PROJMG FORTRAN: AN INTERACTIVE COMPUTER PROGRAM FOR USE WITH THE DEFENSE MANAGEMENT SIMULATION EXERCISE

by

George W. Schultz

March, 1984

Thesis Advisor: M. B. Kline

Approved for public release, distribution unlimited
**Title**: PROJNNG FORTRAN: An Interactive Computer Program for Use with the Defense Management Simulation Exercise

**Author**: George W. Schultz

**Performing Organization Name and Address**: Naval Postgraduate School, Monterey, California 93943

**Controlling Office Name and Address**: Naval Postgraduate School, Monterey, California 93943

**Report Date**: March, 1984

**Number of Pages**: 323

**Security Classification**: Unclassified

**Approval**: Approved for public release, distribution unlimited

**Keywords**: Project Management Training, Contract Negotiation, Contract Negotiation Training

**Abstract**: Contract Negotiation Package (CNP), the supporting computer program for the Defense Management Simulation (DMS), is revised and embedded into a program which makes it user-friendly, and which provides sensitivity analysis capability to it. The program includes a plotting function for the sensitivity analysis. Exercise records are established for review of contracting team performance. Database files are generated which permit teams to submit reports, which provide a baseline for subsequent game sessions, and which...
permit monitor evaluation of team progress.

The text provides a description for the operation of both CNP and PROJNG. It documents the new program, PROJNG.

Appendices include the Fortran code, and Conversational Monitor System (CMS) executive machine language programs for the new programs operation. It contains instruction manuals which depict operation for both programs.
PROJEKING FORTRAN:
An Interactive Computer Program for Use with
the Defense Management Simulation Exercise

by

George W. Schultz
Lieutenant Commander, United States Navy
B.S.E.E., University of New Mexico, 1971

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS TECHNOLOGY
(COMMAND, CONTROL AND COMMUNICATIONS)

from the

NAVAL POSTGRADUATE SCHOOL
March 1984

Author:  

Approved by:  

Thesis Advisor

Second Reader

Chairman, Command, Control and Communications Academic Group

Academic Dean
Contract Negotiation Package (CNP), the supporting computer program for the Defense Management Simulation (DMS), is revised and embedded into a program which makes it user-friendly, and which provides sensitivity analysis capability to it. The program includes a plotting function for the sensitivity analysis. Exercise records are established for review of contracting team performance. Database files are generated which permit teams to submit reports, which provide a baseline for subsequent game sessions, and which permit monitor evaluation of team progress.

The text provides a description for the operation of both CNP and PROJHNG. It documents the new program, PROJHNG.

Appendices include the Fortran code, and Conversational Monitor System (CMS) executive machine language programs for the new programs operation. It contains instruction manuals which depict operation for both programs.
# TABLE OF CONTENTS

I. INTRODUCTION ................................................. 12
   A. HISTORICAL BACKGROUND ................................. 12
   B. SCENARIO .................................................. 13
   C. SIMULATION .............................................. 15
   D. SUMMARY ................................................ 16

II. CMF OPERATION .............................................. 18
   A. PROCEDURES .............................................. 18
   B. BENEFITS ............................................... 22
   C. EFFICIENCIES .......................................... 23
   D. MONITOR FUNCTIONS ..................................... 25
   E. SUMMARY ................................................ 26

III. PROJNNG OPERATION .................................... 27
   A. CMF MODERNIZATION .................................... 27
   B. PROJNNG PACKAGE ....................................... 28
      1. Programs ............................................ 28
      2. Data files ......................................... 29
      3. LINK ................................................. 30
      4. PROJNNG EXEC ....................................... 31
      5. PROFILE EXEC ....................................... 32
   C. DATABASE SECURITY .................................... 33
   D. EXIT .................................................. 34
      1. Fatal Errors ....................................... 35
      2. Security Violation ................................ 36
      3. Menu ................................................ 36
      4. YES/NO Query ...................................... 36
      5. Return to PROJNNG EXEC ........................... 36
   E. PROJNNG PROCEDURES .................................. 37
1. Instructions ........................................... 37
2. Link ................................................. 37
3. PROJMN EXEC Execution ................................ 38
4. Running PROJMN ...................................... 40

F. DATABASE CHANGE ....................................... 41

G. MENU .................................................. 42

H. REPORT SUBMISSION .................................... 44
   1. Progress Reports .................................... 44
   2. Decision Submission ................................ 46

I. CALCULATION CORRECTION ................................ 51

J. SUMMARY ................................................ 53

IV. PROJMN MONITOR OPERATION ................................ 54
   A. CMP MONITOR UNIQUE FUNCTIONS ....................... 54
   B. PROJMN MONITOR UNIQUE FUNCTIONS .................... 56
   C. MONITOR UNIQUE SECURITY SYSTEM ...................... 56
   D. QUERY SUPPRESSION FOR THE MONITOR .................. 57
   E. MONITOR'S MAIN MENU ................................ 59
      1. Change Team Number ................................ 59
      2. The DF Selection Query .............................. 63
      3. The Input Selection Menu ............................ 63
      4. Rerun the Data Calculation ......................... 64
      5. Do a Sensitivity Analysis ............................ 64
      6. Printout Cost * Factors .............................. 64
      7. EXIT ............................................. 64
      8. Flexible Menu ...................................... 65
   F. MONITOR SESSION PROCEDURES ......................... 65
   G. INITIALIZATION ....................................... 66
      1. PROJMN EXEC Operation ............................. 66
      2. Available Files .................................... 66
      3. Purging DATAFILES .................................. 67
   H. BASIC PROGRAM ROUTINES ................................ 69
      1. Start of PROJMN's Run .............................. 70
      2. Summary of Basic Capabilities ...................... 74
I. SENSITIVITY ANALYSIS ROUTINES ................. 74
   1. Sensitivity Analysis Basic Menu
      Options .................................. 74
   2. Sensitivity Analysis Table .................... 78
   3. Menu Expansion .............................. 78
   4. Sensitivity Analysis Plotting .................. 79
   5. Effectiveness Index ............................ 86
   6. Flexible Menu Reset ............................ 86
J. SUMMARY ...................................... 87

V. SUMMARY AND CONCLUSIONS ........................ 89
A. SUMMARY ...................................... 89
B. ENS ........................................... 89
   1. Benefit ..................................... 89
   2. Computer Support .............................. 90
   3. CMP .......................................... 90
C. PROJHNG ....................................... 90
   1. Benefits ..................................... 90
   2. Shortcomings .................................. 91
   3. First Usage ................................... 91
   4. Transportability ............................... 92
D. DOCUMENTATION .................................. 93
   1. CMP .......................................... 93
   2. PROJHNG ....................................... 93
   3. Glossary ....................................... 93
   4. Debugging ..................................... 94

VI. RECOMMENDATIONS FOR FUTURE CONSIDERATION .... 95
A. INCREASED EXERCISE SUPPORT ..................... 95
   1. Discussion ..................................... 95
   2. Recommendation ................................ 96
B. INCLUDE ALL DECISION POINTS .................... 96
   1. Discussion ..................................... 96
   2. Recommendation ................................ 97
C. DEVELOP A CBS TUTORIAL ............... 97
  1. Discussion .................. 97
  2. Recommendation .......... 97
D. VALIDATE PROJMG ................. 98
  1. Discussion .................. 98
  2. Recommendation .......... 98
E. REPLACE THE DATA TABLES WITH EQUATIONS ............. 98
  1. Discussion .................. 98
  2. Recommendation .......... 99
F. INCREASE SENSITIVITY ANALYSIS OPTIONS ............ 100
  1. Discussion .................. 100
  2. Recommendation .......... 100
G. IMPROVED GRAPHICS ................. 100
  1. Discussion .................. 100
  2. Recommendation .......... 102
H. PLATE DISPLAYS ................. 102
  1. Discussion .................. 102
  2. Recommendation .......... 103
I. A SCENARIO GENERATING PROGRAM ............. 104
  1. Discussion .................. 104
  2. Recommendation .......... 104
J. USE PROJMG TO ENCOMPASS OTHER CNP VARIATIONS ............. 105
  1. Discussion .................. 105
  2. Recommendation .......... 105
APPENDIX A: CONTRACT NEGOTIATION PACKAGE (CNP) ............ 106
APPENDIX B: DNS DATASHEETS ............. 123
APPENDIX C: LINKPROJ EXEC ............. 132
APPENDIX D: PROJMG EXEC ............. 134
APPENDIX E: PROFILE EXEC FCR PROJMG ............. 141
<table>
<thead>
<tr>
<th>LIST OF TABLES</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. Development Contract Values Summary .......... 20</td>
</tr>
<tr>
<td>II. Page Data Strings ............................. 21</td>
</tr>
<tr>
<td>III. CCST + FACTORS ............................... 25</td>
</tr>
<tr>
<td>IV. DATABASE PRINTCUT as SELECTION MENU .......... 43</td>
</tr>
<tr>
<td>V. Development Contract Summary .................. 45</td>
</tr>
<tr>
<td>VI. Proposed Values Table .......................... 47</td>
</tr>
<tr>
<td>VII. Final Contract Values Table ................... 48</td>
</tr>
<tr>
<td>VIII. Accepted Values Table ....................... 49</td>
</tr>
<tr>
<td>IX. DATAFILE TEAM03 ............................... 50</td>
</tr>
<tr>
<td>X. CCST + FACTORS ................................. 55</td>
</tr>
<tr>
<td>XI. SELECTION MENU ................................ 58</td>
</tr>
<tr>
<td>XII. CCST + FACTORS ............................... 65</td>
</tr>
<tr>
<td>XIII. Selection Menu ................................ 72</td>
</tr>
<tr>
<td>XIV. Development Contract Summary ................. 76</td>
</tr>
<tr>
<td>XV. Sensitivity Analysis Menu ...................... 77</td>
</tr>
<tr>
<td>XVI. Sensitivity Table ............................. 80</td>
</tr>
<tr>
<td>XVII. Development Contract Summary ................. 109</td>
</tr>
<tr>
<td>XVIII. Page Data Strings ........................... 111</td>
</tr>
<tr>
<td>XIX. Development Contract Summary ................ 112</td>
</tr>
<tr>
<td>XX. Page Data Strings .............................. 115</td>
</tr>
<tr>
<td>XXI. Page Data Strings ............................. 117</td>
</tr>
<tr>
<td>XXII. Development Contract Achieved Value Summary .. 118</td>
</tr>
<tr>
<td>XXIII. Development Contract Achieved Value Summary .. 121</td>
</tr>
<tr>
<td>XXIV. PRCJMNG FILEDEFS ............................ 144</td>
</tr>
<tr>
<td>XXV. PRCJMNG FILES ................................. 146</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Defense Management Simulation Decision Points</td>
<td>14</td>
</tr>
<tr>
<td>3.1</td>
<td>MAIN MENU</td>
<td>42</td>
</tr>
<tr>
<td>4.1</td>
<td>Monitor's Main Menu</td>
<td>59</td>
</tr>
<tr>
<td>4.2</td>
<td>Monitor Mode Menu</td>
<td>60</td>
</tr>
<tr>
<td>4.3</td>
<td>MAIN MENU in Mode 1</td>
<td>61</td>
</tr>
<tr>
<td>4.4</td>
<td>MAIN MENU in Mode 2</td>
<td>61</td>
</tr>
<tr>
<td>4.5</td>
<td>Initial Monitor Display</td>
<td>68</td>
</tr>
<tr>
<td>4.6</td>
<td>Monitor Session File Purge Option</td>
<td>69</td>
</tr>
<tr>
<td>4.7</td>
<td>PROJMMG Initial Displays</td>
<td>70</td>
</tr>
<tr>
<td>4.8</td>
<td>Team Number Responses</td>
<td>71</td>
</tr>
<tr>
<td>4.9</td>
<td>Monitor's Main Menu</td>
<td>74</td>
</tr>
<tr>
<td>4.10</td>
<td>Flight Tests Demonstration</td>
<td>75</td>
</tr>
<tr>
<td>4.11</td>
<td>Monitor's Main Menu in Mode 1</td>
<td>79</td>
</tr>
<tr>
<td>4.12</td>
<td>Plot Character Key Definitions</td>
<td>81</td>
</tr>
<tr>
<td>4.13</td>
<td>Sensitivity Table</td>
<td>82</td>
</tr>
<tr>
<td>4.14</td>
<td>Sensitivity Analysis Percentages Graph</td>
<td>83</td>
</tr>
<tr>
<td>4.15</td>
<td>Monitor's Main Menu in Mode 1</td>
<td>84</td>
</tr>
<tr>
<td>4.16</td>
<td>Selected Vertical Axis</td>
<td>85</td>
</tr>
<tr>
<td>4.17</td>
<td>Selection Menu with Optimized Flight Tests</td>
<td>88</td>
</tr>
</tbody>
</table>
I. INTRODUCTION

A. HISTORICAL BACKGROUND

During the Presidential Administration of John F. Kennedy, Secretary of Defense Robert S. McNamara instituted a major revision for management of defense resources. These procedures were instituted as a revolution to the military procurement system. Among the revisions he instituted were the Planning, Programming and Budgeting System and a management system for acquisition of new major systems.

The impact of these new procedures on the military was to create an immediate need for extensive education of the armed services' acquisition organizations. The personnel in charge of ongoing system procurements, the project managers, needed a means of being educated about the system. New personnel to the field of acquisition also had to have a means of learning about the system.

In response to the services' need, in 1963 the Defense Weapons Systems Management Center developed the Project Management Simulation Exercise (PMSE) to aid in DOD-wide training in project management. The exercise simulated procurement of a missile system to aid the DOD-wide training in PPS.

In the later 1960's, the Industrial College of the Armed Forces (ICAF) obtained the simulation and revised it into the Defense Management Simulation (DMS). DMS was installed at the Naval Postgraduate School in 1971.

The exercise currently has international application. It is used by Israel, Sweden and Singapore to train their defense personnel in the use of acquisition management systems.
Agencies within the U. S. government currently using DMS or related products for training exercises include:

- Industrial College of the Armed Forces
- Naval Postgraduate School
- U. S. Air Force Academy

It is this wide acceptance which has maintained DMS as an active management training exercise.

E. SCENARIC

The exercise developed by DWSMC in 1963 consisted of a time-sequenced set of decision points in the life cycle of a missile system acquisition (see Figure 1.1). The decision points (DPs) simulate the equivalent functions of these major milestones in the system life cycle. Within a prescribed time period each team of students participating in the exercise reviews the information materials for the specific decision point and submits its decision for recommended action at that point in the game, as would be done in actuality. The objective is to optimize the cost effectiveness for procurement, production and development of the proposed new missile system, the Zebra Missile System.
<table>
<thead>
<tr>
<th>ZEPHA PROGRAM SCHEDULE</th>
<th>zero</th>
<th>one</th>
<th>two</th>
<th>three</th>
<th>four</th>
<th>five</th>
<th>six</th>
<th>seven</th>
<th>eight</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DECISION POINTS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conceptual Phase</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Validation Phase</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contract Definition</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Development Phase</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sub-system Design &amp; Test</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proto-type Assembly &amp; Check</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fabricate for Qual Tests</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qualification Tests</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fabricate for Flight Tests</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flight Tests</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Production Phase</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Block I long-lead Procure &amp; Tool</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Block I Mfr. Eng. &amp; Procure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Block I Delivery by lots</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Block II Mfr. Eng. &amp; Procure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Block II Delivery by lots</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deployment Phase</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Deployment by group</td>
<td>1</td>
<td>23</td>
<td>45</td>
<td></td>
<td>678</td>
<td>910</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1.1 Defense Management Simulation Decision Points.
C. SIMULATION

The team recommendations require monitor evaluation. Monitors review the student recommended performance reliabilities, costs, incentives, contractor selection processes, milestone dates and fees. They prepare materials for teams to interact with as they would with the Secretary of Defense and with the contractor. An extensive study of the materials from each team is required by the monitor.

In order for the monitors to not spend as much time on the simulation as the students, the monitors need aid in processing the team data. ICAF developed an Fortran program called Contract Negotiation Package (CNP) to assist the monitor in the evaluation of student team contract negotiations.

The DNS program is more than twenty years old. The program has experienced several modifications and offshoots. Its 1960's programming state-of-the-art remains mostly intact. By 1984 standards of ease of operation and user-friendliness, it is difficult to utilize.

Efforts to modify the program occurred in 1972 and in 1977. The first attempt to upgrade the DNS program in 1972 modified several internal algorithms and produced an offshoot program which provided fewer but more rapid results. In 1977 this version of the program, the Contract Negotiation Package, was again revised. This later revision provided predictions of achieved values to the student teams.

Team access to the program's simulated results grew out of the modernization of management processes. Managers are using the benefits of modern computer systems to improve their own performance. Student team use of the CNP program for the DNS exercise was a logical evolution. CNP use by the teams enables the DNS exercise to reach the level of current
project management technological support in computer-aided acquisition management techniques. The program also remains abreast through student and monitor interest in maintaining the program at the computer support state-of-the-art.

The CNP program available on the IBM-3033 in the W. R. Church Computer Center at the Naval Postgraduate School campus is discussed further in Chapter 2. Its displays reveal CNP's batch program origins. The data input and output formats are batch processing type fields. The lack of interactive capability typifies a program which is batch processed. In its batch processing use, the computer did not have terminals or users with which to interact. Originally CNP had no hardware capable of directly interacting with the users and its design developed accordingly.

Modern menu-driven designs have made user-friendly programming standard practice. As a viable training exercise for acquisition management, the DNS game continues to be valuable. In order to maintain its wide acceptance and expand its efficient utilization, a major revision to the program is needed.

The following chapters discuss a package of programs designed to give CNP an improved interactive program capability.

D. SUMMARY

The DNS training program incorporates an exercise which simulates a project management scenario. DNS is a successful training device which has been in use for 20 years. In support of DNS, the Fortran computer program CNP was developed to provide computer-aided assistance for monitor evaluation of team performance. Later, CNP was released to the student teams in a format which provides them with simulated performance data. It enables team
interaction with the computer. In so interacting, the team gains experience using the computer as an analysis tool and to perform sensitivity analysis ("what if questions") on major project performance and incentive parameters. As a result of CNP's age, its design did not permit adequate interactive capability for easy operation. In order to resolve the need for greater user-friendliness in CNP, this thesis develops, studies and reports on a CNP program revision entitled PRCJMNG FORTRAN.
II. CNP OPERATION

A. PROCEDURES

Proficient use of the Contract Negotiation Program (CNP) depends upon the user's familiarity with both the computer facilities and with the instructions in Appendix A. The "Defense Management Simulation Instructions for Using CNP" presents computer terminal display lines as seen during a terminal session. In following sequentially through the instructions, difficulties are apparent with the program both in user-friendliness and in design. Appendix A demonstrates a session in CNP, and pertains to the following discussion of that program.

The procedures for accessing CNP require the operator to link to the monitor's class disk, then to access the disk, and finally to execute CNP. Accessing the program in this manner normally requires at least a conceptual knowledge of computer command and response terminology. In the event of an accessing malfunction, or if the monitor is using the CNP program's class disk and prevents read-only execution of the disk files, the user may be confronted with unfamiliar terminology. He would need more than a casual knowledge of computer operation to proceed into CNP.

The first function performed by CNP is to input administrative data. Included in the administrative data are the team number, the decision point (DP), and the DP pages for which data are to be entered. Complications with the program quickly arise with these procedures. The data must be input in the exact format prescribed on the instruction sheet. Too many digits, or too many spaces in the data result in erroneous output, and/or error statements. Each
page's data string requires confirmation. This becomes laborious in repeated runs of the program, or repeated executions of the calculations for a Summary Table, Table I.

Frequently explanation is required for the procedure of binary coded response to the query for pages of data to be entered. Coding the page selection for data input can be confusing. Each of the four pages requiring input data must be indicated by '1'(on) or '0'(off) in the four digit code string. Indicating the page number in Arabic characters would be a more commonly expected procedure. Again, an error in entry is irrevocable. The change pages cannot be rescinded once the binary code has been entered.

Having completed the administrative inputs, the team proceeds with providing its proposed DP data. Prior to running CUP, team participants are to have used the DNS Decision Point handouts (see Appendix B) to determine and analyze their tentative DP positions. These results are listed as data lines in the DNS Decision Point Sheets, (Appendix B). Each character in the page's data lines requires precise replication when placed in the page data string input to CUP. The program types the page's data back onto the screen and asks for confirmation. Through relooping to the query for pages of input, the program does allow revision of data. However, each data item to be changed requires an entire page change. The team is confronted with reentering the data string each time. This procedure has proven to be tedious. It frequently results in teams redoing data strings several times to get them correct before a calculation of achieved values can be made. Successive loops through the program have the added check feature of receiving a four line printout of the page data for confirmation. The redundant queries for confirmation of data values have benefit in reducing erroneous data calculations, (see Table II).
Based on the author's experience, about one out of four occasions through the four line recap of data pages, Table II, the team finds an error.

\[ \text{TABLE I} \]
\text{Development Contract Values Summary}

<table>
<thead>
<tr>
<th>INCENTIVE PROVISIONS</th>
<th>INCENTIVE ACHIEVEMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>INCENTIVE AREA</td>
<td>WORST VALUE</td>
</tr>
<tr>
<td>DEV. COST</td>
<td>$53.0M</td>
</tr>
<tr>
<td>FIT TST COMPL</td>
<td>236 WK</td>
</tr>
<tr>
<td>RELIABILITY</td>
<td>75.0%</td>
</tr>
<tr>
<td>ACCURACY</td>
<td>160 YDS</td>
</tr>
<tr>
<td>TOTALS</td>
<td></td>
</tr>
</tbody>
</table>

TOTAL CONTRACT PRICE = $54.9M

The Development Contract Summary Table, Table I, neatly displays the decision point incentive input data as the 'Incentive Provisions'. Only five calculated values are displayed to the team. They are the achieved values for development cost, flight test completion date, reliability, accuracy, and total contract price. Also, displayed on the Summary Table is the computed total of fee percentages obligated. The total fees allocated must be 15%. Allocation of fees to other than 15% results in a warning on the Summary Table. The significance of the Development Contract Summary table and CNP rests in these five achieved values.
Compared with the original CNP design, there are missing values from the Summary Table. It does not print out ten additional calculated values: the fee percentages achieved, their fee cost, and the totals of fees earned and of fees cost. These values were suppressed from the original program table due to their computation by incorrect algorithms. Their suppression is evidenced by the dollar signs at the right of the Summary Table.

