MIX OF MILK, EGGS, AND CHEESE TO GIVE THE MINIMUM NUTRITIONAL DAILY REQUIREMENTS OF VITAMINS A, C, AND D USING THE FOLLOWING DATA:

<table>
<thead>
<tr>
<th>Vitamin</th>
<th>Milk (Gal)</th>
<th>Cheese (Lb)</th>
<th>Eggs (Doz)</th>
<th>Min Daily Rqmt</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>100</td>
<td>10</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>D</td>
<td>10</td>
<td>100</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

COST $1.00  $1.10  $0.50

TYPE (GO OR SKIP)

GO

LET X1 BE THE NUMBER OF GALLONS OF MILK, X2 BE THE NUMBER OF POUNDS OF CHEESE, AND X3 BE THE NUMBER OF DOZENS OF EGGS IN THE DAILY DIET. THE OBJECTIVE IS TO MINIMIZE COST. ATTEMPT TO SET UP THE OBJECTIVE FUNCTION AND THE CONSTRAINTS FOR THIS PROBLEM ON A SEPARATE SHEET OF PAPER. THEN YOU CAN COMPARE YOUR RESULTS WITH THE CORRECT RESULTS.

TYPE (GO OR SKIP)

SKIP

NOW CHOOSE A MAIN AREA TO INVESTIGATE.

TYPE (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, OR 12)

5

THIS SECTION ALLOWS YOU TO PRACTICE THE SIMPLEX ALGORITHM AND SENSITIVITY ANALYSIS.

YOU MAY NOW SELECT A L.P. PROBLEM FOR SOLUTION BY ENTERING THE PARAMETERS AS THEY ARE REQUESTED. YOU HAVE TWO OPTIONS FOR ENTERING THE PARAMETERS. THE REGULAR METHOD IS A STEP BY STEP RESPONSE THAT WILL INSURE ACCURACY. THE EXPEDITE METHOD IS FASTER BUT REQUIRES FAMILIARITY WITH THIS PROGRAM TO BE EFFECTIVE. IT IS RECOMMENDED THAT YOU USE THE REGULAR METHOD FOR THE FIRST FEW PROBLEMS.

SELECT THE OPTION YOU WANT. TYPE (REG OR EXP)

REG

ENTER THE COEFFICIENTS FOR THE OBJECTIVE FUNCTION VARIABLES AS THEY ARE REQUESTED BY TYPING A DECIMAL NUMBER SUCH AS 12. OR +1.2000E+01

X 1 = -4.
X 2 = -5.
X 3 = -2.
X 4 = -11.
X 5 = 0

MORE VARIABLES FOR OBJ. FUNCTION? TYPE (YES OR NO)

NO

TYPE THE COEFFICIENTS FOR THE VARIABLES IN CONSTRAINT EQUATION 1. USE A DECIMAL NUMBER.

X 1 = +1.
X 2 = +1.
X 3 = +1.
X 4 = +1.
X 5 = +0.

MORE VARIABLES FOR EQ. 1?  TYPE (YES OR NO)
ENJTFR CONSTHAI NT EQ. 1 SENSE CODE.
TYPE (1 FOR <=, 2 FOR =, 3 FOR >=)

1

TYPE RIGHTHAND CONSTRAINT VALUE FOR CONSTRAINT EQ. 1 USING DECIMAL NO. SUCH AS 13. OR +1.3000E+01.

MORE CONSTRAINT EQUATIONS? TYPE (YES OR NO)
YES

TYPE THE COEFFICIENTS FOR THE VARIABLES IN CONSTRAINT EQUATION 2. USE A DECIMAL NUMBER.
X 1 = 7.
X 2 = 5.
X 3 = 7.
X 4 = 2.
X 5 = 0.
MORE VARIABLES FOR EQ. 2? TYPE (YES OR NO)
NO

ENTER CONSTRAINT EQ. 2 SENSE CODE.
TYPE (1 FOR <=, 2 FOR =, 3 FOR >=)

1

TYPE RIGHTHAND CONSTRAINT VALUE FOR CONSTRAINT EQ. 2 USING DECIMAL NO. SUCH AS 13. OR +1.3000E+01.

MORE CONSTRAINT EQUATIONS? TYPE (YES OR NO)
YES

TYPE THE COEFFICIENTS FOR THE VARIABLES IN CONSTRAINT EQUATION 3. USE A DECIMAL NUMBER.
X 1 = 3.
X 2 = 5.
X 3 = 10.
X 4 = 15.
X 5 = 0.
MORE VARIABLES FOR EQ. 3? TYPE (YES OR NO)
NO

ENTER CONSTRAINT EQ. 3 SENSE CODE.
TYPE (1 FOR <=, 2 FOR =, 3 FOR >=)

1

TYPE RIGHTHAND CONSTRAINT VALUE FOR CONSTRAINT EQ. 3 USING DECIMAL NO. SUCH AS 13. OR +1.3000E+01.

MORE CONSTRAINT EQUATIONS? TYPE (YES OR NO)
NO

THE PROBLEM YOU HAVE JUST FINISHED ENTERING WILL BE CONDENSED INTO A TABLE FOR YOUR EDITING.
NOTE THAT ROW 1 IS THE OBJECTIVE FUNCTION.

****** ****** ****** EDIT TABLE ****** ****** ****** ****** ******

<table>
<thead>
<tr>
<th>ROW</th>
<th>COL 1</th>
<th>COL 2</th>
<th>COL 3</th>
<th>COL 4</th>
<th>COL 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1.0000E+00</td>
<td>-1.0000E+00</td>
<td>-1.0000E+00</td>
<td>-1.0000E+00</td>
<td>-1.0000E+00</td>
</tr>
<tr>
<td>2</td>
<td>1.0000E+00</td>
<td>1.0000E+00</td>
<td>1.0000E+00</td>
<td>1.0000E+00</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>3</td>
<td>7.0000E+00</td>
<td>7.0000E+00</td>
<td>3.0000E+00</td>
<td>2.0000E+00</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3.0000E+00</td>
<td>5.0000E+00</td>
<td>1.0000E+01</td>
<td>1.5000E+01</td>
<td></td>
</tr>
</tbody>
</table>

CONSTRAINT EQ. SENSE CODE RIGHTHAND SIDE
1  1   1.5000E+01
2  1   1.2000E+02
3  1   1.0000E+02
Any corrections to above numbers?  Type (Yes or No)  

No

Do you want the next or final iteration?  
Type (Next or Final)  

Next

Do you want the next or final iteration?  
Type (Next or Final)  

Final

Optimal solution---Objective function = 9.9286E+01

If you want to stop, type Stop.  
If you want to continue with a new problem, type Go.  
If you want to alter the original problem, type Cur.  
If you want to alter the final iteration, type Iter.  
If you want to check sensitivity, type Sen.  

You may select three cases for sensitivity analysis:  
1. Effects of varying the coefficients in the objective function.  
2. Effects of varying the constraint constants (right-hand side value).  
3. Results of selected discrete changes in any of the variables and/or constants, in any combination.

Select the sensitivity analysis case you want.  
Type (1, 2, or 3)  

Select the objective function variable you want to investigate.  Type a two digit integer such as 03 to identify the variable.
VAR = 02
YOU SELECTED X 2 FOR ANALYSIS.
IS THIS WHAT YOU WANTED? TYPE YES OR NO
YES
ANALYSIS-------"NORMAL VARIABLE" X 2 MAY
RANGE FROM ITS ORIGINAL OBJECTIVE FUNCTION VALUE
OF -5.0000E+00 TO -5.4266E+00 AND STILL
HAVE THE CURRENT SOLUTION OPTIMAL AT 9.9266E+01.

DO YOU WANT TO TRY ANOTHER OBJECTIVE FUNCTION VARIABLE?
TYPE YES OR NO
NO
DO YOU WANT TO CONTINUE WITH SENSITIVITY ANALYSIS?
TYPE YES OR NO
SKIP
NOW CHOOSE A MAIN AREA TO INVESTIGATE.
TYPE (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, OR 12)
10

YOU MAY NOW SELECT A TRANSPORTATION PROBLEM FOR SOLUTION BY
ENTERING PARAMETERS AS THEY ARE REQUESTED. THERE ARE TWO
OPTIONS FOR ENTERING PARAMETERS. THE REGULAR METHOD USES A
STEP BY STEP PROCEDURE THAT INSURES ACCURACY. THE EXPEDITE
METHOD IS FASTER, BUT SHOULD BE USED AFTER YOU ATTAIN FAMILIARITY
WITH THIS PROGRAM.
SELECT THE OPTION YOU DESIRE.
TYPE (REG OR EXP)
REG
ENTER THE NUMBER OF SOURCES AND NUMBER OF DESTINATIONS IN
11 FORMAT; THAT IS, 1, 2, ETC.
THE MAXIMUM NUMBER OF EITHER IS 6.
NUMBER OF SOURCES = 3
NUMBER OF DESTINATIONS = 5

ENTER THE COST OF GOING FROM SOURCE I TO DESTINATION J, C(I,J),
AS THEY ARE REQUESTED. USE FORMAT E11.4 WHICH IS OF THE FORM
12.5 OR 723.47 OR +1.2000E+01. IF YOU USE THE LATTER, IT MUST
BE EXACTLY AS SHOWN.

C(1, 1) = 20.
C(1, 2) = 19.
C(1, 3) = 14.
C(1, 4) = 21.
C(1, 5) = 16.
C(2, 1) = 15.
C(2, 2) = 20.
C(2, 3) = 13.
C(2, 4) = 19.
C(2, 5) = 16.
C(3, 1) = 18.
C(3, 2) = 15.
C(3, 3) = 15.
C(3, 4) = 20.
C(3, 5) = 4000.

WHEN REQUESTED, ENTER THE AMOUNT AVAILABLE AT EACH SOURCE IN
E11.4 FORMAT.

SOURCE 1 = 40.
SOURCE 2 = 60.
SOURCE 3 = 90.

WHEN REQUESTED, ENTER THE AMOUNT REQUIRED AT EACH DESTINATION
IN E11.4 FORMAT.
DESTINATION 1 = 30
DESTINATION 2 = 40
DESTINATION 3 = 70
DESTINATION 4 = 40
DESTINATION 5 = 60

ALL DATA WILL NOW BE REPRODUCED IN TABLE FORM.

<table>
<thead>
<tr>
<th>SOURCE</th>
<th>DESTINATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1 2.0000E+01</td>
<td>1.9000E+01</td>
</tr>
<tr>
<td>2 1.5000E+01</td>
<td>2.0000E+01</td>
</tr>
<tr>
<td>3 1.8000E+01</td>
<td>1.5000E+01</td>
</tr>
<tr>
<td>4 1.8000E+01</td>
<td>1.8000E+01</td>
</tr>
<tr>
<td>5 1.8000E+01</td>
<td>2.0000E+01</td>
</tr>
<tr>
<td>6 1.8000E+01</td>
<td>4.0000E+03</td>
</tr>
</tbody>
</table>

DO YOU DESIRE TO CHANGE ANY OF THE ABOVE?
TYPE (YES OR NO)
NO

THE OPTIMAL SOLUTION IS:
MINIMUM COST IS $2.9400E+03
THE ALLOCATIONS ARE:

X(1, 5) = 4.00000E+01
X(2, 3) = 6.00000E+01
X(3, 1) = 3.00000E+01
X(3, 2) = 4.00000E+01
X(3, 3) = 1.00000E+01
X(3, 4) = 1.00000E+01
X(4, 4) = 3.00000E+01
X(4, 5) = 2.00000E+01

ALL THE OTHER PATHS HAVE ZERO ALLOCATIONS.

DO YOU WISH TO ATTEMPT ANOTHER TRANSPORTATION PROBLEM?
TYPE (YES OR NO)
STOP
YOU HAVE INDICATED YOU DESIRE TO TERMINATE THE LAB.
IS THIS CORRECT?
TYPE (YES OR NO)
YES

LPLAB WILL NOW TERMINATE.
AFTER LPLAB TERMINATES, THE TERMINAL WILL TYPE COMMAND.
YOU MUST THEN ENTER LOGOUT AND PRESS THE RETURN KEY.
THANK YOU.
CHAPTER 5

DETAILED USER INSTRUCTIONS

This chapter may be used as a detailed instructional guide for a Computer-Assisted Instruction Laboratory in Linear Programming (LPLAB). The control program for LPLAB is written in the FORTRAN IV programming language (RUN OPTION) for use on the CDC 6400 at The University of Arizona, Tucson. LPLAB is an interactive laboratory using remote teletype terminals and the time-sharing system of the CDC 6400, INTERCOM 2.

Preliminary Preparation

The laboratory allows operations research students to review linear programming in six areas and to experiment with the SIMPLEX and Transportation Problem algorithms. Prior to playing LPLAB, the student should familiarize himself with the problem size and organization restrictions described in the SIMPLEX algorithm section of this chapter.

The LPLAB program is available through the Systems and Industrial Engineering Office. To use LPLAB, permission must be obtained from The University Computer Center (UCC) regarding the remote teletype terminals and INTERCOM 2. Procedures for using the terminals (telephone number, billing number, terminal identification, etc.) will be provided by the Computer Science or Systems and Industrial Engineering Office and the UCC. The user must ensure that LPLAB and its three
associated files have been catalogued as permanent files on the CDC 6400 before attempting to use the remote terminal.

*Activation and Deactivation of the Laboratory*

Communication between the remote teletype terminal and the central computer must be established before the laboratory can become operational. Similarly, the communication link must be deactivated when the student is finished using the laboratory and the remote terminal. An example is given at the end of the chapter showing the actual dialogue for complete activation and deactivation of the laboratory:

1. Depress the ORIG button, which is located at the lower right of the console below the telephone dial. The button should illuminate and a dial tone will become audible. (If the volume is turned down, the dial tone may not be audible; however, proceed in either case.)

2. Using the telephone dial above the ORIG button, dial the telephone number provided. If the teletype starts printing, then hard line communication has been established. If a busy signal is audible, then someone is already using that telephone number and another number must be obtained. To try another number, depress the CLR button and then the ORIG button again.

3. When hard line communication has been established, INTERCOM 2 will print the following message to the user:

```
CONTROL DATA INTERCOM
DATE 10/15/71
TIME 17.22.01.
```
In practice, the current date and time will be printed instead of the values shown. At this point, the computer is waiting for the user to "log-in." The user should type the following entry and then depress the RETURN key:

LOGIN.

(An entry by the user is not sent to the computer until the RETURN key is depressed. This is also true when using the laboratory itself.) If anything other than the exact six characters shown above are input as the entry, INTERCOM 2 will respond by typing the following message:

THE MYSTERY GUEST WILL PLEASE LOGIN.

The user should then enter the log-in response correctly.

At this point, it is worthwhile to discuss the correction of entries that are detected before the RETURN key is depressed. One method of correcting entries is to backspace over the incorrect character(s) and then retype. Backspacing is accomplished by typing ←, which is entered with the SHIFT and alpha 0 keys. LPG ← OGIN, is sent to the computer as LOGIN, when the RETURN key is depressed. Another method of correcting entries is to "erase" the whole entry and type again. To erase an entry type the letter X while depressing the CTRL key. The carriage will be returned to the left margin, and the paper will be advanced one line, but nothing will be sent to the computer. The correct response can then be entered.

4. Next INTERCOM will respond with:

ENTER 3 CHARACTER USER NAME

and will wait for the response. The user can use any three
alpha-numeric characters as long as the first character is alphabetic.

5. INTERCOM will then request the billing number by typing:

   TYPE VALID BILLING NUMBER

and will wait for the response. If the billing number is not valid, INTERCOM will respond with:

   INVALID BILLING NUMBER.

and the billing number will be requested again.

6. After the billing number is accepted, INTERCOM will request the terminal identification by typing:

   ENTER TERMINAL ID

and the student should respond with the two-character terminal identification that is to be found printed on the insert at the upper right of the console.

7. This concludes the log-in procedure and INTERCOM will respond with:

   10/15/71 17.23.31. LOGGED IN WITH JOBNAME XXXXXXX.
   RESOURCES ALLOWED: FL=XXXXXX, FL=XXXX, FILES-XX, ACCESS
   LEVEL=X. COMMAND-

Actual letters and numbers will be printed where the X's are shown.

The first three characters of the jobname will be the user name entered in step 4. The command message is used by INTERCOM 2 to request direction from the user. The commands necessary to execute the control program of the laboratory will be given in the next step.

8. Eight commands are necessary before LPLAB can be used by the student. They are concerned with:
a. Attaching the four permanent files that contain LPLAB.

b. Setting the time limit.

c. Setting the field length to the maximum value permitted.

d. Insuring the maximum field length remains available.

e. Executing the laboratory control program, LPLAB.

These commands, given in a group with the proper responses underlined, are as follows:

```
COMMAND- ATTACH,LPLAB,LPLAB.
10.26.59.ATTACH,LPLAB,LPLAB.
COMMAND- ATTACH,LP3,LP3.
10.27.15.ATTACH,LP3,LP3.
COMMAND- ATTACH,LP4,LP4.
10.27.29.ATTACH,LP4,LP4.
COMMAND- ATTACH,LP5,LP5.
10.28.19.ATTACH,LP5,LP5.
COMMAND- ETL,60.
COMMAND- EFL,60000.
COMMAND- REDUCE,OFF.
COMMAND- LPLAB.
```

After each ATTACH command is entered, the above messages preceded with the appropriate time are printed. These indicate that the files have been successfully found and attached. If LPLAB, LP3, LP4, and LP5 have not been catalogued as permanent files, the following message would appear after each attempt to attach:

```
**ILLEGAL P.F. REQ
```

This message may also appear if a file name is misspelled. The ATTACH command should then be entered again with the correct spelling.

After entering the eighth command shown above, the student will be under the control of the laboratory.
9. At the conclusion of the laboratory session, the user must deactivate the terminal. After the user has definitely established his desire to terminate with LPLAB, the terminal will type:

LPLAB WILL NOW TERMINATE.
AFTER LPLAB TERMINATES, THE TERMINAL WILL TYPE COMMAND—
YOU MUST THEN ENTER LOGOUT. AND PRESS THE RETURN KEY.
THANK YOU.

19.10.06.STOP
COMMAND—
The user should then type the following entry and depress the RETURN key:

LOGOUT.

After the teletype finishes printing, the user should depress the CLR button. If the user wishes to use the teletype as a typewriter for comments or to critique LPLAB, he need only depress the LCL button and then the CLR button when finished.

Control of LPLAB

This section contains the necessary information needed by the user to control LPLAB. The procedures given here apply after all commands listed in step 8 of the preceding section have been accomplished and the LPLAB program is in control.

LPLAB is completely automatic in its presentation and will progress following periodic commands from the user. LPLAB clearly indicates when it expects a command or input from the user and specifies on each occasion the options from which the user must choose. The command or input that the user must enter is a single word or number that
must be typed on the teletype exactly as requested and transmitted to LPLAB by pressing the RETURN key. The commands and inputs follow naturally from the presentation and require no memorization, no special techniques, and no computer programming knowledge or experience by the user.

All commands and inputs must meet certain general requirements that are listed below:

1. All response (commands or inputs) by the user must be typed on the teletype beginning from wherever LPLAB has positioned the carriage. There are two possible exceptions to this rule:
   a. LPLAB may direct the user to reposition the carriage prior to an input, and
   b. the user may elect to correct a typographical error by voiding the line (see step 3 of the preceding section).

2. All word responses must be identical with the option selected from the list of alternates. No periods or other punctuation marks are allowed.

3. Number responses are of two types: (a) decimal numbers and (b) integer numbers. A decimal number must be represented as either a straight decimal number or as a coded scientific notation number. A straight decimal number may be up to 11 characters including the sign and the decimal point. The sign is optional (a number with no sign is assumed to be positive) but the decimal point is required. Examples of allowable straight decimal numbers are:
   7.;+5;-.00068; 3.0; -17.5769542; 17.5769428
Examples of numbers not allowed as straight decimal numbers are:

7 (no decimal point); +17,57695428 (too many characters)

A coded scientific notation decimal (E format) is illustrated by the following examples of allowable numbers and their values:

+3.0000E+00 = 3,0000; -3.0000E-02 = .30000;
+1.2345E+03 = 1234.5
+1.0000E+10 = 10000000000.

The E format decimal number must contain 11 characters. A sign must precede the number and a sign must follow the E. The E must be present and must occupy the 8th position. The decimal point must be present and must occupy the 3rd position. The number following the E and its sign must be a two-digit integer. Examples of invalid E format numbers are:

1.2345E+00 (no sign before the number)
+1.234E+01 (E out of position)
-1.2345E02 (no sign after the E)

The E format decimal number should be used only when the number to be entered is too large to be represented as a straight decimal number. LPLAB automatically converts all decimal numbers to the E format before printing them.

LPLAB will always accept the command STOP whenever it expects a word reply even though STOP may not be listed as one of the options available at that point. The command STOP will cause LPLAB to deactivate itself and return control to INTERCOM (see step 9 of the previous section for terminating INTERCOM). LPLAB will not recognize STOP as a valid response when it is expecting a number response.
The user can abort at any time and force LPLAB into deactivation by depressing the CTRL key and Z simultaneously, then A. This will return control to INTERCOM. This abort technique should be used only for cases where there seems to be a malfunction which would be indicated by LPLAB responding erratically or not responding at all. If LPLAB becomes inactive for more than about 90 seconds, except when waiting for a user response, the user should assume there has been a malfunction. Prior to aborting, the user may elect to try a remedial technique by typing the command GO. If this fails, the user must abort. Once control has been transferred to INTERCOM, LPLAB may be restarted by typing LPLAB, and pressing the RETURN key.

Normal termination of LPLAB at the desire of the user is done by typing the command STOP and then following the directions given with the final response by the user being "LOGOUT."

**Use of the SIMPLEX Algorithm**

**Section of the LPLAB**

Before entering the parameters of the problem, the user must ensure that the objective function of the problem is expressed as a maximization problem.

For example, if the objective function is

\[
\text{maximize } 4X_1 + 2X_2
\]

rewrite it as

\[
X_0 - 4X_1 - 2X_2 = 0
\]

The objective function that must be entered is then

\[
-4X_1 - 2X_2
\]
The remainder of the equation is not used; it serves only to help the user visualize the relationships. In short, the objective function to be entered must be the negative of the maximization function. As another example, if the problem is to maximize $3X_1 + 4X_2 + 2X_3$, the user must input this as

$$-3X_1 - 4X_2 - 2X_3$$

The constraints can be entered directly as they occur in the statement of the problem. LPLAB will accept $\leq$, $=$, or $\geq$ constraint. Character limitations on the teletype cause it to refer to $\leq$ as $<=$ and to $\geq$ as $=>$. The user must not add surplus, slack, or artificial variables to the constraints. This is done automatically.

After all of the parameters have been entered, an edit table is printed containing all of the parameters that have been entered. The user must check the numbers for accuracy. The user may correct any of the numbers.

After the user is satisfied that the problem has been entered correctly, he directs LPLAB to solve the problem. LPLAB takes the problem as it has been defined by the user and adds slack, surplus, or artificial variables, as appropriate, and rearranges the data into a condensed SIMPLEX tableau. The original variables will be labeled $X_1$ to $X_{40}$, the slack/surplus variables will be $X_{41}$ to $X_{60}$, and the artificial variables will be $X_{61}$ to $X_{80}$. The tableau is printed and is called the first iteration of the algorithm. After the first tableau has been printed, the user has the option of having the next tableau printed showing the results of the next iteration of the algorithm or
having only the final tableau printed showing the optimal solution.
The user has the same option after each subsequent iteration.

The algorithm automatically terminates when an optimal solution has been found. At this point, LPLAB provides the user with several options: 1) he can stop, 2) he can repeat the above process by entering a new problem, 3) he can alter the current problem and cause the algorithm to be repeated, 4) he can alter the final iteration and cause the algorithm to be continued, or 5) he can apply the sensitivity analysis to the current problem. Options 2, 3, and 4 will cause LPLAB to recycle to the appropriate places so the user may make the necessary entries or changes and then cause the basic process to be repeated.

When option 5, sensitivity analysis, has been selected, LPLAB provides three choices: 1) effects of varying the coefficients in the objective function, 2) effects of varying the constraint constants, and 3) results of selected discrete changes in any variable or constant. Choices 1 and 2 of the sensitivity analysis use information from the final tableau and provide ranges over which the test variable or constant may vary before the current solution is no longer optimal. LPLAB prints the ranges of values. These two choices do not alter the final tableau; therefore, any number of the appropriate parameters may be analyzed (only one parameter at a time may be analyzed since multiple ranging analysis is beyond the capability of the laboratory). Choice 3 of the sensitivity analysis causes the initial problem to be altered and a new solution obtained. The user compares the new solution against the old solution. If choice 3 is exercised more than once, the
original problem must be restored before subsequent analysis can be made. Choice 3 is basically the process of resolving the original problem that has been modified but has an additional feature of allowing the user to have only the optimal value of the objective function and a list of the basic variables printed to show the results of the change. Otherwise, the user can select the regular methods of tableau presentation. The user may continue the sensitivity analysis for as long as he desires.

When finished with the sensitivity analysis, the user can stop, enter a new problem, modify the current problem, and repeat the process or select another portion of the laboratory.