To reloop through the input of data and receive another estimate of the achieved values, the team answers an unnecessary query. This question has no application to the teams. The "CSSTFACTORS( NO)" query provides monitor access to the tables of design performance factors achieved and test costs in the form of unit change cost factors. The team's nc("N") response to, or an incorrect response to, this query reloops the program to the top of the main routine. Again, each question: decision point, team, pages, data, yes or no, and confirm must be answered.

It is extremely difficult to exit from the program if the team attempts to stop at any point before the reloop query. At the query "DO YOU WISH TO CONTINUE, Y FOR YES, N
FOR NC", the program provides its only graceful exit. Any other attempts to exit the program must be accomplished by dumping (stopping) the program. Inputs that do not match the data types required for data read-statement formats result in error statements, and with sufficient error occurrence count the program will dump.

B. BENEFITS

The benefits for team utilization of CNP are in its ability to provide some semblance of the achieved values which the main DNS batch program, provides. In so doing, the team uses the computer as a tool with which to view the sensitivity of its incentivization parameters. It can evaluate effects of varying available DNS critical incentive parameters: development cost, flight test completion date, reliability, accuracy and development contract price. This aspect of analysis simulates the real world potential for computers to expedite the computations and as a tool in these assessments. It further provides an environment in which team participation and success depends on understanding the system acquisition concepts that DNS illustrates. In order to expeditiously vary the parameters within the computer the team must have competent knowledge of these DNS concepts.

CNP's best feature appears in its structure. It closely follows the procedures in the DNS Decision Point Handouts. From its inception as an offshoot of the main CMS program, it has inherently followed the decision point sequence and structure of CMS.

------------

"Industrial College of the Armed Forces, Defense Management Simulation Participant's Manual, " Industrial College of the Armed Forces
The wide acceptance of CNP at many institutions can be found in the attributes it brought to DMS. In meeting the objectives of providing a program with the ability to quickly reloop through computations, and the ability to be operated from a terminal, the program fulfills a major need in performing certain sensitivity analyses over the DMS program. Once accessed, the program can be rerun innumerable times without completely reinputting all the data. The data inputs can be accomplished from a terminal. Computer support for DMS on computer terminals has made DMS accessible to more users without adding expense. It has enabled many students to simultaneously utilize DMS as a tool in acquisition management training.

C. DEFICIENCIES

Areas which need improvement in CNP focus on user-friendliness. The first of these is the inability to gracefully exit from the program. Some means must be available to permit the team to stop the program. Currently the program can only be stopped by expending time to finish the session or by generating sufficient syntax errors to force the computer to dump the program. In some cases, sessions have been in sufficient difficulty that the program would not reach the query "DO YOU WISH TO CONTINUE...". For this situation the team is forced to generate error counts which will dump the program. These situations have been exceedingly frustrating for the program users.

A means to change data entries without redoing entire pages is needed. The team must completely redo a DP page in order to change one data item. Changing one item should only require reentering one value. Additionally, data strings from one decision point to another are not readily duplicated for input. Similar strings of data for pages on
DP-3 and DP-4 do not have the same field format. Matching field formats for identical pages in different DP's would enhance the ability of teams to input data strings.

CNP receives all of the input data but cannot retain it between sessions. The team starts from a baseline of no data each time CNP is called. Most significant of the changes needed to modernize CNP is the creation of a database file to retain team data from session to session. A foundation database of the initial paged input could provide a means of replicating session results, of documenting data inputs, and of proceeding to successive DP levels without laboriously reentering previously studied data.

Excessive recurrences of calculating the Summary Table could be eliminated to save team time in the program. Preventing runs which produce Summary Tables without correct incentive fee percentage totals would reduce wasted program run time. Utilizing data that does not meet the DMS parameters calculates erroneous values, and may provide Summary Tables which mislead the team. Repeatedly running calculations to evaluate wide ranges of parameter values without conducting proper preliminary analyses not only detracts from the timeliness of the DMS studies, it defeats the purpose of DMS training teams to understand the concepts of project management. Teams have occasionally found greater ease in working through the program and produce a Summary Table in order to reach the "DO YOU WISH TO CONTINUE...?" query as an expedient to reloop through or exit from the program. This process detracts from the educational benefit and timeliness of DMS.

As part of the training simulation, monitors spend many hours in analyzing the progress of the teams and the positions for contract negotiation. Analysis done by each team must also be accomplished by the monitor to assess student
D. MONITOR FUNCTIONS

CMNP provides one additional table for monitors. COST+FACTORS, Table III, provides the monitor with the cost for a unit increase in a system test category as well as with the design factors achieved.

| TABLE III |
| COST + FACTORS |

| COSTS OF TESTS PER UNIT (IN THOUSANDS OF DOLLARS) |

<table>
<thead>
<tr>
<th>QUALIFICATION TESTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOTOR: 32.034</td>
</tr>
<tr>
<td>AIR FRAME: 126.355</td>
</tr>
<tr>
<td>GUIDANCE: 309.906</td>
</tr>
<tr>
<td>FIRE CONTROL: 938.1E6</td>
</tr>
<tr>
<td>LAUNCHER: 424.555</td>
</tr>
<tr>
<td>FLIGHT TESTS: 316.855</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DESIGN FACTORS ACHIEVED</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPONENT</td>
</tr>
<tr>
<td>MOTOR:</td>
</tr>
<tr>
<td>AIR FRAME:</td>
</tr>
<tr>
<td>GUIDANCE A:</td>
</tr>
<tr>
<td>GUIDANCE B:</td>
</tr>
<tr>
<td>FIRE CONTROL:</td>
</tr>
<tr>
<td>LAUNCHER:</td>
</tr>
</tbody>
</table>

IH0001A PAUSE ; PRESS <<ENTER>> TO CONTINUE.

progress. The monitor must also create positions from which to interact with the teams as the contractor. Within the COST+FACTORS Table, some aid exists for the monitor. However, his understanding of the impact of unit-cost on the contract requires additional analysis time over what the teams spend on incentives analysis. In general, because of limited time the monitor has to spend analyzing each team's
proposals compared with the amount of time put in by the team, coupled with the inconvenience in making repeated CNP runs, most monitors produce an actual DNS table printout corresponding to each team's proposed decision in order to provide more detailed and accurate information for use during contract negotiations. It would be better for CNP to provide the information in a more usable format than to produce the lengthy and time-consuming main exercise reports of the DNS.

E. SUMMARY

The difficulties encountered in CNP detract from its usefulness. Time spent fighting the program problems are excessive compared to the hours available for productive DNS analysis. In an effort to correct these time-sinks in CNP, PROJANG FORTRAN and its associated EXEC routines have been created.
III. PROJHNG OPERATION

A. CNP MODERNIZATION

The task in creating PROJHNG FORTRAN as a revision to CNP was first to retain the same functional structure relationship with DMS. This goal has been accomplished by incorporating the CNP subroutines into a larger program. The expansion has occurred by the added Fortran code for the human-factors engineered subroutines, and new sensitivity analysis features. The human-factors designs provide additional user friendliness to PROJHNG. The program calculations continue to center around the original CNP program routines. Many of the achieved value parameters and performance calculations are accomplished using the same Fortran code as used by CNP. CNP is encompassed in PROJHNG.

Other tasks for PROJHNG FORTRAN were to correct as many of the discrepancies in CNP as possible, to provide an enhanced user-friendly training device and to improve the analysis capability available to the monitor.

In modernizing CNP, the criticisms specified in Chapter 2 were highlighted. The following areas were examined.

1. Link
2. Database
3. Database Security
4. Exit
5. Data Change
6. Menu
7. Report Submission
8. Monitor Access
9. Sensitivity Analysis
10. Calculation Correction
B. PROJMNG PACKAGE

This chapter discusses the features of PROJMNG used to support team analysis of the exercise. The monitor capabilities are discussed in Chapter 4.

1. Programs

PROJMNG consists of more files (executable programs) than does CNP. There are four routines utilized in executing PROJMNG FORTRAN. In addition to the PROJMNG FORTRAN file, the package includes LINKPROJ EXEC (Appendix C), PROJMNG EXEC (Appendix D), and the PROJMNG disk's PROFILE EXEC (Appendix E) which are discussed in detail throughout this thesis.

PROJMNG FORTRAN is designed to function alone provided team files exist and have been defined for PROJMNG FORTRAN as files 9 and 10: "FILEDEF 09 DISK DATAFILE TEAMxx" and "FILEDEF 10 DISK DATACODE TEAMxx". It can be used without the exec routines but at a lower level of performance. PROJMNG can be executed through the routine PROJMNG EXEC.

PROJMNG FORTRAN and PROJMNG EXEC are designed to reside on a publicly accessible disk. In this manner multiple teams can use the program simultaneously. Their database files will be on this one disk where they are conveniently accessible for the program and the monitors. PROJMNG is read from this disk by the team's user disk and executed from the team's disk.

PROJMNG both reads and writes onto its data files. The IBM-3033 public, "all", link access is read-only (R/O) and prevents writing onto the disk. The PROJMNG disk is only read capable when linked to for the exercise. However, the disk linking to PROJMNG can be written onto by the program. A team disk linking to PROJMNG acts as the write disk for the program.
The ability for teams to write directly onto the PROJMNG disk would be unsatisfactory. This access to the program would permit two detrimental effects. The program might be either altered or destroyed by overly curious students entering the files, and team files might also be altered or destroyed. Inadvertent alteration of the files is prevented by student access being limited to read only.

2. Data files

There are four files of data tables that were created for CNP. These tables contain the resource parameters to generate achieved values for the program calculations. These data tables remain in files FT17F001, FT18F001, FT19F001, and FT20F001 as in CNP. No contract simulation calculations can be accomplished without them. The program will dump when it requires a table be read and does not find it. PFCJMNNG will also stop if the tables are not in the required format.

Preformatted database files must be available when the STORE and read routines are called by PROJMNG. Whenever it attempts to write or read these data files and does not find them, the program will dump (stop). Two of the remaining files are these database files. One is the FILE DATAFILE which is preformatted for the DMS data. It serves as a dummy file which can be copied to initiate the team's database, DATAFILE TEAMxx. The other file, FILE DATACODE, serves as a similar dummy file in the creation of the team's security file, DATACCCE TEAMxx.

DATACODE stores the team security code, the security code count for attempts to breach the code, and the USERID of the last team which used the file. Its data enables the program to use routines to check the team and validate the team identity through the security code.
If formatted files for the team are not available on the PROJMNNG disk or the team disk, the dummy files are copied onto the team disk. PROJMNNG EXEC reproduces the two dummy files on the team disk in the names' DATAFILE TEAMxx and DATACODE TEAMxx. They are used interactively by PROJMNNG during the game sessions. The PROJMNNG Instructions (Appendix F) discuss the DATABASE and DATACODE file generation on the team's disk.

When the team finishes its session and exits from PROJMNNG FORTRAN, the last copy of the data which was stored by the program remains on the team's disk. As the computer execution finishes the routine PROJMNNG EXEC, these two files are transmitted to the PROJMNNG disk.

3. LINK

LINKPROJ establishes the link between the team's disk and the PROJMNNG disk. It requires PROJMNNG EXEC in order to run the program.

The routine LINKPROJ is a CMS executive level program which utilizes the link feature to establish the connection between the team's disk and the disk on which PROJMNNG resides. It begins the recording of the team's session, "RECORD ON". This first step in LINKPROJ provides a record of the session for review in the event PROJMNNG does not perform as desired. Once the link has been established, the routine sets the function of the programmable function key 9 (PF9) to enable release of the link. This feature enables the team to release the link after a session has been aborted and the program stopped without completing the LINKPROJ EXEC. -The team would have to abort a session should PROJMNNG be stuck in a continuous loop.- PF9 will not succeed in breaking the link when PROJMNNG is operating normally.
Team members should be aware of the IBM-3033 capability to be dumped (stopped) out of CMS by executing "<ALT>" and "<PA1>" simultaneously. Executing "<PA1>" aborts the program without completing LINKPROJ. The pressing of these two keys on the IBM 3278 terminals will place the virtual system into the CP level of operator interaction. The team can reaccess its "A" disk by executing the command "I CMS" and pressing "<ENTER>"; wait for the display to read "VM READ" and again press "<ENTER>". The disk will again be in CMS and existing links will remain intact. In order to release the link to PROJMNNG press the "<ALT>" and "<PF9>" simultaneously.

LINKPROJ breaks the link. It releases the PROJMNNG disk (0276P). Once the disk is released if no users are accessed to it, the PROJMNNG disk can be logged onto in the read/write mode to enable files to be written onto it.

When LINKPROJ sends the special message to PROJMNNG's disk, "SMSG QACHT LINE" it tells the computer to automatically log (autolog) onto the disk. The disk is activated in the read/write mode only if no other user is on the PROJMNNG disk. The files which the team sent are copied by the disk's PROFILE EXEC.

LINKPROJ also terminates the recording of the session, "RECORD OFF".

4. PROJMNNG EXEC

a. Initiate Files

PROJMNNG EXEC performs the checks of available files. It determines if a formatted data file and a security file are available for the team. If it detects the monitor unique file, DATAINST DISKNUM, it initiates the monitor flag to tell PROJMNNG FORTRAN the user is a monitor.
PROJMNG EXEC checks first for the monitor's file, determines if the user is a team or a monitor. If the user is a team, it checks for the team files on the PROJMG disk and uses them if available. If these files are not found, it will check the team disk for the files and use them. If the files are not found at all, the program copies the dummy files and generates the team label and files.

b. Terminating the Session

If the user is a monitor, PROJMNG EXEC queries him for whether he desires a concatenated listing of all of the team files. If he does, it spools a file with all team data files on it. It then completes its execution and returns operation to LINKPROJ.

For team operation, when the computer leaves PROJMNG FORTRAN control, execution returns to PROJMNG EXEC. In order to complete the session, PROJMNG EXEC has been designed to transmit the data files to the PROJMG disk. This feature is accomplished by spooling the disk, "SPCOL PUNCH CONT CL X 0276P". The punch file created is sent to userid 0276P as a class x file. PROJMNG EXEC punches the data files onto the spool, "PUNCH DATAPILE TEAMxx", and closes the spool, "SPOOL PUNCH CLOSE NOCONT". The file is queued in the reader file of the PROJMG disk (0276P).

5. PROFILE EXEC

When LINKPROJ sends the automatic log-on message to the PROJMG disk, if no other user is on the PROJMNG disk, the disk is activated in the read/write mode. The main computer executes the disk's profile routine. PROFILE EXEC is performed whenever the disk is accessed in the write capable mode. This exec has been uniquely tailored for this PROJMNG specific disk (see Appendix E). It reads up to six class x files queued to the disk. First the PROFILE EXEC
sets the disk reader file to look for class x, "SPOOL RDR CL X", and then reads the files, "READ *". The program has been designed to reset the reader spool to spool any class. Upon completion of writing files onto the disk, the command "SPOOL RDR CL *" returns normal spooling capability.

The PROFILE read function would nominally expect to find a set of files, DATAFILE TEAMxx and DATACODE TEAMxx. The read limit was expanded from two files to six, for three pairs, in expectation that there will be occasions when the disk will remain in use by other teams. When other teams are still using the disk the autolog will be unable to perform its one shot access to the disk. SMSG will not remain queued to access the disk. In order to read those class x files that may have been left waiting, subsequent running of the PROFILE EXEC will read in several more files than those transmitted in the current PROJMAN session. The exec will read up to three additional pairs.

The benefit in using the class x spooling capability is the cue which this class gives. It tells the disk that the file is a PROJMAN file. Normal spooling usage within the IBM 3033 does not use x. 'X' has a uniqueness within the IBM system and will reduce the possibility of PROJMAN EXEC reading non-PROJMAN files.

The files are not read onto the PROJMAN disk while other users remain on the disk. Attempts to write onto the disk in this situation result in garbling the data files.

C. DATABASE SECURITY

The first query the team answers in PROJMAN is to provide the team number.

The second question asks for the team to create a team security code. The team may enter any code of up to eight characters. If the eight characters are all zeros or
contain nothing (nulls) the program will again ask the team to change the code. PROJWNG begins succeeding sessions by asking the team to match the security code stored in database, the team's security code.

The security code system gives the team access to its database while assuring that data has not been altered by anyone except the team or the monitor. This has benefits to the team and to the simulation. The team doesn't have to worry about data being changed accidentally and the monitor has less concern that the team's work might be compromised. The security code is also used for proposal submission. Due to its length and improbability of being entered correctly by mistake, the security code performs the function of assuring the monitor of the team's intent to submit the proposal as indicated.

The monitor can change the database to correct errors in the team files. Corrections may become necessary if the team submits a report and then wants to change it. The proposed and final data reports for a DP are not accessible by the team. Lack of access to submitted reports increases team commitment to the report. It also, provides the monitor with a fixed team position reference point. This is critical if the monitor is to create a contractor position by which to interact with the team.

The submission of reports and their purpose are discussed in greater detail later in this chapter.

D. EXIT

In PROJWNG several methods of stopping the program have been provided. These are:

1. Fatal errors
2. Security violation
3. Menu option
4. Yes/No query

1. Fatal Errors

The computer capability to stop the program when a programmed statement cannot be performed (executed) is referred to as a fatal error. PROJMNNG has selectively modified the fatal error response.

The program termination in response to a null entry (pressing the "<ENTER>" key without providing a character including the character blank ' ') was unsatisfactory. Inadvertent null responses repeatedly dumped users out of the program.

To resolve the fatal error dump (stopping the program), a GLOBAL routine (program which can be called from the IBM 3033's resident routine files) is called. The 'CALL ERRSET' enables the program to ignore fatal errors and proceed with the program. When PROJMNNG finds an error during the process of reading information from the terminal, it performs a jump to the statement specified as the error default by the READ statement. Very serious errors in syntax will continue to be dumped by the computer.

Failure to answer PROJMNNG's questions satisfactorily will cause it to reloop to either the same question or to call out the EXITS routine. If an error is serious enough, the EXITS routine will be called. EXITS allows termination of the program, or if the team wants to continue, EXITS will return to the top of the last query routine and allow the program to reloop. Correct entry of the security code will give the team access to the main program and existing data files with that team's number.
2. **Security Violation**

During the start of a session at the terminal, the user will be queried for his previously entered security code. The program will permit the user to answer the security code query incorrectly only five times during the course of a DMS exercise. At occurrences of incorrect security code entries, a count is kept. On the fifth occasion when the code is entered incorrectly, the counter will tell the program to stop. Successive attempts to enter a team file whose security file counter has reached five will terminate at the security code query.

3. **Menu**

A standard option in the main menu of PRCJNG is EXIT. Selecting the EXIT option directs the program to stop. Again, this operation returns computer execution to the PRCJNG EXEC program. Completion of the PROJMNG EXEC run will return control to the LINKPROJ routine which presents the query for another run.

4. **YES/NO Query**

Each query in PROJMNG which can be answered with a Y for YES or N for NO, can also be answered with an E for EXIT. This E response either calls the program subroutine EXITS or stops the program. EXITS gracefully stops the program by permitting either return of the computer's operation to PROJMNG EXEC or by relooping into the last query routine.

5. **Return to PROJMNG EXEC**

The termination of PROJMNG FORTRAN returns the computer execution to PROJMNG EXEC. This routine executes a transmittal of all team files to the PROJMNG disk.
E. PROCJMNG PROCEDURES

1. Instructions

The procedures used with PROCJMNG FORTRAN are significantly different from CNP. Appendix F contains the PROCJMNG Student Instructions. These instructions replicate the program’s screen displays. Appendix F demonstrates a session using PROCJMNG. Checks and corrective procedures in PROCJMNG render the making of errors more difficult. However, some examples are demonstrated in the PROCJMNG Student Instruction’s tutorial.

2. Link

The instructions ask the team to access PROCJMNG solely by executing the command LINKPROJ. This command runs the program LINKPROJ EXEC. LINKPROJ EXEC is a routine which creates the disk link and accesses PROCJMNG as the user’s 'C' disk, as discussed above. This exec must be placed on the team disk. From there it is able to link and access PROCJMNG’s disk for the team.

Among the capabilities of LINKPROJ are the definition of programmable key PF9. This function can be executed by pressing <<ALT>> and <<PF9>> simultaneously.

LINKPROJ also executes a pair of calls to the macro computer program routine "RECCRD". The first of this pair enables the recording of all session terminal activity. The latter executes after the session has been completed; it terminates the record of the session. Teams should be aware that if the program dumps or they exit the program without performing the “RECCRD OFF”, the record will continue in effect. It can be closed by inputting the command "RECCRD OFF". Termination of the record permits the team to either name and save the session’s record on disk, to edit it, to print it, to quit it ("Q") which places it on the disk.
spool, or to purge the record. Addition of this capability permits review of team session. Undetected difficulties or errors that result in program malfunction can be reviewed.

In order to facilitate rerunning of LINKPRCJ, the routine does query whether the team wishes to run "ANOTHER RUN, Y/N". This feature is one which LINKPROJ holds in common with the other programs in the PROJMNG package of routines. PROJMNG is designed to allow the team to terminate or reloop through program execution.

The running of PROJMNG FORTRAN does not occur from LINKPRCJ. LINKPROJ runs limited functions already specified, and calls the PROJMNG disk's program PROJMNG EXEC.

3. PROJMNG EXEC Execution

PROJMNG EXEC serves several functions in preparing the team disk for the session.

a. Team Disk Status

The PROJMNG EXEC routine provides several queries to the team disk. Through the response error return codes it receives indication of the availability of team data base files and security code files. The purposes accomplished by this procedure are:

(1). File Availability. The error return codes tell PROJMNG EXEC whether there are files available for use. Neither PROJMNG EXEC nor PROJMN FORTRAN can proceed without files from which data can be read or to which data can be written.

(2). File Integrity. Having found a usable DATAFILE on the team disk, PROJMNG EXEC attempts to read the contents. It looks specifically for the team number. The team number cues PROJMNG EXEC not only as to what the team should be called, it also reveals the prior use of the DATAFILE to store data.
k. Monitor File

In a similar manner, PROJMG EXEC determines the availability and validity of the monitor file "DATAINST DISKNUM". The presence of data in the file and the security field format within DATAINST DISKNUM tells PROJMG if the disk gaining access to it belongs to a monitor. Disks belonging to monitors are not intended to be the disk which students access to run PROJMG.

c. DATAFILE Creation

PROJMG EXEC determines the team disk status. If DATAFILE and DATACCDE files are not found, PROJMG EXEC proceeds to determine the team number. It will ask "WHAT IS YOUR TEAM NUMBER?" before it copies the files. After the team number is evaluated for being in the range of 1 to 20, the program generates the "TEAMxx" label. It will accept 1 to 20 with any number of preceding 0's. The TEAMxx label will include up to four numbers in the xx field. This label is used in the process of copying the dummy files. PROJMG copies the dummy files using the labels DATAFILE TEAMxx and DATACCDE TEAMxx as the new file name and file type.