Use of the Transportation Algorithm
Section of the LPLAB

Before using the Transportation Problem Applied section of LPLAB, the user should organize his problem. Organizing will enable him to easily enter the parameters as they are requested. A suggested method is the tableau used in the Transportation Review section of LPLAB. The following example problem is in that form:

<table>
<thead>
<tr>
<th>Source</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Supply</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>15.</td>
<td>20.</td>
<td>13.</td>
<td>19.</td>
<td>16.</td>
<td>60.</td>
</tr>
<tr>
<td>3</td>
<td>18.</td>
<td>15.</td>
<td>18.</td>
<td>20.</td>
<td>M</td>
<td>90.</td>
</tr>
<tr>
<td>Demand</td>
<td>30.</td>
<td>40.</td>
<td>70.</td>
<td>40.</td>
<td>60.</td>
<td></td>
</tr>
</tbody>
</table>
In the above example, all numbers except the designations of the sources and destinations are decimal. These decimal numbers are subject to the same restrictions discussed in the SIMPLEX Algorithm portion of this chapter. Note that the problem violates the restriction on the manual solution that the total amount supplied must equal the total amount demanded. LPLAB will automatically adjust the problem for proper solution if the sums do not equate. The quantity M in the example refers to the situation where there is no shipment from source 3 to demand 5. M is a very large artificial cost of transportation such that the algorithm will not choose it as a feasible shipping path. The user must determine and enter a specific value for M. In general, M should be chosen to be twenty times bigger than the largest shipping cost.

The user may choose one of two methods to enter the problem—Regular or Expedite. If the user chooses the regular option, LPLAB will request the number of sources and the number of destinations in the problem. These quantities must be entered in integer format (single digit with no decimal point) and each must be less than or equal to six. Then the user will be requested to individually enter the cost parameters. Then LPLAB will request the amounts required at each source and destination one at a time and the user will respond after each request. If the user chooses the expedite method of entering parameters, he must first enter the number of sources and destinations. He will then enter the cost parameters of the problem a line at a time as they are requested.
After all parameters are entered, the laboratory will print an edit table for the user to validate all parameters. After the user has approved all entries in the edit table, or has made the necessary corrections, LPLAB will solve the problem and print only the final solution.

This portion of LPLAB does not solve the problem in the manner discussed in detail in the Transportation Problem review section of the laboratory. That solution technique is designed for use with pencil and paper. The LPLAB converts the Transportation matrix of the type previously illustrated to a regular linear programming problem and solves the problem using the SIMPLEX algorithm. This transformation is explained and illustrated in the Transportation Review section of the laboratory. If the user desires to perform an analysis of the effects of varying the problem parameters, he should convert the problem to the standard SIMPLEX form where it can be solved in the SIMPLEX applied portion of the LPLAB. Then he can use the full range of post-optimality analysis options offered.
CHAPTER 6

RECOMMENDATIONS AND CONCLUSIONS

The LPLAB allows the student to review needed material at his own speed without slowing or being impeded by the presence of classmates or incurring the wrath of an impatient instructor. The student may exercise in the algorithms presented using the computer's high speed computational capability. He may vary the various parameters of problems and quickly observe the results of his experimentation. The rapid results of his experiments should give him a considerable insight into the algorithm that would be virtually impossible using the normal manual manipulative techniques.

The LPLAB could be criticized in that some of its sections do little more than present text, a function that could be accomplished at considerably less expense using programmed texts or special purpose teaching machines. However, this is not entirely true. The LPLAB allows the user to move freely into and out of each section and to choose only selected portions of certain sections. It is also true that the current INTERCOM 2 communications system is relatively slow and limits the speed at which a student may advance.

It would enhance the usefulness of the LPLAB if cathode ray tube displays could be employed in a manner similar to that used in the PLATO system discussed earlier in Chapter 2. As was stated, the CDC
6400 computer has the capabilities to use that and other types of peripheral visual display devices.

At present the LPLAB provides the teacher with only the most cursory date concerning the performance of students. The development of an imaginative set of examinations and experiments to fully exploit each section of the LPLAB and correlate the results statistically would enhance the value of the laboratory. The information would be of benefit both to an instructor as well as the student who is trying to determine his own level of development.

The LPLAB is designed to be modified with relative ease to reflect changes in emphasis or to permit addition of material concerning recent advances. It is hoped the LPLAB will be a useful tool to assist the instructor and student in assimilating appropriate material concerning linear programming.
RUN(S)
REWIND(LGO)
CATALOG(LGO,LFLA9,LD=LP3)
COPYCR(INP,LP3)
COPYCR(INP,LP4)
COPYCR(INP,LP5)
REWIND(LP3)
REWIND(LP4)
REWIND(LP5)
CATALOG(LP3,LP3,LD=LP3)
CATALOG(LP4,LP4,LD=LP4)
CATALOG(LP5,LP5,LD=LP5)
REDUCE(OFF)
***EOR***
OVERLAY(WCOL,0,0)
PROGRAM LPLAB(INPUT,OUTPUT,LP3,LP4,LP5,TAPE1=INPUT,TAPE2=OUTPUT,
+TAPE6=LP3,TAPE7=LP4,TAPE9=LP5,TAPE20)
DATA(XFILE=6LTAPF20)

THIS IS THE CONTROL PROGRAM FROM WHICH ANY OF THE SUBSECTIONS OF
LPLA9 MAY BE CALLED.
COMM/TEXT/ TEXT(60), ICHAR, ILINE, ANS
WRITE(2,100)
100 FORMAT(2X,*WELCOME TO LP LAB.*,/,1X,*THIS IS A COMPUTER ASSISTED *,
+LABORATORY IN LINEAR*,/,1X,*PROGRAMMING. THERE ARE ELEVEN *
+TOPICS YOU MAY SELECT.*,/,1X,*WHEN RESPONSES ARE REQUESTED, *
+TYPE THEM EXACTLY AS ASKED FOR.*)
WRITE(2,111)
111 FORMAT(3X,*THE TOPICS ARE AS FOLLOWS:*,/,1X
+1, INTRODUCTORY EXPLANATION OF LP LAB.*,/,1X,
+2, LINEAR PROGRAMMING GENERAL.*,/,1X,
+3, GRAPHICAL PROGRAMMING EXAMPLES.*,/,1X,
+4, REVIEW OF SIMPLEX ALGORITHM.*,/,1X,
+5, SIMPLEX ALGORITHM/SENSITIVITY ANALYSIS APPLIED.*)
WRITE(2,172)
172 FORMAT(2X,*6. REVIEW OF DUAL SIMPLEX ALGORITHM.*,/,1X,
+7, DUAL SIMPLEX APPLIED.*,/,1X,
+8, REVIEW OF TRANSPORTATION PROBLEM ALGORITHM.*,/,1X,
+9, SENSITIVITY ANALYSIS REVIEW.*,/,1X,
+10, TRANSPORTATION ALGORITHM APPLIED.*)
WRITE(2,103)
103 FORMAT(3X,*11. OPTIMAL ASSIGNMENT PROBLEM.*,/,1X,
+12, TERMINATE THE LABORATORY.*)
WRITE(2,110)
110 FORMAT(3X,*NOW CHOOSE A MAIN AREA TO INVESTIGATE.*,/,1X
+TYPE(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 0 OR 12)*)
1105 READ (1,111) REP
111 FORMAT(A10)
IF(REP.EQ.1H1 ) CALL OVERLAY(XFILE,9,0,6HRECALL)
IF(REP.EQ.1H2 ) CALL OVERLAY(XFILE,2,0,6HRECALL)
IF(REP.EQ.1H3 ) CALL OVERLAY(XFILE,11,0,6HRECALL)
IF(REP.EQ.1H4 ) CALL OVERLAY(XFILE,7,0,6HRECALL)
IF(REP.EQ.1H5 ) CALL OVERLAY(XFILE,3,0,6HRECALL)
IF(REP.EQ.1H6 ) GO TO 130
IF(REP.EQ.1H7 ) GO TO 130
IF(REP.EQ.1H8 ) CALL OVERLAY(XFILE,12,0,6HRECALL)
IF(REP.EQ.1H9 ) CALL OVERLAY(XFILE,5,0,6HRECALL)
IF(REP.EQ.1H10 ) CALL OVERLAY(XFILE,1,0,6HRECALL)
IF(REP.EQ.1H11 ) CALL OVERLAY(XFILE,6,0,6HRECALL)
IF(REP.EQ.1H12 ) GO TO 125
IF(ANS.EQ.1H5) GO TO 125
IF(REP.NE.1H1 .AND.REP.NE.1H2 .AND.REP.NE.1H3 .AND.REP.NE.1H4 .AND. 
+REP.NE.1H5 .AND.REP.NE.1H6 .AND.REP.NE.1H7 .AND.REP.NE.1H8 .AND. 
+REP.NE.1H9 .AND.REP.NE.1H10 .AND.REP.NE.1H11 .AND.REP.NE.1H12 )
+WRITE(2,120)
READNO20
120 FORMAT(1X,*I DO NOT UNDERSTAND.*,/)  
GO TO 109
125 WRITE(2,126)
126 FORMAT(IX,*YOU HAVE INDICATED YOU DESIRE TO TERMINATE THE LAB.*,/)  
+/,1X,*IS THIS CORRECT?*,/1X,*TYPE (YES OR NO)*,/)  
127 READ(1,111) REP  
ANS=1H*  
IF(REP ,EQ. 3HYES) GO TO 140  
IF(REP ,EQ. 2HNO) GO TO 109  
WRITE(2,120)  
GO TO 127
130 WRITE(2,129)
129 FORMAT(IX,*NOT COMPLETED YET. CHOOSE ANOTHER.*,/)  
GO TO 1105
140 WRITE(2,141)
141 FORMAT(1X,*LP LAB WILL NOW TERMINATE.*,/)  
+*/ AFT E LP LAB TERMINATES, THE TERMINAL WILL TYPE COMMAND-*/  
++ YOU MUST THEN ENTER LOGOUT AND PRESS THE RETURN KEY,*/  
++ *THANK YOU,*/)  
STOP
END
OVERLAY(TAPE20,2,0)
PROGRAM LINPRO
CCCCCCCCCCCCCCCCCCCC
C THIS SUBROUTINE CONTROLS THE PRESENTATION OF MATERIAL GENERAL  
C TO LINEAR PROGRAMMING.  
COMMON /TEXT/ TEXT(60) , ICHAR, ILINE, ANS  
DIMENSION RESULT(5)  
DATA (RESULT(I), I=1,5)/ 1HT, 2*1HT, 2*1HT/
DO 94 I=1,36  
94 READ(6,100)  
WRITE(2,95)  
95 FORMAT(///,♦ THIS CONTAINS COMMENTS ON GENERAL LINEAR PROGRAMMING.*,//)  
ISCOPE = 0  
DO 99 J=1,6  
READ(6,ICO) (TEXT(I), 1=1,60), ICHAR  
99 WRITE(2,114)(TEXT(I), 1=1,ICHAR)  
WRITE(2,96)  
10 05 READ(1,101) ANS  
IF(ANS .EQ. 4HSTOP) GO TO 113  
IF(ANS .EQ. 2HNO) GO TO 109  
IF(ANS .EQ. 3HYES) GO TO 103  
WRITE (?,112)  
GO TO 1C 10 05  
103 DO 1033 J=1,2  
READ (6,100) (TEXT(I), I=1,60), ICHAR  
1033 WRITE(2,114)(TEXT(I), I=1,ICHAR)  
CCCCCCCCCCCCCCCCCCCC
C THE NEXT STATEMENTS ADMINISTER A FIVE QUESTION TRUE/FALSE EXAM.  
DO 1 14 J=1,5  
READ(6,120)(TEXT(I), I=1,60), ICHAR  
WRITE(2,114)(TEXT(I), I=1,ICHAR)  
1033 READ(1,131) ANS  
IF(ANS .EQ. 4HSTOP) GO TO 113  
IF(ANS ,NE. 1HT ,AND, ANS ,NE. 1HF) GO TO 1035  
IF(ANS ,NE. RESULT(J)) GO TO 1039  
WRITE(2,115)  
GO TO 104
1035 WRITE(2,195)  
GO TO 1033


1039 ISCOPE = ISCORE + 1
104 CONTINUE
   IF(ISCOPE .EQ. 0) WRITE(2,106)
   IF(ISCOPE .EQ. 1) WRITE(2,107)
   IF(ISCOPE .GT. 1) WRITE(2,108) ISCORE
   GO TO 111

C--- THE REMAINDER CONTROL TEXT PRESENTATION.---
109 DO 113 J=1,7
110 READ(6,114)
   GO TO 111
116 READ(6,114)
   DO 110 J=1,7
      PEA0(6,114)
   GO TO 111
1106 READ(1,101) ANS
   IF(ANS .EQ. 3HYES) GO TO 111
   IF(ANS .EQ. 2HNO) GO TO 1108
   IF(ANS .EQ. 4HSKIP .OR. ANS .EQ. 4HSTOP) GO TO 113
      WRITE(2,112)
   GO TO 1106
1118 DO 1109 I=1,24
1109 READ(6,114)
111 READ(6,103)(TEXT(I), I=1,60),ICHAR,ILINE,IRET
      WRITE(2,114)(TEXT(I), I=1,ICHAR)
   GO TO 112
112 WRITE(2,114)(TEXT(I), I=1,ICHAR)
   WRITE(2,96)
   IF(IRET .NE. 0) GO TO 113
   IF(ILINE .EQ. 5) GO TO 1106
1127 READ(1,101) ANS
   IF(ANS .EQ. 2HGO) GO TO 111
   IF(ANS .EQ. 4HSKIP .OR. ANS .EQ. 4HSTOP) GO TO 113
      WRITE(2,1125)
   GO TO 1127
113 REWIND 6
RETURN
100 FORMAT(6A1, I2, 2I, I3,11)
101 FORMAT(A13)
102 FORMAT(2X,*RE-ENTER YOUR ANSWER.*/,* TYPE (YES, NO, OR SKIP)*/)
105 FORMAT(2X,*RE-ENTER YOUR ANSWER.*/,* TYPE (T OR F)*/)
115 FORMAT(1X, *CORRECT*/)
116 FORMAT(1X, *INCORRECT*/)
106 FORMAT(2X, *YOU GOT THEM ALL. CONGRATULATIONS.*/)
107 FORMAT(2X, *YOU MISSED 1. PRETTY GOOD.*/)
108 FORMAT(2X, *YOU MISSED *, I1,*. YOU NEED TO REVIEW LINEAR */
SO PARTS.*/)
114 FORMAT(1X,6A1)
1125 FORMAT(2X,*RE-ENTER YOUR ANSWER.*/,* TYPE (GO OR SKIP)*/)
END

OVERLAY(TAPE20,3,0)

PROGRAM SAL

C--- THIS IS THE SUBROUTINE FOR PRACTICING THE SIMPLEX ALGORITHM AND
C SENSITIVITY ANALYSIS. THIS WAS THE ORIGINAL SAL MAIN PROGRAM.

EXTERNAL SUB1
EXTERNAL SUB2
COMMON|COM1/X(21,81),REDUCE(21,61),NONBASE(60)
INTEGER CONSEN(21),SLACK(21),SUPPLUS(21),ARTIF(21),BASE(21),1
SLA,SLA,SLA,ART,PCONT,HLT
COMMON|COM2/CONSEN,SLACK,SURPLUS,ARTIF,BASE,SLA,SLA,ART,PCONT,HLT
COMMON|COM3/NOROWS,NOCOLS,IFINAL,ITNO
COMMON|COM4/JNOBN,KFLG
COMMON|COM5/ NOCONS
COMMON|COM6/FM10(18),FM11(14),FM12(16),FM13(17),NS1(6),NS2(6)
COMMON/C0M8/ N
COMMON/S1/KS1,KS2,IEXP,IFOUR
COMMON /TEXT/ TEXT(60), ICHAR, ILINE, ANS
DATA NS1/0,0,0,0,0,0/
DATA NS2/0,0,0,0,0,0/
NS1(5) = LOC(SUB1)
NS2(5) = LOC(SUB2)
CALL SYSTEMC(78,NS1)
CALL SYSTEMC(79,NS2)
WRITE(2,95)
95 FORMAT(///,* THIS SECTION ALLOWS YOU TO PRACTICE THE SIMPLEX*,/,
** ALGORITHM AND SENSITIVITY ANALYSIS.*,//)
6000 ISHOP=0
630 FORMAT( 1X,*YOU HAVE TWO OPTIONS FOR ENTERING THE*,/,
1 1X,*PARAMETERS. THE REGULAR METHOD IS A STEP*,/,
1 1X,*BY STEP RESPONSE THAT WILL INSURE ACCURACY.*,//,
1 1X,*THE EXPEDITE METHOD IS FASTER BUT REQUIRES*,/,
1 1X,*FAMILIARITY WITH THIS PROGRAM TO BE EFFECTIVE.*,//,
1 1X,*IT IS RECOMMENDED THAT YOU USE THE REGULAR METHOD FOR*,/,
1 1X,*THE FIRST FEW PROBLEMS.*)
CCCCCCCCCCCCCCCCCCCCCCC
C INITIALIZE
310 DO 100 I=1,3042
100 X(I)=0
DO 101 I=1,110
101 CONSEN(I)=0
ISHORT=0
KFLG=0
IFOUR=0
NOCOLS=0
NOROWS=0
IPPO=0
IFIVAL=0
ITNO=0
WRITE(2,291)
WRITE (2,600)
IF (IPROR .GT. P ) WRITE (2,602)
602 FORMAT( 1X,*YOU MAY NOW ENTER A NEW PROBLEM*)
603 WRITE (2,504)
604 FORMAT( 1X,*SELECT THE OPTION YOU WANT. TYPE (REG OR EXP)*,/) 
CCCCCCCCCCCCCCCCCCC
C READ IN MODE FOR ENTERING DATA
READ (1,219) IEXP
ANS = 1H*
IF (IEXP .EQ. 2HEXP) GO TO 630
IF (IEXP .EQ. 3HEXP) GO TO 320
IF (IEXP .EQ. 4HSTOP ,OP ,IEXP ,EN, 4HSTEP) ANS = IEXP
IF(ANS .EQ. IEXP ) RETURN
WRITE (2,608)
608 FORMAT( 1X,*I DONT UNDERSTAND, TRY AGAIN*)
GO TO 603
327 WRITE(2,204)
324 N=0
IEND=0
CCCCCCCCCCCCCCCCCCCCCCC
C ASSIGN XXX TO KS1 AND KS2 ARE RETURN POINTS AFTER ERROR ROUTINE IF 
C USER ENTERS INVALID FORMAT TYPES.
ASSIGN 325 TO KS1
ASSIGN 326 TO KS2
CCCCCCCCCCCCCCCCCCCCCCC
C NEXT 53 STATEMENTS ASKS USER TO ENTER PROBLEM PARAMETERS AND 
C STORES THEM.
325 N=N+1
326 WRITE(2,215) N
READ(1,206) X(1,N)

IF(IPROR .EQ. 0 .AND. N .LT. 6) GO TO 210
WRITE(2,209)
GO TO 230

210 WRITE (2,211)
230 READ (1,202) TEND
IF (TEND .EQ. 2HNO .OR. N .EQ. 40) GO TO 337
GO TO 325

330 NOCONS = 0
JEND = 0
335 NOCONS = NOCONS + 1
NOROWS = NOCONS + 1
WRITE (2,268) NOCONS
NN=0
JEND=0
ASSIGN 341 TO KS1
ASSIGN 341 TO KS2
340 NN=NN+1
WRITE (2,205) NN
READ (1,206) X(NOROWS,NN)
IF (IPROR .EQ. 0 .AND. NN .LT. 6) GO TO 342
WRITE (2,209)
GO TO 343

342 WRITE (2,198) NOCONS
343 READ (1,202) TEND
IF (TEND .EQ. 2HNO .OR. N .EQ. 40) GO TO 345
GO TO 340
345 ASSIGN 345 TO KS1
ASSIGN 345 TO KS2
IF (IPROR .EQ. 3) GO TO 347
WRITE (2,346) NOCONS
GO TO 348
347 WRITE (2,212) NOCONS
348 READ (1,213) CONSEN(NOROWS)
ASSIGN 344 TO KS1
ASSIGN 344 TO KS2
344 IF (IPROR .EQ. 3) GO TO 350
WRITE (2,349) NOCONS
GO TO 351
350 WRITE (2,214) NOCONS
351 READ (1,216) X(NOROWS,81)
WRITE (2,215)

353 READ (1,202) JEND
ANS = 1H*
IF (JEND .EQ. 3HNO .OR. NOCONS .EQ. 20) GO TO 6
IF (JEND .EQ. 4HSKIP) JEND = JEND
ANS = JEND
IF (ANS .EQ. JEND) RETURN
GO TO 335

C CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C NEXT 49 STATEMENTS PRINTS OUT EDIT TABLE FOR USER TO CHECK
C PARAMETERS HE HAS ENTERED.
6 WRITE(2,250)
250 FORMAT (/,,1X,*THE PROBLEM YOU HAVE JUST FINISHED ENTERING*,/,,
1 1X,*WILL BE CONDENSED INTO A TABLE FOR YOUR EDITING*,/,,
1 1X,*NOTE THAT ROW 1 IS THE OBJECTIVE FUNCTION*,/,,
14(1X,6H***♦***),*EDIT TABLE*,4(1X,6H***♦***),/)
DO 251 I=1,N
ILST=(N+1)-I
IF (X(I,ILST).EQ. 0) GO TO 251
GO TO 253
251 CONTINUE
253 KST=1
KON = 5
254 WRITE (2,257) (J,J=KST,KON)
257 FORMAT (/,'12X,*COL*','I2','4(8X,*COL*','I2))
IJ=ILST
IF (ILST .GT. KON) TJ=KON
DO 253 I=1,NOROWS
WRITE (2,261) I ,  (X(I, J >  ,  J  =  KST , U )
261 FORMAT(IX,*kOW*,I2,5(2X,E11.4))
259 CONTINUE
IF (ILST .LE. KON) GO TO 260
KST=KST+5
KON=KON+5
GO TO 254
260 WRITE (2,264)
264 FORMAT (//,3X,♦CONSTRAINT EQ.*,2X ,♦SENSE COOE^,2X ,*RT.HAND SIOEv ,
DO 265 1=2,NOROWS
ILES=I-1
WRITE (2,267) ILES,CONSEN(I),X (I,31)
267 FORMAT (8X ,12,13X,12,5X ,  El1.4)
265 CONTINUE
WRITE (2,1010)
1010 FORMAT//(8(IX,6H^♦♦♦♦♦))
WRITE (2,263)
263 FORMAT(/,IX,♦ANY CORRECTIONS TO ABOVE NUMBERS? TYPE (YES OR NO)*
1,
READ (1,202) ANS
IF (ANS .EO. 3HYES) GO TO 500
IF (ANS .  EO. 4HST0P .OR. ANS .EO. 4HSKIP) RETURN
CCCCCCCCCCCCCCCCCCCC
C NEXT 110 STATEMENTS ADDS SLACK, SURPLUS, AND ARTIFICIAL VARIABLES
C AND REARRANGES DATA INTO FORM FOR PRINTING AND USE BY SIMPLEX
C ALGORITHM.
270 XMAX =  X(1,1)
PCONT = 0
DO 10 I=1,NOROWS
DO 10 J=1,40
IF(XMAX  .LT. X(I,J))  XMAX=X(I,J)
10 CONTINUE
RTGM=10.*XMAX
DO 30 I=2,NOROWS
IF (CONSEN(I) = 2) 15,25
15 SLA=39+I
SLACK(I)=SLA
X(I,SLA)=1.
GO TO 30
25 SUR=39+I
SUPPLUS(I) = SUR
X(I,SUR)=-1.
23 ART=59+I
ARTIF(I)=ART
X(I,ART) = t.
X(I,ART) = RTGM
30 CONTINUE
DO 35 I=2,NOROWS
IF (ARTIF(I) .EO. G) GO TO 35
DO 34 J=1,81
34 X(I,J)=X(1,J) - RTGM*X(I,J)
35 CONTINUE
DO 41 T=2,NOROWS
IF (SLACK(T) .NE. 0) GO TO 36
BASE(T) = ARTIF(T)
GO TO 40
36 BASE(I) = SLACK(I)
40 CONTINUE
JNONB=0
DO 45 J=1,60
45 IF (X(1,J) .GT. 0) GO TO 45
JNONB=JNONB+1
NONBASE(JNONB) = J
CONTINUE
NOCOLS=JNONB+1
DO 53 K1=1,NROWS
REDUCE(K1,1)=X(K1,81)
DO 53 K2=1,JNONB
J=NONBASE(K2)
REDUCE(K1,K2+1)=X(K1,J)
CONTINUE
ITNO=0
HTLT=0
ENCODE(57,1404,FM10)
ENCODE(133,1104,FM11) NOCONS
ENCODE(57,1404,FM12)
ENCODE(57,1404,FM13)
ENCODE(115,1104,FM10(7)) NOCONS
ENCODE(115,1104,FM12(7)) NOCONS
ENCODE(117,1304,FM13(7)) NOCONS
1104 FORMAT (55H18X,*X*,I2,3(15X,*X*,I2),//,1X,I2,2X,*X*,I2,5(2X,E11.4),I2,
150H(/,5X,*X*,I2,2X,E11.4,2X,E11.4,2X,E11.4,2X,E11.4,2X,E11.4))
1105 FORMAT (93H(/,1X,*IT*,8X,*CURRENT*,//,1X,*NO*,2X,*BASE*,4X,*VALUES
1*X,*X*,I2,2X,*X*,I2,3(2X,E11.4),I2,123H(/,5X,*X*,I2,2X,E11.4,2X,E11.4))
1204 FORMAT (55H18X,*X*,I2,3(15X,*X*,I2),//,1X,I2,2X,*X*,I2,3(2X,E11.4),I2,
141H(/,5X,*X*,I2,2X,E11.4,2X,E11.4,2X,E11.4,2X,E11.4))
1304 FORMAT (55H18X,*X*,I2,2(15X,*X*,I2),//,1X,I2,2X,*X*,I2,4(2X,E11.4),I2,
150H(/,5X,*X*,I2,2X,E11.4,2X,E11.4,2X,E11.4,2X,E11.4))
1404 FORMAT (57H(/,1X,*IT*,8X,*CURRENT*,//,1X,*NO*,2X,*BASE*,4X,*VALUES
1X,,I2)
55 ITNO=ITNO+1
IF (KFLAG ,GT. 1) GO TO 3549
53 IF (JNONB ,GT. 4) GO TO 62
GO TO (65,6,63,62),JNONB
56 MPE = JNONB-4
JSTART = 5
JSTART1 = JSTART+1
54 IF (MPE ,GT. 4) GO TO 60
GO TO (57,58,59,67), MPE
57 INC=JSTART
IN1 = INC + 1
WRITE (2,FM11) (NONBASE(J),J=JSTART,INC),ITNO,BASE(1),REDUCE(1,1),
1 REDUCE(1,J),BASE(1),REDUCE(1,1),REDUCE(I,J),J=JSTART1,IN1,
1 I=2,NROWS)
GO TO 67
58 INC=JSTART+1
IN1 = INC + 1
WRITE (2,FM12) (NONBASE(J),J=JSTART,INC),ITNO,BASE(1),REDUCE(1,1),
1 (REDUCE(1,J),J=JSTART1,IN1),BASE(1),REDUCE(1,1),REDUCE(I,J),
1 J=JSTART1,IN1,I=2,NROWS)
GO TO 67
59 INC=JSTART+2
IN1 = INC + 1
WRITE (2,FM13) (NONBASE(J),J=JSTART,INC),ITNO,BASE(1),REDUCE(1,1),
1 (REDUCE(1,J),J=JSTART1,IN1),BASE(1),REDUCE(I,1),REDUCE(I,J),
42