In the event that more than 20 team files are required, PROJMG will accept team file labels with additional leading zeroes. Teams with files labeled TEAM01 and TEAM0001 can both be created.

The availability of the DATAFILES also gives PROJMG EXEC the team number. Lack of DATAFILES is synonymous with no team number. The team number is determined only once during the course of the gameplay. It is PROJMG EXEC that makes that query. It forms the team number into the "xx" portion of the files' type labels. PROJMG stacks the team number into its virtual reader to enable PROJMG FORTRAN to copy the team number. PROJMG FORTRAN uses the
team number to label displays and, when storing the database, places the number at the head of the record. It is this storing process and filling in of the first DATAFILE field which later enables PROJMN EXEC to find the team number without additional queries.

4. Running PROJMN

PROJMN EXEC completes determining if the disk belongs to a team or monitor, gets the team number or sets the monitor's flag of team number 21 (FL21), insures the availability of write files or creates them, and then loads and starts PROJMN FORTRAN's text file. The team database capability has been set for 20 teams. The data file for monitors was chosen in excess of the team database design size of 20. Having chosen TEAM21 to be the monitor's file label, it was convenient to utilize the label TEAM21 as a flag between PROJMN EXEC and PROJMN FORTRAN. It indicates to PROJMN FORTRAN that the computer operator for the session has been determined by PROJMN EXEC to be a monitor. Upon termination of PROJMN FORTRAN, PROJMN EXEC again takes control. If the operator is a monitor, the EXEC allows the monitor to request the creation of a concatenated file of all current team files on the PROJMN disk. For team disks, the exec sends a copy of the files DATAFILE and DATAACCDE to the master disk. This file transmission enables the monitor to determine team progress and to accept the DP decision reports. PROJMN erases unnecessary residual files such as the LOAD MAF and then returns execution control to LINKFORJ to perform the "ANOTHER RUN, Y/N?" query. The team's response to this question will either terminate the session or reloop the program to the beginning of LINKFORJ.
After data files have been created, PROJMNG uses the DATAFILES as a baseline of team information. Once the file is created the program assumes the data is accurate and satisfactory for use in all subsequent sessions. All further Selection Menu displays will display the EMS data items as they are currently found in the team's file DATAFILE TEAMxx.

At the beginning of the team's first session all of their data items must be submitted. A database is generated from which successive sessions will obtain their baseline of information. The process of asking for all data items occurs through the program's subroutine PAGEIN.

The pages of the EMS decision sheets, Appendix F, list the data entries necessary for CNP. These same entries are queried once in PROJMNG by PAGEIN as demonstrated in Appendix F. Each page of data queried coincides with the page and the data item of the decision sheets. Available parameter ranges are listed with the item name in PAGEIN. Correct completion of a page of data permits the program to check the data input for errors. If the program does not understand the input, or requires revision, a *****WARNING***** is displayed and questions designed to correct the discrepancy are asked. If the data entered cannot be read by PROJMNG the EXIT query "DO YOU WISH TO CONTINUE: Y/N?" will occur. Teams experiencing confusion, or wishing to exit the program during PAGEIN, can do so by pressing <<ENTER>> without any input. During PAGEIN this action will enable the team to use the exit query to stop the program.

The automatic call to the PAGEIN routine occurs only when the DATAFILE contains 0's in all of the data base data fields. Successive sessions in PROJMNG by a team utilize
the database DATAFILE to fulfil the data field input function. PAGEIN can be called at any time from the Selection Menu. This option permits the team to make mass changes to data items.

Once the program has a set of data it will produce a table containing those values, Table IV. The SELECTION MENU gives a review of the current data values and acts as a selection menu for items the team wishes to change. SELECTION MENU presentation is achieved by either completion of PAGEIN, by beginning a subsequent session, by completion of a DP proposal submission, or by team choice from the MAIN MENU.

DC YOU WISH TO:

1. RECEIVE THE TABLE OF ACHIEVED VALUES.
   *****WARNING***** AFTER THE TENTH DP-3 RUN
   AN ADDITIONAL DEVELOPMENT COST OF $100,000
   WILL BE INCURRED FOR EACH ADDITIONAL RUN.
   THIS IS RUN 1 OF YOUR DP-3.

2. INPUT SELECTION MENU.
3. SUBMIT A CONTRACT PROPOSAL
4. EXIT

Figure 3.1 MAIN MENU.

G. MENU

Continuing sequentially through an initial terminal session with PROJMNGL, the next feature which differentiates this program from CNF is the MAIN MENU, Figure 3.1. From this routine in the main program, any of the options performed by PROJMNGL for the team can be executed. This menu places the PROJMNGL FORTRAN program in the category of being a menu-driven program.
The options available in the **MAIN MENU** are to print the table of achieved values, to return to the database table, to submit a proposal, or to exit.

The achieved values option directs the program to execute the routines associated with creating the Development Contract Summary table, Table V. First the data entries for fee percentages are checked. The fee percentages are added together and evaluated to determine if they total 15%. If they do not equal 15%, **PROJMG FORTRAN** requests corrections through a sequence which displays each current fee percentage value and asks for a value to be submitted. The program performs the computations for the table created in subroutine **PROSUM**. Then the table is displayed. Each time the computations are made the run is counted. After ten runs, the program adds a fee of $100,000 onto the contract price for each subsequent run at that DP level.

The second option returns the display of the data table. It provides the team with the table, Selection Menu, of data items which are used for reference in changing data values.

The third capability from the **MAIN MENU** is proposal submission. Submission of proposals is discussed in the next section.

To enable the team to stop the program, the fourth option in **MAIN MENU** allows them to exit.

**B. REPORT SUBMISSION**

1. **Progress Reports**

Submission of progress reports has been readily accomplished in **PROJMG EXEC**. The process of completing **PROJMG EXEC** when leaving **PROJMG FORTRAN** serves to execute the commands for file transmission as discussed.
### TABLE V
Development Contract Summary

<table>
<thead>
<tr>
<th>INCENTIVE AREA</th>
<th>INCENTIVE PROVISIONS</th>
<th>INCENTIVE ACHIEVEMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WORST VALUE</td>
<td>BEST VALUE</td>
</tr>
<tr>
<td>DEV. COST</td>
<td>$ 53.0M</td>
<td>$ 47.0M</td>
</tr>
<tr>
<td>FLT TST COMPLI</td>
<td>238 WK</td>
<td>202 WK</td>
</tr>
<tr>
<td>RELIABILITY</td>
<td>75.0%</td>
<td>81.0%</td>
</tr>
<tr>
<td>ACCURACY</td>
<td>160 YDS</td>
<td>140 YDS</td>
</tr>
<tr>
<td>TOTALS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TOTAL CONTRACT PRICE = $ 54.9M

180001A PAUSE; PRESS <ENTER> TO CONTINUE.
in subsection 3.B.4.2. The process of data file transmission to the PROJMGNG disk provides those files for access by the monitor. In the monitor's session, all team data files which have been sent to the PROJMGNG disk are copied onto the monitor's personal disk for read/write use. This feature was also discussed in 3.B.5. and is covered in greater detail in Chapter 4 with the discussion of monitor functions.

2. **Decision Submission**

Submission of the team's decision occurs with the transmittal of the team files to the PROJMGNG disk. Within the data fields transmitted, there is a label field which contains the status of the team's work on a given decision point. This field is either blank, "00000000", "PROPCSED" or "FINAL". These labels are sequentially exclusive. They specify the exact status for the DP level of the set of database fields in which they are contained. Each DP level has four lines of data in the DATAPILE TEAMxx. The DP status label is the last field of data in a set of lines.

The DP field labels are assigned in the FINISH subroutine. By accepting the values in the table of proposed values for the contract proposal, Proposed Values Table (Table VI), or of proposed final values as the final approved contract, Final Contract Values Table (Table VII), the team causes the label to be written. These labels occur only when the team enters its security code to validate the proposed values. Completion of the value assignments to the database for transmittal appears with the table of accepted values, Table VIII.

The label is written as the last field in each of DP level database records. These labels can be seen in the 33rd field of each set of four lines of a DP data, Table IX.
<table>
<thead>
<tr>
<th>NO.</th>
<th>ITEM</th>
<th>CURRENT VALUE</th>
<th>DP-3</th>
<th>NO.</th>
<th>ITEM</th>
<th>CURRENT VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>CONTRACTOR</td>
<td>1</td>
<td></td>
<td>16</td>
<td>GUIDANCE QUAL TESTS</td>
<td>6.00</td>
</tr>
<tr>
<td>2.</td>
<td>GUIDANCE APPROACH</td>
<td>1</td>
<td></td>
<td>17</td>
<td>FLIGHT TESTS</td>
<td>17.00</td>
</tr>
<tr>
<td>3.</td>
<td>GUIDANCE CONFIGURATION</td>
<td>2</td>
<td></td>
<td>18</td>
<td>MAXIMUM COST</td>
<td>53.00</td>
</tr>
<tr>
<td>4.</td>
<td>MAINTAINABILITY ENG $</td>
<td>1.75</td>
<td></td>
<td>19</td>
<td>MINIMUM COST</td>
<td>47.00</td>
</tr>
<tr>
<td>5.</td>
<td>VALUE ENG M$</td>
<td>0.75</td>
<td></td>
<td>20</td>
<td>MAX COST INCENTIVE %</td>
<td>4.00</td>
</tr>
<tr>
<td>6.</td>
<td>PARALLEL DEVELOP M$</td>
<td>0.0</td>
<td></td>
<td>21</td>
<td>LATEST WEEK</td>
<td>238.00</td>
</tr>
<tr>
<td>7.</td>
<td>MOTOR RELIABILITY %</td>
<td>92.00</td>
<td></td>
<td>22</td>
<td>EARLIEST WEEK</td>
<td>202.00</td>
</tr>
<tr>
<td>8.</td>
<td>AIRFRAME RELIABILITY</td>
<td>97.00</td>
<td></td>
<td>23</td>
<td>MAX DELIVER INCENTIVE%</td>
<td>3.50</td>
</tr>
<tr>
<td>9.</td>
<td>LAUNCHER RELIABILITY</td>
<td>98.50</td>
<td></td>
<td>24</td>
<td>MAXIMUM RELIABILITY %</td>
<td>81.00</td>
</tr>
<tr>
<td>10.</td>
<td>FIRE CONTROL ERROR YDS</td>
<td>80.00</td>
<td></td>
<td>25</td>
<td>MINIMUM RELIABILITY %</td>
<td>75.00</td>
</tr>
<tr>
<td>11.</td>
<td>GUIDANCE ERROR YDS</td>
<td>70.00</td>
<td></td>
<td>26</td>
<td>MAX RELIABILITY %</td>
<td>75.00</td>
</tr>
<tr>
<td>12.</td>
<td>MOTOR QUAL TESTS</td>
<td>25.00</td>
<td></td>
<td>27</td>
<td>MAXIMUM ERROR YDS</td>
<td>160.00</td>
</tr>
<tr>
<td>13.</td>
<td>AIRFRAME QUAL TESTS</td>
<td>5.00</td>
<td></td>
<td>28</td>
<td>MINIMUM ERROR YDS</td>
<td>140.00</td>
</tr>
<tr>
<td>14.</td>
<td>LAUNCHER QUAL TESTS</td>
<td>5.00</td>
<td></td>
<td>29</td>
<td>MAX ERROR INCENTIVE %</td>
<td>4.00</td>
</tr>
<tr>
<td>15.</td>
<td>FIRE CONTROL QUAL TESTS</td>
<td>3.00</td>
<td></td>
<td>30</td>
<td>WEEK FOR IOT 10</td>
<td>1337.00</td>
</tr>
</tbody>
</table>

**DO YOU WISH TO ENTER THIS DATA AS YOUR PROPOSED DP-3 DECISION?**

****CAUTION****CAUTION****CAUTION****CAUTION****CAUTION****

YOU MAY ONLY INPUT A DECISION ONCE.

ENTER YOUR TEAM SECURITY CODE IF YOU WISH TO COMMIT TO A DECISION; OR "CONT".
### TABLE VII
Final Contract Values Table

<table>
<thead>
<tr>
<th>No.</th>
<th>Item</th>
<th>Current Value @ DP-3</th>
<th>No.</th>
<th>Item</th>
<th>Current Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>CONTRACTOR</td>
<td>1</td>
<td>16.</td>
<td>GUIDANCE QUAL TESTS</td>
<td>6.00</td>
</tr>
<tr>
<td>2.</td>
<td>GUIDANCE APPROACH</td>
<td>1</td>
<td>17.</td>
<td>FLIGHT TESTS</td>
<td>17.00</td>
</tr>
<tr>
<td>3.</td>
<td>GUIDANCE CONFIGURATION</td>
<td>2</td>
<td>18.</td>
<td>MAXIMUM COST</td>
<td>$53.00</td>
</tr>
<tr>
<td>4.</td>
<td>MAINTAINABILITY ENG $/h</td>
<td>1.75</td>
<td>19.</td>
<td>MINIMUM COST</td>
<td>$47.00</td>
</tr>
<tr>
<td>5.</td>
<td>VALUE ENG $/h</td>
<td>0.75</td>
<td>20.</td>
<td>MAX COST INCENTIVE %</td>
<td>4.00</td>
</tr>
<tr>
<td>6.</td>
<td>PARALLEL DEVELOP $/h</td>
<td>0.00</td>
<td>21.</td>
<td>LATEST WEEK</td>
<td>238.00</td>
</tr>
<tr>
<td>7.</td>
<td>MOTOR RELIABILITY %</td>
<td>92.00</td>
<td>22.</td>
<td>EARLIEST WEEK</td>
<td>202.00</td>
</tr>
<tr>
<td>8.</td>
<td>AIRFRAME RELIABILITY %</td>
<td>97.00</td>
<td>23.</td>
<td>MAX DELIVER INCENTIVE %</td>
<td>3.50</td>
</tr>
<tr>
<td>9.</td>
<td>LAUNCHER RELIABILITY %</td>
<td>98.50</td>
<td>24.</td>
<td>MAXIMUM RELIABILITY %</td>
<td>81.00</td>
</tr>
<tr>
<td>10.</td>
<td>FIRE CONTROL ERROR YDS</td>
<td>80.00</td>
<td>25.</td>
<td>MINIMUM RELIABILITY %</td>
<td>75.00</td>
</tr>
<tr>
<td>11.</td>
<td>GUIDANCE ERROR YDS</td>
<td>70.00</td>
<td>26.</td>
<td>MAX RELIABILITY INCENTIVE %</td>
<td>3.50</td>
</tr>
<tr>
<td>12.</td>
<td>MOTOR QUAL TESTS</td>
<td>25.00</td>
<td>27.</td>
<td>MAXIMUM ERROR YDS</td>
<td>160.00</td>
</tr>
<tr>
<td>13.</td>
<td>AIRFRAME QUAL TESTS</td>
<td>6.00</td>
<td>28.</td>
<td>MINIMUM ERROR YDS</td>
<td>140.00</td>
</tr>
<tr>
<td>14.</td>
<td>LAUNCHER QUAL TESTS</td>
<td>3.00</td>
<td>29.</td>
<td>MAX ERROR INCENTIVE %</td>
<td>4.00</td>
</tr>
<tr>
<td>15.</td>
<td>FIRE CONTROL QUAL TESTS</td>
<td>3.00</td>
<td>30.</td>
<td>WEEK FOR LOT 10</td>
<td>337.00</td>
</tr>
</tbody>
</table>

**IMPORTANT**

***You must check the above entries with the approved final DP-3.***

Enter your team security code if you wish to commit to a decision; or "cont".
### Table VIII

**Accepted Values Table**

<table>
<thead>
<tr>
<th>No.</th>
<th>Item</th>
<th>Current Value</th>
<th>Proposed Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CONTRACTOR</td>
<td>1</td>
<td>16. G. Q. Tests</td>
</tr>
<tr>
<td>2</td>
<td>GUIDANCE APPROACH</td>
<td>1</td>
<td>17. FLIGHT TESTS</td>
</tr>
<tr>
<td>3</td>
<td>GUIDANCE CONFIGURATION</td>
<td>2</td>
<td>18. MAX COST</td>
</tr>
<tr>
<td>4</td>
<td>MAINTAINABILITY ENG $H$</td>
<td>1.75</td>
<td>19. MINIMUM COST</td>
</tr>
<tr>
<td>5</td>
<td>VALUE ENG $M$</td>
<td>0.75</td>
<td>20. MAX COST INCENTIVE</td>
</tr>
<tr>
<td>6</td>
<td>PARALLEL DEVELOP $M$</td>
<td>0.0</td>
<td>21. LATEST WEEK</td>
</tr>
<tr>
<td>7</td>
<td>MOTOR RELIABILITY %</td>
<td>92.00</td>
<td>22. EARLIEST WEEK</td>
</tr>
<tr>
<td>8</td>
<td>AIRFRAME RELIABILITY %</td>
<td>97.00</td>
<td>23. MAX DELIVER INCENTIVE</td>
</tr>
<tr>
<td>9</td>
<td>LAUNCHER RELIABILITY %</td>
<td>98.50</td>
<td>24. MAXIMUM RELIABILITY</td>
</tr>
<tr>
<td>10</td>
<td>FIRE CONTROL YDS $Y$</td>
<td>80.00</td>
<td>25. MINIMUM RELIABILITY</td>
</tr>
<tr>
<td>11</td>
<td>GUIDANCE ERROR $Y$</td>
<td>70.00</td>
<td>26. MAX RELIABILITY INCENTIVE</td>
</tr>
<tr>
<td>12</td>
<td>MOTOR QUAL TESTS $S$</td>
<td>25.00</td>
<td>27. MAXIMUM ERROR YDS</td>
</tr>
<tr>
<td>13</td>
<td>AIRFRAME QUAL TESTS $E$</td>
<td>6.00</td>
<td>28. MINIMUM ERROR YDS</td>
</tr>
<tr>
<td>14</td>
<td>LAUNCHER QUAL TESTS $S$</td>
<td>3.00</td>
<td>29. MAX ERROR INCENTIVE</td>
</tr>
<tr>
<td>15</td>
<td>FIRE CONTROL QUAL TESTS $S$</td>
<td>3.00</td>
<td>30. WEEK FOR LOT 10</td>
</tr>
</tbody>
</table>

*See your monitor if you have a problem.*

**IH0001A PAUSE; PRESS <<ENTER>> TO CONTINUE.**
<table>
<thead>
<tr>
<th></th>
<th>3 3 112</th>
<th>1.75</th>
<th>0.75</th>
<th>0.0</th>
<th>92.00</th>
<th>97.00</th>
<th>88.50</th>
<th>89.00</th>
<th>70.00</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>25.00</td>
<td>6.00</td>
<td>3.00</td>
<td>3.00</td>
<td>6.00</td>
<td>17.00</td>
<td>47.00</td>
<td>4.00</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>238.00</td>
<td>202.00</td>
<td>3.50</td>
<td>81.00</td>
<td>75.00</td>
<td>3.50</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>160.00</td>
<td>140.00</td>
<td>4.00</td>
<td>337.00</td>
<td>6.00</td>
<td>FINAL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>238.00</td>
<td>202.00</td>
<td>3.50</td>
<td>81.00</td>
<td>75.00</td>
<td>3.50</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>160.00</td>
<td>140.00</td>
<td>4.00</td>
<td>337.00</td>
<td>6.00</td>
<td>PROPOSED</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0 0 0 0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td>1981P</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>00000000</td>
</tr>
</tbody>
</table>
Use of the security code in this feature was designed for 'human factor' engineering. The conscious process of entering the team's security code indicates the user's deliberate entry of the data. Inadvertent acceptance of the database for the team's position cannot occur.

The team does retain the responsibility for reading and understanding the data displayed in the table of data, Table VII.

The process of the team submitting a final contract proposal provides the program with the position agreed to during contract negotiations. Submission of the final report indicates to the program that the team has completed a DP level. PROJ.MNG assumes the team is ready to proceed to the next DP level, for example DP-4. It utilizes the previous DP level data entries from the submitted final report and prepares the new DP data table. The team is sequenced to the next higher level whenever a final proposal is detected for a given DP level.

I. CALCULATION CORRECTION

As discussed in Chapter 2 several of the computations which were designed for CNP to accomplish were not satisfactory. Consequently their display was suppressed.

In PROJ.MNG these calculations have been reformulated. Their algorithm is based on the linear extrapolation of a fee over the range of its allotted parameter values. The contract cost for the fee incentive is calculated by:

1. Assume achieving optimum performance results in total fee payment.
2. Assume achieving the worst performance expected results in no fee payment.
3. Determine the ratio of satisfactory achievement; the
difference between the achieved value and the worst
value expected divided by the range of value between
worst and optimum values expected.
4. Multiply the ratio determined by the maximum fee
percentage offered for the incentive.
5. Multiply the resulting percentage by the target
development cost, the average value of the maximum
and minimum desired development cost.

The following equation is used in PROJMNNG to determine
fee ratio.

FEERATIO = FEEMAX * (ABS(WORST - ACHIEVED)) / (RANGE)  (eqn 3.1)

This formulation for fee costs has proven to be more
satisfactory than that of CNP.

The difficulty in CNP arises from numerical manipulation.
Parameters in CNP frequently have their power-of-ten
changed to permit calculations at different resource multi-
plies. This is particularly true in the case of achieved
development cost. Frequent multiplication by 1,000,000 with
division by 1,000,000 to return the dollar values to
millions of dollars incurs computer roundoff. The roundoff
errors when repeated several times become sufficiently large
to be apparent in the ratio calculations.

PROJMNNG does not use the numbers which the computer used
in computing achieved value resources to determine the fee
values. Its numbers are taken directly from the database
array, and are only mathematically manipulated by this one
calculation. This process results in only one iteration of
roundoff error.
PROJMNNG is not able to eliminate computer rounding error, but the rounding error effect has been significantly reduced.

J. SUMMARY

PROJMNNG's design has its foundation in CNP's routines which were designed around DMS. It uses this foundation from CNP, and develops an extended program. The program extension uses the current IBM-3033 facilities to drive user-directed options in a manner which provides the team control over program execution for completion of the DMS exercise. These features for team use of PROJMNNG make PROJMNNG easier to use in analyzing team positions. Chapter 4 discusses the analysis capabilities of PROJMNNG which are available to the monitor.
IV. RECOMMENDING MONITOR OPERATION

A. CNP MONITOR UNIQUE FUNCTIONS

CNP has one routine capable of providing support to the monitor for his evaluations of the scenario progress. This singular routine is not available to the teams. The monitor can access the COST + FACTORS Table, Table X as discussed in the previous chapter. Access to the monitor routine occurs by the computer operator's response to the query for "COST + FACTORS ?". This response is coded to provide security of the COST + FACTORS table from team access. In order to preserve its integrity, the security code will not be given or demonstrated here. Correct security code responses to both the first "COST + FACTORS ?" and a second query of "COST + FACTORS ?" are necessary. The first response is a three character code word. The second query response is based on a data field used in the computations which has already been entered for the session. It constitutes a variable security code.