1J = JSTAR1, IM1, A = 2, NOROWS)
GO TO 67

60 INC = JSTART + 3
IN1 = INC + 1
WRITE(2, FM10) (NONBASE(J), J = JSTART, INC), IN1, BASE(I), REDUCE(1),
1{REDUCE(1, J), J = JSTART, IN1, BASE(I), REDUCE(1), REDUCE(1, J),
1J = JSTART, IN1, A = 2, NOROWS)
IF (MORE .LE. 4) GO TO 67
MORE = MORE + 4
JSTART = JSTART + 4
JSTAR1 = JSTART + 1
GO TO 54

62 WRITE(2, FM10) (NONBASE(J), J = 1, 4), IN1, BASE(I),
1{REDUCE(1, J), J = 1, 5), (BASE(I), REDUCE(I, J), J = 1, 5),
1I = 2, NOROWS)
IF (JNONB .GT. 4) GO TO 56
GO TO 67

63 WRITE(2, FM12) (NONBASE(J), J = 1, 3), IN1, BASE(I),
1{REDUCE(1, J), J = 1, 4), (BASE(I), REDUCE(I, J), J = 1, 4),
1I = 2, NOROWS)
GO TO 67

64 WRITE(2, FM11) NONBASE(1), IN1, BASE(I),
1{REDUCE(1, J), J = 1, 2), (BASE(I), REDUCE(I, J), J = 1, 2),
1I = 2, NOROWS)
GO TO 67

67 IF (PCONT . NE. 0) GO TO 360
68 WRITE(2, 216)
READ(1, 217) ANS
IF (ANS . EQ. 4HNEXT) GO TO 355
IF (ANS . EQ. 4HSTOP, OR, ANS . EQ. 4HSKIP) RETURN
3549 IFINAL = 1

C NEXT 12 STATEMENTS CONTROLS PRINTING OF TABLEAUS.
355 CALL SIMPLEX
IF(KFLG . EQ. 3) GO TO 53
IF (PCONT . EQ. 1, 909, IFINAL . EQ. 1) GO TO 53
367 GO TO (370, 373, 390, 391), PCONT
370 IF (KFLG . EQ. 3) GO TO 9001
WRITE(2, 1405) REDUCE(1, 1)
GO TO 400
380 WRITE(2, 1006)
GO TO 400
390 WRITE(2, 1007)
GO TO 400

391 IF (KFLG . EQ. 3) GO TO 9004
WRITE(2, 1008) REDUCE(1, 1)
410 WRITE(2, 218)
READ(1, 219) ANS
IPROB = IPROB + 1
ISHORT = 1
IF (ANS . EQ. 2HGO) GO TO 310
IF (ANS . EQ. 3HCUP) GO TO 500
IF (ANS . EQ. 4HITER) GO TO 900
IF (ANS . EQ. 4HSTOP, OR, ANS . EQ. 4HSKIP) RETURN
IF (ANS . EQ. 3HSEN . AND. (PCONT . EQ. 1, OR, PCONT . EQ. 4)) GO TO 431
IF (ANS . EQ. 4HSEN . AND. (PCONT . EQ. 2, OR, PCONT . EQ. 3)) GO TO 402
WRITE(2, 2410)
410 FORMAT (1X,*I DONT UNDERSTAND*)
GO TO 400
402 WRITE(2, 2403)
403 FORMAT (1X,*THE SENSITIVITY ANALYSIS DOES NOT APPLY TO THE CURRENT


401 CALL SENSIT
IF(KFLG.GT.9) GO TO 270
GO TO 400
C NEXT 10 STATEMENTS ALLOWS USER TO CHANGE THE DATA IN THE EDIT TABLE.
500 WRITE (2,502)
502 FORMAT (1X,*TO CORRECT A VARIABLE COEFFICIENT, TYPE COEF*,/, 1X,*TO CORRECT A CONSTRAINT EQUATION SENSE CODE, TYPE SENSE*,/, 1X,*TO CORRECT A CONSTRAINT RIGHHTAND SIDE VALUE, TYPE RIGHT*,/) 
503 READ (1,219) ANS
IF (ANS .EQ. 4HCOEF) GO TO 509
IF (ANS .EQ. 5HSENSE) GO TO 540
IF (ANS .EQ. 5HRIGHT) GO TO 560
IF (ANS .EQ. 3HALL) GO TO 5681
IF (ANS .EQ. 4HSTOP .OR. ANS .EQ. 4HSKIP) RETURN
WRITE (2,504)
504 FORMAT (1X,*I DON'T UNDERSTAND*** TYPE COEF OR SENSE OR RIGHT*,/) 
GO TO 503
508 IF (IEXP .EQ. 3HEXP) GO TO 516
WRITE (2,511)
511 FORMAT (1X,*TO CORRECT A VARIABLE COEFFICIENT YOU MUST LOCATE*,/, 1X,*THE NUMBER IN THE ABOVE TABLE BY ROW AND COLUMN. NOTE*,/, 1X,*THAT ROW 1 IS FOR THE OBJECTIVE FUNCTION, ROW 2 FOR*,/, 1X,*THE 1ST CONSTRAINT EQUATION, ETC., AND COL 1 IS FOR*,/, 1X,*VARIABLE 1, COL 2 FOR VARIABLE 2, ETC.*,/) 
512 WRITE (2,514)
514 FORMAT (1X,*TYPE THE ROW AND COLUMN NUMBERS AS REQUESTED*,/, 1X,*USE TWO DIGIT INTEGERS ONLY SUCH AS 03 OR 12 ETC.*)
ASSIGN 515 TO K1
ASSIGN 516 TO K2
516 WRITE (2,519)
518 FORMAT (* ROW=*)
READ (1,520) KROW
520 FORMAT (1I)
ASSIGN 5211 TO K1
ASSIGN 5231 TO K2
521 WRITE (2,522)
522 FORMAT (* COL=*)
READ (1,520) KCOL
IF (IEXP .EQ. 3HEXP) GO TO 525
WRITE (2,524) KROW,KCOL,X(KROW,KCOL)
524 FORMAT (1X,*YOU JUST IDENTIFIED ROW *,I2,* AND COL *,I2, 1* FOR *E11.4,/, 1X,*IS THIS THE NUMBER YOU WANT TO CORRECT? TYPE YES OR NO*,/) 
READ (1,219) ANS
IF (ANS .EQ. 2HNO) GO TO 516
IF (ANS .EQ. 4HSTOP .OR. ANS .EQ. 4HSKIP) RETURN
ASSIGN 525 TO K1
ASSIGN 525 TO K2
525 WRITE (2,526)
526 FORMAT (1X,*TYPE THE NEW NUMBER YOU WANT*,/,* NEW=*)
READ (1,216) XNEW
WRITE (2,528) XNEW,KROW,KCOL
528 FORMAT (1X,*YOU HAVE ASKED FOR *E11.4,* TO BE ENTERED*,/, 1X,*AT ROW *,I2,* AND COL *,I2,/, 1X,*IS THIS WHAT YOU WANT? TYPE YES OR NO*,/) 
READ (1,219) ANS
IF (ANS .EQ. 2HYES) GO TO 530
IF (ANS .EQ. 4HSTOP .OR. ANS .EQ. 4HSKIP) RETURN
GO TO 525
530 X(KROW,KCOL)=XNEW
WRITE (2,532)
FORMAT (1X,*ANY MORE CORRECTIONS TO COEFFICIENTS? TYPE YES OR NO
1*/*)
READ (1,219) ANS
IF (ANS .EQ. 3) RETURN
IF (ANS .EQ. 4) STOP
RETURN
WRITE (2,534)

FORMAT (1X,*TO CORRECT CONSTRAINT EQ. SENSE, TYPE SENSE*,/,
1 1X,*TO CORRECT RIGHHAND SIDE VALUES, TYPE RIGHT*,/
1 1X,*IF NO MORE CORRECTIONS, TYPE ALL*,/) 
GO TO 503

WRITE (2,542)
ASSIGN 540 TO KS1
ASSIGN 540 TO KS2

FORMAT (1X,*TYPE THE CONSTRAINT EQ. NUMBER AS TWO DIGIT INTEGER*,
1/, 1 1X,*SUCH AS 03 OR 15*,/,* EQ.*,)
READ (1,520) KEQ
ASSIGN 5421 TO KS1
ASSIGN 5421 TO KS2

WRITE (2,544)

FORMAT (1X,*TYPE THE CORRECT CONSTRAINT SENSE CODE AS 1 OR 2 OR 3
1*,/,* CODE=*)
READ (1,213) KCODE
WRITE (2,546) KEQ,KCODE

WRITE (2,548)
CONSEN(KEQ+1)=KCODE
WRITE (2,551)

FORMAT (1X,*ANY MORE CONSTRAINT SENSE CORRECTIONS? TYPE YES OR NO*,/) 
READ (1,219) ANS
IF (ANS .EQ. 3) RETURN
IF (ANS .EQ. 4) STOP
RETURN

WRITE (2,555)

FORMAT (1X,*TO CORRECT RIGHHAND SIDE VALUES, TYPE RIGHT*,/
1 1X,*IF THERE ARE NO MORE CORRECTIONS, TYPE ALL*,/) 
GO TO 503

WRITE (2,562)
ASSIGN 560 TO KS1
ASSIGN 560 TO KS2
READ (1,520) KEQ
ASSIGN 5632 TO KS1
ASSIGN 5632 TO KS2

WRITE (2,566)

FORMAT (1X,*TYPE THE CORRECTED RIGHHAND VALUE*,/* VAL=*)
READ (1,276) XRT
WRITE (2,564) XRT,KEQ

FORMAT (1X,*YOU HAVE ASKED FOR *,E11.4,* TO BE THE *,/,
1 1X,*RIGHHAND VALUE FOR EQ. *,I2/*,
1 1X,*IS THIS WHAT YOU WANT? TYPE YES OR NO*,/) 
READ (1,219) ANS
IF (ANS .EQ. 3) RETURN
IF (ANS .EQ. 4) STOP
RETURN

WRITE (2,565)

FORMAT (1X,*ANY MORE RIGHHAND SIDE CORRECTIONS? TYPE YES OR NO*,
2*)
READ (1,219) ANS
IF (ANS .EQ. 3HYES) GO TO 560
IF (ANS .EQ. 4HSSTOP .OR. ANS .EQ. 4HSKIP) RETURN
WRITE (2,568)
568 FORMAT (1X,*ANY OTHER CORRECTIONS? TYPE YES OR NO*,/) READ (1,219) ANS
IF (ANS .EQ. 3HYES) GO TO 500
IF (ANS .EQ. 4HSSTOP .OR. ANS .EQ. 4HSKIP) RETURN
WRITE (2,1012)
1012 FORMAT(1X,*YOU WANT A REVISED TABLE PRINTED*,/) READ (1,219) ANS
IF (ANS .EQ. 3HYES) GO TO 6
IF (ANS .EQ. 4HSSTOP .OR. ANS .EQ. 4HSKIP) RETURN
WRITE (2,514)
514 FORMAT(1X,*YOU SHOULD PENCIL IN YOUR CORRECTIONS*,/) GO TO 270

CCCCCCCCCCCCCCCCCCCC
C NEXT 26 STATEMENTS ARE THE EXPEDIT ROUTINE FOR ENTERING DATA.
630 WRITE (2,632)
ASSIGN 633 TO KS1
ASSIGN 630 TO KS2
632 FORMAT(1X,*TYPE NO. OF VARIABLES IN OBJ. FUNCTION (03 ETC).*),/
READ (1,520) N
WRITE (2,634)
634 FORMAT(1X,*TYPE ALL COEFFICIENTS OF OBJ. FUNCT. IN ORDER*,/) ASSIGN 6341 TO KS1
ASSIGN 6341 TO KS2
READ (1,206) (X(1,1),1=1,N)
6342 WRITE (2,638)
638 FORMAT(1X,*TYPE NO. OF CONSTRAINT EQUATIONS (C3 ETC)*),/
ASSIGN 6342 TO KS1
ASSIGN 6342 TO KS2
READ (1,520) NOCONS
NPOWS = NOCONS + 1
DO 650 I = 2,NPOWS
ICONS=I-1
WRITE (2,640 ICONS)
640 FORMAT (1X,*TYPE COEFF, SENSE CODE(1 IS <=, 2 IS =, 3 IS >=),*),/
1 1X,*AND **T HAND SIDE, IN ORDER, FOR CONSI. EQ. *,I2,*)
1 1X,*DEFRESS RETURN KEY BEFORE STARTING.*)
ASSIGN 6401 TO KS1
ASSIGN 6401 TO KS2
6401 READ (1,206) (X(I,J),J=1,N)
ASSIGN 6402 TO KS1
ASSIGN 6402 TO KS2
6402 READ (1,213) CONSEN(I)
ASSIGN 6403 TO KS1
ASSIGN 6403 TO KS2
6403 READ (1,206) X(I,81)
650 CONTINUE
GO TO 6

CCCCCCCCCCCCCCCCCCCC
C NEXT 23 STATEMENTS ARE ROUTINE FOR ALTERING FINAL ITERATION
C TABLEAU.
900 IF (IEXP .EQ. 3HEXP) GO TO 904
WRITE (2,902)
902 FORMAT(1X,*YOU MAY ALTER THE NUMBERS IN THE FINAL*,/) 1 1X,*IDENTIFYING THE APPROPRIATE ROW AND*,/) 1 1X,*COLUMN, YOU MUST DETERMINE THE ROW AND COLUMN*,/) 1 1X,*NUMBERS BY COUNTING SEQUENTIALLY.*)
904 WRITE (2,905)
905 FORMAT(1X,*TYPE TWO DIGIT INTEGERS (03 ETC.) FOR ROW AND COL*,/)
1 IX, * ROW = *
ASSIGN 904 TO KS1
ASSIGN 904 TO KS2
READ (1,520) IROW
9051 WRITE (2,906)
906 FORMAT (* COL=*)
ASSIGN 9051 TO KS1
ASSIGN 9051 TO KS2
READ (1,520) ITCOL
WRITE (2,907) IROW,ITCOL,REDUCE(IROW,ITCOL)
908 FORMAT(1X,*ROW *,I2,* AND COL *,I2,* LOCATES *,E11.4,*,*,*),
1 IX,*IS THIS WHAT YOU WANT TO ALTER? TYPE YES OR NO*,/,
READ (1,219) ANS
IF (ANS .EQ. THYES) GO TO 910
IF (ANS .EQ. STOP OR ANS .EQ. HKSKIP) RETURN
913 WRITE (2,912)
912 FORMAT(1X,*TYPE NEW VALUE YOU WANT TO ENTER*,/,
1*NEW=*)
ASSIGN 910 TO KS1
ASSIGN 911 TO KS2
READ (1,206) REDUCE(ITROW,ITCOL)
WRITE (2,914)
914 FORMAT(1X,*ANY OTHER ALTERATIONS? TYPE YES OR NO*,/)
READ (1,219) ANS
IF (ANS .EQ. THYES) GO TO 904
IF (ANS .EQ. STOP OR ANS .EQ. HKSKIP) RETURN
GO TO 53
CCCCCCCCCCCCCCCCCC
C NEXT 4 STATEMENTS CONTROL PRINTING OF CONDENSED DATA DURING
C SENSITIVITY PRINTOUT.
9001 WRITE(2,9003)REDUCE(1,1),(BASE(I),I=2,NOROWS)
9003 FORMAT(1X,*THE OPTIMAL SOLUTION = *,E11.4,*,*,*),
1 IX,*BASE IS X *,2013)
GO TO 400
9004 WRITE(2,9006)REDUCE(1,1),(BASE(I),I=2,NOROWS)
9006 FORMAT(1X,*AN ALTERNATIVE OPTIMAL SOLUTION = *,E11.4,*,*,*),
1 IX,*BASE IS X *,2013)
GO TO 400
193 FORMAT(1X,*MORE VARIABLES FOR EQ. *,I2,*> TYPE(YES OR NO)*),/
201 FORMAT(1X,*YOU MAY NOW SELECT A L.P. PROBLEM FOR SOLUTION*),/
1 IX,*BY ENTERING THE PARAMETERS AS THEY ARE REQUESTED,*)
222 FORMAT(A10)
204 FORMAT(1X,*ENTER THE COEFFICIENTS FOR THE OBJECTIVE FUNCTION*,/,
1 IX,*VARIABLES AS THEY ARE REQUESTED BY TYPING A DECIMAL*,/,
1 IX,*NUMBER SUCH AS 12. OR +1.23E+01*)
205 FORMAT(* X*,I2,*>)
246 FORMAT(E11.4)
204 FORMAT(1X,*TYPE THE COEFFICIENTS FOR THE VARIABLES IN*,/,
1 IX,*CONSTRAINT EQUATION*,I2,*> USE A DECIMAL NUMBER,*)
209 FORMAT(1X,*MORE VARIABLES? TYPE (YES OR NO)*),/
211 FORMAT(1X,*MORE VARIABLES FOR OBJ. FUNCTION? TYPE (YES OR NO)*),/
212 FORMAT(1X,*ENTER CONSTRAINT EQ. *,I2,* SENSE CODE,*),/,
1 IX,*TYPE (1 FOR <=, 2 FOR =, 3 FOR >=*)*,/)
213 FORMAT(I1)
214 FORMAT(1X,*TYPE RIGHTHAND CONSTRAINT VALUE FOR CONSTRAINT*,/,
1 IX,*EQ. *,I2,* USING DECIMAL NO. SUCH AS 13. OR +1.33E+01*),/
215 FORMAT(1X,*MORE CONSTRAINT EQUATIONS? TYPE (YES OR NO)*),/
216 FORMAT(1X,*DO YOU WANT THE NEXT OR FINAL ITERATION?*,/,
1 IX,*TYPE (NEXT OR FINAL)*)*,/)
217 FORMAT(A5)
218 FORMAT(* IF YOU WANT TO STOP, TYPE STOP*),/
1 IX,*IF YOU WANT TO CONTINUE WITH A NEW PROBLEM, TYPE GO*),/
1 IX,*IF YOU WANT TO ALTER THE ORIGINAL PROBLEM, TYPE CUR*),/
SUBROUTINE SIMPLEX

C THIS SUBROUTINE APPLIES SIMPLEX ALGORITHM TO PROBLEM ENTERED
C BY USER AND FINDS SOLUTION.
C IFINAL USED AS FLAG TO CONTROL PRINTING OF EACH ITERATION OR
C ONLY FINAL ITERATION.
COMMON/COM1/X(21,51),REDUCE(21,61),NONBASE(60)
INTEGER CONSEN(21), SLACK(21),SUPPLUS(21),ARTIF(21),BASE(21),
1 SLA,SUR,ART,PCONT,HLT
INTEGER NEG(41)
REAL MIN, DATA
COMMON/COM2/CONSEN,SLACK,SURPLUS,ARTIF,BASE,SLA,SUP,ART,PCONT,HLT
COMMON/COM3/NROWS,NCOLS,IFINAL,ITNO
COMMON/COM4/NUMNP,KFLG

DO 63 I=2,NOCOLS
63 NEG(I)=0
66 ROW1MIN = REDUCE(1,2)
JCOLMN=2
DO 71 J=2,NOCOLS
IF (NEG(J),EQ,-1) GO TO 70
IF(ROW1MIN,LE,REDUCE(1,J)) GO TO 70
POW1MIN = REDUCE(1,J)
JCOLMN=J
70 CONTINUE
IF (ROW1MIN,EQ,0) GO TO 100
IF (ROW1MIN,GT,J) GO TO 71
DO 72 I=2,NOROWS
IPOS=I
IF (REDUCE(I,JCOLMN),GT,0,AND,REDUCE(I,1),GE,0) GO TO 74
72 CONTINUE
GO TO 50
74 MINRAT=REDUCE(IPOS,1)/REDUCE(IPOS,JCOLMN)
IROW=IPOS
IPOS=IPOS+1
DO 75 I=IPOS,NOROWS
IF (REDUCE(I,JCOLMN),LE,0,OR,REDUCE(I,1),LT,0) GO TO 75
RATIO=REDUCE(I,1)/REDUCE(I,JCOLMN)
IF (MINRAT,LE,RATIO) GO TO 75
MINRAT=RATIO
IROW=I
75 CONTINUE
69 PIVIT = REDUCE(IROW,JCOLMN)
DO 76 J=1,NOCOLS
76 REDUCE(IROW,J)=REDUCE(IROW,J)/PIVIT
DO 80 I=1,NOROWS
IF (I,EQ,IROW) GO TO 80
DO 79 J=1,NOCOLS
IF (J,EQ,JCOLMN) GO TO 79
78 REDUCE(I,J)=REDUCE(I,J)-REDUCE(IROW,J)*REDUCE(I,JCOLMN)
79 CONTINUE
REDUCE(I,JCOLMN) = - REDUCE(I,JCOLMN) / PIVIT
80 CONTINUE
SUBROUTINE SENSIT

C THIS SUBROUTINE PROVIDES THE SENSITIVITY ANALYSIS FOR PROBLEM
C USER HAS ENTERED AND PROGRAM HAS SOLVED.