In CNP, an additional monitor capability is provided through the confirmation query, "CONFIRM ?", at the DP number verification sequence, "IS THIS DP NUMBER ** xx;"Y"=YES OR "N"=NC ?". Response to this query with a fixed single letter, places the operator in the "INSTRUCTOR" mode. Analysis of the program's Fortran code indicates that the INSTRUCTOR mode simply suppresses the queries for confirmation and the repeat back display of input data. Answering the statements "IS THIS DP NUMBER ** xx;"Y"=YES OR "N"=NC ?" with a "T" reduces the displayed queries. It eliminates the repeated displays of "RECEIVED .......: xxxxx; CONFIRM ?". It also eliminates the repeat back printout of input page data.
### TABLE X

**COST + FACTORS**

<table>
<thead>
<tr>
<th>Components</th>
<th>Cost Per Unit (Thousands of Dollars)</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUALIFICATION TESTS</td>
<td></td>
</tr>
<tr>
<td>MOTOR</td>
<td>32.034</td>
</tr>
<tr>
<td>AIR FRAME</td>
<td>126.355</td>
</tr>
<tr>
<td>GUIDANCE</td>
<td>309.906</td>
</tr>
<tr>
<td>FIRE CONTROL</td>
<td>938.186</td>
</tr>
<tr>
<td>LAUNCHER</td>
<td>424.555</td>
</tr>
<tr>
<td>FLIGHT TESTS</td>
<td>316.855</td>
</tr>
</tbody>
</table>

**Design Factors Achieved**

<table>
<thead>
<tr>
<th>Component</th>
<th>Table Row</th>
<th>Factor Achieved</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOTOR</td>
<td>25</td>
<td>0.825</td>
</tr>
<tr>
<td>AIR FRAME</td>
<td>15</td>
<td>0.958</td>
</tr>
<tr>
<td>GUIDANCE A</td>
<td>22</td>
<td>0.660</td>
</tr>
<tr>
<td>GUIDANCE E</td>
<td>25</td>
<td>0.566</td>
</tr>
<tr>
<td>FIRE CONTROL</td>
<td>16</td>
<td>0.846</td>
</tr>
<tr>
<td>LAUNCHER</td>
<td>7</td>
<td>0.963</td>
</tr>
<tr>
<td>LAUNCHER</td>
<td>7</td>
<td>0.963</td>
</tr>
</tbody>
</table>

The loss of page data repeat back and confirmation, and the format of Table I have not been appealing to users of the program. The "T" letter code system was not human factor engineered. Loss of the data item confirmation sequence increases the occasions when inaccurate results are received in the Development Contract Summary Table. Users
have not found benefit in these options. Additional experience with CNP is necessary to use these features successfully. Consequently, use of the "T" feature has not been used. The COST + FACTORS table has been found to be less than adequate.

B. PROJECT MONITOR UNIQUE FUNCTIONS

In PROJMONG a variety of options are available for the monitor. They include:

1. Monitor Unique Security System
2. Query Suppression for the Monitor
3. Monitor's MAIN MENU
4. File Access Options
5. CCST + FACTORS
6. Sensitivity Analysis
7. Graphing
8. Copies of Student Files

This chapter will review the intended capability and operation of these monitor unique functions.

C. MONITOR UNIQUE SECURITY SYSTEM

The PROJMONG EXEC routine looks through the computer operator's disk for a file titled "DATAINST DISKNUM". Having located this file, the computer assumes that the operator is a monitor, and proceeds to validate the monitor identity. The program contains three file checks to determine if the operator is a monitor. Unsuccessful completion of the first two checks will cause the computer to ignore the DATAINST DISKNUM file and assume the computer user is a student.
The first of these checks is the file format. The file first line must contain the userid. The second line must contain a monitor specified security code. This monitor security code is queried in the same manner in which the team security code is queried for a team disk.

The third verification is a check of the stored monitor security code against the individual's ability to reproduce that code. A code match allows the program to proceed. Failure to accurately match the code causes the program to stop. PROJMG EXEC takes control to finish its execution. The user can reaccess PROJMG FORTRAN in his response to "ANOTEER RUN, 'Y'/'N'?".

D. QUERY SUPPRESSION FOR THE MONITOR

Successfully accessing the program as a monitor suppresses the student's Main Menu, discussed in chapter 3. Consequently, the monitor does not receive the student's Main Menu after each alteration of the Selection Menu. Based on the assumption that the computer operator is a monitor the program immediately follows the Selection Menu with the Development Contract Summary, Table XI.

Table XI demonstrates the display sequence for the Selection Menu with the execution of option "0". The monitor's response to the Selection Menu with a zero causes the immediate execution of the achieved value calculations. Other selections from this table are performed similarly to those done for the team. The changes entered by the monitor in this first mode will not be stored in the database by the program. In monitor mode 1, the call to the routine STORE does not occur. Only the array of data in the disk's buffer will be altered. The monitor sees the computer response to his entry of a "0" as an immediate display of the Development Contract Summary.
### TABLE XI
**SELECTION MENU**

<table>
<thead>
<tr>
<th>NO.</th>
<th>ITEM</th>
<th>CURRENT VALUE</th>
<th>NO.</th>
<th>ITEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CONTRACT</td>
<td>1</td>
<td>16.</td>
<td>GUIDANCE QUAL TESTS</td>
</tr>
<tr>
<td>2</td>
<td>GUIDANCE APPROACH</td>
<td>1</td>
<td>17.</td>
<td>FLIGHT TESTS</td>
</tr>
<tr>
<td>3</td>
<td>GUIDANCE CONFIGURATION</td>
<td>2</td>
<td>18.</td>
<td>MAXIMUM COST M$</td>
</tr>
<tr>
<td>4</td>
<td>MAINTAINABILITY ENG M$</td>
<td>1.75</td>
<td>19.</td>
<td>MINIMUM COST M$</td>
</tr>
<tr>
<td>5</td>
<td>VALUE ENG M$</td>
<td>0.75</td>
<td>20.</td>
<td>MAX COST INCENTIVE %</td>
</tr>
<tr>
<td>6</td>
<td>PARALLEL DEVELOP M$</td>
<td>0.0</td>
<td>21.</td>
<td>LATEST WEEK</td>
</tr>
<tr>
<td>7</td>
<td>MOTOR RELIABILITY</td>
<td>92.00</td>
<td>22.</td>
<td>EARLIEST WEEK</td>
</tr>
<tr>
<td>8</td>
<td>AIRFRAME RELIABILITY</td>
<td>97.00</td>
<td>23.</td>
<td>MAX DELIVER INCENTIVE</td>
</tr>
<tr>
<td>9</td>
<td>LAUNCHER RELIABILITY</td>
<td>98.50</td>
<td>24.</td>
<td>MAXIMUM RELIABILITY %</td>
</tr>
<tr>
<td>10</td>
<td>FIRE CONTROL ERROR YDS</td>
<td>80.00</td>
<td>25.</td>
<td>MINIMUM RELIABILITY %</td>
</tr>
<tr>
<td>11</td>
<td>GUIDANCE ERROR YDS</td>
<td>70.00</td>
<td>26.</td>
<td>MAX RELIABILITY INCENTIVE</td>
</tr>
<tr>
<td>12</td>
<td>MOTOR QUAL TESTS</td>
<td>25.00</td>
<td>27.</td>
<td>MAXIMUM ERROR YDS</td>
</tr>
<tr>
<td>13</td>
<td>AIRFRAME QUAL TESTS</td>
<td>6.00</td>
<td>28.</td>
<td>MINIMUM ERROR YDS</td>
</tr>
<tr>
<td>14</td>
<td>LAUNCHER QUAL TESTS</td>
<td>3.00</td>
<td>29.</td>
<td>MAX ERROR INCENTIVE %</td>
</tr>
<tr>
<td>15</td>
<td>FIRE CONTROL QUAL TESTS</td>
<td>3.00</td>
<td>30.</td>
<td>WEEK FOR LCT</td>
</tr>
</tbody>
</table>

**Q. NO:** 31. **CHANGE BY PAGE**

**SELECT AN ITEM WHICH NEEDS TO BE CHANGED FOR DP-3.**

<=="NOTE: The query '?' response "0".

---

**DP-3 "DEVELOPMENT CONTRACT SUMMARY" TEAM 3**

<table>
<thead>
<tr>
<th>INCENTIVE PROVISIONS</th>
<th>INCENTIVE ACHIEVEMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>INCENTIVE AREA</td>
<td>WORST VALUE</td>
</tr>
<tr>
<td>DEV. COST</td>
<td>$53.0M</td>
</tr>
<tr>
<td>FIT. TST. COMPLI</td>
<td>236 WK</td>
</tr>
<tr>
<td>RELIABILITY</td>
<td>75.0%</td>
</tr>
<tr>
<td>ACCURACY</td>
<td>160 YDS</td>
</tr>
<tr>
<td><strong>TOTALS</strong></td>
<td></td>
</tr>
<tr>
<td><strong>TOTAL CONTRACT PRICE</strong></td>
<td></td>
</tr>
</tbody>
</table>
E. MONITOR'S MAIN MENU

The PROJMNG monitor's MAIN MENU contains options to allow the monitor access to any of the routines in the program. It has a unique feature in its flexible length. This feature provides menu listings to the monitor which are determined by his status in the program. If the monitor has just entered PROJMNG or has changed the team number, and has not completed the computations for other display values, the program will suppress those options from the menu display. Options which require calculations or selections which have not been processed for the team are assumed to present results which are either not applicable or misleading.

<table>
<thead>
<tr>
<th>MAIN MENU:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. CHANGE TEAM NUMBER</td>
</tr>
<tr>
<td>2. THE DP SELECTION QUERY</td>
</tr>
<tr>
<td>3. THE INPUT SELECTION MENU</td>
</tr>
<tr>
<td>4. RERUN THE DATA CALCULATION</td>
</tr>
<tr>
<td>5. DO A SENSITIVITY ANALYSIS</td>
</tr>
<tr>
<td>6. PRINTOUT COST FACTORS</td>
</tr>
<tr>
<td>7. EXIT</td>
</tr>
</tbody>
</table>

Figure 4.1  Monitor's Main Menu.

Figure 4.1, demonstrates the basic monitor's Main Menu.

1. Change Team Number

Option 1 when selected will return the monitor to the program's team input routine (TEAMIN). The execution of this routine begins by displaying a menu, Figure 4.2. The selections on this menu are:

59
SELECT WHICH INSTRUCTOR MODE YOU DESIRE.
1. RUN TEAM SCENARIOS WHICH ARE ON THE MONITOR'S DISK.
2. CHANGE TEAM FILES.
   THIS MODE WILL ALTER THE STUDENT'S FILES.
3. RUN THE MONITOR'S TEAM FILE 'TEAM21'.
4. EXIT

Figure 4.2 Monitor Mode Menu.

a. Run Team Scenarios from the PROJMNG Disk

Selecting this choice enables the monitor to run evaluations and displays of the teams whose DATAFILES have been written on the PROJMNG disk. PROJMNG has at this point already written these files to the individual monitor's disk for use. As the monitor selects a team to review, the program determines if the team file exists. If the file exists, the monitor may proceed with the evaluation. If no files exist for the team requested, PROJMNG responds with 'FILES DO NOT EXIST FOR TEAMxx', and proceeds to ask for a different team number, 'WHAT TEAM NUMBER?', until a team number is submitted for which files are present.

This option selects a mode of operation in which team data files are not changed by the monitor. The menus displayed are only those associated with the monitor's options. The Monitor's Main Menus are flexible menus. The discussion of this flexibility and of the basic menu will be provided in the following sections of this chapter. These differences are demonstrated by the two monitor menu displays for mode 1, Figure 4.3, and for mode 2, Figure 4.4.
b. Change Team Files

Option 2 has an entirely different affect on the database. Again, the files accessed are those previously in existence. The monitor receives a different version of the MAIN MENU. The added option in this menu, "7. FILE TEAM PROPOSAL" permits the monitor to change the DP report level for any of the team's DPs irregardless of the team's current level.

The monitor can perform all of the functions he could in mode 1 plus changing the report level and placing these selections in the team database. This mode should only
be used to correct team errors. Team database manipulation for any other reason will lead to team confusion.

The query which finalizes the replacement of the team database, "SHOULD THE REWRITTEN TEAM xx FILE BE SENT TO THE PROJMNNG disk: Y/N ?". Unless this query is answered in the affirmative, the mode 2 capability to overwrite the team file is not performed.

Answering yes to the query to rewrite the team file directs PROJMNNG through a routine which performs three calls to the subroutine FRCTCMS which permits CMS executive level commands to be executed from within a Fortran program. The three commands in this routine are the same as the spool commands in PROJMNNG EXEC which send the file DATAFILE TEAMxx to the PROJMNNG disk. The disk punch file is spooled as a class x file. The DATAFILE TEAMxx is punched into the spocl. Finally the spool is closed and sent to the PROJMNNG disk.

The program does allow the DATAFILes on the monitor's personal disk to be rewritten whenever mode 2 operation is selected. Transmission of these files to the master PROJMNNG disk can be accomplished after responding in the negative to the query "SHOULD THE REWRITTEN TEAM xx FILE BE SENT TO THE PROJMNNG disk: Y/N ?". Sending these files to the master disk must be a deliberate act of the monitor. The monitor may either use one of the macro routines in general usage on the IBM-3033 for sending files between users, "SEND". Or, he may spool the files for immediate read by the PROJMNNG file using the following sequence of steps. (The lower case characters in the following command demonstration are optional inputs.)

1. "SPOCl PUNCH CCNT CLASS X 0276P"
2. "PUNCH DATAFILE TEAM(XX)", where xx is the two digit team number.
3. "SPOCl PUNCH CLOSE CCNT"
In mode 1 or mode 2 operation, only the Development Contract Summary table resembles the same display for the teams.

c. Run the monitor's team file "TEAM21"

The third option provides a team file for the monitor. TEAM21 is the monitor's own team file to simulate the team interaction with the program. This provides a scratch pad file for the monitor's use.

d. EXIT

Option 4 from the monitor's Main Menu stops the program. Entering a '4' in response to this menu directs the program to stop. PROJMGN EXEC resumes control of the computer processing.

2. The DP Selection Query

Number 2 on the monitor's mode menu reloops the program to the DP query for which DP the monitor desires to evaluate. This query will not prevent the monitor from accessing DF levels for which the teams have not submitted files. The level which the team has completed is displayed on the Selection Menu, with the DP query, and on the proposed report table. This information should be known by the monitor based on the progress of the DMS scenario.

3. The Input Selection Menu

Again, the menu routine simply reloops the program execution to a previously executed question to permit the monitor to change his mode of analysis. In this case he returns to the team Selection Menu. The last team and DP inputs entered during the monitor's session will apply to the team database which is displayed.
Charges to a team's list of data in the program buffer which have been entered during a session remain applicable until the team number is changed. Any variations to the Selection Menu which have been selected for the team currently being evaluated will be in effect. In mode one, a team's data will revert to the PROJ/MNG disk's team datafile values when changing team numbers. Mode 2 will keep the changed values until the PROJ/MNG disk values are reccpied down onto the monitor's disk.

4. **Rerun the Data Calculation**

   This option causes PROJ/MNG to redo the achieved value calculations. The display for this selection is a repeat of the Development Cost Summary.

5. **To a Sensitivity Analysis**

   This option creates a major option for exploration by the monitor. It is discussed as part of the sequential demonstration of the monitor functions which follows in Section I below.

6. **Printout Cost + Factors**

   Similar to the capability for COST + FACTORS demonstrated by CNP, PROJ/MNG has an equivalent feature for the monitor. Table XII shows the improved layout of the data. The table has the benefit over CNP's version by not requiring a separate code to access it. The table is simply accessed as a menu selection from the monitor's Main Menu.

7. **EXIT**

   This is another example of PROJ/MNG's ability to stop the program.
TABLE XII
COST + FACTORS

COSTS OF TESTS PER UNIT (IN THOUSANDS OF DOLLARS)

QUALIFICATION TESTS
- MOTOR : 32,034
- AIR FRAME : 126,355
- GUIDANCE : 309,906
- FIRE CONTROL : 938,189
- LAUNCHER : 424,555
- FLIGHT TESTS : 316,855

DESIGN FACTORS ACHIEVED

<table>
<thead>
<tr>
<th>COMPONENT</th>
<th>TABLE</th>
<th>ROW</th>
<th>FACTOR</th>
<th>ACHIEVED</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOTOR</td>
<td>1</td>
<td>25</td>
<td>0.825</td>
<td>0.877</td>
</tr>
<tr>
<td>AIR FRAME</td>
<td>2</td>
<td>15</td>
<td>0.958</td>
<td>0.980</td>
</tr>
<tr>
<td>GUIDANCE A</td>
<td>5</td>
<td>22</td>
<td>0.666</td>
<td>0.630</td>
</tr>
<tr>
<td>GUIDANCE B</td>
<td>6</td>
<td>22</td>
<td>0.589</td>
<td>0.000</td>
</tr>
<tr>
<td>FIRE CONTROL</td>
<td>4</td>
<td>10</td>
<td>0.845</td>
<td>0.802</td>
</tr>
<tr>
<td>LAUNCHER</td>
<td>3</td>
<td>7</td>
<td>0.938</td>
<td>0.988</td>
</tr>
</tbody>
</table>

IHC001A PAUSE; PRESS <<ENTER>> TO CONTINUE.

8. Flexible Menu

The options presented in this section are the basic capabilities of PROJMN. However, additional capabilities become possible as the evaluation progresses. With the completion of the sensitivity analysis additional tables and graphs are within the capability of PROJMN. These flexible menu capabilities are explained and demonstrated in Section I of this chapter.

F. MONITOR SESSION PROCEDURES

The remainder of the discussion on menus will be presented sequentially in the following sections. Their relative position in the textual sequence corresponds to their occurrence in the process of stepping through the program's routines.
These sequential steps of PROJMNG are demonstrated as the program would be operated. The material in this demonstration is preliminary to the preparation of an instructor's manual. The sections are:

1. Initialization
2. Basic routines of the Main Menu
3. Sensitivity analysis
4. Logoff unique routines

G. INITIALIZATION

1. PROJMNG EXEC Operation

The monitor's access to the PROJMNG package of programs is also accomplished by LINKPROJ. The determination that the disk accessing PROJMNG is a monitor is made, as discussed in Chapter 3. Having completed the checks of the monitor's file DATAINST DISKNUM, the PROJMNG EXEC program conducts a check of files for a DATAFILE TEAM21. This file must be present on the monitor's disk if he is going to use it in PROJMNG. PROJMNG EXEC either finds the file or creates one in order to have the file ready for PROJMNG FORTRAN's read and write instructions.

2. Available Files

The lack of presence of the files which PROJMNG EXEC looks for is an automatic function of the IBM-3033. This capability was retained to display to the monitor a list of teams from which he has not received reports, as shown in Figure 4.5. Offsetting of the display from the left margin was performed in order to highlight between the computer printout and the typed queries.
A benefit of the "PRESS <<ENTER>>" set of statements is the ability to begin a display on a fresh screen. Displays which should appear as one screen occur readily when the pause has finished the previous display to keep it displayed while the user is reading it. The pause can be followed by an immediate command in FRTCMS to 'CLRSCRN ', clear the screen. This sequence of commands leaves a fresh screen for the next screen display. The first line that PROJWNG prints out appears as the top line of a display (see Figure 4.7).

Cueing the user to press <<ENTER>> identifies a pause which has been placed in order to organize screen displays. The pause in Figure 4.5 enables the display of the teams 'not found' to be isolated on one display. There can be up to 21 teams displayed, as for example at the start of an exercise scenario. Without the pause to segregate the displays, all 21 teams would not appear on one screen. The single screen full of one type of data aids in user ability to study and analyze the data. Splitting the display to a second view eliminates the first screen's data and taxes the user's recall to do any comparisons. Isolating the data onto one screen eliminates these disruptions to the user's study of the data. This technique of segregating related information onto a screen has been used throughout this program.

3. Purging DATAFILES

In order to provide a convenient method of purging files from the completed exercises, a routine was generated which permits the monitor to purge all of the existing files, to conduct normal operation of the program or to exit. These options are depicted in the display as shown in Figure 4.6. The purge option purges only the available 'A' disk DATAFILE TEAMxx and DATACODE TEAMxx. Should the
BEGIN RECORDING OF TERMINAL SESSION
* NOTE!: YOU ARE NOW LINKED TO PROJMG ON YOUR 192 DISK, MODE C
* NOTE!: PRESS PFCS TO BREAK THE LINK
PRESS <<ENTER>> WHEN YOU ARE READY TO CONTINUE.

*****************************************************
** EXECUTION IS IN PROGRESS. WAIT, DO NOT PRESS ENTER.*****
FILE "DATAFILE TEAM21 A" NCI FOUND.
FILE "DATAFILE TEAM11 C" NCI FOUND.
FILE "DATAFILE TEAM12 C" NCI FOUND.
FILE "DATAFILE TEAM13 C" NCI FOUND.
FILE "DATAFILE TEAM14 C" NCI FOUND.
FILE "DATAFILE TEAM15 C" NCI FOUND.
FILE "DATAFILE TEAM16 C" NCI FOUND.
FILE "DATAFILE TEAM17 C" NCI FOUND.
FILE "DATAFILE TEAM18 C" NCI FOUND.
FILE "DATAFILE TEAM19 C" NCI FOUND.
FILE "DATAFILE TEAM20 C" NCI FOUND.

Figure 4.5 Initial Monitor Display.

PROJMG EXEC purge function be performed on the PROJMG FORTRAN's disk the entire exercise scenario would be eliminated. Only database files pre-existing on team files or monitor's files could be reaccessed for PROJMG. With a new play of the exercise, it is expected that a new group of teams will be accessing PROJMG. No conflict with former team databases should be experienced once the monitor's and the PROJMG's disks have been purged.

Normal operation initiates PROJMG FORTRAN execution. The standard procedure during a game exercise is designed to respond to this query with an "N". The word length difference between PURGE and N has been designed to make inadvertent destruction of the team files more difficult.

"E" again provides a means of stopping the program. In this instance, "E" will stop PROJMG EXEC and return the computer to running the LINKPROJ program. The final display
of LINKPROJ, "ANOTHER RUN, 'Y' OR 'N'?", would be the query following the request to stop.