EXTERNAL SUM
EXTERNAL SUB2
COMMON/COM1/X(21,61),REDUCE(21,61),NONBASE(60)
INTEGER CONSEN(21),SLACK(21),SURPLUS(21),ARTIF(21),BASE(21),
1 SLA,SUR,ART,PCONT,HLT
COMMON/COM2/CONSEN,SLACK,SURPLUS,ARTIF,BASE,SLA,SUR,ART,PCONT,HLT
COMMON/COM3/NOROWS,NOCOLS,IFINAL,ITNO
COMMON/COM4/JSNONB,KFLG
COMMON/S1/KS1,KS2,IEXP,IFOUR
COMMON/TEXT/TEXT(60),ICCHAR,ILINE,ANS
DIMENSION FM10(16),PM11(14),FM12(16),FM13(17),NS1(6),NS2(6)
DATA NS1/0,0,0,0,0,0/
DATA NS2/1,0,0,0,0,0/
DATA NS1/S1,SUR,SUR,SUR,SUR,SUR/
DATA NS2/S2,S2,S2,S2,S2,S2/
KFLG=0
WRITE(2,720)
720 FORMAT(2,720)
721 WRITE(2,722)
722 FORMAT(1X,*SELECT THE SENSITIVITY ANALYSIS CASE YOU WANT.*),
1 1X,*TYPE (1, 2, OP 3)*,/
READ(1,724) KASE
ANS = 1H*
IF (KASE .EQ. 1H1) GO TO 728
IF (KASE .EQ. 1H2) GO TO 770
IF (KASE .EQ. 1H3) GO TO 900
IF (KASE .EQ. 4HSTOP .OR. KASE .EQ. 4HSKIP) ANS = KASE
IF (ANS .EQ. KASE) RETURN
WRITE (2,726)
726 FORMAT (1X,*I DON'T UNDERSTAND?*)
GO TO 721

CCCCCCCCCCCCCCCCCCCCCCCCC
C NEXT 61 STATEMENTS ARE OBJECTIVE FUNCTION ANALYSIS.
728 WRITE (2,730)
730 FORMAT (1X,*SELECT THE OBJECTIVE FUNCTION VARIABLE* ,/,
1 1X,*YOU WANT TO INVESTIGATE. TYPE A TWO DIGIT INTEGER* ,/,
1 1X,*SUCH AS 03 TO IDENTIFY THE VARIABLE.*,/,
1 *VAR=* )
IF (FOUR = 0)
ASSIGN 728 TO KS1
ASSIGN 723 TO KS2
READ (1,732) KVAR
732 FORMAT (12)
WRITE (2,734) KVAR
734 FORMAT (1X,*YOU SELECTED X*,I2,* FOR ANALYSIS.*,/,
1 1X,*IS THIS WHAT YOU WANTED? TYPE YES OR NO*,/) READ (1,724) KONFR
ANS = 1H*
IF (KONFR .EQ. 3HYES) GO TO 736
IF (KONFR .EQ. 4HSTOP .OR. KONFR .EQ. 4HSKIP) ANS = KONFR
IF (ANS .EQ. KONFR) RETURN
GO TO 728
736 DO 740 I = 1,JNONB
KNON = I
IF (NONBASE(KNON) .EQ. KVAR) GO TO 748
740 CONTINUE
DO 742 I = 2,NOPOWS
KBASE = I
IF (BASE(KBASE) .EQ. KVAR) GO TO 760
742 CONTINUE
WRITE (2,744) KVAR
744 FORMAT (1X,*X*,I2,* HAS NO MEANING FOR THIS PROBLEM.*,/,
1 1X,*DO YOU WANT TO TRY ANOTHER VARIABLE? TYPE YES OR NO*,/)
READ (1,724) KONFR
ANS = 1H*
IF (KONFR .EQ. 3HYES) GO TO 728
IF (KONFR .EQ. 4HSTOP .OR. KONFR .EQ. 4HSKIP) ANS = KONFR
IF (ANS .EQ. KONFR) RETURN
745 WRITE (2,746)
746 FORMAT (1X,*DO YOU WANT TO CONTINUE WITH SENSITIVITY ANALYSIS?*,/,
1 1X,*TYPE YES OR NO*,/)
READ (1,724) KONFR
ANS = 1H*
IF (KONFR .EQ. 3HYES) GO TO 721
IF (KONFR .EQ. 4HSTOP .OR. KONFR .EQ. 4HSKIP) ANS = KONFR
IF (ANS .EQ. KONFR) RETURN
KFLG = 1
RETURN
748 IF (NONBASE(KNON) .LE. 40) GO TO 752
WRITE (2,750) NONBASE(KNON)
750 FORMAT (1X,*ANALYSIS - - - - - - NONBASIC VARIABLE X*,I2,* MAY*,/,
1 1X,*RANGE FROM ITS ORIGINAL OBJECTIVE FUNCTION VALUE*,/,
1 1X,*VAR=* )
WRITE (2,754) NONBASE(KNON), X(1,KVAR), OMAX, REDUCE(1,1)
754 FORMAT (1X,*ANALYSIS - - - - - - NONBASIC VARIABLE X*,I2,* MAY*,/,
1 1X,*RANGE FROM ITS ORIGINAL OBJECTIVE FUNCTION VALUE*,/,
1 1X,*VAR=* )
WRITE (2,756)
    FORMAT (/ , 1X,*DO YOU WANT TO TRY ANOTHER OBJECTIVE FUNCTION VARIABLE*/
1LE>? , 1X,*TYPE YES OR NO*/)
GO TO 745
767 IF (BASE(KPSE) .LE. 40) GO TO 761
WRITE (2,756) BASE(KPSE)
GO TO 756
761 TESTN = 1.E+99
TESTP = -1.E+99
DO 764 I=2,NOCOLS
IF (REDUCE(KPSE,1) .EQ. 0 ) GO TO 764
IF (REDUCE(1,1) .EQ. 0 ) GO TO 763
TEST = -REDUCE(1,1) / REDUCE(KPSE,1)
IF (TEST .LT. C) GO TO 762
IF (TEST .GE. TESTN) GO TO 764
TESTN = TEST
JNTST = I
GO TO 764
762 IF (TEST .LE. TESTP) GO TO 764
TESTP = TEST
JPTST = I
GO TO 764
763 TESTP = 0.
JPTST = I
764 CONTINUE
TESTL = X(1,KVAR) - TESTN
TESTH = X(1,KVAR) - TESTP
OBJL = REDUCE(1,1) + REDUCE(KPSE,1) * TESTP
OBJH = REDUCE(1,1) + REDUCE(KPSE,1) * TESTN
WRITE(2,756) BASE(KPSE),TESTL,TESTH,OBJL,OBJH,BASE(KPSE)
766 FORMAT( 1X,*ANALYSIS BASIC VARIABLE X* MAY RANGE FROM *, E11.4, TO *, E11.4, IN THE ORIGINAL OBJECTIVE*, /
1X,*HOWEVER, THE OBJECTIVE FUNCTION VALUE WILL RANGE*, /
1X,*FROM *, E11.4, TO *, E11.4, AS * X* V A R I E S.*)
GO TO 776
CCCCCCCCCCCCCCCC
C NEXT 60 STATEMENTS ARE FOR RIGHT HAND CONSTANT ANALYSIS.
776 WRITE (2,772)
772 FORMAT (1X,*SELECT THE CONSTRAINT YOU WANT TO INVESTIGATE*, /
1X,*REFER BACK TO THE EDIT TABLE TO IDENTIFY THE*, /
1X,*CONSTRAINT EQUATION NUMBER AND THE RIGHTHAND SIDE VALUE*, /
1X,*TYPE THE EQ. NUMBER AS A TWO DIGIT INTEGER SUCH AS 04*, /
1* EQ NO=*)
IF(QUF=0
ASSIGN 770 TO KS1
ASSIGN 770 TO KS2
READ (1,732) KEQ
IF (KEQ ,LE. NOPROWS-1) GO TO 778
WRITE (2,774) KEQ
775 WRITE (2,776)
776 FORMAT (1X,*DO YOU WANT TO SELECT ANOTHER EQUATION FOR ANALYSIS* /
1X,*IF THE RIGHTHAND SIDE CONSTRAINT? TYPE YES OR NO*/)
READ (1,724) KONFR
ANS = 1H*
IF (KONFREQ.3HYES) GO TO 770
IF(KONFR .EQ. 4HSTOP .OR. KONFR .EQ. 4HSKIP) ANS = KONFR
IF(ANS .EQ. KONFR ) RETURN
WRITE (2,746)
GO TO 747
778 DO 782 I=1,JNIANV
KCOLR = I + 1
IF (NONBASE(I) .EQ. (45 + KEO)) GO TO 790
IF (NONBASE(I) .EQ. (60 + KEO)) GO TO 790
782 CONTINUE
DO 785 I=2,NOPOWS
KROWN = I
IF (BASE(I) .EQ. (59 + KEO)) GO TO 800
IF (BASE(I) .EQ. (59 + KEO)) GO TO 806
786 CONTINUE
790 RTESTP = 1.5 + 99
RTESTN = -1.E+99
IF (REDUCE(I,1) .EQ. 0.) GO TO 793
IF (REDUCE(I,1) .EQ. (59 + KEO)) GO TO 800
DO 794 I=2,NOPOWS
RTEST = -REDUCE(I,1) / REDUCE(I,KCOLR)
IF (RTEST .LT. 0.) GO TO 792
IF (RTEST .GE. RTESTP) GO TO 794
RTESTP = RTEST
PJNTST = I
GO TO 794
792 IF (RTEST .LE. RTESTN) GO TO 794
RTESTN = RTEST
PJNTST = I
GO TO 794
793 RTESTN = J.
PJNTST = I
794 CONTINUE
RTESTL = X(KEQ+1,81) + RTESTN
RTESTH = X(KEQ+1,81) + RTESTP
R09JL = REDUCE(1,1) + REDUCE(1,KCOLR) * RTESTN
R09JH = REDUCE(1,1) + REDUCE(1,KCOLR) * RTESTP
WRITE(2,798) KEQ,X(KEQ+1,31),RTESTL,RTESTH,R09JL,R09JH
798 FORMAT(1X,*ANALYSIS---THE RIGHTHAND SIDE CONSTRAINT OF*,/,
1IX,*EQUATION *,I2,*(ORIGINAL VALUE *,E11.4,*)*,/,
1IX,*MAY RANGE FROM *,E11.4,* TO *,E11.4,* AND STILL*,/,
1IX,*THE OBJECTIVE FUNCTION VALUE WILL RANGE FROM*,/,
1IX,*E11.4,* TO *,E11.4,* AS THE RIGHTHAND SIDE VARIES.*)
GO TO 775
300 RTESTN = REDUCE(KROWN,1)
RTESTL = X(KEO+1,81) + RTESTN
WRITE (2,3(2) KEO,X(KEO+1,91),RTESTL)
A02 FORMAT (1X,*ANALYSIS---THE RIGHTHAND SIDE CONSTRAINT OF*,/,
1IX,*EQUATION *,I2,*(ORIGINAL VALUE *,E11.4,*)*,/,
1IX,*MAY RANGE FROM *,E11.4,* TO AN UNLIMITED UPPER BOUND*)
GO TO 775
806 WRITE (2,806)
806 FORMAT (1X,*THE CURRENT SOLUTION YOU ARE USING IS NOT A*,/,
1IX,*FEASIBLE SOLUTION, THEREFORE ANY ANALYSIS IS NOT VALID*)
GO TO 775
CCCCCCCCCCCCCCCCCCCC
C NEXT 70 STATEMENTS FOR DISCRETE CHANGE ANALYSIS.
300 IF (IFOUN .GT. 0) GO TO 914
WRITE (2,906)
906 FORMAT (1X,*TO CHANGE VARIABLE COEFFICIENTS, TYPE COEF*,/,
1 1X,*TO CHANGE RIGHHTHAND CONSTRAINT CONSTANTS, TYPE CONST*/,
READ (1,724) KONFR
ANS = 1H*
IF (KONFR .EQ. 4HCOEF) GO TO 910
IF (KONFR .EQ. 5HC0NST) GO TO 938
IF (KONFR .EQ. 4HSKIP) ANS = KONFR
IF(ANS .EQ. 4H KONFR ) RETURN
WRITE (2,908)
908 FORMAT (1X,*I DONT UNDERSTAND??*)
GO TO 904
910 WRITE (2,912)
912 FORMAT (1X,*TYPE TWO DIGIT INTEGERS (03 ETC.) TO LOCATE THE*,/,
1 1X,*COEFFICIENT IN THE EDIT TABLE ABOVE.*)
914 WRITE (2,915)
916 FORMAT (* ROW=*)
ASSIGN 914 TO KS1
ASSIGN 914 TO KS2
READ (1,732) KROW
918 WRITE (2,920)
920 FORMAT (* COL=*)
ASSIGN 913 TO KS1
ASSIGN 913 TO KS2
READ (1,732) KCOL
922 WRITE (2,924)
924 FORMAT (1X,*TYPE A DECIMAL NO. (12. OR +1,203DE+01) FOR NEW VALUE,
1X,*NEW=*)
ASSIGN 922 TO KS1
ASSIGN 922 TO KS2
READ (1,926) XNEW
926 FORMAT (E11.4)
WRITE (2,928) KROW,KCOL,XNEW
928 FORMAT (1X,*YOU WILL CHANGE ROW *,I2,* COL *,I2,* TO *,E11.4,/,1 1X,*IS THIS WHAT YOU WANT> TYPE (YES OR NO)*,*/)
READ (1,724) KONFR
ANS = 1H*
IF (KONFR .EQ. 3HYES) GO TO 939
IF(KONFR .EQ. 4HSKIP) ANS = KONFR
IF(ANS .EQ. 4H KONFR ) RETURN
GO TO 914
930 X(KROW,KCOL)=XNEW
931 WRITE (2,932)
932 FORMAT (1X,*ANY MORE COEFFICIENT CHANGES? TYPE (YES OR NO)*,*/)
READ (1,724) KONFR
ANS = 1H*
IF (KONFR .EQ. 3HYES) GO TO 914
IF(KONFR .EQ. 2HNO) GO TO 934
IF(KONFR .EQ. 4HSKIP) ANS = KONFR
IF(ANS .EQ. 4H KONFR ) RETURN
WRITE (2,908)
GO TO 931
934 WRITE (2,936)
936 FORMAT (1X,*ANY RIGHHTHAND SIDE CONSTRAINT CHANGES? TYPE (YES OR NO)*,*/)
READ (1,724) KONFR
ANS = 1H*
IF (KONFR .EQ. 3HYES) GO TO 914
IF (KONFR .EQ. 2HNO) GO TO 956
IF(KONFR .EQ. 4HSKIP) ANS = KONFR
IF(ANS .EQ. 4H KONFR ) RETURN
WRITE (2,908)
GO TO 934
938 WRITE (2,940)
940 FORMAT (1X,*TYPE TWO DIGIT INTEGER (03 ETC.) FOR CONSTANT*,/,
1 1X,*EQUATION NUMBER FROM EDIT TABLE*,/,
ET Eq. No. =*1*
ASSIGN 934 TO KS1
ASSIGN 935 TO KS2
READ (1,732) KEO
WRITE (2,944) KEO
FORMAT( 1X,*TYPE DECIMAL NUMBER (12, OP +1.2003E+01) FOR NEW VALUE
1X,* FOR R.HAND SIDE OF CONSTRAINT EQ. *I2,*,*,*,*/ NEW=*)
ASSIGN 942 TO KS1
ASSIGN 942 TO KS2
READ (1,926) XNEW
WRITE (2,946) KEO,XNEW
FORMAT( 1X,*YOU WILL CHANGE CONSTRAINT EQ. NUMBER *,I2,/, 1X,* TO *
E11.4,*. IS THIS WHAT YOU WANT? TYPE (YES OR NO)*,*/)
READ (1,724) KONFR
ANS = 1H*
IF (KONFR ,EQ, 3HYES) GO TO 948
IF (KONFR ,EQ, 2HNO) GO TO 938
IF(KONFR ,EQ, 4HSTOP OR, KONFR ,EQ, 4HSKIP) ANS = KONFR
IF(ANS ,EQ, KONFR ) RETURN
WRITE (2,904)
GO TO 945
X(KEQ+1,81)=XNEW
WRITE (2,950)
FORMAT IX,* ANY MORE CONSTRAINT CONSTANT CHANGES? TYPE (YES OR NO)*,*/)
READ (1,724) KONFR
ANS = 1H*
IF (KONFR ,EQ, 3HYES) GO TO 938
IF (KONFR ,EQ, 2HNO) GO TO 952
IF(KONFR ,EQ, 4HSTOP OR, KONFR ,EQ, 4HSKIP) ANS = KONFR
IF(ANS ,EQ, KONFR ) RETURN
WRITE (2,308)
GO TO 949
WRITE (2,954)
FORMAT( 1X,*ANY COEFFICIENT CHANGES? TYPE (YES OR NO)*,*/)
READ (1,724) KONFR
ANS = 1H*
IF (KONFR ,EQ, 3HYES) GO TO 910
IF (KONFR ,EQ, 2HNO) GO TO 956
IF(KONFR ,EQ, 4HSTOP OR, KONFR ,EQ, 4HSKIP) ANS = KONFR
IF(ANS ,EQ, KONFR ) RETURN
WRITE (2,2908)
GO TO 952
WRITE (2,958)
FORMAT( 1X,*YOU MAY SELECT THREE DISPLAY MODES FOR THE RESULTS*,*/
1 X,*OF THE CHANGES YOU JUST MADE.*,*/
1 X,*1, FULL TABLEAU FOR ALL ITERATIONS.*,*/
1 X,*2, FULL TABLEAU FOR ONLY FINAL ITERATION.*,*/
1 X,*3, VALUE OF OBJECTIVE FUNCTION AND LIST OF BASIS.*,*/)
ITNO=0
WRITE (2,960)
FORMAT( 1X,*SELECT THE MODE YOU WANT. TYPE (1, 2, OR 3)*,*/)
READ (1,724) KONFR
ANS = 1H*
IF (KONFR ,EQ, 1H1) GO TO 962
IF (KONFR ,EQ, 1H2) GO TO 964
IF (KONFR ,EQ, 1H3) GO TO 956
IF(KONFR ,EQ, 4HSTOP OR, KONFR ,EQ, 4HSKIP) ANS = KONFR
IF(ANS ,EQ, KONFR ) RETURN
WRITE (2,908)
GO TO 959
KFLG=1
RETURN
KFLG=2
RETURN
966 KFLG=3
RETURN
END
SUBROUTINE SUB1
COMMON/S1/KS1,KS2,IEXP
COMMON/COMMON/COMMON/COMMON
SUB1 AND SUB2 ARE ERROR ROUTINES TO CATCH ERRORS IN USER-INPUT
SUB1 NUMBERS WITH BAD OR ILLEGAL FORMAT. PREVENTS A FATAL ERROR
FROM TERMINATING THE PROGRAM. IF USER INPUT 12,36 INSTEAD
OF 12.36 THE ERROR ROUTINES WILL ALLOW USER TO CORRECT THE ERROR
AND CONTINUE EXECUTING.
IF (IEXP .EQ. 3*HEX) GO TO 5
WRITE (2,1)
1  FORMAT( 1X, *YOU HAVE MADE A MISTAKE IN THE ABOVE NUMBER, */,
1 1X, *TRY AGAIN.*)
GO TO KS1
5  WRITE (2,7)
7  FORMAT( 1X,*YOU HAVE MADE A MISTAKE IN ONE OF*,/,
1 1X,*THE NUMBERS IN THE ABOVE LIST*,*/,
1 1X,*RETYPE THE ENTIRE LIST*,*/,
1 1X,*FIRST DEPRESS THE RETURN KEY.*)
GO TO KS1
3  CONTINUE
RETURN
END
SUBROUTINE SUB2
COMMON/S1/KS1,KS2,IEXP
IF (IEXP .EQ. 3*HEX) GO TO 6
WRITE (2,2)
2  FORMAT( 1X,*YOU HAVE MADE A MISTAKE IN THE ABOVE NUMBER, */,
1 1X,*THE MAGNITUDE IS TOO BIG FOR THE COMPUTER*,*/,
1 1X,*CHECK THE POSITION OF THE E AND TRY AGAIN.*)
GO TO KS2
6  WRITE (2,9)
9  FORMAT( 1X,*ONE OF THE NUMBERS IN THE ABOVE STRING*,/,
1 1X,*IS TOO BIG FOR THE COMPUTER. CHECK THE POSITION*,/,
1 1X,*OF THE E AND RETYPE THE ENTIRE LIST*,/,
1 1X,*FIRST DEPRESS THE RETURN KEY.*)
GO TO KS2
4  CONTINUE
RETURN
END
OVERLAY (TAPE20,4,0)
PROGRAM PRESIMP
COMMON/COMMON/COMMON/COMMON
COMMON TEXT/ TEXT(60), ICHA, ILINE, ANS
DIMENSION QUIZ1(5), QUIZ2(8)
DATA (QUIZ1(I), I=1,5)/1HT, 1HF, 1HT, 1HF, 4*1HT/
DATA (QUIZ2(I), I=1,8)/1HT, 1HF, 1HT, 1HF, 4*1HT/
DO 99 I=1,151
99 READ(6,103)
WRITE(2,95)
95 FORMAT(/,/* THIS IS A REVIEW OF THE SIMPLEX ALGORITHM,*/
WHERE = 3
99 READ(5,121) (TEXT(I), I=1,63), ICHAR, ILINE
WRITE (2,101) (TEXT(I), I=1,ILINE)
DO 102 J=2,ILINE
READ(6,103) (TEXT(I), I=1,ILINE)
WRITE(2,131)
102 WRITE(2,131)
96 FORMAT(/)
IWHERE = IWHERE + ILINE
READ(1,103) ANS
IF(ANS .EQ. 4HSTOP) GO TO 155
IF(ANS .EQ. 4H/VCHR) GO TO 124
IF(ILINE .EQ. 7) GO TO 111
IF(ILINE .EQ. 6) GO TO 114
IF(ANS .EQ. 2HGO) GO TO 99
WRITE(2,110) GO TO 1025
IF(ANS .EQ. 6H/CONVEX) GO TO 99
IF(ANS .EQ. 2HGO) GO TO 112
GO TO 113
READ(6,100)
IWHERE = IWHERE + 39
GO TO 125
IF(ANS .EQ. 6HC/REV) GO TO 117
IF(ANS .EQ. 7H/CHICKEN) GO TO 115
WRITE(2,110) GO TO 1025
DO 116 J=1,8
READ(6,100) IWHERE = IWHERE + 8
GO TO 125
CCCCcccCccccccccccccccccccccccc
C THE FOLLOWING ADMINISTER A FIVE QUESTION TRUE/FALSE EXAM.
J=2
ISCOPE = 0
DO 118 I=1,5
DO 1195 KK=1,J
READ(5,103) (TEXT(K), K=1,60), ICHAR
WRITE(2,111) (TEXT(K), K=1,ICHAR)
WRITE(2,96)
DO 1171 READ(1,103) ANS
IF(ANS .NE. 1H/AND. ANS .NE. 1H/Go TO 177
IF(ANS .EQ. QUIZ(I)) GO TO 1175
IF(ANS .EQ. 4HSTOP) GO TO 155
WRITE(2,134)
ISCOPE = ISCOPE - 1
GO TO 1189
WRITE(2,110) GO TO 1171
WRITE(2,135)
DO 1189 I=1,5
DO 1185 KK=1,J
READ(5,103) (TEXT(K), K=1,60), ICHAR
WRITE(2,111) (TEXT(K), K=1,ICHAR)
WRITE(2,96)
CONTINUE
IWHERE = IWHERE + 8
IF(ISCOPE .EQ. 0) WRITE(2,120)
IF(ISCOPE .EQ. 1) WRITE(2,121)
IF(ISCOPE .EQ. 2) WRITE(2,122) ISCOPE
CCCCccccccccccccccccccccccccccc
C THE FOLLOWING CONTROL RELLOCATION TO THE PROPER POINT IF THE USER
C DESIRES TO SKIP TO THE REVIEW SECTION.
GO TO 125
J=75 - IWHERE
IF(J .LE. L) GO TO 132
DO 1245 I=1,J
READ(6,103) (TEXT(I), I=1,60), ICHAR, ILINE, IRET
WRITE(2,101) (TEXT(I), I=1,ICHAR)
DO 130 J=2,ILINE
WRITE(2,101) (TEXT(I), I=1,ICHAR)
WRITE(2,96)
WHERE = WHERE + ILINE

131 READ (1,117) ANS
IF(ANS .EQ. 4HSKIP ,09 , ANS .EQ. 4HSTOP) GO TO 155
IF(ANS .EQ. 6HREVIEW) GO TO 124
IF(IRET .EQ. 1) GO TO 133
IF(IRET .EQ. 2) GO TO 135
IF(ANS .EQ. 2HGO) GO TO 125
132 WRITE (2,110)
GO TO 131
133 IF(ANS .EQ. 4HELP) GO TO 125
IF(ANS .EQ. 5HSTEADY) GO TO 140
IF(ANS .NE. 2HGO) GO TO 132
DO 134 J=1,24
134 READ (6,100)
GO TO 125
135 IF(ANS .EQ. 5HPIVOT) GO TO 125
IF(ANS .NE. 2HGO) GO TO 132
DO 136 J=1,41
136 READ (6,100)
CONTINUE

THE FOLLOWING ADMINISTER AN EIGHT QUESTION TRUE/FALSE EXAM.

140 ISCORE = 0
DO 150 I=1,8
150 READ (6,109) (TEXT(J) , J=1,60), ICHAR, ILINE
WRITE (2,121) (TEXT(J), J=1, ICHAR)
DO 151 K=2,ILINE
READ (6,109) (TEXT(J), J=1,60), ICHAR
151 WRITE (2,121) (TEXT(J), J=1, ICHAR)
WRITE (2,96)
READ (1,103) ANS
IF(ANS .EQ. 4HSTOP) GO TO 155
IF(ANS .NE. 1HRT AND . ANS .NE. 1HF) GO TO 152
IF(ANS .EQ. 0QUIZ2(I)) GO TO 153
ISCORE = ISCORE + 1
WRITE (2,104)
GO TO 150
152 WRITE (2,110)
GO TO 1515
153 WRITE (2,105)
150 CONTINUE
IF(ISCORE .EQ. 8) WRITE (2,120)
IF(ISCORE .EQ. 7) WRITE (2,121)
IF(ISCORE .GE. 1) WRITE (2,122) ISCORE
155 REWIND 6
RETURN

FORMAT (6041, I2, 2X, I3, I1)
1*1 FORMAT (1X, 6041)
103 FORMAT (A10)
104 FORMAT (1X, *INCORRECT*/)
105 FORMAT (1X, *CORRECT*/)
110 FORMAT (1X, *I DO NOT UNDERSTAND. P'TYPE YOUR ENTRY))*
12J FORMAT (1X, *YOU GOT THEM ALL. CONGRATULATIONS,*/
121 FORMAT (1X, *YOU MISSED 1. PRETTY GOOD,*/
122 FORMAT (1X, *YOU MISSED *,I1, *. LOOK IN A TEXT FOR HELP,*/
END
OVERLAY (TAPL20,7,0)
PROGRAM PRETRAN

THIS SUBROUTINE CONTROLS PRESENTATION OF MATERIAL REVIEWING THE
TRANSPORTATION ALGORITHM.
COMMON/TEXT/ TEXT(60), ICHAR, ILINE, ANS
DIMENSION RESULT(8)
DATA (RESULT(I), I=1,8) / 1HT, 2*1HF, 1HT, 4*1HF/
WRITE (2,95)
THIS IS A REVIEW OF THE TRANSPORTATION ALGORITHM.