"WAIT FOR EXECUTION TO BEGIN" became a necessary printout as the result of long waits for execution to begin on the NPS IBM-3033. (One monitor reported experiencing a 20 minute delay during very heavy use of the computer facility, with the "WAIT FOR EXECUTION..." label reminding him that the program was in the queue to be run.)

```
*****WARNING*****
*****ANSWERING 'PURGE' TO THE FOLLOWING QUERY WILL*****
*****PURGE ALL EXISTING STUDENT DATAFILES*****
*****ON THE DISK YOU ARE USING.*****

DC YOU WISH TO INITIALIZE ALL TEAM FILES
FOR A NEW EMS EXERCISE?
'PURGE' =PURGE ALL TEAM DATAFILES
'N' =PROCEED WITH NORMAL OPERATION
'E' =STOP

====>n
EXECUTION BEGINS...

WAIT FOR "EXECUTION TO BEGIN".
```

Figure 4.6 Monitor Session File Purge Option.

These initial queries, printouts and displays by PROJMNNG in the monitor mode of operation should appear to the reader as very similar to those presented to the teams. It should also be noticed in Figure 4.7 that the program does not ask for a team security code input to create the team's code. Instead, it has already received the monitor's security code from PROJMNNG EXEC. The monitor's code is found by PROJMNNG FORTRAN in DATAINST DISKNUM.

H. BASIC PROGRAM ROUTINES

The program routines discussed in this section perform similar processes in both CNP and PROJMNNG.
1. **Start of PROJMNNG's Run**

The stacked data items are read by PROJMNNG as its first step in the routine for team number input "TEAMIN".

---

**TO TERMINATE THE PROGRAM AT ANY POINT, TYPE "Y"=EXIT IN RESPONSE TO ANY "YES/NO" QUERY.**

**CAUTION** IN RESPONSE TO A QUERY, PRESSING "<ENTER>" WITHOUT PROVIDING DATA WILL DUMP THE PROGRAM.

PLEASE ENTER YOUR SECURITY CODE.

---

**Figure 4.7 PROJMNNG Initial Displays.**

Having received the FL21 flag of TEAM21, and validated the monitor's security code, PROJMNNG can proceed. If the security code should be entered incorrectly, the PROJMNNG execution is terminated. The monitor receives a warning of the error, and is asked to reinitiate his startup of the program. He may either answer 'Y' to "ANOTHER RUN, 'Y' OR 'N'?" or reenter "LINKPROJ" after PROJMNNG EXEC and LINKPJMNG have finished execution.

PROJMNNG FORTRAN asks for the team number when a monitor is running the program. Team selection within the program is necessary in order to permit the evaluation of more than one team. Whereas a team will only enter the program with one team scenario to perform, the monitor is able to change teams without reinitiating the program.

**Figure 4.8 demonstrates the response given by PROJMNNG when it cannot find files for the team number requested in either options 1 or 2 of the monitor's Main Menu.** The read cue '?' indicates to the operator that an
input is expected. In the operations demonstrated throughout this thesis '===' indicates the user input in response to the EBCJM.NG queries. Figure 4.8 also presents a case in which the program does find the requested files and proceeds to query the monitor for which DP level he desires to evaluate.

```
WHAT TEAM NUMBER?
====> 2  RECORDS DO NOT EXIST FOR TEAM 2
WHAT TEAM NUMBER?
====> 3  WHAT DECISION POINT DO YOU WISH TO ANALYZE?
====> 3
```

Figure 4.8 Team Number Responses. The queries illustrated in Figure 4.8 complete the functions of the program routine TEAMIN. The program returns to the main program text and enters the SELECT routine which displays the Selection Menu as it did for the teams, Table XIII. Table XIII functions the same as before except that the entries are not stored in the team's database file. The changes are made only to the monitor's disk data buffer arrays. In option 2 of the monitor's main menu, the changes are also stored on the monitor's copy of the team files in his file listing. These changes are not applied to the PROJMNG disk files unless the monitor specifically sends the changed files to that disk.

In response to option '0' in Table XIII PROJMNG immediately completes the calculation of fee percentages in the program routine PRCTCK. Successful completion of the percentage check is followed with the calculation of achieved values. This process occurs with the program calls to the routines:
TABLE XIII
Selection Menu

<table>
<thead>
<tr>
<th>NO.</th>
<th>ITEM</th>
<th>CURRENT VALUE</th>
<th>DP-3</th>
<th>NO.</th>
<th>ITEM</th>
<th>CURRENT VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CONTRACTOR</td>
<td>1</td>
<td></td>
<td>16</td>
<td>GUIDANCE CUAL TESTS</td>
<td>6.00</td>
</tr>
<tr>
<td>2</td>
<td>GUIDANCE APPROACH</td>
<td>1</td>
<td></td>
<td>17</td>
<td>FLIGHT TESTS</td>
<td>17.00</td>
</tr>
<tr>
<td>3</td>
<td>GUIDANCE CONFIGURATION</td>
<td>2</td>
<td></td>
<td>18</td>
<td>MAXIMUM COST</td>
<td>M$: 53.00</td>
</tr>
<tr>
<td></td>
<td>MAINTAINABILITY ENG M$</td>
<td>1.75</td>
<td></td>
<td>19</td>
<td>MINIMUM COST</td>
<td>M$: 47.00</td>
</tr>
<tr>
<td>4</td>
<td>VALUE ENG M$</td>
<td>0.75</td>
<td></td>
<td>20</td>
<td>MAX COST INCENTIVE %</td>
<td>4.00</td>
</tr>
<tr>
<td>5</td>
<td>PARALLEL DEVELOP M$</td>
<td>0.0</td>
<td></td>
<td>21</td>
<td>LATEST WEEK</td>
<td>238.00</td>
</tr>
<tr>
<td>6</td>
<td>MOTOR RELIABILITY %</td>
<td>92.00</td>
<td></td>
<td>22</td>
<td>EARLIEST WEEK</td>
<td>202.00</td>
</tr>
<tr>
<td>7</td>
<td>AIRFRAME RELIABILITY %</td>
<td>97.00</td>
<td></td>
<td>23</td>
<td>MAX DELIVER INCENTIVE</td>
<td>3.50</td>
</tr>
<tr>
<td>8</td>
<td>LAUNCHER RELIABILITY %</td>
<td>98.50</td>
<td></td>
<td>24</td>
<td>MAXIMUM RELIABILITY %</td>
<td>81.00</td>
</tr>
<tr>
<td>9</td>
<td>FIRE CONTROL ERROR YDS</td>
<td>80.00</td>
<td></td>
<td>25</td>
<td>MINIMUM RELIABILITY %</td>
<td>75.00</td>
</tr>
<tr>
<td>10</td>
<td>GUIDANCE ERROR YDS</td>
<td>70.00</td>
<td></td>
<td>26</td>
<td>MAX RELIABILITY INCENTIVE</td>
<td>3.50</td>
</tr>
<tr>
<td>11</td>
<td>MOTOR QUAL TESTS</td>
<td>25.00</td>
<td></td>
<td>27</td>
<td>MAXIMUM ERROR YDS</td>
<td>160.00</td>
</tr>
<tr>
<td>12</td>
<td>AIRFRAME QUAL TESTS</td>
<td>6.00</td>
<td></td>
<td>28</td>
<td>MINIMUM ERROR YDS</td>
<td>140.00</td>
</tr>
<tr>
<td>13</td>
<td>LAUNCHER QUAL TESTS</td>
<td>3.00</td>
<td></td>
<td>29</td>
<td>MAX ERROR INCENTIVE %</td>
<td>4.00</td>
</tr>
<tr>
<td>14</td>
<td>FIRE CONTROL QUAL TSTs</td>
<td>3.00</td>
<td></td>
<td>30</td>
<td>WEEK FOR LOT 10</td>
<td>337.00</td>
</tr>
</tbody>
</table>

****0. NONE********31. CHANGE BY PAGE****
SELECT AN ITEM WHICH NEEDS TO BE CHANGED FOR DP-3.
1. **ZEW** - sets all non-database variables used in the computations to zero.

2. **INPUT3** - reads the data tables in files FT17F001, FT18F001, FT19F001 and FT20F001.

3. **PRESET** - uses the database values to make initial adjustments to the achieved value indices.

4. **TRADE** - evaluates the team's input tradeoffs and uses them to generate affects which to apply in calculating the achieved value indices.

5. **DESSES** - degrades the performance factors based on requests for system test.

6. **DESES** - degrades the performance for the effects of selected engineering values and performs the calculations for production costs. (This feature, production cost computation, was not used in CNP and has not been activated in PROJNNG.)

7. **SE** - applies resource parameters to the achieved value calculations.

8. **MCD9** - combines the calculation results from the previous three routines to derive a total cost factor for development.

9. **RERUN** - selectively sets recursive variables to zero to enable the program to finish the calculations of one DP and to reloop to compute the values for successive DP's.

10. **REPORT** - completes the calculations after all DP level computations have been completed.

11. **PECSUM** - performs conversion of the achieved values to the appropriate power of ten for display to the user and generates the display, Table XIV.
2. **Summary of Basic Capabilities**

To this point, the basic capabilities of PRCJMNG which it has in common with CNP have been discussed. The improved displays, access and queries discussed parallel those features given to the students, with exception of student access to COSI + FACTORS. In the next section, the 'flexible menu' feature of the monitor's main menu, and the associated capabilities for sensitivity analysis, for data displays, and for analysis graphing are described.

I. **SENSITIVITY ANALYSIS ROUTINES**

1. **Sensitivity Analysis Basic Menu Options**

Figure 4.9 shows again the monitor's menu in mode 1 operation for comparison to the expanded menu which is now described. In order to accomplish the calculation of a sensitivity analysis, option 5 from the menu is selected. In performing this selection, a flag is set 'on' to indicate the availability of values for use in the other displays of the expanded menu options.

```
MAIN MENU:
1. CHANGE TEAM NUMBER
2. THE DP SELECTION QUERY
3. THE INPUT SELECTION MENU
4. REEUN THE DATA CALCULATION
5. DO A SENSITIVITY ANALYSIS
6. PRINTOUT COST+FACTORS
7. EXIT
```

*Figure 4.9 Monitor's Main Menu.*
Selection of number 5 from the menu immediately accesses the program routine SWSITIV. This subroutine provides a selection menu (Table XV), queries parameters and parameter data for variation in the analysis, edits the parameter data to remain within DNS scenario limitations, divides the variable parameter into 10 steps for 11 levels of evaluation, runs eleven cycles through the achieved values calculations, displays the table of achieved values at each level, and stores these values in an array for later use in redisplaying the table and for plotting the computations.

Following the display of Table XV, the monitor makes a selection of which DNS data item to analyze. His selection determines a series of questions which PROJMNG displays in order to resolve the analysis parameter details. Figure 4.10 shows one of these queries. In this case the Flight Tests will be varied. The maximum and minimum values permitted are displayed with the queries for range values. These are not only cues to the user but also are the limitations which PROJMNG imposes.

If the selection exceeds the scenario design limits, PROJMNG will take one of two actions. In the first case, PROJMNG will reset the value to be within the range.

```
====>14
INPUT THE VALUES TO VARY FLIGHT TESTS BETWEEN.
WHAT IS THE LOWER VALUE: >10. ?
====>10
WHAT IS THE UPPER VALUE: <25. ?
====>25
```

Figure 4.10 Flight Tests Demonstration.


<table>
<thead>
<tr>
<th>INCENTIVE AREA</th>
<th>INCENTIVE PROVISIONS</th>
<th>INCENTIVE ACHIEVEMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WORST VALUE</td>
<td>BEST VALUE</td>
</tr>
<tr>
<td>DEV. COST</td>
<td>$53.0M</td>
<td>$47.0M</td>
</tr>
<tr>
<td>FIT TST COMPLI</td>
<td>238 WK</td>
<td>202 WK</td>
</tr>
<tr>
<td>RELIABILITY</td>
<td>75.0%</td>
<td>81.0%</td>
</tr>
<tr>
<td>ACCURACY</td>
<td>160 YDS</td>
<td>140 YDS</td>
</tr>
<tr>
<td>TOTALS</td>
<td>15.0%</td>
<td></td>
</tr>
</tbody>
</table>

TOTAL CONTRACT PRICE = $54.9M
TABLE XV
Sensitivity Analysis Menu

<table>
<thead>
<tr>
<th></th>
<th>ITEM</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MAINTAINABILITY ENG COST</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>VALUE ENGINEERING COST</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>PARALLEL GUIDANCE COST</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>MOTOR RELIABILITY RANGE</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>AIRFRAME RELIABILITY RANGE</td>
<td>17</td>
</tr>
<tr>
<td>6</td>
<td>LAUNCHER/GSE RELIABILITY</td>
<td>18</td>
</tr>
<tr>
<td>7</td>
<td>FIRE CONTROL IMPACT ERROR</td>
<td>19</td>
</tr>
<tr>
<td>8</td>
<td>GUIDANCE IMPACT ERROR</td>
<td>20</td>
</tr>
<tr>
<td>9</td>
<td>MOTOR QUAL TESTS</td>
<td>21</td>
</tr>
<tr>
<td>10</td>
<td>AIRFRAME QUAL TESTS</td>
<td>22</td>
</tr>
<tr>
<td>11</td>
<td>LAUNCHER/GSE QUAL TESTS</td>
<td>23</td>
</tr>
<tr>
<td>12</td>
<td>FIRE CONTROL QUAL TESTS</td>
<td></td>
</tr>
</tbody>
</table>

*****0.* NONE*****
permitted. If the second case occurs, the user will be requested for input. This occurs when PROJMNG doesn't understand the limits requested.

2. Sensitivity Analysis Table

The Sensitivity Table, Table XVI provides a simple column display of the achieved values computed at each step of the analysis as a function of the related parameters. The steps are labeled in the lefthand column. The second column lists the value of the selected parameter for the step. Some of the parameters will result in duplicate steps. PROJMNG divides the range of the parameter values by 10 which can result in a fraction. In creating a step for the table, the parameter value can be successively rounded to the same value. PROJMNG is telling the monitor, it has recognized the parameter as requiring an integer value. It has provided the closest integer to the value. The duplication is the occurrence of the closest integer as the same number on successive occasions. The first occasion PROJMNG rounded up to the value and on the second occasion it rounds down.

The remaining columns, except ZI, are self explanatory. They correlate with the achieved values data given in the table of Development Contract Summary, Table XIV. ZI is explained later in this section.

Upon completion of the table, PROJMNG displays the cue of "IHC001A PAUSE ;PRESS <<ENTER>> TO CONTINUE". This holds the display for review by the monitor and reminds the monitor of the action to perform after reviewing the table.

3. Menu Expansions

Pressing <<ENTER>> causes the program to continue. The sequential routing of PROJMNG returns the monitor's MAIN
MENU display in the main program. Figure 4.11 demonstrates the flexible menu with the new options. These options are discussed as we proceed further into PROJMNG.

**Figure 4.11 Monitor's Main Menu in Mode 1.**

Number 7 on the menu returns to the last display of the Sensitivity Table viewed. PROJMNG's subroutine SNSITIV is reentered through a call (entry) at the end of the subroutine, ENTRY SNSPET (enter SNSITIV and print the table). SNSITIV's array of stored values enables PROJMNG to reprint the table without any calculations.

4. **Sensitivity Analysis Plotting**

The second new feature available through the main menu is the capability to graph the table results. The stored values in the array are passed to the PLTSCH subroutine in a data buffer, COMMON ELOTVA. The graph is 20 lines high and fills the entire screen. Consequently a pause statement at the bottom of the graph could not appear on the same screen with the display.
### TABLE XVI

**Sensitivity Table**

<table>
<thead>
<tr>
<th>#</th>
<th>SENSITIVITY</th>
<th>DEVELOPMENT</th>
<th>FLIGHT TEST</th>
<th>RELIABILITY</th>
<th>ACCURACY</th>
<th>FEE</th>
<th>CONTRACT</th>
<th>E1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FLIGHT TESTS</td>
<td>% ACHIEVED</td>
<td>% ACHIEVED</td>
<td>% ACHIEVED</td>
<td>% ACHIEVED</td>
<td>% ACHIEVED</td>
<td>% ACHIEVED</td>
<td>% ACHIEVED</td>
</tr>
<tr>
<td>0</td>
<td>10.00</td>
<td>4.00</td>
<td>41.22</td>
<td>3.50</td>
<td>197.00</td>
<td>0.00</td>
<td>1.36</td>
<td>153.00</td>
</tr>
<tr>
<td>1</td>
<td>11.00</td>
<td>4.00</td>
<td>41.22</td>
<td>3.50</td>
<td>197.00</td>
<td>0.00</td>
<td>1.62</td>
<td>151.00</td>
</tr>
<tr>
<td>2</td>
<td>12.00</td>
<td>4.00</td>
<td>41.22</td>
<td>3.50</td>
<td>197.00</td>
<td>0.00</td>
<td>2.29</td>
<td>146.99</td>
</tr>
<tr>
<td>3</td>
<td>13.00</td>
<td>4.00</td>
<td>41.22</td>
<td>3.50</td>
<td>197.00</td>
<td>1.29</td>
<td>4.65</td>
<td>146.89</td>
</tr>
<tr>
<td>4</td>
<td>14.00</td>
<td>3.58</td>
<td>47.62</td>
<td>3.50</td>
<td>199.00</td>
<td>2.37</td>
<td>3.65</td>
<td>144.78</td>
</tr>
<tr>
<td>5</td>
<td>15.00</td>
<td>2.97</td>
<td>48.54</td>
<td>3.50</td>
<td>199.00</td>
<td>2.96</td>
<td>3.36</td>
<td>143.67</td>
</tr>
<tr>
<td>6</td>
<td>16.00</td>
<td>1.75</td>
<td>50.38</td>
<td>3.50</td>
<td>200.00</td>
<td>3.50</td>
<td>3.85</td>
<td>140.12</td>
</tr>
<tr>
<td>7</td>
<td>17.00</td>
<td>1.15</td>
<td>61.28</td>
<td>3.40</td>
<td>203.00</td>
<td>3.50</td>
<td>82.61</td>
<td>139.84</td>
</tr>
<tr>
<td>8</td>
<td>18.00</td>
<td>0.17</td>
<td>62.74</td>
<td>2.63</td>
<td>211.00</td>
<td>3.50</td>
<td>84.08</td>
<td>137.84</td>
</tr>
<tr>
<td>9</td>
<td>19.00</td>
<td>0.00</td>
<td>63.96</td>
<td>2.04</td>
<td>217.00</td>
<td>3.50</td>
<td>84.90</td>
<td>137.84</td>
</tr>
<tr>
<td>10</td>
<td>20.00</td>
<td>0.00</td>
<td>55.47</td>
<td>0.88</td>
<td>229.00</td>
<td>3.50</td>
<td>85.31</td>
<td>136.84</td>
</tr>
</tbody>
</table>

IHCO01A PAUSE; PRESS <<ENTER>> TO CONTINUE.

IHCO01A PAUSE; PRESS <<ENTER>> TO CONTINUE.
In order to alleviate this difficulty the pause has been eliminated. The display appears on one screen with the addition of a blank line which forces the IBM to hold the display under the condition of a screen overflow, as indicated by the "MORE" in the screen lower right corner. The screen can be held only temporarily in this IBM display mode. If the operator desires to extend the hold for closer analysis of the graph, pressing <<ENTER>> will change the IBM display mode to "HOLDING" which can remain indefinitely. Once the operator has finished with the "MORE" display mode, or the "HOLDING" mode he can continue by clearing the screen. On the IBM 3278 this is accomplished by pressing <<ALT>> and <<CLEAR>> simultaneously.

The definition key for characters is displayed on the graph, as given in Figure 4.12. A series of print statements has been placed in FLTSCH to list the key in the righthand graph margin.

```
* = DEVELOPMENT COST
T = FLIGHT TEST COMPLETION DATE
R = RELIABILITY
A = ACCURACY
F = TOTAL FEE PERCENTAGE
C = TOTAL COST
E = EFFECTIVENESS INDEX
S = Several curves intersect
```

Figure 4.12 Plot Character Key Definitions.

The table displayed by "8. PLOT LAST SENSITIVITY ANALYSIS ACHIEVED VALUES." divides the screen height of twenty lines into the largest value which is to be displayed. This functions as the ordinate scale for the graph, Figure 4.13. FLTSCH displays the entire set of data. In the next selection, "9. PLOT LAST SENSITIVITY ANALYSIS"
Figure 4.13 Sensitivity Table.
Figure 4.14  Sensitivity Analysis Percentages Graph.
FEE PERCENTAGES." the vertical axis is provided with a scale of 0.5 per line. The height of the ordinate becomes 10, which will display the fees up to 10%. The benefit of this charge in scales is in the presentation of a magnified view in the y-axis between the range of 0 and 10 where the fee percentages are concentrated. Figure 4.14 demonstrates the percentage graph.

Graphing can also be scaled by the monitor. Choosing Sensitivity Menu option number 10 directs PROJMNNG to ask for the ranges which are to be used for the y-axis. Figure 4.15 demonstrates the interactive exchange between the program and user to create the desired y-scale. The program accepts any values for the minimum and maximum values for the axis. The program again divides this range into 20 steps. The step size becomes the y-axis scale. If the scale is less than 0.05 per line, PROJMNNG resets the scale to 0.05. Figure 4.16 demonstrates the graph resulting from the steps in figure 4.15.

```
MAIN MENU:
1. CHANGE TEAM NUMBER
2. THE EP SELECTION QUERY
3. THE INPUT SELECTION MENU
4. REFUN THE DATA CALCULATION
5. DO A SENSITIVITY ANALYSIS
6. PRINTCUT COST*FACTORS
7. PRINT LAST SENSITIVITY TABLE.
8. PRINT LAST SENSITIVITY ANALYSIS ACHIEVED VALUES.
9. PLOT LAST SENSITIVITY ANALYSIS FEE PERCENTAGES.
10. PLOT WITH SELECTED Y-AXIS SCALES.
11. USE OPTIMUM EI VALUE FROM LAST SENSITIVITY ANALYSIS FOR EP-3 DATA.
12. EXIT
```

Figure 4.15 Monitor's Main Menu in Mode 1.
<==NOTE This is the selection from the MAIN MENU to view this graph.
LOWEST VALUE FOR THE Y-AXIS

HIGHEST VALUE FOR THE Y-AXIS

<table>
<thead>
<tr>
<th>Value</th>
<th>Symbol</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td></td>
<td>Y-AXIS = PERCENT</td>
</tr>
<tr>
<td>1.40</td>
<td></td>
<td>x-AXIS = TESTS</td>
</tr>
<tr>
<td>1.80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* = DEV COST
T = TESTCOMP
R = RELIABIL
W = ACCURACY
% = TOTIFEE
* = EFFINDEX
E = MULTIPTS

Figure 4.16 Selected Vertical Axis.
5. **Effectiveness Index**

Selection 11 on the Sensitivity Menu provides the monitor with a means of directly copying into the data array (Selection Menu) the best sensitivity parameter value for the highest EI value. SNSITIV determines the best occurrence of the highest value of Effectiveness Index. The EI with the lowest parameter value is used for cost analysis, accuracy analysis, and completion date analysis. For parameters involving 'reliability', the EI of the highest parameter value is stored in EICPT. It will not make a change of the optimum selection stored in those cases where the minimum value is zero. It stores the sensitivity parameter's step number in variable EICPT. When option 11 is selected, the sensitivity parameter being evaluated has its value changed to that of the evaluation step which had been saved in EIOPT. Figure 4.17 demonstrates the revised Selection Menu.

6. **Flexible Menu Reset**

Should the monitor change the team, the DP level or the data in the database, the calculations made by SNSITIV are no longer applicable. PROJMNG flags the use of the subroutines which generate these changes. Through the running of these subroutines, the flag which expands the Sensitivity Menu is reset to 'off'. The option to replot graphs or tables based on the previous set of data is invalidated.

Selection of the mode 2 option to "FILE TEAM PROPOSAL" generates the monitor mode flag MR. For mode 2, MR = 2. This flag causes the menu statement to file the proposal, and flags the program when leaving mode 2 either at exit or on return to the Monitor's Mode Menu to ask if the rewritten file should be sent to the PROJMNG disk.
J. SUMMARY

The PROJMNNG FORTRAN program retains its basic computations and organization in common with the original CNP program. These basic features have been enhanced by improved user/program interaction, and by the reformulation of the achieved value fee percentage computation algorithm.

PROJMNNG FORTRAN has an additional benefit over CNP in the extended sensitivity analysis capability it provides to the monitor. Both the sensitivity tables and graphs enhance the monitor's ability to evaluate the team's progress when using PROJMNNG.
<table>
<thead>
<tr>
<th>No.</th>
<th>Item</th>
<th>Current Value @ DP-3</th>
<th>No.</th>
<th>Item</th>
<th>Current Value @ DP-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Contractor</td>
<td>1</td>
<td>16</td>
<td>Guidance Qual Tests</td>
<td>6.00</td>
</tr>
<tr>
<td>2</td>
<td>Guidance Approach</td>
<td>1</td>
<td>17</td>
<td>Flight Tests</td>
<td>17.00</td>
</tr>
<tr>
<td>3</td>
<td>Guidance Configuration</td>
<td>2</td>
<td>18</td>
<td>Maximum Cost M$</td>
<td>53.00</td>
</tr>
<tr>
<td>4</td>
<td>Maintainability ENG M$</td>
<td>1.75</td>
<td>19</td>
<td>Minimum CCSI M$</td>
<td>47.00</td>
</tr>
<tr>
<td>5</td>
<td>Value ENG M$</td>
<td>0.75</td>
<td>20</td>
<td>Max Cost Incentive %</td>
<td>4.00</td>
</tr>
<tr>
<td>6</td>
<td>Parallel Develop M$</td>
<td>0.0</td>
<td>21</td>
<td>Latest Week</td>
<td>238.00</td>
</tr>
<tr>
<td>7</td>
<td>Motor Reliability %</td>
<td>92.00</td>
<td>22</td>
<td>Earliest Week</td>
<td>202.00</td>
</tr>
<tr>
<td>8</td>
<td>Airframe Reliability %</td>
<td>97.00</td>
<td>23</td>
<td>Max Deliver Incentive %</td>
<td>3.50</td>
</tr>
<tr>
<td>9</td>
<td>Launcher Reliability %</td>
<td>98.50</td>
<td>24</td>
<td>Max Reliability %</td>
<td>81.00</td>
</tr>
<tr>
<td>10</td>
<td>Fire Control Error YDS</td>
<td>80.00</td>
<td>25</td>
<td>Minimum Reliability %</td>
<td>75.00</td>
</tr>
<tr>
<td>11</td>
<td>Guidance Error YDS</td>
<td>70.00</td>
<td>26</td>
<td>Max Reliability Incentive %</td>
<td>3.50</td>
</tr>
<tr>
<td>12</td>
<td>Motor Qual Tests</td>
<td>25.00</td>
<td>27</td>
<td>Maximum Error YDS</td>
<td>160.00</td>
</tr>
<tr>
<td>13</td>
<td>Airframe Qual Tests</td>
<td>6.00</td>
<td>28</td>
<td>Minimum Error YDS</td>
<td>154.00</td>
</tr>
<tr>
<td>14</td>
<td>Launcher Qual Tests</td>
<td>3.00</td>
<td>29</td>
<td>Max Error Incentive %</td>
<td>4.00</td>
</tr>
<tr>
<td>15</td>
<td>Fire Control Qual Tests</td>
<td>3.00</td>
<td>30</td>
<td>Week for Lot 10</td>
<td>337.00</td>
</tr>
</tbody>
</table>

Troubleshooting:

- None

Figure 4.17 Selection Menu with Optimized Flight Tests.
V. SUMMARY AND CONCLUSIONS

A. SUMMARY

The DNS exercise provides a time proven means of educating project managers in the potentials, sensitivities and incentives of acquisition management. This same capability is the significance of CNP and of PROJMNNG as training aids in the DNS exercise. The support of project management training by computer programs has existed for the past twenty years. This thesis has upgraded the available computer program support for DNS. PROJMNNG provides DNS with a user-friendly interactive tool to assist in the acquisition management training process.

B. DNS

DNS has proven to be an effective training exercise. It provides personnel in the Department of Defense, and personnel in other system acquisition environments, with relevant management training.

1. Benefit

The benefit of DNS as a training exercise is in the requirement it places on students to understand acquisition management concepts. It provides a sequenced set of events to develop, produce and deploy a missile. The Decision Point's serve as planning milestones that the student teams must prepare for in order to successfully manage their acquisition project. Their understanding of the concepts of acquisition management is challenged by the exercise in a pseudo-real time scenario.
2. **Computer Support**

During the twenty years in which DMS has been available, there has been a revolution in the tools available to support project management. The ability of computers to rapidly organize, process and correlate data has changed the acquisition management environment.

3. **CNP**

DMS has adapted somewhat to management evolution by its incorporation and development of CNP.

CNP is an interactive Fortran program which has been used for the past decade. In its current configuration, it is difficult to use. Since it is not user friendly.

C. **PROJMN**

In order for DMS to remain a relevant management training exercise, it must be kept current in its state-of-the-art management concepts, training facility, and computer support. The Fortran program PROJMN has been created to provide DMS a program which is more user-friendly. It is not just an off-shoot of CNP. It incorporates portions of CNP by virtue of utilizing most of the subroutines and algorithms of CNP. PROJMN improves the Fortran support for DMS in its refinement of CNP and by expanding the programmed support for analysis.

1. **Benefits**

PROJMN's benefits include:

- program access
- program exit
- session record
- data file of input parameters
- menus for operation selection
• expanded sensitivity analysis
• tables and graphs to study parameter sensitivity
• user-friendly displays
• monitor-team communication interaction through datafile exchange

2. **Shortcomings**

**PROJMG** has not been completely validated. Its achieved values have not been tested sufficiently to determine its ability to react in a pseudo-realistic manner to parameter variations.

Additionally, the programming facilities utilized in **PROJMG** are not as modern as they could be. The **PRCJMG** capability has been hampered by the limitations of the IBM-3033. Areas in which these limitations have impacted **PROJMG** are discussed in Chapter 6, **Recommendations**.

3. **First Usage**

**PRCJMG** has undergone a field test. It was used to support the Winter 1984 course in **Project Management**, AS-3501, taught at the Naval Postgraduate School by Lcdr. J. Ferris. This field test resulted in the following program enhancements:

• Ability to mass purge data files
• Ability of the monitor to send files to the **PROJMG** disk
• The SMSG autolog feature operates only when no users are accessed to the **PROJMG** disk
• All displays are formatted for 80 character wide personal computer (PC) display.
• All exec query responses are tested for input errors.
• **PROJMG** EXEC tests files for format before replacing the team disk copy. The file used
for the team database is sequentially the first one which exhibits the correct format from either the PROJMN disk or the team, and defaults to the file "FILE DATAPFILE".

During this field test, the major problem in facilitating the program occurred in the autolog procedure. The SMSG automatic log-on of the PROJMN disk occurred regardless of users being on the disk. This situation resulted in files being read into the disk without proper organization. The files subjected to this garbling were unusable.

SMSG is now programmed to write the team files onto the PECJMNG disk only when the disk is not accessed by any user.

4. Transportability

The wide usage of DIS in the U. S. and other countries provided interest in making PROJMN capable of use on various computer systems. CMS unique features and IBM extensions to fortran have been discussed in this thesis to enable adaptation. In its present form PROJMN relies heavily on CMS and its executive language facilities to manipulate the data files outside of the Fortran program environment. These facilities require assessment on a case by case basis in order for PROJMN to be usable on any system other than IBM's CMS with the FRGCMS extension.

a. FORTRAN

The program has been compiled both on the FORTRAN H and FORTRAN GI compilers of the MPS IBM-3033. PROJMN is unable to compile in WATFIV (WATERLOO FORTRAN). The complication which precludes the WATFIV compilation is the call to FRGCMS. It is desirable to not remove the FRGCMS calls. They provide a major enhancement to the program by
enabling the program to clear the screen, 'CLRSCRN'. Loss of this feature would disorganize the PROJMNG displays.

E. DOCUMENTATION

1. CNP

A major problem in working with a program is its documentation. CNP has several styles of documentation. There are periodic lines of text to label or explain the programs operation. These comments were beneficial in reconstructing some of the program operation, but inadequate in deciphering the program operation.

The best documentation present in CNP were the names selected for variables. Each of the input variables, a majority of the computational variables and the routine names were carefully selected to provide a relation with their data definition or with the process they represent. The variables in most of CNP are acronyms.

2. PROJMNG

PROJMNG contains additional comments throughout the program code taken from CNP. Frequent comments have been added to explain the processes accomplished by groups of code and routines. All of the terms added to the program by PROJMNG are meaningful acronyms for the routines and variables.

3. Glossary

Appendix H provides a glossary of the terms used in both CNP and PROJMNG. Compilation of the glossary provides a documentation method which supplements the benefit obtained from selecting variables which are acronyms for the content definition. Some definitions for CNP terms in the glossary remain undetermined.
4. Debugging

Terms used in a program without adequate documentation may interact in unexpected areas of the program. With 3063 variables in the program, CNP is a maze of connections between variables. Tracking the details of terms used in CNP was difficult. The two worst aspects of terms in CNP are multiple definitions for a term and manipulation of the variables to change the value's power-of-ten multiple. For example, RATIO is used in CNP to determine the percentage of a fee to be awarded, it is also used as a value of millions of dollars paid for fees, and it is used in a resource ratio calculation. An example of the manipulation of variables is the array CTC(8). This array is used in millions of dollars and in dollars.

The disadvantage in using mixed definitions for a variable is the impact which occurs on calculations which depend on the variable. If a calculation is based on a specific multiple of the variable and it has varying values dependent on which subroutine used it last, the programmer cannot utilize the variable. He must consider the inability to predict the occurrence of changes to the variables definition.
VI. RECOMMENDATIONS FOR FUTURE CONSIDERATION

The planning, design and execution of this revision to CNP have generated the following recommendations.

- Increase the program support to the exercise
- Expand the program to support more Decision Points in the DMS exercise
- Develop a DMS computer assisted tutorial
- Validate the achieved values
- Change the data tables to formulas
- Increase the sensitivity analysis capabilities
- Improve the graphics resolution
- Change the menu displays to 'plate' displays
- Develop a program to create project management exercise scenarios for actual system acquisitions
- Use PROCJNG to encompass other CNP program variations

A. INCREASED EXERCISE SUPPORT

1. Discussion

The wide availability of personal computers provides project managers and students with improved data processing capability. They now have computation capability available to them instantaneously throughout the acquisition process for storing and analyzing their own specific acquisition data. If DMS is to continue to provide project managers with training at the level now available in the management environment, it must have software facility equivalent to at least that of their personal systems. It is feasible to make the DMS program compatible with PCs. DMS must be supported
PROJMNG FORTRAN: AN INTERACTIVE COMPUTER PROGRAM FOR USE WITH THE DEFENSE MANAGEMENT SIMULATION EXERCISE (U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA G W SCHULTZ UNCLASSIFIED MAR 84 F/G 9/2
at the state-of-the-art. It must be contemporary in providing a program which has the degree of computer facility which the managers experience in day-to-day use and in actual acquisition planning. As this capability develops, the program supporting DNS should expand to:

- Support all DNS Decision Point levels
- Interface with student and manager PCs
- Support actual acquisition scenarios

2. **Recommendation**

Design new program capabilities to permit DNS adaptation for broader application to PC's and to change the acquisition management process.

B. **INCLUDE ALL DECISION POINTS**

1. **Discussion**

Both CNP and PROJMAN support the DNS exercise only for DF-3 and DP-4. The present DNS scenario includes six Decision Points. All of the DP's require team analysis of the exercise parameters and submission of Decision Sheets (see Appendix B). These characteristics are conducive to incorporation of these DP's into other PROJMAN like analyses.

Use of an interactive computer program to support DP's 1 and 2 would provide an opportunity for each team to do a better analysis in these early Decision Points, and to initiate the team database for later use. Inclusion of all DP's in the program would permit documentation of the entire team exercise performance in one computer record.
2. **Recommendation**

A future computer support program for DMS should be designed to expand exercise support from the PROJMN and CNP capability of DP 3 and 4 evaluation to all exercise DP's.

C. **DEVELOP A DMS TUTORIAL**

1. **Discussion**

Computer tutorials have become a popular method of training. The availability of computer assets for CNP or PROJMN indicates the presence of computer hardware to support a tutorial program. Student interaction with a computer supported DMS tutorial could be used to introduce the student to the DMS exercise, to perform sample problems, and to give examples.

The one-on-one capability for student response to a DMS tutorial should be limited to exercise orientation and familiarization. DF's 2 through 6 provide an opportunity for a team-work experience. This team interaction should remain an integral part of the DMS exercise.

2. **Recommendation**

Incorporate the DMS training scenario into an acquisition management tutorial. The tutorial is seen as providing:

- scenario presentation to the individual
- review of initial positions
- aid to students to remedy difficulty with concepts
D. VALIDATE PROJMNG

1. Discussion

Correlation of achieved values between the CNP/PROJMNG programs and the DNS data values has not been demonstrated. An analysis should be conducted in order to determine if the algorithms in PROJMNG provide realistic approximations to the full DNS batch program.

Evidence exists of differences between some CNP and PROJMNG results. Either program may be at fault. PROJMNG's errors result from its acceptance of the algorithms and calculations contained in the original CNP program. The values computed by CNP cannot be assumed to be accurate.

Current classroom usage of PROJMNG has been predicated on the assumption that these values indeed perform as should be expected. This assumption was based on the incorporation of the Fortran code for the achieved values calculation from CNP into PROJMNG.

2. Recommendations

A series of evaluations should be conducted to compare the actual DNS batch outputs and with the performance of PROJMNG.

E. REPLACE THE DATA TABLES WITH EQUATIONS

1. Discussion

The tables of values included in CNP/PROJMNG have limited range in which performance data can be obtained. Beyond these limits data values are not available for calculations. In most cases, CNP does not warn the user that he has performed a calculation which is not a feasible solution. In most cases, PROJMNG forces the team to resubmit data items which are outside the range capabilities.
discussed in the DMS instructions. There are parameters for which limits are not specified in the DMS handouts, and which can result in unusual calculations such as achieving a flight test completion date of 0.

There are two reasons for redesigning the program resource value algorithms which are used to calculate achieved values. First is to provide a continuous function instead of a step function. The table values read for calculations in both CNP and PROJHNG do not provide continuous value functions. Second the values have a limited range. Student selections which are beyond the capability of the table result in erroneous achieved values, such as the flight test example above.

The results provided by CNP and PROJHNG, as discussed in Chapter 1, are not actual results. They represent approximate results to reflect the real world issue of contract risk.

The programming of equations to obtain the resource values will result in the need to validate the achieved values predictions. Validating the values achieved as a result of changing the resource allocation algorithms would be of equal complexity to validating PROJHNG's current results. Both validation projects would verify the program's capability to produce realistic predicted contract performance values. In the interest of reducing the expenditure of effort between these two validations, the effort would be best applied in one project. To validate PROJHNG and later require revalidation for a new algorithm would obviate the former.

2. **Recommendation**

   First, redesign and implement the equations for determining resource allocations. Second, validate the entire new program package of PROJHNG for DMS support.
F. INCREASE SENSITIVITY ANALYSIS OPTIONS

1. Discussion

The options available in the sensitivity analysis menu do not have as dramatic an effect in demonstrating the impact of incentives and other parameters as had been originally expected. Sensitivity curves showing definite peaks or valleys in the achieved effectiveness index are not demonstrated as readily as had been anticipated. The graphic presentation and sensitivity selection options should be designed to provide the user with a clear depiction of the effect of incentive variation. The user should be able to readily determine from the sensitivity analysis products the benefits or disadvantages of changing his decision above or below his current position.

2. Recommendations

Determine other analysis options which may have greater value to the monitors. Conduct a series of analyses to determine if they affect the sensitivity, and to eliminate those options which do not cause an achieved value fluctuation.

G. IMPROVED GRAPHICS

1. Discussion

One area of computer, hardware and software development which is rapidly changing is graphics. The graph subroutine in PROJHNG was specifically designed for the program and has a number of limitations. No other IBM-3033 graphing program was found which provides multiple curve graphing, selectable scales and the number of variations which PROJHNG provides. A (continuous) curve plot is needed to enhance demonstration of parameter variations. The
graphic detail which is needed to improve the display does not appear to be within the capability of the IBM-3278 terminals which are currently a widely available computer peripheral for student use, and which are particularly available in large quantities at NPS. What is needed is the ability to control screen display for continuous line plots. This type of resolution for the graphics on the IBM-3033 is not within the IBM-3278 terminals capability. Terminals do exist which provide greater screen control and which can be programmed to plot continuous lines for the IBM.

Display detail is undergoing evolution. The advantages of color for highlighting presentations adds a perceptible benefit in human ability to comprehend. Color can be used to add a perception of relative weight of significance between curves and between points on a curve. The use of characters and figure presentations which exceed the standard typewriter characters in ASCII presentations increase the pictorial communication of data. Both of these factors -color graphics and detailed display pixel control- are receiving widespread attention in interactive computer systems and should be considered as enhancements for computer support programs which will be successors to CNP and PROJHNG.

A benefit to plotting continuous curves is the ability to separate points of curve intersection. Another benefit is the ability of continuous curves to show the connection between points in the same curve. The PROJHNG plots readily demonstrate the difficulty in presenting the different curves whose plots have a coinciding point.

In the short term, use of Fortran capable displays on the Tektronix 618 or the Ramtek does not appear reasonable. Several factors degrade the significance of implementing operation on a more complex graphic system. One is the added processing time and cost for a display. The
second reason to not consider graphing the program on a graphics display system is the rapid rate of technical development currently occurring. Based on the presence of new IBM color graphics terminals at NPS, the facility at the W. R. Church Computer Center may be expected to provide high resolution color graphics for users in the not to distant future. In this case the effort to recode the PROJANG program for high resolution graphics on the Tektronix or the Bantek graphics terminals would be obviated both in comparative cost and in utility.

Another factor which should be considered before programming DNS support on a graphics unique system, such as DISSPIA, is whether the system will allow DNS to remain transportable. CHP and PROJANG were designed to be cost-effective by their potentially wide distribution and usage. The common availability of the systems on which these two programs are based permits them to be widely copied. Use of unique display systems would reduce the transportability and preclude spreading the cost over a wider user population.

2. **Recommendations**

Future computer programs to support DNS should demonstrate multi-color plotting capability and continuous line graphing.

B. **PLATE DISPLAYS**

1. **Discussion**

The Selection Menu display sets a pattern of presentation of data throughout the PROJANG program. Menu heading and columns of data labels remain consistent in the monitor modes of operation, in the student mode of operation and in the display of data for the report submission tables. With the capability to produce plate displays, these factors
could be set into a file to be overlaid onto the screen. Items in the table are excellent candidates for a plate information display. The plate display capability would enable data item value change on the screen without reprinting the display and without the need to indicate which item is to be changed. The user will be able to place the terminal cursor on the data item value, change the values as desired, and submit the entire change as one input.

Not only would the plate display simplify data change and operation, it would also eliminate the need for four of the subroutines used in PROJMG. The two change routines, the PAGEIN subroutine and the FINISH subroutines could be combined into one menu with the Selection Menu format.

Plate display capability would also be useful for the Sensitivity Analysis Menu. The format of this display could be changed into the same format as the Selection Menu. Much of the sensitivity analysis routine could be eliminated with the saving in lines-of-code which the plate display represents.

The IBM CMS II version currently being implemented at the Naval Postgraduate School (NPS) contains this capability. Plate display capability for this program is expected to be available within the coming academic quarter.

2. **Recommendation**

Adapt PROJMG to display plates as soon as CMS II is implemented.
I. A SCENARIO GENERATING PROGRAM

1. Discussion

Within the computer simulation environment, conceptual level programs have been demonstrated which permit the user to create his own decision support and analysis scenario. Application of this technology to DMS is envisioned as providing a program which can simulate any actual acquisition scenario. These concept level programs utilize a hierarchy of menus to query the user for arrays to establish the program dimensions. The program then queries the user for array size, array name, variables names, parameter ranges for the variables and relative significance (weight) of the variables within the parameter. A hierarchy of menu-driven routines guides the user to create a program which supports the scenario to be studied. These programs enable the user to vary input variables in a matrix of parameters. The user constructs a scenario based on his concept of the parameters, on his feeling of their interaction and their relative significance. Based on these user perceptions such a program creates predictions for the effects of the parameter interaction. These predictions are heavily weighted by user bias to parameter interaction. Usually the program results predict what the user envisioned because of the impact of his biases. The main benefit of scenario creation has proven to be the opportunity for the user to conduct a detailed management study of the situation being modelled.