READ(9,100) (TEXT(I), I=1,60), ICHAR, ILINE, IRET
WRITE(2,101) (TEXT(I), I=1,60), ICHAR
DO 102 J=2, ILINE
READ(9,103) (TEXT(I), I=1,60), ICHAR
WRITE(2,101) (TEXT(I), I=1,60), ICHAR
WRITE(2,96)

FORMAT(/
1027 READ(1,103) ANS
IF(IRET .NE. 0) GO TO 105
IF(ANS .EQ. 2HGO) GO TO 99
IF(ANS .EQ. 4HSKIP .OR. ANS .EQ. 4HSTOP .OR. ANS .EQ. 6HRETURN)
+GO TO 105
1935 WRITE(2,104)
GO TO 1027

THE FOLLOWING CONTROL LOCATION THE THE TEXT DEPENDING ON THE
SUBSECTION SELECTED.
105 REWIND 9
K=0
IF(ANS .EQ. 1H1) K=118
IF(ANS .EQ. 1H2) K=279
IF(ANS .EQ. 1H3) K=169
IF(ANS .EQ. 1H4) K= 66
IF(ANS .EQ. 1H5) K= 46
IF(ANS .EQ. 1H6 .OR. ANS .EQ. 4HSTOP .OR. ANS .EQ. 4HSKIP) RETURN
IF(ANS .EQ. 6HRETURN) GO TO 130
IF(K .EQ. 0) GO TO 1035
113 DO 113 J=1,K
READ(9,103) TEXT(I), I=1,60), ICHAR
IF(ANS .NE. 1H5) GO TO 99
ISCORE = 0

ADMINISTER AN EIGHT QUESTION TRUE/FALSE EXAM.
DO 115 J=1,3
READ(9,103) TEXT(I), I=1,60), ICHAR
WRITE(2,101) (TEXT(I), I=1, ICHAR)
WRITE(2,96)
READ(1,103) ANS
IF(ANS .NE. 1HF .AND. ANS .NE. 1HT) GO TO 1165
IF(ANS .NE. RESULT(J)) GO TO 118
WRITE(2,117)
GO TO 115
1165 WRITE(2,114)
GO TO 1165
119 WRITE(2,119)
ISCORE = ISCORE + 1
115 CONTINUE
IF(ISCORE .EQ. 2) WRITE(2,120)
IF(ISCORE .EQ. 1) WRITE(2,121)
IF(ISCORE .GT. 1) WRITE(2,122) ISCORE
WRITE(2,123)
134 WRITE(2,124)
IRET = 1
GO TO 1027
140 FORMAT(6G1, I2, 2X, I3, I1)
121 FORMAT(1X, 60A1)
103 FORMAT(A11)
104 FORMAT(1X,*I DO NOT UNDERSTAND. RETYPE YOUR RESPONSE.*/)
117 FORMAT(1X,*CORRECT*/)
119 FORMAT(IX, '*INCORRECT*/)
120 FORMAT(IX, '*YOU GOT THEM ALL. CONGRATULATIONS***/)
121 FORMAT(IX, '*YOU MISSED ONE. PRETTY GOOD.*/)
122 FORMAT(IX, '*YOU MISSED *, I1, *, YOU NEED MORE WORK IN THIS *
   **AREA.*/)
123 FORMAT(* THIS COMPLETES THE SELF TEST.*)
124 FORMAT(* CHOOSE ANOTHER TRANSPORTATION AREA TO INVESTIGATE.*/
   ** TYPE (1, 2, 3, 4, 5, OR 6)*/)
END

PROGRAM SENSE

C THIS SUBROUTINE CONTROLS PRESENTATION OF THE MATERIAL REVIEWING
C SENSITIVITY ANALYSIS.
COMMON/TEXT/ TEXT(60), ICHAR, ILINE, ANS
WRITE(2,95)
95 FORMAT(/,
   ' THIS IS A REVIEW OF SENSITIVITY ANALYSIS.*/
99 READ(7,100) (TEXT(I),I=1,60), ICHAR, ILINE,IPET
WRITE(2,101) (TEXT(I),I=1,ICHAR)
DO 102 J = 2, ILINE
   READ(7,100) (TEXT(I),I=1,60), ICHAR
WRITE(2,101) (TEXT(I),I=1,ICHAR)
WRITES(2,96).
96 FORMAT(//)
1025 READ(1,133) ANS
   IF(IPET .EQ. 1 OR. ANS .EQ. 4HSKIP OR. ANS .EQ. 6HRETURN OR. ANS .EQ. 4HSTOP) RETURN
   IF(ANS .EQ. 2HGO) GO TO 99
1025 WRITE (2,104)
GO TO 1025

PROGRAM ASSIGN

C THIS SUBROUTINE CONTROLS PRESENTATION OF MATERIAL DISCUSSING THE
C OPTIMAL ASSIGNMENT PROBLEM.
COMMON/TEXT/ TEXT(60), ICHAR, ILINE, ANS
DO 94 I=1,121
94 READ(7,100)
WRITE(2,95)
95 FORMAT(/,
   ' THIS IS A REVIEW OF THE OPTIMAL ASSIGNMENT *
C THIS SUBROUTINE ALLOWS THE USER TO ENTER AND SOLVE A TRANSPORTATION
C PROBLEM CONSISTING OF NO MORE THAN SIX SOURCES AND SIX DESTINATIONS.

EXTERNAL SUB1
EXTERNAL SUB2
COMMON/COM1/X(21,31), REDUCE(21,61), MONBASE(60)
INTEGER CJNSEN(21), SLACK(21), SURPLUS(21), ARTIF(21), BASE(21),
STRU.SUR, ART, PCONT, HLT
COMMON/COM2/CONS, SLACK, SURPLUS, ARTIF, BASE, SLA, SUR, ART, PCONT, HLT
COMMON/COM4/NOINF, KFLG
COMMON/COM5/1KS1, KS2, IEXP, IFOUR
COMMON /TEXT/ TEXT(60), ICHAR, ILINE, ANS
DIMENSION M$1(6), NS2(6)
DATA NS1/0, L, 0, 0, C, 0/
DATA NS2/0, C, 0, 3, 0, 0/
NS1(5) = LOC$F(SUB1)
NS2(6) = LOC$F(SUB2)
CALL SYSTEMC(73, M$1)
CALL SYSTEMC(79, NS2)
ILINE = 9
6998 IF (ILINE .NE. 0) GO TO 6999
WR1TE(2,7307)
WRITE(2,7001)
DO 109 I = 1, 3042
109 X(I) = 0
6999 WRIT£(2,7307)
7000 FORMAT(///" YOU MAY NOW SELECT A TRANSPORTATION PROBLEM FOR *
+ solution by */ , 1X , *ENTERING PARAMETERS AS THEY ARE REQUESTED , **,
+ there are two */ , 1X , *OPTIONS FOR ENTERING PARAMETERS. THE *,
+ regular method uses a */ , 1X , *STEP BY STEP PROCEDURE THAT *
+ insures accuracy, the expedite*
7001 FORMAT(1X, "METHOD IS FASTER, BUT SHOULD BE USED AFTER YOU *
+ ATTAIN FAMILIARITY*, */ , 1X , *WITH THIS PROGRAM,*)
7002 FORMAT(1X, "SELECT THE OPTION YOU DESIRE", */ , 1X,
+ type (REG OR EXP)*/ , )
ILINE = 1
7004 READ(1,7002) ANS
7005 FORMAT(A10)
IF(ANSEQ.3HREG.OP.ANS.EQ.3HEXP)GOTO7JC5
IF(ANSEQ.4HSTOP.OR.ANS.EQ.4HSKIP)RETURN
WRITE(2,7303)
GOTO7604
7603FORMAT(1X,*IDONOTUNDERSTAND.REENTERYOURRESPONSE.*),/
7605ASSIGN7311TOKSI
ASSIGN7011TOKSI
IEXP=0
CCCCCCCCCCCCCCCCCCCCCCCC
CENTER THE AMOUNT OF SOURCES AND DESTINATIONS INVOLVED.
7005FORMAT(*X,*ENTER THE NUMBER OF SOURCES AND NUMBER OF DESTINATION
+SY*IN*,/1X,*I1FORMAT THAT IS,1,2,ETC.*/,/IX,+
*THE MAXIMUM NUMBER OF EITHER IS 6,*),/
7006FORMAT(1X,*NUMBER OF SOURCES = *),/
7009FORMAT(1X,*NUMBER OF DESTINATIONS = *),/
WRITE(2,7106)
7111WRITE(2,7309)
READ(1,7010)ISOURCE
IF(ISOURCE.GT.6)GOTO7028
7010FORMAT(I1)
ASSIGN7012TOKSI
ASSIGN7012TOKSI
7012WRITE(2,7309)
READ(1,7010)IDEST
IF(IDEST.GT.6)GOTO7028
IF(ANSEQ.3HEXP)GOTO7030
CCCCCCCCCCCCCCCCCCCC
CENTER THE TRANSPORTATION COST OF GOING FROM EACH SOURCE TO EACH
CDESTINATION.
7013FORMAT(3X,*ENTER THE COST OF GOING FROM SOURCE I TO DESTINATION
+JI*IN*,/1X,*11FORMAT THAT IS,1.2,ETC.*,/IX,
*LATTER, IT MUST*,/*.*EXACTLY AS SHOWN,*),/
ASSIGN714TOKSI
ASSIGN714TOKSI
KA=ISOURCE+IDEST+1
DO7015I=1,ISOURCE
DO7015J=1,IDEST
K=(I-1)*IDEST+J
7014WRITE(2,7016)I,J
READ(1,7017)X(KA,K)
7015CONTINUE
7016FORMAT(1X,*C(*,I1*,*,I1,*)=*),/
7017FORMAT(E11.4)
WRITE(2,7320)
7020FORMAT(3X,*WHEN REQUESTED, ENTER THE AMOUNT AVAILABLE AT EACH SOUR
+CE*IN*,/,*E11.4FORMAT,*),/
KA=ISOURCE+IDEST+1
ASSIGN7027TOKSI
ASSIGN7023TOKSI
DO7021I=1,ISOURCE
7023WRITE(2,7922)I
READ(1,7017)X(I,KA)
7021CONTINUE
7022FORMAT(1X,*SOURCE*,II,*,II,*)=*),/
WRITE(2,7925)
7025FORMAT(*WHEN REQUESTED, ENTER THE AMOUNT REQUIRED AT EACH DEST
+INATION*,/,*E11.4FORMAT,*),/
ASSIGN7029TOKSI
ASSIGN7029TOKSI
DO7026I=1,IDEST
7029WRITE(2,7927)I
\[ K_A = ISOURCE \times IDEST + 1 \]

7026 READ(1,X) I(SOURCE+I,KAA)

7027 FORMAT(* DESTINATION*, I1 = *)

CC CC CCC CCCCC CCCCC CCCCC CCC

CC NOW ALL ARE ENTERED IN THE REG FORMAT.

GO TO 7046

7028 WRITE(2,7030)

7030 FORMAT(* TOO BIG, SIX IS THE MAXIMUM NUMBER, */)

GO TO KS1

CC CCCCC CCCCC CCCCC CCCCC CCCCC

C EXEDITE ENTRY POINT.

7031 IEXP=3HEXP

ASSIGN 7333 TO KS1

ASSIGN 7033 TO KS2

DO 7032 J=1,ISOURCE

7032 READ(1,X) I(KAA+1,J)

ASSIGN 7037 TO KS1

ASSIGN 7037 TO KS2

WRITE(2,7035)

7035 FORMAT(* ENTER, IN ORDER, THE COSTS OF GOING FROM SOURCE *, I1, + TO ALL DESTINATIONS, */ , * USE E11.4 FORMAT, */)

KAA = ISOURCE + IDEST + 1

DO 7036 K=1, IDEST

7036 Y(I,KAA)

ASSIGN 7042 TO KS1

ASSIGN 7042 TO KS2

7037 WRITE(2,7040)

7040 FORMAT(* ENTER, IN ORDER, THE AMOUNT OF RESOURCES AVAILABLE AT EACH SOURCE, */ , * USE E11.4 FORMAT, */)

KAA = ISOURCE + IDEST + 1

DO 7041 I=1, IDEST

7041 READ(1,X) I(I,KAA)

ASSIGN 7045 TO KS1

ASSIGN 7045 TO KS2

7042 WRITE(2,7045)

7045 FORMAT(* ALL DATA WILL NOW BE REPRODUCED IN TABLE FORM, */ * )

IF(IDEST .GT. 5) GO TO 7950

K = IDEST

WRITE(2,7047) (J,J=1,K)

7047 FORMAT(*, I4X, * DESTINATION*, I1, 12X, I1, 4(12X, I12))

KAA = ISOURCE + IDEST + 1

DO 7050 I=1, ISOURCE

7050 WRITE(2,7048) I, (X(KAA,KAA+J), J=1,K)

7048 FORMAT(1X, I1, 5(2X, E11.4))

GO TO 7055

7055 K = 1

KAA= ISOURCE + IDEST + 1

WRITE(2,7049) (J,J=1,K)

7049 FORMAT(*, I4X, * DESTINATION*, I1, 12X, I1, 4(12X, I12))

KAA= (I-1)*IDEST

WRITE(2,7048) I, (X(KAAA,KAA + J), J=1,K)

K = K+6

7051 K = K+4

WRITE(2,7053)

7053 FORMAT(* DESTINATION*, I1, * SOURCE 6*)

K = 6
KAAA = ISOURCE + IDEST + 1
DO 7952 I = 1, ISOURCE
WRITE(2,7954) I, X(KAAA,K)
7952 K = K + 6
7954 FORMAT(3X,I1, 4X, E11.4)
7955 WRITE(2,7952)
7052 FORMAT(/, * SOURCE*, I1X, * DESTINATION*)
K = ISOURCE
KAAA = ISOURCE * IDEST + 1
IF(ISOURCE .LT. IDEST) K = IDEST
DO 7056 I = 1, K
IF(I .LE. ISOURCE .AND. I .LE. IDEST) WRITE(2,7057) I, X(I,KAAA), I,
+ X(ISOURCE + I - 1, KAAA)
IF(I .LT. ISOURCE .AND. I .GT. IDEST ) WRITE(2,7057) I, X(I,KAAA)
7056 IF(I .GT. ISOURCE .AND. I .LE. IDEST) WRITE(2,7058) I, X(ISOURCE + I-1,
+ KAAA)
7057 FORMAT(I1X,2(I1,2X,E11.4,2X))
7058 FORMAT(17X, II, 2X, E11.4)
00 TO 7053
CCC CCC CCC CCC CCC CCC
C THIS PORTION CONTROLS THE MODIFICATION OF ANY PARAMETERS IN THE
C EDIT TABLE.
7053 WRITE(2,7059)
7059 FORMAT(/, * DO YOU DESIRE TO CHANGE ANY OF THE ABOVE? */
+ * TYPE (YES OP NO) */
7060 READ(1,7061) ANS
IF(ANS .EQ. 'HST0P' OR. ANS .EQ. 'HSKIP') RETURN
IF(ANS .EQ. '2HNO') GO TO 7130
IF(ANS .EQ. '3HYES') GO TO 7065
WRITE(2,7003)
GO TO 7063
7065 WRITE(2,7066)
7066 FORMAT(/, * IF YOU DESIRE TO CHANGE A COST, TYPE COST*, /
+ * IF YOU DESIRE TO CHANGE A RESOURCE, TYPE RESOURCE*, /
+ * IF YOU DESIRE TO CHANGE A DESTINATION, TYPE DEST*/)
7067 READ(1,7068) ANS
IF(ANS .EQ. 'HST0P' OR. ANS .EQ. 'HSKIP') RETURN
IF(ANS .EQ. '4HCOST') GO TO 7070
IF(ANS .EQ. '3HRESOURCE') GO TO 7080
IF(ANS .EQ. '4HDEST') GO TO 7085
WRITE(2,7003)
GO TO 7067
7070 WRITE(2,7071)
7071 FORMAT(/, * YOU INDICATED YOU DESIRE TO CHANGE A TRANSPORTATION COST
+ I, */ * A COST IS FROM SOURCE I TO DESTINATION J, */ *
+ * ENTER THE SOURCE IN I1 FORMAT,* / * SOURCE NUMBER = *)
ASSIGN 7072 TO KS1
ASSIGN 7072 TO KS2
7072 READ(1,7073) I
WRITE(2,7073)
7073 FORMAT(/, * ENTER THE DESTINATION IN I1 FORMAT,* / *
+ * DESTINATION = *)
ASSIGN 7075 TO KS1
ASSIGN 7075 TO KS2
7075 READ(1,7076) J
WRITE(2,7076) I, J, I, J
7076 FORMAT(/, * NOW ENTER THE NEW COST OF GOING FROM SOURCE *, I1,
+ * TO DESTINATION *, I1, */ IN E11.4 FORMAT,* /
+ * COST*, I1, *, I1, *) = *)
ASSIGN 7077 TO KS1
ASSIGN 7077 TO KS2
K = (I-1)*IDEST+J
KAAA = ISOURCE + IDEST + 1
7077 READ(1,7077) X(KAAA,K)
7277 WRITE(2,7078)
7078 FORMAT(/,'DO YOU DESIRE TO MAKE FURTHER CORRECTIONS?*/,'* TYPE (YES OR NO)*/')
7079 READ(1,7092) ANS
   IF(ANS .EQ. '4HSTOP', .00, ANS .EQ. '4HSKIP') RETURN
   IF(ANS .EQ. '3HYES') GO TO 7065
   IF(ANS .EQ. '2HNO') GO TO 7096
   WRITE(2,7093)
   GO TO 7079
7080 WRITE(2,7181)
7081 FORMAT(/,'YOU INDICATED YOU DESIRE TO CHANGE A RESOURCE QUANTITY.*Printer the RESOURCE NUMBER IN II FORMAT.*','* RESOURCE IS *)
   ICHAR = 0
   GO TO 7088
7085 WRITE(2,7086)
7086 FORMAT(/,'YOU INDICATED YOU DESIRE TO CHANGE A DESTINATION QUANTITY.*Printer the DESTINATION NUMBER IN II FORMAT.*','* DESTINATION IS *)
   ICHAR = 1
7098 ASSIGN 7096 TO KS1
   ASSIGN 7099 TO KS2
7090 READ(1,7C10) I
   IF(ICHAR .EQ. 0) WRITE(2,7091) I, I
   IF(ICHAR .EQ. 1) WRITE(2,7092) I, I
7091 FORMAT(/,'ENTER THE CHANGE IN THE RESOURCE *,II,* QUANTITY IN E11',
   +,4 FORMAT.*,*/,* RESOURCE *,II,* = *)
7092 FORMAT(/,'ENTER THE CHANGE IN THE DESTINATION *,II,* QUANTITY IN ',
   +E11,4 FORMAT.*,*/,* DESTINATION *,II,* = *)
   ASSING 7095 TO KS1
   Assign 7095 TO KS2
   KA = ISOURCE * IDEST + 1
7095 IF(ICHAR .EQ. 0) READ(1,7017) X(I,KA)
   IF(ICHAR .EQ. 1) READ(1,7117) X(I+ISOURCE, KA)
   GO TO 7277
7096 WRITE(2,7197)
7097 FORMAT(/,'DO YOU DESIRE TO HAVE THE EDIT TABLE RE-PRINTED?*/,'* TYPE (YES OR NO)*/')
7098 READ(1,7C12) ANS
   IF(ANS .EQ. '4HSTOP', .00, ANS .EQ. '4HSKIP') RETURN
   IF(ANS .EQ. '2HNO') GO TO 7100
   IF(ANS .EQ. '3HYES') GO TO 7046
   WRITE(2,7093)
C THE EDIT TABLE HAS BEEN APPROVED BY THE USER.
C A DUMMY SOURCE OR DESTINATION IS ADDED IF NEEDED.
   GO TO 7098
CCCCCCcccccccccccccccc
7100 X(21,81) = 0
   X(20,81) = 0
   KA = ISOURCE * IDEST + 1
   DO 7110 I = 1, IDEST
7113 X(21,81) = X(21,81) + X(ISOURCE+I, KA)
   DO 7110 I = 1, IDEST
7111 X(20,81) = X(20,81) + X(I, KA)
   IF(X(21,81) .GT. X(20,81)) GO TO 7114
   IF(X(21,81) .EQ. X(20,81)) GO TO 7120
   DO 7112 I = 1,40
   X(I,KA + ISOURCE+2,I) = X(I,KA+ISOURCE+1,I)
7112 X(I,IDEST+ISOURCE+1,I) = 0
   IDEST = IDEST + 1
   X(I,IDEST+ISOURCE, KA) = X(23,81) - X(21,81)
   DO 7300 I = 1,20
   X(I,KA + ISOURCE) = X(I,KA)
7300 X(I,KA) = 0
   GO TO 7120
7114 DO 7113 I = 1,40
  X(IDEST+ISOURCE+2,I) = X(IDEST+ISOURCE+1,I)
7113 X(IDEST+ISOURCE+1,I) = 0
    J = ISOURCE + IDEST + 2
    DO 7115 I = 1, IDEST
7115 X(J-I, KA) = X(J-I-I, KA)
    ISOURCE = ISOURCE + 1
    X(ISOURCE , KA ) = X(21,81) - X(20,81)
    DO 7301 I=1,20
    X(I,KA+IDEST) = X(I,KA)
7301 X(I,KA) = 0
C NEXT THE CONSTRAINT EQUATIONS ARE SET UP SO THAT THE PROBLEM
C CAN BE SOLVED USING THE SIMPLEX ALGORITHM IN SUBROUTINE SIMPLER.
7120 DO 7200 I=1, ISOURCE
    DO 7200 J=1, IDEST
    K = (I-1)*IDEST+J
    DO 7201 I=1, ISOURCE
      K=(I-1)*IDEST+J
    7201 X(I SOU RCE , J ,  K ) = 1.0
    X(21,1) = ISOURCE + IDEST
    X(21,2) = ISOURCE * IDEST
    X(21,3) = ISOURCE
    X(21,4) = IDEST
    CALL SIMPLER
    ISOURCE = X(21, 3 )
    IDEST = X(21,4)
    DO 7212 1  =  1,ISOURCE
C THE FOLLOWING PRINT THE FINAL SOLUTION TO THE TRANSPORTATION
C PROBLEM AND DETERMINE IF THE USER HAS ANOTHER PROBLEM HE
C DESIRES TO SOLVE.
    DO 7212 J=1,IDEST
    K=(J-M)*IDEST
    ANS = X(23, K)
    IF(ANS ,EQ. 0) GO TO 7212
    WRITE(2,7213) I,J,ANS
    7212 CONTINUE
    7213 FORMAT( *X( ,  ,  *= ,  Ell.4 )
    WRITE(2,7214)
7214 FORMAT(//,** ALL THE OTHER PATHS HAVE ZERO ALLOCATIONS,**,///,
          ** DO YOU WISH TO ATTEMPT ANOTHER TRANSPORTATION PROBLEM?,//,
          ** TYPE (YES OR NO)*,//)
7215 READ(1,7002) ANS
    IF(ANS ,EQ. 4HSTOP .OP. ANS ,EQ. 4HSKIP .OR. ANS .EQ. 2HNO) RETURN
    IF(ANS ,EQ. 3HYES) GO TO 6993
    WRITE(2,7003)
    GO TO 7215
    END
SUBROUTINE SUB1
C SU1 AND SUB2 ARE ERROR ROUTINES TO CATCH ERRORS IN USER-INPUT
C NUMBERS WITH BAD OR ILLegal FORMAT. PREVENTS A FATAL ERROR
C FROM TERMINATING THE PROGRAM. IE, IF USER INPUT 12,36 INSTEAD
C OF 12.36 THE ERROR ROUTINES WILL ALLOW USER TO CORRECT THE ERROR
C AND CONTINUE EXECUTING.
COMMON/Sl/KS1,KS2,IEXP
  IF (IEXP .EQ. 3HXF) RETURN
  WRITE(2,1) 1
1 FORMAT( 1X,*YOU HAVE MADE A MISTAKE IN THE ABOVE NUMBER*, 1X,*TRY AGAIN,* )
  GO TO KS1
WRITE (2,7)
FROMAT (1X,"YOU HAVE MADE A MISTAKE IN ONE OF",/,
 1 1X,"THE NUMBERS IN THE ABOVE LIST",/,
 1 1X,"TYPE THE ENTIRE LIST",/,
 1 1X,"FIRST DEPRESS THE RETURN KEY")
GO TO KS1
7 CONTINUE
RETURN
END
SUBROUTINE SUB2
COMMON/1/KS1,KS2,EXP
IF (EXP .EQ. 3HFEXP) GO TO 6
WRITE (2,2)
2 FORMAT (1X,"YOU HAVE MADE A MISTAKE IN THE ABOVE NUMBERS",/,
 1 1X,"THE MAGNITUDE IS TOO BIG FOR THE COMPUTER",/,
 1 1X,"CHECK THE POSITION OF THE E AND TRY AGAIN")
GO TO KS2
WRITE (2,3)
3 FORMAT (1X,"ONE OF THE NUMBERS IN THE ABOVE STRING",/,
 1 1X,"IS TOO BIG FOR THE COMPUTER. CHECK THE POSITION",/,
 1 1X,"IF THE E AND RETYPE THE ENTIRE LIST",/,
 1 1X,"FIRST DEPRESS THE RETURN KEY")
GO TO KS2
CONTINUE
RETURN
END
SUBROUTINE SIMPLER
CCCCCCCCCCCCCCCCCCCCCCCC
C THIS SUBROUTINE APPLIES THE SIMPLEX ALGORITHM TO THE
C TRANSPORTATION PROBLEM.
INTEGER P,C,0,BASIC,PC,PR,R,BASIC,C,VAR,U,V,K
COMMON/COM1/A(21,31),BASIC(31),P(30),R(30),AA(30),W(30),VAP(30),
+CC(30)
M = A(21,1)
N = A(21,2)
CCCCCCCCCCCCCCCCCCCC
C COEFFICIENT MATRIX IS M X N, MM DENOTES WHETHER PROBLEM IS MAX OR MIN
B=N+1
CCCCCCCCCCCCCCCCCCCC
C COLUMN 0 = N+1 IS FOR THE B VECTOR
C = M+1
CCCCCCCCCCCCCCCCCCCC
C ROW C = M+1 IS FOR THE COST ROW
D=M+2
CCCCCCCCCCCCCCCCCCCC
C ROW D = M+2 IS FOR THE W ROW
C THE NEXT 9 STEPS ENTER THE DATA AND MAKE THE B VECTOR NON-NEGATIVE
DO 3 I=1,G
BASIC(I) = 0+I
W(I) = 1.0
IF (A(I,0) .GE. 0.0) 6,3,3
6 DO 7 J=1,R
7 A(I,J) = (-1.0)*A(I,J)
3 CONTINUE
DO 9 J=1,G
9 A(C,J) = (-1.0)*A(C,J)
CCCCCCCCCCCCCCCCCCCC
C THE NEXT 90 LOOP COMPUTES ROW M+2 (THE W ROW)
DO 12 J=1,R
VAR(J) = J
A(0,J) = 6.0
DO3121=1,4
12 A(0,J) = A(J,J) + A(I,J)*W(I)
CCCCCCCCCCCCCCCCCCCC
BEGIN PHASE ONE