2. Recommendation

Develop a program whose design would permit project managers to create program simulations tailored to assess their specific real-world scenarios.
J. USE PROJHNG TO ENCOMPASS OTHER CNP VARIATIONS

1. Discussion

The design of PROJHNG around CNP routines with its own routines performing the features which make it different from CNP makes it possible for PROJHNG to encompass other CNP variations. Any CNP style program which has the same common buffer of data could utilize PROJHNG's enhancements. The buffer of common variables in PROJHNG has been edited from that contained in CNP. The CNP buffer can be restored into PROJHNG, and does permit PROJHNG operation.

The design of PROJHNG routines to support the calculations obtained in CNP suggest the potential of PROJHNG as a set of routines to enhance the user-friendliness of other programs. Programs which could generate the same data items as CNP could be used as a nucleus program whose data would be used by PROJHNG routines to conduct sensitivity analysis.

Programs which are candidates for consideration are:

- the main DNS batch game
- the latest ICAF DNS version (SAFE 84)
- the Swedish version called PNM/ZEbra

2. Recommendation

Investigate the use of PROJHNG as is or modified to be compatible with the latest ICAF DNS version (SAFE 84) and the Swedish version called PNM/ZEbra. Test the concept of using PROJHNG as an enhancement to the other management acquisition programs.
<table>
<thead>
<tr>
<th>DP-3</th>
<th>NOTES</th>
</tr>
</thead>
</table>

Log onto the computer

----> LINK 0541P 191 192
DASD 192 FORCED R/O; R/V BY 1681P
R(0102); T=0.01/0.01 17:59:02

----> ACC 92 C
C (192) R/O

----> CNP
EXECUTION BEGINS.
TYPE DP NUMBER 3 OR 4

----> 3
THIS IS DP NUMBER ***** 3 *****
Y OR N

----> Y
TO CONFIRM, TYPE 1 FOR OK, 0 IF
CORRECTION MUST BE MADE.
TEAM NUMBER XXX

----> 003
RECEIVED - TEAM NUMBER: 3
CONFIRM:

----> 1
NEW OR CHANGES TO INPUT DATA
RETYPE WHOLE PAGE CONTAINING ONE
0 (ZERO) FOR NO - 1 (ONE) FOR YES
TYPE IN COMBINATION, 4 0'S AND 1'S FOR
PAGES 1 to 4

----> 1111
TYPE PAGE 1

----> 1120 7500750000
RECEIVED FOR PAGE 1:
1 2
1.75
0.75
0.0
CONFIRM:

----> 1

LINK TO CNP PROGRAM

CALLS CNP

FIRST TIME YOU MUST TYPE ALL 1'S

107
TYPE PAGE 2:

RECEIVED FOR PAGE 2:
92.0
97.0
98.5
90
70
25
6
6
1
7
CONFIRM:

TYPE PAGE 3:

RECEIVED FOR PAGE 3:
53.0
47.0
5.0
238
202
3.0
91
85
3.5
CONFIRM:

TYPE PAGE 4:

RECEIVED FOR PAGE 4:
160
140
3.5
337
CONFIRM:

ERROR MADE
If not corrected it will result in an impossible output
<===TYPE '0' TO REINPUT PAGE 2
<table>
<thead>
<tr>
<th>INCENTIVE AREA</th>
<th>INCENTIVE PROVISIONS</th>
<th>INCENTIVE ACHIEVEMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WORST VALUE</td>
<td>BEST VALUE</td>
</tr>
<tr>
<td>DEV. COST</td>
<td>$ 53.0M</td>
<td>$ 47.0M</td>
</tr>
<tr>
<td>FLT TST COMPL</td>
<td>230 WK</td>
<td>202 WK</td>
</tr>
<tr>
<td>RELIABILITY</td>
<td>85.0%</td>
<td>91.0%</td>
</tr>
<tr>
<td>ACCURACY</td>
<td>160 YDS</td>
<td>140 YDS</td>
</tr>
<tr>
<td>TOTALS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL CONTRACT PRICE = $ 54.7M</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
COST + FACTORS (NO)
DO YOU WISH TO CONTINUE, Y FOR YES, N FOR NO

Y TYPE DP NUMBER 3 OR 4

====> 3
THIS IS DP NUMBER ****** 3 ******

====> Y
TO CONFIRM, TYPE 1 FOR OK, 0 IF
CORRECTION MUST BE MADE.
TEAM NUMBER

====> 003
RECEIVED - TEAM NUMBER: 3
CONFIRM:

====> 1
NEW OR CHANGES TO INPUT DATA
RETYPE WHOLE PAGE CONTAINING ONE
0 (ZERO) FOR NO - 1 (ONE) FOR YES
TYPE IN COMBINATION, 4 0S AND 1S FOR
PAGES 1 TO 4

====> 0110
TYPE PAGE 2
====> 880970985007025633617
RECEIVED FOR PAGE 2:
88.6
87.0
90.4
80.1
76.0
73.6
73.0
72.6
6.3
1.6
CONFIRM:

====> 1
TYPE PAGE 3:
====> 5424800502382020258781040
RECEIVED FOR PAGE 3:
54.2
48.8
4.0
3.8
230
202
2.5
0.7
1.5
1.0
CONFIRM:
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>1</th>
<th>2</th>
<th>1.75</th>
<th>0.75</th>
<th>0.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>202</td>
<td>88.00</td>
<td>97.00</td>
<td>90.50</td>
<td>80.00</td>
<td>70.00</td>
<td>25.00</td>
</tr>
<tr>
<td>10303</td>
<td>54.20</td>
<td>48.80</td>
<td>5.00</td>
<td>238.00</td>
<td>202.00</td>
<td>2.50</td>
</tr>
<tr>
<td>10304</td>
<td>160.00</td>
<td>140.00</td>
<td>3.50</td>
<td>337.00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CHECK CAREFULLY (This is a repeat of the input data).
<table>
<thead>
<tr>
<th>INCENTIVE ACHIEVEMENTS</th>
<th>TEAM 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>INCENTIVE VALUE</td>
<td>$70,200</td>
</tr>
<tr>
<td>MIX-FEE ALLOWED</td>
<td>$99,700</td>
</tr>
<tr>
<td>TOTAL</td>
<td>$169,900</td>
</tr>
</tbody>
</table>

**TOTAL CONTRACT PRICE = $56.2M**
DP-4

Repeat login procedure.

--- LINK 0541P 191 192
DASD 192 FORCED R/O; R/W BY 1681P
--- ACC 92 0 (192) R/O
--- CNP R; T=0.01/0.01 17:59:08
--- 003 EXECUTION BEGINS.
TYPE DP NUMBER 3 OR 4
--- 0 Y THIS IS DP NUMBER 4 ******
--- Y OR N TO CONFIRM, TYPE 1 FOR OK, 0 IF
TEAM NUMBER XXX
--- 003 RECEIVED - TEAM NUMBER: 3
CONFIRM:
--- 1 HAS CORRECT DP-3 DATA BEEN ENTERED
--- Y, N
--- 1120 TYPE PAGE 1 DP-3 INFO
--- 1120 7500750000
--- 1 TYPE PAGE 2 DP-3 INFO
--- 860970985080070025833177
CONFIRM:
--- 1 TYPE PAGE 3 DP-3 INFO
--- 542400002352025878014
CONFIRM:
--- 1 TYPE PAGE 4 DP-3 INFO
--- 160140035337
CONFIRM:

NOTES

LINK TO CNP PROGRAM

CALLS CNP

--- MUST ENTER N EACH TIME YOU LOGON.

FIRST TIME THROUGH YOU MUST
ENTER FINAL DP-3 DATA.

CHECK EACH INPUT CAREFULLY
BEFORE CONFIRMING
<table>
<thead>
<tr>
<th>Table XX</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Page Data Strings</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>1.75</th>
<th>0.75</th>
<th>0.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>PG-1</td>
<td>88.00</td>
<td>97.00</td>
<td>90.50</td>
<td>80.00</td>
<td>70.00</td>
</tr>
<tr>
<td>PG-2</td>
<td>17.00</td>
<td>16.00</td>
<td>14.00</td>
<td>12.00</td>
<td>10.00</td>
</tr>
<tr>
<td>PG-3</td>
<td>54.20</td>
<td>48.80</td>
<td>5.00</td>
<td>28.00</td>
<td>202.00</td>
</tr>
<tr>
<td>PG-4</td>
<td>166.00</td>
<td>140.00</td>
<td>1.50</td>
<td>337.00</td>
<td>337.00</td>
</tr>
</tbody>
</table>

115
CONFIRM:
NEW OR CHANGES TO INPUT DATA
RETYPE WHOLE PAGE CONTAINING ONE
O (END) FOR NO - 1 (ONE) FOR YES
TYPE IN COMBINATION, 4 05 AND 15 FOR
PAGES 1 to 4

1110
TYPE PAGE 1
1120 RECEIVED FOR PAGE 1:
1.1
2.00
0.75
0.0

CONFIRM:

6809
TYPE PAGE 2:
690 RECEIVED FOR PAGE 2:
86.0
97.5
94.5
80
70
27
9.0
3.0
1.0

CONFIRM:

5328
TYPE PAGE 3:
540 RECEIVED FOR PAGE 3:
27.0
6.0
3.0
189
87
81
4.0

CONFIRM:

*** WARNING ***
*** INCENTIVE LIMITS WERE REVERSED ***
*** DECISION SHEET MUST BE CORRECTED ***

DOUBLE CHECK OF CORRECT DP-3
ALL OUTPUT DEPENDS ON THIS.
FIRST TIME YOU MUST
ENTER ALL '1' S (IN THIS
EXAMPLE, ENTRY WAS '1110'
BECAUSE PAGE-4 IS THE
SAME AS FOR DP-3.
THIS PRODUCES AN ERRONEOUS RESULT.
SEE BELOW.

189) TEAM
225) REVERSED LIMITS AND
PUT IN 189 BEFORE 225.
CNP MAKES THE CORRECTION
AND GIVES A WARNING
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>1</th>
<th>2</th>
<th>2.00</th>
<th>0.75</th>
<th>0.0</th>
<th>13.00</th>
<th>1.50</th>
<th>87.00</th>
<th>81.00</th>
<th>4.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>10301</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2.00</td>
<td>0.75</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10302</td>
<td>66.00</td>
<td>97.00</td>
<td>98.50</td>
<td>80.00</td>
<td>70.00</td>
<td>25.00</td>
<td>6.00</td>
<td>3.00</td>
<td>3.00</td>
<td>6.00</td>
<td></td>
</tr>
<tr>
<td>10303</td>
<td>53.20</td>
<td>47.80</td>
<td>6.00</td>
<td>225.00</td>
<td>189.00</td>
<td>1.50</td>
<td>87.00</td>
<td>81.00</td>
<td>4.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10304</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

NOTE: CMP FILLED IN ALL 0'S FOR PAGE 4.
# TABLE XXII

Development Contract Achieved Value Summary

<table>
<thead>
<tr>
<th>INCENTIVE AREAS</th>
<th>INCENTIVE PROVISIONS</th>
<th>INCENTIVE ACHIEVEMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WORST VALUE</td>
<td>BEST VALUE</td>
</tr>
<tr>
<td>DEV. COST</td>
<td>$53.2M</td>
<td>$47.0M</td>
</tr>
<tr>
<td>FLT. TST. COMPL</td>
<td>225 WK</td>
<td>180 WK</td>
</tr>
<tr>
<td>RELIABILITY</td>
<td>81.0%</td>
<td>87.0%</td>
</tr>
<tr>
<td>ACCURACY</td>
<td>0 YDS</td>
<td>0 YDS</td>
</tr>
<tr>
<td>TOTALS</td>
<td>11.5%</td>
<td></td>
</tr>
</tbody>
</table>

TOTAL MAX FEE ALLOWED DOES NOT EQUAL TOTAL CONTRACT PRICE = $53.1M
NOTE: CNP will give a warning if fee does not equal 15%.
In final batch printout, the program will reallocate percentage among all variables.
'Y' for iteration to get correct data in.

Because DP-3 data is already in

COST + FACTORS (NO)

DO YOU WISH TO CONTINUE, Y FOR YES, N FOR NO

TYPE DP NUMBER 3 OR 4

THIS IS DP NUMBER 4

TO CONFIRM, TYPE 1 FOR OK, 0 IF
CORRECTION MUST BE MADE.
TEAM NUMBER XXX

RECEIVED - TEAM NUMBER: 3
CONFIRM:

HAS CORRECT DP-3 DATA BEEN ENTERED
Y, N

NEW OR INPUT CHANGES TO INPUT DATA
RETYPE WHOLE PAGE CONTAINING ONE
0 (ZERO) FOR NO - 1 (ONE) FOR YES
TYPE IN COMBINATION, 4 OS AND 1S FOR
PAGES 1 TO 4

TYPE PAGE 1
RECEIVED FOR PAGE 1:
1
2
0.00
0.75
0.0

CONFIRM:

TYPE PAGE 2:
RECEIVED FOR PAGE 2:
88.0
97.0
90.5
80
70
27
6
3
2
13

CONFIRM:
### TABLE XXIII
Development Contract Achieved Value Summary

<table>
<thead>
<tr>
<th>INCENTIVE AREA</th>
<th>INCENTIVE PROVISIONS</th>
<th>INCENTIVE ACHIEVEMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WORST VALUE</td>
<td>BEST VALUE</td>
</tr>
<tr>
<td>DEV. COST</td>
<td>$53.2M</td>
<td>$47.8M</td>
</tr>
<tr>
<td>FLT TST COMPL.</td>
<td>225 WK</td>
<td>189 WK</td>
</tr>
<tr>
<td>RELIABILITY</td>
<td>81.0%</td>
<td>87.0%</td>
</tr>
<tr>
<td>ACCURACY</td>
<td>160 YDS</td>
<td>140 YDS</td>
</tr>
<tr>
<td>TOTALS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| TOTAL CONTRACT PRICE = $ 54.3M
COST + FACTORS ( NO)
DO YOU WISH TO CONTINUE, Y FOR YES,
N FOR NO
Y: FOR ITRATION
N: FOR STOP
DEFENSE MANAGEMENT SIMULATION

DECISION SHEET 3

1. Of the total funds approved by the Secretary of Defense, the following distribution is made for procurement of the ZEBRA Weapon System:

   a. Development Phase
      $____.____ Million [1-4]

   b. Production Phase
      $____.____ Million [5-8]

2. The Development Contractor selected for ZEBRA System development is: (enter appropriate contractor number)

   a. Midas Missile Corporation (contractor 1)
   b. Apex Aerospace, Incorporated (contractor 2)

3. The basic guidance approach selected for the ZEBRA System is: (enter appropriate approach number)

   a. Radar Inertial Guidance (G-1) (approach 1)
   b. All-Inertial Guidance (G-2) (approach 2)

4. The guidance configuration selected for development is: (enter appropriate configuration number)

   a. "A" as primary, "B" as parallel (configuration 1)
   b. "A" alone (configuration 2)
   c. "B" as primary, "A" as parallel (configuration 3)
   d. "B" alone (configuration 4)

5. The development contract to be negotiated contains the following features:

   a. Funds for the following activities. These funds are included in the total target development cost:

      (1) Maintainability Engineering $____.____ Million [12-15]
      (2) Value Engineering $____.____ Million [16-19]
      (3) Parallel Guidance Development (enter $ where not desired) $____.____ Million [20-23]
DECISION SHEET 3

b. The following minimum technical performance characteristics: (demonstrated at completion of flight tests)

(1) Motor Sub-system Reliability

(2) Airframe Sub-system Reliability

(3) Launcher/GSE Sub-system Reliability

(4) Fire Control Sub-system Impact Error

(5) Guidance Sub-system Impact Error

(c. The following numbers of sub-system Qualification Tests:

(1) Motor Sub-system (20-40)

(2) Airframe Sub-system (3-9)

(3) Launcher/GSE Sub-system (2-6)

(4) Fire Control Sub-system (2-4)

(5) Guidance Sub-system (3-9)

(d. The following number of Flight Tests: (10-25)

6. Decision Point

Team Number

Card Number

125
7. The incentive provisions for this development contract (subject to negotiations) are as follows:

NOTE: If any area is not incentivized, enter the target value for both the maximum and minimum fee values, enter 0 for the maximum fee percentage.

a. Development Program Cost:

- Maximum Cost (minimum fee value) $_______ Million [1-3]
- Minimum Cost (maximum fee value) $_______ Million [4-6]
- Maximum Fee Percentage ________ % [7-9]

b. Schedule; Completion of Flight Tests:

- Latest Week (minimum fee value) Week No. ________ [10-12]
- Earliest Week (maximum fee value) Week No. ________ [13-15]
- Maximum Fee Percentage ________ % [16-18]

c. System Reliability:

- Maximum Reliability (maximum fee value) ________ % [19-20]
- Minimum Reliability (minimum fee value) ________ % [21-22]
- Maximum Fee Percentage ________ % [23-25]
Team Number ________

DEd: 

SHEET 3

d. System Impact Error:

Maximum Error (poorest accuracy)  (minimum fee value)  ______ yards [26-28]
Minimum Error (best accuracy)  (maximum fee value)  ______ yards [29-31]
Maximum Fee Percentage  ______ % [32-34]

8. Production Decisions:

a. The desired date for completion of deployment of lot TEN is:  Week No. ______ [44-46]

b. Fabrication of Block I missiles is to start after:

(1) Qualification Tests  [concurrent (fabrication 1)]
(2) Flight Tests  [fly-before-buy (fabrication 2)]  (1 or 2) ______ [47]

9. Class:

decision Point
Team Number
Card Number
1. The Development Contractor selected for ZEBRA System development is:
   (enter appropriate contractor number)
   a. Midas Missile Corporation (contractor 1) __ [1]
   b. Apex Aerospace, Incorporated (contractor 2)

2. The basic guidance approach selected for the ZEBRA System is:
   (enter appropriate approach number)
   a. Radar Inertial Guidance (G-1) (approach 1) __ [2]
   b. All-Inertial Guidance (G-2) (approach 2)

3. The final guidance configuration selected for development and production is:
   (enter appropriate configuration number)
   a. "A" configuration (configuration 2) __ [3]
   b. "B" configuration (configuration 4)

4. The revised development contract contains the following features:
   a. Funds for the following activities:
      (If you do not desire to expend additional funds in these areas, enter the appropriate amounts expended at DP-3. If you do desire to expend additional funds in these areas, enter the total amounts you now desire.)
      (1) Maintainability Engineering $______ Million [4-6]
      (2) Value Engineering $______ Million [7-9]
   b. [For Monitor's Use Only] 0 0 0 [10-12]
c. The following minimum technical performance characteristics:
(demonstrated at completion of flight tests)

(1) Motor Sub-system Reliability _______ % [33-35]
(2) Airframe Sub-system Reliability _______ % [36-38]
(3) Launcher/GSE Sub-system Reliability _______ % [39-41]
(4) Fire Control Sub-system Impact Error _______ yards [42-44]
(5) Guidance Sub-system Impact Error _______ yards [45-47]

d. The following numbers of sub-system Qualification Tests:

(1) Motor Sub-system (20-40) ______ [48-49]
(2) Airframe Sub-system (3-9) ______ [50]
(3) Launcher/GSE Sub-system (2-6) ______ [51]
(4) Fire Control Sub-system (2-4) ______ [52]
(5) Guidance Sub-system (3-9) ______ [53]

e. The following number of Flight Tests: (10-25) ______ [54-55]
6. The incentive provisions for this development contract (subject to negotiations) are as follows:

NOTE: If any area is not incentivized, enter the target value for both the maximum and minimum fee values, enter 0 for the maximum fee percentage.

a. Development Program Cost:

Maximum Cost (minimum fee value) $____ Million [1-3]
Minimum Cost (maximum fee value) $____ Million [4-6]
Maximum Fee Percentage [7-9]

b. Schedule; Completion of Flight Tests:

Latest Week (minimum fee value) Week No. [10-12]
Earliest Week (maximum fee value) Week No. [13-15]
Maximum Fee Percentage [16-18]

c. System Reliability:

Maximum Reliability (maximum fee value) % [19-20]
Minimum Reliability (minimum fee value) % [21-22]
Maximum Fee Percentage % [23-25]
d. System Impact Error:

Maximum Error (poorest accuracy) (minimum fee value) _______ yards [26-28]

Minimum Error (best accuracy) (maximum fee value) _______ yards [29-31]

Maximum Fee Percentage _______ % [32-34]

7. Production Decisions:

a. The Development Contractor is to procure the following number of lots of long-lead time components in preparation for Block I production:

[44-45]

b. The earliest Lot TEN deployment date for which fixed price proposals are desired is:

Week No. _______ [46-48]

c. Fabrication of Block I missiles is to start after:

(1) Qualification Tests [concurrent (fabrication 1)]

(2) Flight Tests [fly-before-buy (fabrication 2)]

(1 or 2) _______ [49]

8. Class

Decision Point _______ [73-74]

Team Number _______ [76]

Card Number _______ [77-78]

______ [80]
*THIS EXEC FACILITATES STUDENT LINK TO THE 0276P_DISK FOR EXECUTION OF
*THE PROJECT MANAGER SIMULATION "PROJMG FORTRAN".

&CONTROL OFF
&AGAIN
CP LINK 0276P 191 192 RR
ACCESS 192 C
CP SET PF09 IMM REL 192 (DET) &CP SET PF09
CLRSRN
EXEC RECORD ON
&TYPE * NOTE!: YOU ARE NOW LINKED TO PROJMG ON YOUR 192 DISK, MODE C
&TYPE * NOTE!: PRESS PF09 TO BREAK THE LINK
EXEC PROJMG
REL 192 (DET)
CP SMSC QACHT KLINE
&BEGTYPE

&END
&READ ARGS
&IF &1 EQ . &SKIP
&IF &1 EQ Y &GOTO &AGAIN
EXEC RECORD OFF
&END
"PROJMC EXEC" PROVIDES ENTRY CHECKS FOR THE MAIN PROGRAM "PROJMC"
* AMONG THE CHECKS ARE THOSE HIGHLIGHTED BY COMMENTS BELOW. THEY FACIL-
* ITATE "TEAM" STORED FILE PROCESSING.

********

*CONTROL OFF
SET BLIP *
SET CMSTYPE HT
GLOBAL TXTLIB FORTMOOD2 MOD2EEN IMSLP NONIMSL CMSLIB

FILEDEF 5 TERMINAL (PERM
FILEDEF 6 TERMINAL (PERM
FILEDEF 13 DISK DATAINST DISKNUM A (PERM
FILEDEF 17 DISK FILE FT17F001 C (PERM
FILEDEF 18 DISK FILE FT18F001 C (PERM
FILEDEF 19 DISK FILE FT19F001 C (PERM
FILEDEF 20 DISK FILE FT20F001 C (PERM
FILEDEF 23 DISK FILE FT23F001 C (PERM
*FILE DATAINST--IDENTIFIES THE DISK AS BELONGING TO A MONITOR/INSTRUCTOR
*FILES 17,18,19 & 20 -- ARE DATA FOR ACHIEVED VALUE DETERMINATION IN
* EXECUTING THE PROGRAM'S SIMULATED INTERACTION WITH A
* CONTRACTOR..........................