THE NEXT 7 STEPS FIND THE MAX ELT, OS, IN ROW M+2
PC IS THE COLUMN NUMBER OF THE PIVOT COLUMN

DS = A(0, 1)
PC = 1
DO 22 J = 1, N
IF (A(0, J) = DS) 22, 21, 21
21 DS = A(0, J)
PC = J
22 CONTINUE

IF (DS - O) DS = DS, 23, 23, 24

OS = A(0, 1)
PC = J
DO 22 J = 1, N
IF (A(0, J) = OS) 22, 21, 21
21 OS = A(0, J)
PC = J
22 CONTINUE

IF (OS > 0) 23, 23, 24
IF (OS < 0) 23

THE NEXT 14 STEPS FIND THE MINIMUM RATIO FOR DETERMINING THE PIVOT ROW
PR IS THE ROW NUMBER OF THE PIVOT ROW
C THE FIRST 7 STEPS INSURE THAT ONLY POSITIVE RATIOS ARE CONSIDERED

K = 0
DO 26 I = 1, M
IF (A(I, PC) < 0) 26, 26, 23
28 K = K + 1
P(K) = A(I, PC) / A(I, PP)
R(K) = I
26 CONTINUE

PR = R(1)
P = P(1)
DO 34 L = 1, K
IF (P(L) = PR) 32, 34, 34
32 PR = P(L)
34 CONTINUE

PE = A(PR, PC)

THE NEXT DO LOOP SETS THE PIVOT ELT = 1

AA(J) = (1, R1) A(I, J)
DO 33 J = 1, N
33 AA(J) = A(J, PC)

THE NEXT DO LOOP ZEROS THE REST OF THE PIVOT COLUMN

DO 35 I = 1, N
CC(I) = A(I, PC)
DO 35 J = 1, N
35 AA(J) = A(J, PC) * CC(I)

THE NEXT DO LOOP ADJUSTS ROW M+2

DO 37 J = 1, N
A(PR, J) = AA(J)
A(D, J) = 0
37 D = D + 1

GO TO 25 ERROR

WRITE (2, 77) 
77 FORMAT (/// , * THERE IS NO FEASIBLE SOLUTION TO THIS PROBLEM *)

C THE NEXT DO LOOP ZEROS ALL COLUMNS WITH A NEGATIVE IN ROW M+2

DO 27 J = 1, N
IF (A(J, D) < 0) 27, 27, 27
27 D = D + 1

D = D + 1
36 \text{AI} \text{J}=i,0 \\
30 \text{CONTINUE}\text{CCCCCC}\text{CCCCCCCCCCCCC}
C \text{BEGIN PHASE TWO}\text{CCCC}\text{CCCCCCCCCCCCC}
C \text{THE NEXT 7 STEPS FIND THE MAX ELT, CS, IN ROW M+1}\text{CCCCCCC}
C \text{PC IS THE COLUMN NUMBER OF THE PIVOT COLUMN}\text{CCCCCCCCCCCCC}
40 \text{CS} = \text{AI(C,J)} \\
\text{PC}=1 \\
\text{DO 42 J=1,N} \\
\text{IF(AI(J,J)-CS)42,41,41} \\
41 \text{CS} = \text{AI(C,J)} \\
\text{PC}=J \\
42 \text{CONTINUE}\text{CCCCCCCCCCCCC}
C \text{IF CS IS POSITIVE, RECOMPUTE THE MATRIX (STEP 224)}\text{CCCCCCCCCCCCC}
C \text{IF CS IS NOT POSITIVE, THE ALGORITHM IS FINISHED, PRINT OUT Z (STEP 43)}\text{CCCCCCCCCCCCC}
C \text{THE NEXT 7 STEPS FIND THE MINIMUM RATIO FOR DETERMINING THE PIVOT ROW}\text{CCCCCCCCCCCCC}
C \text{PR IS THE ROW NUMBER OF THE PIVOT ROW}\text{CCCCCCCCCCCCC}
C \text{THE FIRST 7 STEPS INSURE THAT ONLY POSITIVE RATIOS ARE CONSIDERED}\text{CCCCCCCCCCCCC}
224 K=0 \\
\text{DO226} I=1,N \\
\text{IF(AI(I,PC)<=0)225,225,225} \\
225 K=K+1 \\
\text{P(K)} = \text{AI(I,J)}/\text{AI(I,PC)} \\
\text{R(K)} = I \\
226 \text{CONTINUE}\text{CCCCCCCCCCCCC}
C \text{THE NEXT DO LOOP SETS THE PIVOT ELT =1}\text{CCCCCCCCCCCCC}
C \text{THE NEXT DO LOOP ZEROS THE REST OF THE PIVOT COLUMN}\text{CCCCCCCCCCCCC}
233 \text{AA(J)} = \text{AI(PR,J)} \\
\text{DO235} J=1,C \\
\text{AA(J)} = \text{AI(I,PC)} \\
\text{DO 235 J=1,C} \\
235 \text{AA(J)} = \text{AI(I,PC)} \\
\text{DO 237 J=1,B} \\
237 \text{AA(J)} = \text{AI(PR,J)} \\
\text{GO TO 40}\text{CCCCCCCCCCCCC}
C \text{THE OPTIMAL SOLUTION IS}\text{STOP} \\
7210 \text{THE ALLOCATIONS ARE}\text{STOP} \\
7211 \text{CONTINUE} \\
\text{RETURN} \\
\text{OVERPLAY(TAPE20,11,0)} \\
\text{PROGRAM INTRO}\text{CCCC}\text{CCCCCCCCCCCCC}
C THIS SUBROUTINE CONTROLS THE PRESENTATION OF MATERIAL EXPLAINING LPLA9 OPERATION.

COMMON /TEXT/ TEXT(60), ICHAR, ILINE, ANS
WRITE(2,95)
95 FORMAT(\\\\\\\\** THIS IS AN INTRODUCTORY EXPLANATION OF THIS LAB.\*/\\\\\\\\**)
99 READ(6,100) (TEXT(I),I= 1,63), ICHAR, ILINE
WRITE(2,114) (TEXT(I), I=1,ICHAR)
DO 101 J=2, ILINE
READ(6,100) (TEXT(I), I=1,63), ICHAR
101 WRITE(2,114) (TEXT(I), I=1, ICHAR)
WRITE(2,96)
96 FORMAT(\\)
IF(ILINE.EQ.11) GO TO 104
111 READ(1,1"2) ANS
IF(ANS.EQ.2HGO) GO TO 99
IF(ANS.EQ.4HSKIP .OR. ANS.EQ.4HSTOP) GO TO 104
WRITE (2,103)
GO TO 111
104 REWIND 6
RETURN
100 FORMAT(60A1, 12, 2X, 13)
102 FORMAT(A10)
103 FORMAT(2X, 'RE-ENTER YOUR ANSWER. TYPE (GO OR SKIP)*/)
114 FORMAT(IX,60A1)
END

PROGRAM GRAPR

C THIS SUBROUTINE CONTROLS PRESENTATION OF THE GRAPHICAL PROGRAMMING EXAMPLE.

COMMON /TEXT/ TEXT(60), ICHAR, ILINE, ANS
DO 94 I=1,103
94 READ(6,101)
WRITE(2,95)
35 FORMAT(\\\\\\\\** THIS IS A GRAPHICAL LINEAR PROGRAMMING EXAMPLE.\*/\\\\\\\\**)
99 READ(6,100) (TEXT(I),I=1,63), ICHAR , ILINE
WRITE(2,114) (TEXT(I), I=1,ICHAR)
DO 101 J=2, ILINE
READ(6,100) (TEXT(I), I=1,63), ICHAR
101 WRITE(2,114) (TEXT(I), I=1, ICHAR)
WRITE(2,96)
96 FORMAT(\\)
IF(ILINE.EQ.6) GO TO 107
115 READ(1,102) ANS
IF(ANS.EQ.2HGO) GO TO 99
IF(ANS.EQ.4HSKIP .OR. ANS.EQ.4HSTOP) GO TO 107
WRITE (2,103)
GO TO 115
107 REWIND 6
RETURN
100 FORMAT(60A1, 12, 2X, 13)
102 FORMAT(A10)
105 FORMAT(2X, 'RE-ENTER YOUR RESPONSE. TYPE (GO OR SKIP)*/)
114 FORMAT(IX, 60A1)
END
**EOP**

The following general comments apply to your use of LP LAB.

1. Always begin typing your response from the position where the carriage has stopped.
2. To correct an error you can back space using upper case 0 and then retype over the error or you can void an entire line by depressing the Ctrl key and the X simultaneously. This will move the carriage to the left margin of a new line. You will begin typing at that position.
3. You must type all responses exactly as shown when selecting a reply from a list of options.
4. You will be using two types of numbers (integer and decimal). Integers may be one or two digits and must not have a decimal point. If a two digit integer is requested, it must be entered in the form (03). Decimal numbers can be either straight decimal or E format. For example +135.472 is equivalent to +1.3647E+2. The straight decimal number must contain a decimal point and may contain up to 11 characters counting the . and the sign if sign omitted. Numbers must have exactly the format shown above. The sign must be present.
5. At any time during problem solving you can terminate LP LAB by typing STOP whenever LP LAB is expecting a word command. LP LAB will not recognize STOP if it is expecting a number.
6. LP LAB can be aborted any time by depressing the Ctrl key and the Z simultaneously and then the A.
7. If LP LAB is inactive for more than about 90 sec. (except when waiting for your response) it may be malfunctioning. You may abort if you desire! But prior to aborting, you can try once to correct the problem by typing GO.

Linear Programming (LP) deals with the problem of allocating limited resources among competing activities in an optimal manner. All mathematical functions in LP must be linear functions. Would you like to try a test to see if you understand linear and non-linear functions?

Type (YES, NO, or SKIP).

After each of the following, type T for true if it is a linear function, F for false if it is not.

- **X1+X2+X3=14** Type (T or F)
- **(X1)CUBE+2X3=7** Type (T or F)
- **SQUARE ROOT(X1)+X2=7** Type (T or F)
- **X1+X2+X3=(7)SQUARED** Type (T or F)
- **14X1+22.3X3-72.39X4<=4.123** Type (T or F)

The mathematical statement of a general LP problem is as follows:

Find values of X1, X2, ..., XN which maximize (or minimize).

The linear function Z=C1X1+C2X2+...+CNXN

Subject to the restrictions:

A11X1+A12X2+...+A1NXN=B1
A21X1+A22X2+...+A2NXN<=B2
... ...
AM1X1+AM2X2+...+AMNXN=BM

Type (GO or SKIP)

Where Aij, Bj, and Cj are given constants, all Xi>=0, and * is a relational operator in the constraint equations may be
The function $Z$ that is to be optimized is the objective function. Would you like to see the formulation of a simple LP problem? Type (yes, no, or skip).

Objective: determining the minimum cost mix of milk, eggs, and beef to give the minimum nutritional daily requirements of vitamins A, C, and D using the following data:

- Milk (gallons): $1 \times 100 = 100$
- Beef (pounds): $1 \times 100 = 100$
- Eggs (dozens): $1 \times 100 = 100$

Vitamin requirements:

- Vitamin A: $1 \times 1 = 1$
- Vitamin C: $1 \times 10 = 10$
- Vitamin D: $1 \times 10 = 10$

Cost:

- Milk: $1.10$
- Beef: $0.50$
- Eggs: $1.10$

Type (go or skip).

Let $X_M$ be the number of gallons of milk, $X_B$ be the number of pounds of beef, and $X_E$ be the number of dozens of eggs in the daily diet. The objective is to minimize cost. Attempt to set up the objective function and the constraints for this problem on a separate sheet of paper. Then you can compare your results with the correct results.

Minimize $Z = 1.3X_M + 1X_B + 0.5X_E$ subject to the constraints:

1. $X_M + X_B + X_E \geq 1$
2. $100X_M + 11X_B + 9X_E \geq 50$
3. $10X_M + 100X_B + 10X_E \geq 10$

Note the direction of the inequalities since there must be a minimum amount of each type of vitamin. Again, $X_M \geq 0$, $X_B \geq 0$, and $X_E \geq 0$.

The following restrictions limit the application of linear programming:

1. Proportionality. The objective function and every constraint function must be linear.
2. Additivity. Any mix of elements must not change the original relationships. If a $C_i$ is associated with an $X_i$, the $C_i$ must not change no matter what quantity of $Y_i$ is specified.
3. Divisibility. Fractions of variables must be allowed.
4. Deterministic. All coefficients in the LP model (AI, BJ, CI) are assumed to be known constants.
5. All $X_i \geq 0$. Note that if you require a different relationship, say $X_i = -10$, you may use the relationship $X_i = X_i + 10$ in the linear program where $X_i = X_i + 10$.

Consider a linear function $Y = AX_1 + BX_2 + CX_3$.

Traditional optimization requires taking partial derivatives with respect to each variable, setting the resulting equations to zero, and solving them simultaneously to give an optimum for each variable. Type (go or skip).

There are no values of $X_i$ in a linear function which make the partial derivatives vanish—a necessary condition for an internal optimum. Therefore, since there are no discontinuities in the function, the optimum can only lie at a boundary. If extreme values for a linear function are desired, restrictions must be imposed. These are the constraint equations.

Prior to examining the multidimensional solution techniques, consider a two-dimensional linear programming problem that may be solved graphically.

Maximize $Z = 2X_1 + 5X_2$ subject to

- $X_1 \leq 4$
- $X_2 \leq 6$
- $3Y_1 + 2Y_2 \leq 18$
- As usual, $X_1 \geq 0$ and $X_2 \geq 0$.

Depicting the problem graphically.

Type (go or skip).
PRIOR TO PROCEEDING, CHOOSE THE VALUES OF (X1, X2) YOU FEEL MAXIMIZE 7.

BY EXAMINATION, THE CORRECT ANSWER IS (2, 6) FOR A MAXIMUM VALUE OF Z BEING 36. NOTE THAT THE MAXIMUM OCCURRED AT A NODAL OR INTERSECTION POINT. UNFORTUNATELY, THE GRAPHICAL METHOD IS UNUSABLE WITH THREE VARIABLES AND UNUSABLE WITH FOUR OR MORE. THE SIMPLEX METHOD IS THE MULTIVARIABLE SOLUTION TECHNIQUE.

THIS SECTION IS DIVIDED INTO TWO PARTS—DEFINITIONS NECESSARY TO THE SIMPLEX METHOD AND REVIEW OF THE SIMPLEX METHOD ITSELF, TO SKIP THE DEFINITION PORTION AND GO DIRECTLY TO SIMPLEX REVIEW, TYPE REVIEW.

FEASIBLE SOLUTION: A FEASIBLE SOLUTION IS A VALUE OF (X1, X2, ..., XN) FOR WHICH ALL RESTRICTIONS OF THE PROBLEM ARE SATISFIED (THAT IS, ANY PART OF THE SHADED AREA IN THE TWO DIMENSIONAL GRAPHICAL PROGRAMMING EXAMPLE).

OPTIMAL SOLUTION: AN OPTIMAL SOLUTION IS A FEASIBLE SOLUTION THAT OPTIMIZES THE OBJECTIVE FUNCTION.

THERE ARE FOUR FUNDAMENTAL PROPERTIES UPON WHICH THE SIMPLEX METHOD IS BASED:

PROPERTY 1: THE COLLECTION OF FEASIBLE SOLUTIONS CONSTITUTE A CONVEX SET. (TO RECEIVE AN EXPLANATION OF CONVEX SETS, TYPE CONVEX.)

DEFINITION: A SET OF POINTS IS CALLED A CONVEX SET IF ALL POINTS ON THE STRAIGHT LINE SEGMENT JOINING ANY TWO POINTS IN THE SET BELONG TO THE SET.

A TRIANGULAR SET IS CONVEX.

* * * * * *

AN L SHAPED SET IS NOT CONVEX.
NOTE THAT A LINE CAN BE DRAWN FROM A TO B SUCH THAT POINTS ON THAT LINE ARE OUTSIDE THE SET.

TYPE (GO, SKIP, OR REVIEW)  

THOSE WERE TWO DIMENSIONAL EXAMPLES. THREE DIMENSIONAL EXAMPLES OF CONVEX SETS ARE A SPHERE AND A CUBE. NOTE THAT A SPHERE IS NON-LINEAR WHILE A CUBE IS LINEAR. TYPE READY IF YOU WOULD LIKE TO TEST YOURSELF ON CONVEX SETS. TYPE CHICKEN TO RETURN TO THE DISCUSSION OF PROPERTY TWO.

TYPE (READY, CHICKEN, OR REVIEW)  

AFTER EACH OF THE FOLLOWING, TYPE T FOR TRUE OR F FOR FALSE.

A CIRCLE IS A CONVEX SET. TYPE (T OR F)  

A KIDNEY SHAPED SET IS A CONVEX SET. TYPE (T OR F)  

A RECTANGLE IS A CONVEX SET. TYPE (T OR F)  

A WASHER (CIRCLE WITH A HOLE IN IT) IS A CONVEX SET. TYPE (T OR F)  

A LINEAR PROGRAMMING FEASIBLE SOLUTION DOES NOT HAVE TO BE PART OF A CONVEX SET. TYPE (T OR F)  

PROPERTY 2: IF A FEASIBLE SOLUTION EXISTS, THEN A BASIC FEASIBLE SOLUTION EXISTS WHERE THE SET OF BASIC FEASIBLE SOLUTIONS CORRESPONDS TO THE EXTREME POINT (NODAL POINTS IN THE GRAPHICAL EXAMPLE) OF THE SET OF FEASIBLE SOLUTIONS. TYPE (GO, SKIP, OR REVIEW)  

PROPERTY 3: THERE EXISTS A FINITE (COUNTABLE) NUMBER OF BASIC FEASIBLE SOLUTIONS.

PROPERTY 4: IF THE OBJECTIVE FUNCTION HAS A FINITE OPTIMUM (DOES NOT GO TO POSITIVE OR NEGATIVE INFINITY), THEN AT LEAST ONE OPTIMAL SOLUTION IS A BASIC FEASIBLE SOLUTION.

A KEY REASON FOR THE ABOVE PROPERTIES IS THE COMPLETE LINEARITY OF THE LINEAR PROGRAMMING MODEL. TYPE (GO OR SKIP)  

THE SIMPLEX METHOD STARTS AT A BASIC FEASIBLE SOLUTION AND MOVES TO AN ADJACENT EXTREME POINT THAT PRODUCES A VALUE OF THE OBJECTIVE FUNCTION NEAKER OPTIMUM. IT CONTINUES MOVING ALONG EDGES FROM EXTREME POINT TO EXTREME POINT IN AN ITERATIVE MANNER. WHEN NO ADJACENT EXTREME POINT PROVIDES A MORE OPTIMUM VALUE OF THE OBJECTIVE FUNCTION, THE PROCEDURE STOPS. TYPE (GO OR SKIP)  

TO PROVIDE A QUICK REVIEW OF THE SIMPLEX ALGORITHM, A SIMPLE PROBLEM WILL BE SOLVED USING THE ALGORITHM:  

PROBLEM: MAXIMIZE 5X1 + 6X2  
SUBJECT TO .2X1 + .3X2 <= 1.8  
.2X1 + .1X2 <= 1.2  
.3X1 + .3X2 <= 2.4

NOTE: EVEN IF IT IS DESIRED TO MINIMIZE THE OBJECTIVE FUNCTION, ONE CAN STILL USE THE SIMPLEX ALGORITHM FOR MAXIMIZING. THE CONVERSION FROM MINIMIZATION TO MAXIMIZATION IS MADE BY MULTIPLYING THE OBJECTIVE FUNCTION BY (-1). TYPE (GO OR SKIP)  

THE ALGORITHM REQUIRES ALL VALUES OF BI (RIGHT HAND SIDE OF THE CONSTRAINT EQUATION) TO BE NON-NEGATIVE AND ALL CONSTRAINT EQUATIONS TO BE EQUALITIES. THERE ARE TRICKS TO ACCOMPLISH THIS:

1. CONSTRAINT EQUATION INEQUALITY (=<)  
ADD A SLACK VARIABLE TO CHANGE TO AN EQUALITY.

EXAMPLE:

CONTRAINT IS X1 =< 2  
CAN BE WRITTEN X2 = 2 - X1 WHERE X2 >= 0

SUCH THAT X1 * X2 = 2  
X2 IS A SLACK VARIABLE. TYPE (GO OR SKIP)  

2. CONSTRAINT EQUATION INEQUALITY (=>)  
ADD A SURPLUS VARIABLE TO CHANGE TO AN EQUALITY.

EXAMPLE:

CONTRAINT IS X1 = 2  
CAN BE WRITTEN -X2 = 2 - X1 WHERE X2 >= 0
SUCH THAT X1 - X2 = 2 X2 IS A SURPLUS VARIABLE.

TYPE (GO OR SKIP) 19

4. NEGATIVE XI WITH AN INEQUALITY constrained.

INCREASE A SLACK OR SURPLUS VARIABLE TO CONVERT THE INEQUALITY TO AN EQUALITY, THEN MULTIPLY THE EQUATION BY (-1).

EXAMPLE:

CONSTRAINT IS X1 - X2 >= -2

ADD SURPLUS VARIABLE X1 - X2 - X3 = -2

MULTIPLY BY (-1)

-X1 + X2 + X3 = 2 ALL XI >= 0

TYPE (GO OR SKIP) 18

NOW BACK TO THE EXAMPLE!

INTRODUCE SLACK (OR SURPLUS) VARIABLES AND LET THE OBJECTIVE FUNCTION BE X0. THEN REWRITE AS

X0 - 5X1 - 6X2

.2X1 + .3X2 + 1X41 = 1.8

.2X1 + .1X2 + 1X42 = 1.2

.3X1 + .7X2 + 1X43 = 2.4

TYPE (GO OR SKIP) 19

NOTE THAT OBJECTIVE FUNCTION (X0) IS SET EQUAL TO 0 AND X1 AND X2 TRANPOSED TO LEFT OF > SIGN WITH THEIR SIGN REVERSED. INITIALLY LET X1 = X2 = 0.

TYPE (GO OR SKIP) 18

THE ABOVE DATA CAN BE CONDENSED INTO A TABLEAU:

IT CURRENT

NO BASE VALUES X1 X2 ROW

1 X3 0 -5 -6 1

X41 1.8 .2 + .3 2

X42 1.2 .2 + .1 3

X43 2.4 .3 + .3 4

TYPE (GO OR SKIP) 19

X3, X41, X42, X43 ARE IN SOLUTION (THEY ARE CALLED THE BASIC VARIABLES OR BASIS) WITH VALUES ShOWN.

X1 AND X2 ARE THE NONBASIC VARIABLES AND ARE = 0.

TO RECEIVE ADDITIONAL INFORMATION ABOUT BASIS, TYPE HELP.

TYPE (GO, HELP, OR SKIP) 25


X1 X2 X41 X42 X43 9 ROW

-5 -7 9 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

+2 +3 1 0 0 1.8 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2

+2 +1 0 1 0 1 0 1 0 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2

+3 +3 0 0 0 2.4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4

TYPE (GO OR SKIP) 19

NOTE: A BASIC VARIABLE HAS COEFFICIENT OF 1 AND ALL OTHER COEFFICIENTS IN THAT COLUMN ARE 0. BASIC VARIABLES HAVE THE VALUE OF THE ENTRY OF THE ROW IN WHICH THE COEFFICIENT IS 1.

Example: In this example the slack variables form the initial basis, or the basis for iteration 1, and the variables have the following values:

x1 = x2 = 0
x41 = 1.8
x42 = 1.2
x43 = 2.4

THE TABLEAU PRESENTED EARLIER IS A CONDENSATION OF THE FULL COEFFICIENT MATRIX.

TYPE (GO OR SKIP) 58

THE ALGORITHM SAYS TO CHECK ROW 1 VARIABLES FOR NEGATIVE COEFFICIENTS AND SELECT THE MOST NEGATIVE VARIABLE TO ENTER THE BASIS. IF ALL COEFFICIENTS ARE POSITIVE OR ZERO, THE optimum solution has been reached. TO DETERMINE THE VARIABLE TO EXIT,
CALCULATE FOR EACH ROW THE RATIO OF THE CURRENT VALUES TO THE COEFFICIENTS OF THE ENTERING VARIABLE AND SELECT THE MINIMUM RATIO (IGNORE NEGATIVE NUMBERS OR ZEROS IN THE DENOMINATOR). TYPE (GO OR SKIP) AS AN EXAMPLE, THE CALCULATIONS FOR THE INITIAL SELECTION ARE PRESENTED:

<table>
<thead>
<tr>
<th>Row</th>
<th>Ratio</th>
<th>Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.8</td>
<td>X_1</td>
</tr>
<tr>
<td>2</td>
<td>1.2</td>
<td>X_2</td>
</tr>
<tr>
<td>3</td>
<td>2.4</td>
<td>X_3</td>
</tr>
</tbody>
</table>

Therefore the X_3 element of row 2 will be designated the pivot element and X_2 will enter the basis replacing X_4. A tie for minimum value may occur. Regardless of the tie value chosen, the algorithm will proceed to the optimal solution. Type (GO OR SKIP) to explain a pivot, consider the full matrix of the sample problem:

<table>
<thead>
<tr>
<th>X_1</th>
<th>X_2</th>
<th>X_41</th>
<th>X_42</th>
<th>X_43</th>
<th>B_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>35.0</td>
<td>-1.0</td>
<td>20.0</td>
<td>1</td>
</tr>
<tr>
<td>X_2</td>
<td>6.0</td>
<td>67.3</td>
<td>3.3</td>
<td>1.7</td>
<td>2</td>
</tr>
<tr>
<td>X_42</td>
<td>0.6</td>
<td>13.3</td>
<td>-3.3</td>
<td>3.3</td>
<td>3</td>
</tr>
<tr>
<td>X_43</td>
<td>0.6</td>
<td>10.0</td>
<td>-1.0</td>
<td>0.0</td>
<td>4</td>
</tr>
</tbody>
</table>

To explain a pivot, consider the full matrix of the sample problem:

THE B COLUMN IN ROW 1 CONTAINS THE CURRENT VALUE OF THE OBJECTIVE FUNCTION. IN THE OTHER ROWS THE B COLUMN IS THE VALUE OF THE BASIC VARIABLE IN THAT ROW. X_2 WAS SELECTED TO ENTER THE BASIS IN ROW 2. THEREFORE ALL ELEMENTS IN ROW 6 ARE DIVIDED BY THE X_2 VALUE TO MAKE THE X_2 VALUE IN THAT ROW EQUAL 1.

EXAMPLE:

<table>
<thead>
<tr>
<th>X_1</th>
<th>X_2</th>
<th>X_41</th>
<th>X_42</th>
<th>X_43</th>
<th>B_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>35.0</td>
<td>-1.0</td>
<td>20.0</td>
<td>1</td>
</tr>
<tr>
<td>X_2</td>
<td>6.0</td>
<td>67.3</td>
<td>3.3</td>
<td>1.7</td>
<td>2</td>
</tr>
<tr>
<td>X_42</td>
<td>0.6</td>
<td>13.3</td>
<td>-3.3</td>
<td>3.3</td>
<td>3</td>
</tr>
<tr>
<td>X_43</td>
<td>0.6</td>
<td>10.0</td>
<td>-1.0</td>
<td>0.0</td>
<td>4</td>
</tr>
</tbody>
</table>

TYPE (GO OR SKIP) THE NEXT STEP IS TO REDUCE THE COEFFICIENTS IN COLUMN X_2 TO 0 IN ALL ROWS BUT ROW 2. FOR ROW 3 MULTIPLY ROW 2 (THE ROW WITH THE NEW BASIC VARIABLE) BY (-0.1), (THE NEGATIVE OF THE X_2 VALUE FOR X_2), ADD THAT TO ROW 3, PLACING THE FINAL RESULT IN ROW 3. TYPE (GO OR SKIP) THEN IN A SIMILAR MANNER, MULTIPLY ROW 2 BY (-3), ADD THAT TO ROW 4, AND PLACE THE FINAL RESULT IN ROW 4. PROCEED SIMILARLY FOR THE OBJECTIVE FUNCTION. THE RESULTANT COEFFICIENT MATRIX IS:

<table>
<thead>
<tr>
<th>X_1</th>
<th>X_2</th>
<th>X_41</th>
<th>X_42</th>
<th>X_43</th>
<th>B_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1.0</td>
<td>0</td>
<td>26.0</td>
<td>0</td>
<td>36.0</td>
<td>1</td>
</tr>
<tr>
<td>67</td>
<td>1</td>
<td>3.33</td>
<td>6</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>-3.3</td>
<td>1</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>-10</td>
<td>0</td>
<td>-1.0</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

TYPE (GO OR SKIP)
THE TERM FOR THE ABOVE OPERATION, IN ADDITION TO BEING CALLED A PIVOT OPERATION, IS \( x_2 \) HAS ENTERED THE BASIS. NOTE THE VALUE FOR \( x_1 \) IN ROW 1 IS THAT FOR THE OBJECTIVE FUNCTION WITH \( x_2=6 \) AND \( x_1=0 \). \( x_1 \) IS NOT YET IN THE BASIS.

TYPE (GO OR SKIP)

NOTE THAT FOR THIS ITERATION NUMBER (IT NO) \( x_0 \), THE VALUE OF THE OBJECTIVE FUNCTION, HAS INCREASED TO 36.0. CHECKING ROW 1 SHOWS THAT THE OPTIMAL SOLUTION HAS NOT BEEN REACHED (\( x_1 \) HAS A NEGATIVE COEFFICIENT). APPLYING THE ALGORITHM AGAIN WILL BRING \( x_1 \) INTO THE BASIS AND EXIT \( x_4 \). YOU SHOULD CONVINCE YOURSELF THAT THE MINIMUM RATIO IS \( 1.60 \) DIVIDED BY \( 0.13 \).

TYPE (GO OR SKIP)

IT CURRENT

NO BASE VALUES \( x_4 \) \( x_4 \) ROW

<table>
<thead>
<tr>
<th>( x_1 )</th>
<th>40.6</th>
<th>17.4</th>
<th>7.7</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_2 )</td>
<td>2.9</td>
<td>5.1</td>
<td>-5.1</td>
<td>2</td>
</tr>
<tr>
<td>( x_1 )</td>
<td>4.6</td>
<td>-2.6</td>
<td>7.7</td>
<td>3</td>
</tr>
<tr>
<td>( x_4 )</td>
<td>1.1</td>
<td>-1.3</td>
<td>4.8</td>
<td>4</td>
</tr>
</tbody>
</table>

THIS IS THE FINAL ITERATION. ALL ENTRIES IN ROW 1 ARE POSITIVE AND THE ALGORITHM TERMINATES. TYPE (GO OR SKIP)

SIMPLEX REQUIRES THAT THERE BE A BASIS IN THE INITIAL TABLEAU AND THAT THE OBJECTIVE FUNCTION HAVE ZERO COEFFICIENTS FOR ALL BASIC VARIABLES. IF THERE ARE CONSTRAINT EQUATIONS WITH EQUALITIES, SLACK (OR SURPLUS) VARIABLES MAY NOT BE ADDED. HOWEVER, ARTIFICIAL VARIABLES, VERY DIFFERENT FROM SLACK VARIABLES, MAY BE ADDED TO FORM THE INITIAL BASIS. TYPE (GO OR SKIP)

A PROCEDURE FOR OBTAINING AN INITIAL BASIS TO AN LP PROBLEM IS TO ADD BOTH SLACK AND ARTIFICIAL VARIABLES WHERE REQUIRED. HOWEVER, UNLIKE SLACK OR SURPLUS VARIABLES, ARTIFICIAL VARIABLES MUST *NOT* BE IN THE FINAL SOLUTION. THE BIG M METHOD INSURES THE LATTER CONDITION IF AN OPTIMAL BASIC FEASIBLE SOLUTION EXISTS. THE METHOD IS SIMPLE AND WILL BE OUTLINED RATHER THAN PRESENTING AN EXAMPLE.

TYPE (GO OR SKIP)

1. ASSUME A MAXIMIZATION PROBLEM OF THE FORM ILLUSTRATED PREVIOUSLY IN THIS SECTION.

2. FOR EACH ARTIFICIAL VARIABLE IN THE INITIAL BASIS, ASSIGN A LARGE POSITIVE COEFFICIENT (CALLED \( M \)) TO THAT VARIABLE IN THE OBJECTIVE FUNCTION. THE VALUE OF \( M \) MUST BE LARGE ENOUGH TO INSURE THAT THE ARTIFICIAL VARIABLE WILL NOT BE IN THE FINAL SOLUTION. TYPE (GO OR SKIP)

3. PERFORM A LIMITED PIVOT OPERATION TO REDUCE THE BASIC VARIABLE COEFFICIENTS IN THE OBJECTIVE FUNCTION TO ZERO.

4. PROCEED WITH THE REGULAR SIMPLEX PROCEDURE.

TYPE READY TO TRY A SHORT TEST TO SEE HOW WELL YOU UNDERSTAND THE SIMPLEX METHOD! TYPE (READY OR SKIP)

AFTER EACH STATEMENT, TYPE T FOR TRUE OR F FOR FALSE.

THE INITIAL OBJECTIVE FUNCTION (\( x_0 \)) ROW MUST INCLUDE ONLY NON-BASIC VARIABLES. TYPE (T OR F)

THE VALUE OF THE OBJECTIVE FUNCTION WILL FLUCTUATE TO GREATER AND LOWER VALUES AS THE ALGORITHM PROGRESSES TO AN OPTIMAL SOLUTION. TYPE (T OR F)

THE SIMPLEX ALGORITHM REQUIRES A CONVEX SET TO INSURE ACCURATE RESULTS. TYPE (T OR F)

ALL OPTIMAL SOLUTIONS ARE BASIC FEASIBLE SOLUTIONS. TYPE (T OR F)

INEQUALITIES ARE NOT PERMITTED IN THE SIMPLEX ITERATIONS. TYPE (T OR F)
PIVOT OPERATIONS INVOLVE A CHANGE OF BASIS. TYPE (T OR F) 44
A FEASIBLE SOLUTION SATISFIES ALL RESTRICTIONS OF THE PROBLEM. TYPE (T OR F) 54

**PRACTICAL LP PROBLEMS ARE NOT SOLVED AFTER THE SIMPLEX SOLUTION IS REACHED. PARAMETERS OF A PROBLEM ARE SOMETIMES KNOWN, BUT ARE ESTIMATED. SENSITIVITY ANALYSIS ENABLES THE PROBLEM FORMULATOR TO DETERMINE THE MOST CRITICAL PARAMETERS AND ESTIMATE THOSE MOST CAREFULLY. TYPE (GO OR SKIP)**

AS AN EXAMPLE, THE FOLLOWING INITIAL SIMPLEX TABLEAU IS GIVEN:

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-4</td>
<td>-5</td>
<td>-9</td>
<td>-11</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>120</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>5</td>
<td>10</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
</tbody>
</table>

WHERE X5, X6, AND X7 ARE SLACK VARIABLES ADDED TO ELIMINATE INEQUALITIES AND TO FORM AN INITIAL BASIS. ROW 1 IS THE OBJECTIVE FUNCTION AND ROWS 2-3 ARE THE CONSTRAINT EQUATIONS. THE FINAL SOLUTION TABLEAU IS:

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.43</td>
<td>.43</td>
<td>1.57</td>
<td>1.57</td>
<td>0</td>
<td>.71</td>
<td>99.29</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>.71</td>
<td>1.43</td>
<td>0</td>
<td>-.14</td>
<td>7.14</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>-.86</td>
<td>1.86</td>
<td>-.972</td>
<td>1</td>
<td>.57</td>
<td>64.3</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>.29</td>
<td>1.72</td>
<td>-.43</td>
<td>0</td>
<td>.14</td>
<td>7.86</td>
</tr>
</tbody>
</table>

IT IS NOT NORMALLY NECESSARY TO FULLY REWORK THE PROBLEM EVERY TIME A MINOR CHANGE IS MADE IN ONE OF THE INITIAL TABLEAU VALUES. CHOOSE ONE OF THE FOLLOWING CASES TO INVESTIGATE:

1. CHANGE IN OBJECTIVE FUNCTION COST COEFFICIENT (C1).
2. CHANGE IN RIGHT HAND VALUES (RH) IN THE CONSTRAINT EQUATIONS.
3. CHANGE IN A COEFFICIENT (AI) IN THE CONSTRAINT EQUATIONS.
4. SKIP BACK TO THE MAIN PROGRAM.

TYPE (1, 2, 3, OR 4) FOR WHAT RANGE OF VALUES OF A C1, WILL THE CURRENT VARIABLES REMAIN IN THE BASIS, AND WHAT WILL BE THE EFFECT ON THE VALUE OF THE OBJECTIVE FUNCTION?

THERE ARE TWO CASES:

1. COST COEFFICIENT OF A VARIABLE NOT IN THE BASIS.
2. COST COEFFICIENT OF A VARIABLE IN THE BASIS.

TYPE (GO OR RETURN) FOR CASE 1, USE X2 FROM THE EXAMPLE TABLEAUS FOR ILLUSTRATION. DEFINE D AS THE PERMISSIBLE CHANGE IN THE COEFFICIENT C2. THEREFORE, IN THE INITIAL ITERATION, THE COEFFICIENT FOR X2 IN ROW 1 BECOMES -(5+D), SINCE ROW 6 IS ALWAYS OPERATED ON AND NEVER AN OPERATOR, THE COEFFICIENT OF X2 IN THE FINAL ITERATION IS +(1.43-D).

SINCE THE SOLUTION IS OPTIMAL UNLESS A COST COEFFICIENT IS NEGATIVE, THE PERMISSIBLE RANGE OF D IS D<-.43, SINCE THE LIMITS ON C2 ARE C2<+.43.

FOR CASE 2, USE X3 IN THE EXAMPLE TABLEAUS AS AN ILLUSTRATION. DEFINE D AS THE PERMISSIBLE CHANGE IN THE INITIAL VALUE OF C3. THEREFORE, IN THE INITIAL TABLEAU, THE X3 COST COEFFICIENT IS MODIFIED TO -(9+D) AND IN THE FINAL TABLEAU C3 IS -. TYPE (GO OR RETURN) BY DEFINITION, A BASIC VARIABLE MUST HAVE A COEFFICIENT IN THE FIRST ROW. PERFORM A MINIMUM PIVOT OPERATION MULTI-
PLYING ROW 4 BY D, ADDING IT TO ROW 1, AND PLACING THE RESULT IN ROW 1. THE ROW 1 RESULT IS:

\(0.43 + 2.290)X^2 + (1.57 + 1.723)X^3 + (1.86 - .030)X^5 + (1.71 + 1.40)X^6\)

DETERMINE MAXIMUM LIMITS ON D SUCH THAT NO COST COEFFICIENT GOES NEGATIVE.

TYPE (GO OR RETURN)

THE RESULT IS \(-5.5 = D = 4.33\) GIVING A PERMISSIBLE RANGE OF VALUES FOR C3 OF \(9 - 5.5 = C3 = 9 + 4.33\).


THIS FINISHES CASE 1. CHOOSE ANOTHER CASE TO INVESTIGATE.

FOR WHAT RANGE OF VALUES OF A BI WILL THE CURRENT VARIABLES REMAIN IN THE BASIS, AND WHAT WILL BE THE EFFECT ON THE VALUE OF THE OBJECTIVE FUNCTION?

THERE ARE TWO CASES TO CONSIDER:

1. A SLACK VARIABLE IS IN THE BASIS OF ROW I.
2. A NON-SLACK VARIABLE IS IN THE BASIS OF ROW I.

CASE 1 IS TRIVIAL. ROW 7 OF THE EXAMPLE TABLEAUS WILL ILLUSTRATE. D IS THE PERMISSIBLE CHANGE IN THE INITIAL VALUE OF BI. THE KEY IS KEEPING THE BI POSITIVE (THAT IS, FEASIBLE). IF A BI GOES NEGATIVE, THE SOLUTION IS INFEASIBLE. SO FOR CASE 1, \(-46.43 = D = \infty\) GIVES A PERMISSIBLE RANGE OF D AS \(73.54 = D = \infty\).

CASE 2 IS LITTLE HARDER. IN THE INITIAL TABLEAU SET UP BI+D, OR USING ROW 2 OF THE EXAMPLE TABLEAU AS AN ILLUSTRATION, \((15+D)\). NOTE THAT D HAS A COEFFICIENT OF 1 AS DOES X5 (THE BASIC VARIABLE) IN THE INITIAL TABLEAU.

NOW GO TO THE FINAL TABLEAU. X5 HAS ITS FINAL COEFFICIENTS COMPUTED SO USE THOSE FOR D IN THE VARIOUS ROWS AS BELOW:

ROW 2: \(7.14 + 1.470\)
ROW 3: \(46.43 - 8.720\)
ROW 4: \(7.86 - .430\)

COMPUTE A RANGE OF VALUES FOR D THAT WILL DRIVE NONE OF THE BI NEGATIVE.

TYPE (GO OR RETURN)

PERFORM ALL ITERATIONS, THEN EXAMINE EVERY D TO INSURE NO BI OR CI IS DRIVEN NEGATIVE. THEN CONSTRUCT IN ALLOWABLE RANGE FOR D AND THE BI AS WAS DONE IN THE PREVIOUS CASES. THE PROPER USE OF MATRIX MULTIPLICATION MAKES THE ANALYSIS SOMEWHAT LESS PAINFUL, BUT A FULL TREATMENT OF THE SUBJECT IS NOT POSSIBLE IN THIS PROGRAM. YOU ARE REFERRED TO A TEXTBOOK.

THIS FINISHES CASE 2. CHOOSE ANOTHER CASE TO INVESTIGATE.

FOR WHAT RANGE OF VALUES OF AN AIJ WILL THE CURRENT VARIABLES REMAIN IN THE BASIS, AND WHAT WILL BE THE EFFECT ON THE OBJECTIVE FUNCTION?

COMPUTATION FOR THIS CASE DOES NOT READILY FALL OUT AS IT DID IN THE PREVIOUS TWO. DEFINE THE CHANGE IN AN AIJ AS D. IN OUR EXAMPLE USE A23 SUCH THAT IN ROW 3 THE COEFFICIENT OF X3 IS NOW \(3 + D\) IN THE INITIAL TABLEAU.

TYPE (GO OR RETURN)

PERFORM ALL ITERATIONS, THEN EXAMINE EVERY D TO INSURE NO AI OR CI IS DRIVEN NEGATIVE. THEN CONSTRUCT IN ALLOWABLE RANGE FOR D AND THE AIJ AS WAS DONE IN THE PREVIOUS CASES. THE PROPER USE OF MATRIX MULTIPLICATION MAKES THE ANALYSIS SOMEWHAT LESS PAINFUL, BUT A FULL TREATMENT OF THE SUBJECT IS NOT POSSIBLE IN THIS PROGRAM. YOU ARE REFERRED TO A TEXTBOOK.

THIS FINISHES CASE 3. CHOOSE ANOTHER CASE TO INVESTIGATE.

ASSIGNMENT PROBLEMS TYPICALLY ASSIGN N PERSONS TO N JOBS IN AN OPTIMAL MANNER. THE PROCEDURE ALSO CAN ASSIGN
MACHINES TO JOBS, HUSBANDS TO WIVES, ETC., AS LONG AS THERE IS A VALUE OF SOME SORT CONNECTED WITH EACH ASSIGNMENT.

TYPE (GO OR SKIP)

SINCE THE ASSIGNMENT PROBLEM IS A SPECIAL CASE OF THE TRANSPORTATION PROBLEM, THE NOTATION OF THE TRANSPORTATION PROBLEM DISCUSSED EARLIER WILL BE USED. HOWEVER, THE FOLLOWING RESTRICTIONS APPLY:

1. \( X_{ij} \) MAY EQUAL ONLY 1 OR 0.

2. THE ASSIGNMENT MATRIX IS SQUARE, THAT IS, \( M = N \).

3. EVERY PERSON IS ASSIGNED ONE AND ONLY ONE JOB.

TYPE (GO OR SKIP)

AS AN EXAMPLE, DETERMINE AN ASSIGNMENT THAT GIVES GREATEST MARITAL HAPPINESS, WHEREopIDE I HAS RATED BACHELOR J ON A SCALE FROM 1 TO 9. UNFORTUNATELY, THERE ARE 4 WOMEN AND 5 MEN. TO MEET THE \( M = N \) CRITERION WE PROVIDE A FICTITIOUS WOMAN WHO PROVIDES 0 HAPPINESS. THE RATINGS ARE ARRANGED IN A TABLE AS FOLLOWS:

<table>
<thead>
<tr>
<th>Egbert</th>
<th>Fred</th>
<th>Gaston</th>
<th>Horace</th>
<th>Igor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amanda</td>
<td>8</td>
<td>5</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>Reaula</td>
<td>1</td>
<td>7</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Chloe</td>
<td>3</td>
<td>1</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>Delia</td>
<td>7</td>
<td>6</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>Nobody</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
</tr>
</tbody>
</table>

NOTE THAT AMANDA WILL NOT MARRY FRED REGARDLESS.

TYPE (GO OR SKIP)

THE ALGORITHM FOR SOLVING THE OPTIMAL ASSIGNMENT PROBLEM MINIMIZES THE TOTAL ASSIGNMENT VALUE, SINCE THE USUAL SOLUTION REQUIRES THAT COST BE MINIMIZED. TO CONVERT THE PROBLEM TO ONE SOLVABLE BY MINIMIZATION, INVENT THE RATINGS: THAT IS, \( C_{ij}^* = 10 - C_{ij} \) AND REPLACE EACH \( C_{ij} \) WITH THE CORRESPONDING \( C_{ij}^* \).

TYPE (GO OR SKIP)

Egbert  Fred  Gaston  Horace  Igor

Amanda  2    M    5      4    1    54
Beaula  9    3      6      3    54
Chloe   7    9      3      5    4    54
Delia   3    4      1      4    5    54
Nobody  10   10     10     10   10   54

SINCE AMANDA AND FRED ARE INCOMPATIBLE, M IS A LARGE VALUE (20 FOR THIS PROBLEM) TO INSURE A MATCH DOES NOT OCCUR.

TYPE (GO OR SKIP)

STEP 1: SUBTRACT FROM EACH COLUMN, THE MINIMUM TERM SUCH THAT THERE IS A 0 ENTRY IN EACH COLUMN.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>17</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>9</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

TYPE (GO OR SKIP)

STEP 2: SUBTRACT FROM EACH ROW, THE MINIMUM TERM SUCH THAT THERE IS A 0 ENTRY IN EACH ROW.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>17</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

TYPE (GO OR SKIP)

STEP 3: EXAMINE EACH ROW AND COLUMN IN TURN. IF A
ROW/COLUMN HAS EXACTLY ONE 0 IN IT, RESERVE THAT ROW/COLUMN FOR ASSIGNMENT, SINCE THERE CAN BE ONLY ONE ASSIGNMENT IN EACH ROW AND COLUMN, DELETE THE DUPLICATE, UNNEEDED POSITIONS FROM CONSIDERATION. REPEAT THE ABOVE UNTIL ALL ZERO POSITIONS ARE RESERVED OR ELIMINATED. IN THE EXAMPLE, 6 (0) SIGNIFIES A RESERVED ZERO AND X SIGNIFIES AN ELIMINATED ZERO.

| FIRST RESERVE (A,E) ELIMINATE (A,H), (A,I) | 51 12 |
| SECOND RESERVE (9,F) | 23 |
| THIRD RESERVE (C,H) ELIMINATE (D,H), (N,H) | 52 |

RESULTING IN ASSIGNMENT AS FOLLOWS:

<table>
<thead>
<tr>
<th>A</th>
<th>(0)</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>(0)</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>(0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td>(0)</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ONLY 4 OF THE 5 ALLOCATIONS HAVE BEEN MADE.

TYPE (GO OR SKIP)

STEP 4: MARK ALL ROWS THAT DO NOT HAVE ASSIGNMENTS.

STEP 5: DRAW A LINE THROUGH EACH UNMARKED ROW AND EACH MARKED COLUMN.

FOR THE EXAMPLE PROBLEM:

<table>
<thead>
<tr>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>17</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>7</td>
<td>6</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>N</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

TYPE (GO OR SKIP)

STEP 6: SELECT THE SMALLEST ELEMENT NOT COVERED BY A LINE, THAT IS, NOT IN **. SUBTRACT IT FROM ALL UNCOVERED ELEMENTS, ADD IT TO ELEMENTS AT POINTS WHERE LINES INTERSECT. THEN GO TO STEP 3.

TYPE (GO OR SKIP): 1 IS THE SMALLEST ELEMENT UNCOVERED. THE NEW VALUE MATRIX IS:

<table>
<thead>
<tr>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>9</td>
<td>17</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>7</td>
<td>6</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>N</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

PERFORM STEP 3.

TYPE (GO OR SKIP)

RESERVE (A,E) ELIMINATE (A,I) 34 12

RESERVE (9,F) ELIMINATE (N,F) 34

RESERVE (C,G) ELIMINATE (C,H), (D,G) 42

RESERVE (N,H) 14

RESULTING IN ASSIGNMENT AS FOLLOWS:

<table>
<thead>
<tr>
<th>A</th>
<th>(0)</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>(0)</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>(0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td>(0)</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
PERFORM STEPS 4 AND 5.  TYPE (GO OR SKIP) 57

STEP 4A: MARK ROW D.
STEP 4B: MARK COL G.
STEP 4C: MARK ROW N.
STEP 4D: MARK ROW B.
STEP 4E: MARK ROW A.

THE RESULTANT ARRAY IS:

<table>
<thead>
<tr>
<th></th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>17</td>
<td>4</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>7</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

PERFORM STEP 6:  TYPE (GO OR SKIP) 57

SELECT 1 AS THE MINIMUM UNCOVERED VALUE. THE RESULTANT VALUE ARRAY IS:

<table>
<thead>
<tr>
<th></th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>18</td>
<td>5</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>6</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

PERFORM STEP 3:  TYPE (GO OR SKIP) 57

RESERVE (A,C) ELIMINATE (A,I), (A,N)
RESERVE (B,F) ELIMINATE (B,I), (N,F)
RESERVE (C,G) ELIMINATE (C,G)
RESERVE (D,N)
RESERVE (N,H)

SINCE ALL ARE ASSIGNED, THIS IS THE OPTIMAL SOLUTION WITH
THE FOLLOWING RECOMMENDED NUPIALS:

AHAMDA-EGERT 8
BLEAULA-FRED 7
CHLOE-IGOR 6
DELLA-GASTON 9
POOR HORACE 0
TOTAL BLISS 30

***END***

THE TRANSPORTATION PROBLEM DETERMINES THE OPTIMAL
SHIPMENT OF A COMMODITY FROM SOURCE TO DESTINATION (DEMAND)
POINT. TYPICALLY THERE ARE MULTIPLE SOURCES AND DESTI-
NATIONS AND A COST ASSOCIATED WITH GOING FROM SOURCE I TO
DESTINATION J. TOTAL DEMAND MUST EQUAL TOTAL RESOURCES
AVAILABLE. THE FOLLOWING SYMBOLOGY IS USED:

AI UNITS PRODUCED AT SOURCE I.
BJ UNITS DESIRED AT DESTINATION J.
XIJ AMOUNT SHIPPED FROM SOURCE I TO DESTINATION J.
Cij UNIT COST OF SHIPMENT FROM I TO J.
M NUMBER OF SOURCES (I=1,2,...,M).
H NUMBER OF DESTINATIONS (J=1,2,...,N).

TYPE (GO OR SKIP)

THE TRANSPORTATION PROBLEM IS WRITTEN IN CONCISE TABLE:

<table>
<thead>
<tr>
<th>SOURCES</th>
<th>DESTINATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>C11</td>
</tr>
<tr>
<td>3</td>
<td>C12</td>
</tr>
<tr>
<td>N</td>
<td>C1N</td>
</tr>
</tbody>
</table>
THE CIJ ARE INCLUDED IN THE ABOVE FOR REFERENCE, BUT DO NOT CHANGE DURING APPLICATION OF THE ALGORITHM.

TYPE (GO OR SKIP)

THE FOLLOWING RELATIONSHIPS MUST HOLD FOR THE TRANSPORTATION PROBLEM ALGORITHM TO WORK:

1. SUM OF THE XIJ ACROSS A ROW MUST EQUAL THE ROW AI.
2. SUM OF THE XIJ DOWN A COLUMN MUST EQUAL THE COLUMN BJ.
3. LASTLY, THESE THREE QUANTITIES MUST BE EQUAL
   A. SUM OF ALL THE AI.
   B. SUM OF ALL THE BJ.
   C. SUM OF ALL THE XIJ.

TYPE (GO OR SKIP)

CHOOSE ONE OF THE FOLLOWING AREAS TO INVESTIGATE. YOU MAY RETURN TO THIS POINT AT ANY TIME THE PROGRAM REQUESTS INPUT BY TYPING RETURN.

1. EXAMPLE OF A TRANSPORTATION PROBLEM (WILL BE USED IN PARTS 2, 3, AND 4).
2. FINDING AN INITIAL BASIC FEASIBLE SOLUTION (IBFS).
3. PERFORMING THE ALGORITHM.
4. DEGENERACY.
5. SELF TEST ON 1, 2, 3, AND 4 ABOVE.
6. SKIP TO THE MAIN PROGRAM.

CHOOSE ONE OF THE SIX TRANSPORTATION AREAS TO INVESTIGATE.

TYPE (1, 2, 3, 4, 5, OR 6).

THE FOLLOWING ARE TRUE/FALSE QUESTIONS. TYPE T FOR TRUE AND F FOR FALSE, AS APPROPRIATE:

TOTAL SUPPLY MUST EQUAL TOTAL DEMAND FOR THE TRANSPORTATION ALGORITHM TO PERFORM PROPERLY.

LINEARITY IS NOT REQUIRED FOR THE TRANSPORTATION PROBLEM.

THERE ARE (N+N) INDEPENDENT VARIABLES IN THE TRANSPORTATION PROBLEM WHERE N IS THE NUMBER OF COLUMNS AND M IS THE NUMBER OF ROWS.

THERE MUST ALWAYS BE EXACTLY (N+M-1) BASIC VARIABLES IN ANY TRANSPORTATION ARRAY.

A DEGENERATE SOLUTION IS ONE IN WHICH THERE ARE MORE THAN (N+M-1) BASIC VARIABLES.

A DEGENERATE SOLUTION PROHIBITS USE OF THE ALGORITHM.

SELECTION CRITERIA FOR SELECTING NEW BASIC VARIABLES Requires the largest CIJ*=(CIJ-(UI*VJ)) of the non-basic cells to be chosen to enter the basis. Type (T OR F) TRUE (T OR F) non-degenerate solutions were examined in section 8.

3. A DEGENERATE SOLUTION OCCURS WHEN THERE ARE LESS THAN (N+M-1) BASIC VARIABLES. DEGENERACY CAN OCCUR EITHER IN DETERMINING THE BASIC FEASIBLE SOLUTION OR DURING APPLICATION OF THE ALGORITHM. HOWEVER, THERE MUST BE EXACTLY (N+M-1) BASIC CELLS (SOME OF WHICH MAY BE ESSENTIALLY ZERO) FOR THE ALGORITHM TO PROCEED TO AN OPTIMAL SOLUTION.

TYPE (GO OR RETURN)

TO ILLUSTRATE, CONSIDER THE EXAMPLE MODIFIED BY CHANGING THE ROW 3 SUPPLY FROM 90 TO 70. THE INITIAL BASIC FEASIBLE SOLUTION WOULD THEN BE:

SOUPCE DESTINATION
1
2
3

SUPPLY
40
40
47
40
47
10
70
47
DEMAND 30 40 50 40 60 47

TYPE (GO OR RETURN) 20

Obviously all steps are not shown. Cell (3,1) is selected to enter the basis. The maximum value of 30 can drive both cell (2,1) and cell (2,3) to zero. However, only one may leave the basis since the algorithm requires exactly (m*n-1) basic variables. Type (GO OR RETURN) 58

Choose either cell to be replaced. In the example, cell (2,1) is arbitrarily chosen to leave the basis. Cell (3,3) is zero, but we assign it a value S (very small) such that it is negligible in performing computations, but identifies the basic variable. The allocation array is now as follows:

TYPE (GO OR RETURN) 20

DESTINATION 31 8

SOURCE 1 2 3 4 5 39
1 40 4
2 50 10 4
3 40 4
4 40 4

Type (GO OR RETURN) 20

Compute the u_i and v_j and select the new basic variables.

DESTINATION 31 8

SOURCE 1 2 3 4 5 39
1 40 4
2 50 10 4
3 40 4
4 40 4

Type (GO OR RETURN) 20

A degenerate initial basic feasible solution must also have S valued variables added until there are (n+m-1) basic variables. These additional variables may be added arbitrarily so long as the allocations are in independent positions. That is, the closed loop test of section 2 is met.

This completes this section. Choose another transportation area to investigate.

Type (GO OR RETURN) 20

Cost in dollars/unit of production at each plant is:

PLANT 1 2 3 4 5 34
1 20 19 14 21 16 52
2 15 20 13 19 16 52
3 18 15 18 20 16 52

Type (GO OR RETURN) 20

Since the total demand exceeds total production (sum of the a_i does not equal the sum of the b_j) the problem must be modified by adding a dummy source. The demands are then considered as upper bounds. The final cost and routing table is listed below (no x_i,j are shown).

Type (GO OR RETURN) 20

DESTINATION 31 8

SOURCE 1 2 3 4 5 SUPPLY 47
1 20 19 14 21 16 40 47
2 15 20 13 19 16 60 47

There are three plants that each produce 5 different products (except plant 3, which cannot produce product 5). The production capacity (supply of a_i) of each plant and the assumed sales (demand or b_j) are as follows:

PLANT 1 2 3 4 5 39
1 40 1 33 52
2 60 2 40 52
3 90 3 72 52
4 45 4 45 52
5 60 5 60 52

Type (GO OR RETURN) 20

Cost in dollars/unit of production at each plant is:

PLANT 1 2 3 4 5 34
1 20 19 14 21 16 52
2 15 20 13 19 16 52
3 18 15 18 20 16 52

Type (GO OR RETURN) 20

Source 1 2 3 4 5 Supply 5
1 20 19 14 21 16 40 47
2 15 20 13 19 16 60 47
THE M IN CELL (3,5) OPERATES AS DOES THE M IN THE SIMPLEX. IT IS A LARGE ENOUGH UNIT COST TO INSURE X35 IS 0 IN THE FINAL SOLUTION. SIMILARLY, THE ZEROS IN ROW 4 ARE ANALOGOUS TO SLACK VARIABLES IN THE SIMPLEX.

This example can also be set as a standard linear program. The formulation is as follows:

\[
\begin{align*}
\min & \quad 2x_{11} + 19x_{12} + 14x_{13} + 21x_{14} + 16x_{15} + 15x_{21} + 2x_{22} + 55x_{23} + 13x_{24} + 19x_{25} + 16x_{31} + 15x_{32} + 18x_{33} + 20x_{34} + 6x_{35} \\
\text{subject to:} & \quad x_{11} + x_{12} + x_{13} + x_{14} + x_{15} = 40 \\
& \quad x_{21} + x_{22} + x_{23} + x_{24} + x_{25} = 60 \\
& \quad x_{31} + x_{32} + x_{33} + x_{34} + x_{35} = 90 \\
& \quad x_{11} + x_{21} + x_{31} \leq 30 \\
& \quad x_{12} + x_{22} + x_{23} \leq 40 \\
& \quad x_{13} + x_{23} + x_{33} \leq 70 \\
& \quad x_{14} + x_{24} + x_{34} \leq 40 \\
& \quad x_{15} + x_{25} \leq 60 \\
& \quad \text{all } x_{ij} \geq 0
\end{align*}
\]

This completes this section. Choose another transportation area to investigate.

The example transportation problem has a non-degenerate initial basic feasible solution derived using the Vogel method. The occupied cells form an initial basis of \((m+n-1)\) variables. As in the simplex algorithm, variables must be chosen to enter and leave the basis until the optimal solution is reached.

For each basic variable (occupied cell) set up an equation:

\[c_{ij} = u_i + v_j\]

The result will be \((n+m-1)\) equations with \((n+m)\) unknowns. Therefore one of the \(u_i\) or \(v_j\) is assigned an arbitrary value at the start. The first steps of the transportation algorithm procedure are:

1. Set up an equation \(c_{ij}-(u_i+v_j)=0\) for each occupied cell (basic variable).
2. Assign an arbitrary value (normally 0) to one of the \(u_i\) or \(v_j\).
3. Compute the remaining \(u_i\) and \(v_j\) using the basic \(c_{ij}\).
4. Using the relation \(c_{ij}^* = c_{ij} - (u_i + v_j)\) compute \(c_{ij}^*\) for each non-basic cell using the \(u_i\) and \(v_j\) computed in 3.

The example problem will illustrate the first four steps.

First set up an array of \(c_{ij}\) marking the occupied (basic) cells by \((c_{ij})\).
TYPE (GO OR RETURN)

USING STEP 4, THE CALCULATION OF CIJ* FOR THE NON-BASIC POSITIONS USING CIJ* = CIJ - (U_I + V_J) IS:

SOURCE 1 2 3 4 5
1 5 9 1 5
2 10 3
3 -2 -1 M-21
4 1 6 3

TYPE (GO OR RETURN)

STEPS 1 THRU 4 ARE NOW COMPLETE.

5. EXAMINE THE CIJ*, IF ALL ARE POSITIVE, THE SOLUTION IS OPTIMAL. IF ONE OR MORE ARE NEGATIVE, CHOOSE THE MOST NEGATIVE TO ENTER THE BASIS. TYPE (GO OR RETURN)

CELL (3,1) IS CHOSEN TO ENTER THE BASIS.

6. THE CELL TO LEAVE THE BASIS IS THE ONE THAT REACHES ZERO FIRST AS THE NEW CELL IS ALLOCATED UNITS.

DESTINATION

SOURCE 1 2 3 4 5
1 40 4
2 30- 20+ 10
3 + 43 50- 10
4 40 10

TYPE (GO OR RETURN)

CELL (3,1) HAS A PLUS TO SHOW IT IS TO BE INCREASED. SINCE THE ARRAY MUST REMAIN BALANCED, AS CELL (3,1) IS INCREASED, CELLS (2,1), (2,3), AND (3,3) MUST BE MODIFIED AS SHOWN. CELL (3,1) IS INCREASED TO 30, DRIVING CELL (2,1) FROM THE BASIS.

DESTINATION

SOURCE 1 2 3 4 5
1 40 4
2 30- 20+ 10
3 + 43 50- 10
4 40 10

CONTINUE THE PROBLEM. WHEN YOU HAVE FINISHED STEPS 1 THROUGH 4, TYPE (GO OR RETURN).

BASIC VARIABLES ARE SHOWN (CIJ), NON-BASIC CELLS CONTAIN CIJ* = CIJ - (U_I + V_J).

DESTINATION

SOURCE 1 2 3 4 5
1 7 9 1 5 (16) -5
2 10 (13) 3 (16) -5
3 (18) (15) (19) -1 M-21 6
4 3 6 3 (0) (0) -21
VJ 18 15 18 21 21

NOW PERFORM STEPS 5 AND 6. SELECT CELL (3,4) TO ENTER THE BASIS, CELL (2,5) IS DRIVEN OUT. THE RESULTING BASIC ALLOCATION IS:

DESTINATION

SOURCE 1 2 3 4 5
1 40 4
2 60
3 40 10 10
4 30 20

NOW PERFORM STEPS 1 THROUGH 4 ON THE NEW SOLUTION.

TYPE (GO OR RETURN)

THE ARRAY OF NON-BASIC AND BASIC CELLS IS:

DESTINATION

SOURCE 1 2 3 4 5
1 6 8 0 5 (16) -4
2 2 10 (13) 4 1 -5
3 (18) (15) (18) (20) M-20 6

NOW PERFORM STEPS 5 AND 6. SELECT CELL (3,4) TO ENTER THE BASIS, CELL (2,5) IS DRIVEN OUT. THE RESULTING BASIC ALLOCATION IS:

DESTINATION

SOURCE 1 2 3 4 5
1 40 4
2 60
3 40 10 10
4 30 20

NOW PERFORM STEPS 1 THROUGH 4 ON THE NEW SOLUTION.

TYPE (GO OR RETURN)

THE ARRAY OF NON-BASIC AND BASIC CELLS IS:

DESTINATION

SOURCE 1 2 3 4 5
1 6 8 0 5 (16) -4
2 2 10 (13) 4 1 -5
3 (18) (15) (18) (20) M-20 6

NOW PERFORM STEPS 5 AND 6. SELECT CELL (3,4) TO ENTER THE BASIS, CELL (2,5) IS DRIVEN OUT. THE RESULTING BASIC ALLOCATION IS:

DESTINATION

SOURCE 1 2 3 4 5
1 40 4
2 60
3 40 10 10
4 30 20

NOW PERFORM STEPS 1 THROUGH 4 ON THE NEW SOLUTION.

TYPE (GO OR RETURN)

THE ARRAY OF NON-BASIC AND BASIC CELLS IS:

DESTINATION

SOURCE 1 2 3 4 5
1 6 8 0 5 (16) -4
2 2 10 (13) 4 1 -5
3 (18) (15) (18) (20) M-20 6
APPLYING STEP 5, THE SOLUTION IS OPTIMAL.

THIS COMPLETES THIS SECTION. CHOOSE ANOTHER TRANSPORTATION AREA TO INVESTIGATE.

THE TRANSPORTATION PROBLEM IS AN EXTENSION OF THE SIMPLEX METHOD, AND THE GENERAL REQUIREMENT OF BASIC FEASIBILITY EXISTS. FOR REFERENCE, THE EXAMPLE COST AND ROUTING TABLE IS LISTED.

**Transportation Problem**

<table>
<thead>
<tr>
<th>Source</th>
<th>Demand</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>40</td>
</tr>
<tr>
<td>3</td>
<td>70</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
</tr>
<tr>
<td>5</td>
<td>60</td>
</tr>
</tbody>
</table>

**Supply**

| 1      | 20     |
| 2      | 0      |
| 3      | 0      |
| 4      | 0      |
| 5      | 40     |

**Cost**

<table>
<thead>
<tr>
<th>Source</th>
<th>Destinations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20 19 14 21 16 40</td>
</tr>
<tr>
<td>2</td>
<td>26 13 19 16 60</td>
</tr>
<tr>
<td>3</td>
<td>18 15 18 20 60</td>
</tr>
<tr>
<td>4</td>
<td>0 0 0 0 50</td>
</tr>
</tbody>
</table>

**Demand**

<table>
<thead>
<tr>
<th>Destinations</th>
<th>1 2 3 4 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 0 0 20 20</td>
</tr>
<tr>
<td>2</td>
<td>0 0 0 20 40</td>
</tr>
<tr>
<td>3</td>
<td>0 20 36 0 90</td>
</tr>
<tr>
<td>4</td>
<td>30 20 0 0 50</td>
</tr>
</tbody>
</table>

**Starting at Cell (4,1) Trace a Pencil Line Through All Non-Zero Entries.**

**Northwest Corner Rule**

1. Allocate the maximum feasible amount to the cell in the upper left-hand corner.
2. Move one step right if there is any remaining supply. Otherwise, move one cell down.
3. Continue until all is allocated.

**Solver:**

The correct solution will be presented next.
DEMAND 30  40  70  40  60
        50  50
47
47

THERE IS NO REASON TO EXPECT THIS METHOD TO BE ANYWHERE NEAR THE OPTIMAL SOLUTION. TYPE (GO OR RETURN) 59

THE VOGEL APPROXIMATION METHOD OF FINDING AN INITIAL BASIC FEASIBLE SOLUTION IS BASED ON THE COST COEFFICIENTS (Cij) FOR EACH CELL. IT NORMALLY WILL PROVIDE A NEAR OPTIMAL SOLUTION. 58

DEFINITION: A ROW DIFFERENCE OR COLUMN DIFFERENCE IS THE ABSOLUTE ARITHMETIC DIFFERENCE BETWEEN THE SMALLEST CIJ AND NEXT-TO-SMALLEST CIJ IN THAT ROW OR COLUMN. 59

TYPE (GO OR RETURN) 20

THE PROCEDURE IS AS FOLLOWS:
1. SET UP A COST AND REQUIREMENTS TABLE.
2. CALCULATE ROW AND COLUMN DIFFERENCES FOR EACH ROW AND COLUMN.
4. ALLOCATE THE MAXIMUM POSSIBLE NUMBER OF UNITS TO THE CELL POSITION WITH THE SMALLEST COST (CIJ).

TYPE (GO OR RETURN) 20

5. ALLOCATE ZERO IN THE OTHER POSITIONS OF THE ROW OR COLUMN FROM WHICH YOU HAVE USED UP ALL PERMISSIBLE DEMAND OR SUPPLY.

TYPE (GO OR RETURN) 54

6. ELIMINATE FULLY ALLOCATED ROWS OR COLUMNS FROM FURTHER CONSIDERATION.

7. STOP IF ALL ROWS AND COLUMNS ARE FULLY ALLOCATED, OTHERWISE GO TO STEP 2 AND REPEAT.

NOW APPLY THE VOGEL METHOD TO THE EXAMPLE PROBLEM:

SOURCE   1    2    3    4    5    SUPPLY
1    20    19    14    21    16    10    40    2
2    15    20    13    19    16    60
3    18    15    18    20    M    90
4    0    0    0    0    0    50

DEMAND    30    40    70    40

TYPE (GO OR RETURN) 46

NOTE THAT THE AVAILABLE SUPPLY UNITS IN ROW 4 HAVE BEEN DECREASED FROM 53 TO 10. NOW YOU CALCULATE THE ROW AND COLUMN DIFFERENCES. TYPE (GO OR RETURN) 20

YOUR RESULTS SHOULD BE:

ROW DIFFERENCE COLUMN DIFFERENCE
1    2    1    15
2    2    2    15
3    3    3    13
4    6    5    16

SELECT THE PROPER ROW OR COLUMN. TYPE (GO OR RETURN) 58
COLUMN 5 SHOULD BE SELECTED. HOW MUCH SHOULD BE ALLOCATED?
TO WHAT ELEMENT?
TYPE (GO OR RETURN)
ALLOCATE 10 UNITS TO CELL (4,4) AND DELETE ROW 4.
PERFORM THE NEXT FULL ITERATION.
THE RESULTS ARE:
ROW DIFFERENCE COLUMN DIFFERENCE
1 2 1
2 2 2
3 3 3
SELECT COLUMN 2 AND ALLOCATE 40 UNITS TO CELL (3,2).
DROP COLUMN 2. PERFORM THE NEXT FULL ITERATION.
TYPE (GO OR RETURN)
THE RESULTS ARE:
ROW DIFFERENCE COLUMN DIFFERENCE
1 2 1
2 3 3
3 4 4
SELECT COLUMN 1 AND ALLOCATE 30 UNITS TO CELL (2,1).
DROP COLUMN 1. PERFORM THE NEXT FULL ITERATION.
TYPE (GO OR RETURN)
THE RESULTS ARE:
ROW DIFFERENCE COLUMN DIFFERENCE
1 2 1
2 3 3
3 4 4
SELECT ROW 3 (SINCE M IS VERY LARGE) AND ALLOCATE 50
UNITS TO CELL (3,3). DROP ROW 3. PERFORM THE NEXT FULL
ITERATION.
THE RESULTS ARE:
ROW DIFFERENCE COLUMN DIFFERENCE
1 2 1
2 3 3
3 4 4
SELECT ROW 2 (SELECTION WAS ARBITRARY, IT COULD HAVE
BEEN ROW 3) AND ALLOCATE 20 UNITS TO CELL (2,3). DROP
COLUMN 3 AND, BY INSPECTION, ALLOCATE 40 UNITS TO CELL
(1,5) AND 10 UNITS TO CELL (2,5). TYPE (GO OR RETURN)
THE RESULTANT FINAL TRANSPORTATION ARRAY INITIAL
BASIC FEASIBLE SOLUTION IS:

<table>
<thead>
<tr>
<th>SOURCE</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>SUPPLY</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>40</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>DEMAND</td>
<td>40</td>
<td>60</td>
<td>60</td>
<td>60</td>
<td>60</td>
<td></td>
</tr>
</tbody>
</table>

THIS COMPLETES THIS SECTION. CHOOSE ANOTHER TRANSPORTATION AREA TO INVESTIGATE.
TYPE (1, 2, 3, 4, 5, OR 6)
***EOP***
SELECTED BIBLIOGRAPHY


Iverson, K. E., The Role of Computers in Teaching, Queens University, Kingston, Ontario, 1968.