********

SET CMSTYPE HT
* (THE FOLLOWING COPYFILES ARE PERFORMED TO PERMIT FORTRAN PROGRAM
* EXECUTION TO CHANGE THE TEAM FILES)

USERID
* (USERID VARIABLES ARE USED AS SECURITY CHECKS)
&READ VARS &DUMMY &USERID
&USER = 0
* (CHECK FOR THE PRESENCE OF AN INSTRUCTOR PECULIAR FILE)
STATE DATAINST DISKNUM A
&RET = &REICODE
&IF &RET NE 0 &GOTO -STUDENT
* (CHECK FOR A "P" IN THE FIRST LINE FIFTH POSITION)
FINDSIAK DATAINST DISKNUM A 5 P
&IF &REICODE NE 0 &GOTO -STUDENT
&READ VARS &USER
&IF &USER NE &USERID &GOTO -STUDENT
&CONTINUE
PRESS <ENTER> WHEN YOU ARE READY TO CONTINUE.

&END
&READ
CLRSCRN
&BEGIN

****** EXECUTION IS IN PROGRESS. WAIT. DO NOT PRESS ENTER.******

&END
FILEDEF 11 DISK DATAINST DISKNUM A
   (ESTABLISH THE INSTRUCTOR UNIQUE FLAG OF 21)
&FLAG = TEAM21
STATE DATAFILE TEAM21 A
&IF &RETCODE NE 0 COPYFILE FILE DATAFILE C DATAFILE TEAM21 A (REPLACE
&NM = 0
   #
   &NEXTIN
   &NM = &NM + 1
&CHART = TEAM&NM
&IF &NM LT 10 &CHART = TEAM0&NM
STATE DATAFILE &CHART C
&IF &RETCODE EQ 0 COPYFILE DATAFILE &CHART C = = A (REPLACE
&IF &NM LT 20 &GOTO &NEXTIN
FILEDEF 9 DISK DATAFILE TEAM21
&BEGIN
PRESS <ENTER> WHEN YOU ARE READY TO CONTINUE.

&END
&READ
-PROCES
CLRSCRN

*****************************************************************************

THE FOLLOWING LINES PROVIDE A ROUTINE TO PURGE TEAM DATAFILES FROM THE DISK ON WHICH THE USER HAS 'A' ACCESS. THE DESIGN OF THIS FEATURE IS TO PURGE ALL TEAM FILES FROM THE DISK ONTO WHICH TEAM FILES ARE STORED BY THIS EXEC ROUTINE. THIS OPERATION WILL BE PARTICULARLY BENEFICIAL IN HELPING MONITORS TO INITIALIZE THE DISK FOR A NEW GAME. THIS FUNCTION IS ACCOMPLISHED BY ANSWERING 'Y', AFFIRMATIVE, TO PURGE TEAM DATAFILES WHILE LOGGED ONTO THE PROJEC MOUNTED FILE.

*****************************************************************************
**WARNING**

ANSWERING 'Y' (YES) TO THE FOLLOWING QUERY WILL:

PURGE ALL EXISTING STUDENT DATAFILES

ON THE DISK YOU ARE USING.

DO YOU WISH TO INITIALIZE ALL TEAM FILES FOR A NEW DMS EXERCISE?

'P' = PURGE ALL TEAM DATAFILES

'N' = PROCEED WITH NORMAL OPERATION

'E' = STOP

&END
&DMS MAKEBUF
&READ VARS &N
&DMS DROPBUF
&IF .&N EQ . &GOTO -PROEX
&IF &N EQ PURGE &GOTO -REDUM
&IF &N EQ E &GOTO -END
&GOTO -PRORUN
-PROEX
&PROEX
&DO YOU WISH TO CONTINUE; Y/N ?
&END
&READ VARS &NOGO
&IF &NOGO EQ . &GOTO -INSTEND
&IF &NOGO EQ Y &GOTO -PROCES
&GOTO -INSTEND
-REDUM
-STATE DATAFILE * A
&IF &RETCODE ME 0 &GOTO -END
-ERASE DATAFILE * A
-ERASE DATACODE * A
&IF &RETCODE ME 0 &GOTO -END
&GOTO -REDUM
-STUDENT

(SET THE TEAM NUMBER)

-STATE DATAFILE * A
&RET = &RETCODE
&IF &RET ME 0 &GOTO -NEXNUM

(CHECK FOR A "3" IN THE FIRST LINE FOURTH POSITION)
FINDSTAK DATAFILE * A
&READ VARS &O &TWO &THREE
&IF &ONE EQ 0 &GOTO -NEWNUM
&GOTO -TEAMNUM
-NEWNUM
&BETYPE
&END
CMS MAKEBUF
&READ VARS &ONE
CMS DROPBUF
&IF &ONE EQ . &GOTO -EXITIM
&IF &ONE GE 21 &GOTO -EXITIM
&IF &ONE LE 0 &GOTO -EXITIM
&IF &ONE GT 0 &GOTO -TEAMNUM
-EXITIM
&BETYPE
DO YOU WANT TO CONTINUE; Y/N ?
&END
&READ VARS &NOCO
&IF &NOCO EQ . &GOTO -INSTEND
&IF &NOCO EQ Y &GOTO -NEWNUM
&GOTO -INSTEND
-TEAMNUM
&1 = &ONE
&CHAR = TEAM&1
&IF &1 LT 10 &CHAR = TEAM&1
(ESTABLISH THE STUDENT'S WORKING FILES)
FILEDEF 9 DISK DATAFILE &CHAR * (PERM
&IF &FLAG ME TEAM2 FILEDEF 11 DISK DATACODE &CHAR * (PERM
* (CHECK THE INSTRUCTOR FILES FOR A PREVIOUSLY STORED TEAM DATABASE)
-TEAMFILE
STATE DATAFILE &CHAR C
&IF &RETCODE ME O &GOTO -DISKMM
FINDSTAK DATAFILE &CHAR C
&READ VARS &THREE
&IF &THREE EQ 0 &GOTO -DISKMM
&IF &THREE EQ 0 &GOTO -DISKMM
COPYFILE DATAFILE &CHAR C = = A (REPLACE)
-ADISKMM
STATE DATAFILE &CHAR A
&IF @RETCODE EQ 0 &GOTO -CODE
COPYFILE FILE DATAFILE C DATAFILE &CHAR A (REPLACE)
-CODE
STATE DATAFILE &CHAR C
&IF @RETCODE NE 0 &GOTO -ADISKC
FINDSTK DATAFILE &CHAR C
&READ VARS &THREE
&IF &THREE EQ . &GOTO -ADISKC
&IF &THREE EQ 0 &GOTO -ADISKC
COPYFILE DATAFILE &CHAR C = = A (REPLACE)
-ADISKC
STATE DATAFILE &CHAR A
&IF @RETCODE EQ 0 &GOTO -PROGRUN
COPYFILE FILE DATAFILE C DATAFILE &CHAR A (REPLACE)
SET CNSTYPE RI
-PROGRUN
&BEGTYPE

WAIT FOR "EXECUTION TO BEGIN".

&END
SET BLIP OFF
&STACK AONE
&STACK &FLAG
************************************************************
LOAD PROJNRG (START
**************************************************************************
&IF &READFLAG EQ STACK DESBUF
SET BLIP *
* (THE FORTRAN PROGRAM MANIPULATION OF FILES PREVENTS GENERATION OF NEW
* DATA FILES DURING THE PROGRAM EXECUTION, THE FOLLOWING ROUTINE PERMITS
* THE OPERATOR TO GENERATE AT LEAST ONE NEW DATABASE FILE BY PROGRAM RE-
* DEFINITION OF FILEDEF AS APPLIED TO THE THREE DUMMY FILES GENERATED
* AT THE TOP OF THIS "EXEC".)
* (IF THE OPERATOR IS NOT AN INSTRUCTOR, BYPASS DATAFILE PRINTOUT
* CAPABILITIES)
&IF @USERID NE &USER &GOTO -NOTINS
* (GET THE TEAM NUMBER)
&BEGTYPE
DO YOU WISH TO RECEIVE A LISTING OF TEAMS ON THE CLASS DISK,
Y=YES?
&END
&READ VARS &ANS
&IF &ANS NE &GO TO -INST END
&IF &ANS NE Y &GO TO -INST END

&NM = 0

CP SPOOL PUNCH CONT TO &USER
-NEXT NM
&NM = &NM + 1
&CHART = TEAM &NM
&IF &NM LT 10 &CHART = TEAM 0 &NM
  PUNCH DATAFILE &CHART C
&IF &NM LT 20 &GO TO -NEXT NM
CP SPOOL PUNCH CLOSE NO CONT
CP SPOOL PUNCH OFF
READCARD INST FILE A
&GO TO -INST END
-NOT INS

(DATA DUMP THE TEAM'S DATABASE FILES TO THE INSTRUCTOR)
CP SPOOL PUNCH CL X CONT TO 0276 P
PUNCH DATAFILE &CHAR AO
&PUNCH &USER ID
CP SPOOL PUNCH CLOSE NO CONT
CP SPOOL PUNCH OFF
CP SPOOL PUNCH CL X CONT TO 0276 P
PUNCH DATACODE &CHAR AO
&PUNCH &USER ID
CP SPOOL PUNCH CLOSE NO CONT
CP SPOOL PUNCH CL A
CP SPOOL PUNCH OFF
-INST END
SET CMSTYPE HT
ERASE LOAD MAP A
SET CMSTYPE RT
SET BLIP OFF
-END
<table>
<thead>
<tr>
<th>TRACE</th>
</tr>
</thead>
<tbody>
<tr>
<td>The following warning alerts operators logging onto the disk</td>
</tr>
<tr>
<td>* that they should access by &quot;ACC (NOPROF)&quot; and not permit the</td>
</tr>
<tr>
<td>* system to run the Profile Exec since it has a logoff command</td>
</tr>
<tr>
<td>* on its bottom line. Normal Profile type functions are performed</td>
</tr>
<tr>
<td>* by executing a separate command of &quot;MYPROF&quot; which performs the</td>
</tr>
<tr>
<td>* routines in MYPROF EXEC.</td>
</tr>
</tbody>
</table>

```
&BEGIN

******WARNING****** <<ACC (NOPROF)>>,<<MYPROF>> VICE <<PROFILE>>

&END
CP SPOOL RDR CL X
READ *
READ *
READ *
READ *
READ *
READ *
READ *
READ *
CP SPOOL RDR CL *
CP LOGOFF
```
A. INTRODUCTION

This pamphlet contains general instructions for student operation of the program "PROJMOV FORTRAN" which is designed as a companion to the Industrial College of the Armed Forces' "DEFENSE MANAGEMENT SIMULATION" exercise.

The program functions as a computerized analysis tool assisting a project manager in evaluating his assessment of the goals and incentives for his contract to procure a new missile system. It does not provide answers to the simulation's main batch computer program.

Significant effort has been placed in making the program as easy to operate as possible. The user-friendly design permits the simulation's participants to correct, to rerun, to submit reports to the monitor, and to exit the game with the minimum of difficulty.

Additionally, consideration has been given to protecting the individual team's results. A generic security system guards access to the team's files to prevent unintentional destruction of the database, or database plagiarism. Proper entry of the team security code enables the team to access its stored database and permits submission of a team proposal to the class monitor. Attempts to circumvent these measures are counted and recorded in the database. After five erroneous entries of the security code the team's data
files are not accessible by the students. The monitor must be contacted to reset the counter. In order to document the access and attempts to access a team database, the routine PROJHNG EXEC transmits the database to the monitor's computer disk with the USERID and USERNAME from the disk running PROJHNG EXEC.

In the event you dump (exit) out of the program, other files may appear on your disk such as PROJECT MODULE, LOAD MAP, FILE DATAFILE, and FILE DATACODE. Please erase these extra files. The FILEDEFS in Table XXIV will be generated do not allow files on your disk which are not part of PROJHNG to use these file names.

<table>
<thead>
<tr>
<th>TABLE XXIV</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROJHNG FILEDEFS</td>
</tr>
<tr>
<td>1. FILE 2</td>
</tr>
<tr>
<td>2. FILE 6</td>
</tr>
<tr>
<td>3. FILE 9</td>
</tr>
<tr>
<td>4. FILE 11</td>
</tr>
<tr>
<td>5. FILE 17</td>
</tr>
<tr>
<td>6. FILE 18</td>
</tr>
<tr>
<td>7. FILE 19</td>
</tr>
<tr>
<td>8. FILE 20</td>
</tr>
</tbody>
</table>
during the execution of PROJMG EXEC.

The following sequence of operations demonstrates the procedures involved in the operation of "PROJMG FORTRAN". Several files are involved in the game operation. They are listed in order to enable the participant to manage his student disk. Problems with file definition conflict should be considered. Do not leave non-game files with the same file name on your disk.
flagging of the program for access to the next DP level. Please leave this file on your disk during your use of the game.

5. DATACODE TEAMxx contains your team's security code and number of times it has been incorrectly entered. Once you terminate a run of the simulation, the file is sent to the monitor who uses it at his copy. On the monitor's PROJHNG disk, it provides the primary code file for the team. The monitor's copy is the primary copy. In order to preclude program abort if DATACODE cannot be found on your disk. DO NOT TAMPER WITH THESE FILES or difficulty will occur in the program's correlation of your team's security code.

B. INSTRUCTIONS

PROJHNG FORTRAN is available on the Naval Postgraduate School IBM-3033 computer in the 'W. R. CHURCH COMPUTER CENTER'.

The text which follows is a tutorial demonstrating the features of the PROJHNG package of programs and data files. These features are demonstrated as they would occur on the IBM-3033 using an IBM 3278 terminal. Follow the procedures as they appear in the text. Use your own team number and team security code.
In the following examples, please perform the operations highlighted in the left margin by =>. 

The three column divisions below are arranged with the operator inputs in the left column. Computer responses and the headings for both explanation and for *****EXAMPLES***** are in the center column. The right column contains remarks to aid in understanding the procedures being executed.
ENTER YOUR NEW CODE.

PROJMG determines if you have a code of 8 zeros and requires a code input to be made. Select a code of up to 8 characters and enter it. If you already have a code, the program will ask you to enter it for a comparative validation. The program will now proceed to check the database for previous information. If it finds nothing in the files, it will sequence through each item from the participant's decision point data sheets asking for the inputs. At the end of each page it will check the entries against the input requirements and ask for corrections as necessary.
<table>
<thead>
<tr>
<th>OPERATOR</th>
<th>PROGRAM</th>
</tr>
</thead>
</table>

#### logon

BEGIN RECORDING OF TERMINAL SESSION

* NOTE!* YOU ARE NOW LINKED TO PROJMN ON YOUR 192 DISK

MODE C

* NOTE!* PRESS P109 TO BREAK THE LINK

FILE 'DATAFILE A' NOT FOUND.

WHAT IS YOUR TEAM NUMBER?

#### linkproj

FILE 'DATAFILE TEAM01 C' NOT FOUND.

FILE 'DATAFILE TEAM01 A' NOT FOUND.

FILE 'DATACODE TEAM01 C' NOT FOUND.

FILE 'DATAFILE TEAM01 A' NOT FOUND.

PRESS <<ENTER> WHEN YOU ARE READY TO CONTINUE

EXECUTION BEGINS...

PROJMN EXEC checks to see if you are an instructor (monitor).

If not, and if it finds no TEAMxx data files, it will create the files needed.

You should enter your team's number here, wait for completion of this process.

Determines if a DATAFILE TEAMxx has already been submitted to the monitor.

PROJMN EXEC creates all of its needed files on your A disk.

There may be a pause of several minutes in completing this process.

**DO NOT PANIC.**

---

**TO TERMINATE THE PROGRAM AT ANY POINT, TYPE "E"=EXIT IN RESPONSE TO ANY "YES/NO" QUERY.**

**CAUTION IN RESPONSE TO A QUERY, PRESSING <<ENTER>> WITHOUT PROVIDING DATA WILL DUMP THE PROGRAM**

---

YOU ARE ENCOURAGED TO CHANGE YOUR SECURITY CODE.

---

All security codes have been initialized.
*****INPUT PAGE 1 DP-3 INFO*****

CONTRACTOR NUMBER,(1 OR 2)?:
GUIDANCE APPROACH,(1 OR 2)?:
GUIDANCE CONFIGURATION,(1-4); NOT 1 OR 3 AFTER DP-3?
MAINTENABILITY ENGINEERING FUNDS IN $M:?
VALUE ENGINEERING FUNDS IN $M:?
PARALLEL GUIDANCE FUNDS IN $M:?

DATA CHECKS

Parallel guidance selections 1 or 3 will be rejected after DP-3.

Query for parallel guidance funds.
If guidance configuration selected is 2 or 4, or after DP-3, parallel guidance funds will not be queried. They are zeroed.
Checks of data values will occur at the end of each page. Necessary corrections will be queried.
Sample *****WARNINGS***** of data errors are demonstrated below.
### INPUT PAGE 2 DP-3 INFO.

<table>
<thead>
<tr>
<th>Question</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOTOR RELIABILITY?</td>
<td>&gt;90</td>
</tr>
<tr>
<td>AIRFRAME RELIABILITY?</td>
<td>&gt;97</td>
</tr>
<tr>
<td>LAUNCHER/GSE RELIABILITY?</td>
<td>&gt;98.5</td>
</tr>
<tr>
<td>FIRE CONTROL ACCURACY?</td>
<td>&gt;80</td>
</tr>
<tr>
<td>GUIDANCE ACCURACY?</td>
<td>&gt;70</td>
</tr>
<tr>
<td>MOTOR QUAL TESTS:20-40?</td>
<td>&gt;25</td>
</tr>
<tr>
<td>AIRFRAME QUAL TESTS:3-9?</td>
<td>&gt;6</td>
</tr>
<tr>
<td>LAUNCHER/GSE QUAL TESTS:2-6?</td>
<td>&gt;3</td>
</tr>
<tr>
<td>FIRE CONTROL QUAL TESTS:2-4?</td>
<td>&gt;3</td>
</tr>
<tr>
<td>GUIDANCE SYSTEM QUAL TESTS:3-9?</td>
<td>&gt;1</td>
</tr>
<tr>
<td>FLIGHT TESTS:10-25?</td>
<td>&gt;7</td>
</tr>
</tbody>
</table>

### DATA CHECKS

### CORRECTIONS

**WARNING**

Guidance qualification tests must be greater than 3 and less than 9. Your qual tests= 1. Previous value 1.00, input a value?
*****INPUT PAGE 3 DP-3 INFO.******

= 53
= 47
= 238
= 202
= 3
= 91
= 92
= 3.5

*****WARNING*****
MAXIMUM SYSTEM RELIABILITY IS LESS THAN MINIMUM RELIABILITY: MAX = 91.; MIN = 92. FOR MAXIMUM RELIABILITY, PREVIOUS VALUE 91.00, INPUT a VALUE.

FOR MINIMUM RELIABILITY,

Following are additional examples of correction check warnings.

The program automatically queries pairs of values which are for the same data element.
Le Lai Liu * ZLJ W = oz Ii. - o-
DATABASE PRINTOUT as SELECTION MENU

<table>
<thead>
<tr>
<th>NO.</th>
<th>ITEM</th>
<th>CURRENT VALUE</th>
<th>NO.</th>
<th>ITEM</th>
<th>CURRENT VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CONTRACTOR</td>
<td>1</td>
<td>16</td>
<td>GUIDANCE QUAL TESTS</td>
<td>6.00</td>
</tr>
<tr>
<td>2</td>
<td>GUIDANCE APPROACH</td>
<td>1</td>
<td>17</td>
<td>FLIGHT TESTS</td>
<td>17.00</td>
</tr>
<tr>
<td>3</td>
<td>GUIDANCE CONFIGURATION</td>
<td>1</td>
<td>18</td>
<td>MAXIMUM COST</td>
<td>83.00</td>
</tr>
<tr>
<td>4</td>
<td>MAINTAINABILITY ENG MS</td>
<td>1.75</td>
<td>19</td>
<td>MINIMUM COST</td>
<td>47.00</td>
</tr>
<tr>
<td>5</td>
<td>VALUE ENG</td>
<td>0.75</td>
<td>20</td>
<td>MAX COST INCENTIVE</td>
<td>5.00</td>
</tr>
<tr>
<td>6</td>
<td>PARALLEL DEVELOP</td>
<td>0.4</td>
<td>21</td>
<td>LATEST WEEK</td>
<td>238.00</td>
</tr>
<tr>
<td>7</td>
<td>MOTOR RELIABILITY</td>
<td>90.00</td>
<td>22</td>
<td>EARLIEST WEEK</td>
<td>202.00</td>
</tr>
<tr>
<td>8</td>
<td>AIRFRAME RELIABILITY</td>
<td>97.00</td>
<td>23</td>
<td>MAX DELIVER INCENTIVE</td>
<td>3.00</td>
</tr>
<tr>
<td>9</td>
<td>LAUNCHER RELIABILITY</td>
<td>98.50</td>
<td>24</td>
<td>MAXIMUM RELIABILITY</td>
<td>91.00</td>
</tr>
<tr>
<td>10</td>
<td>FIRE CONTROL ERROR YDS</td>
<td>80.00</td>
<td>25</td>
<td>MINIMUM RELIABILITY</td>
<td>85.00</td>
</tr>
<tr>
<td>11</td>
<td>GUIDANCE ERROR YDS</td>
<td>70.00</td>
<td>26</td>
<td>MAX RELIABLE INCENTIVE</td>
<td>3.50</td>
</tr>
<tr>
<td>12</td>
<td>MOTOR QUAL TESTS</td>
<td>29.00</td>
<td>27</td>
<td>MAXIMUM ERROR YDS</td>
<td>160.00</td>
</tr>
<tr>
<td>13</td>
<td>AIRFRAME QUAL TESTS</td>
<td>6.00</td>
<td>28</td>
<td>MINIMUM ERROR YDS</td>
<td>140.00</td>
</tr>
<tr>
<td>14</td>
<td>LAUNCHER QUAL TESTS</td>
<td>5.00</td>
<td>29</td>
<td>MAX ERROR INCENTIVE</td>
<td>3.50</td>
</tr>
<tr>
<td>15</td>
<td>FIRE CONTROL QUAL TESTS</td>
<td>1.00</td>
<td>30</td>
<td>WEEK FOR LOT 10</td>
<td>337.00</td>
</tr>
</tbody>
</table>

****NO. NONE****31, INPUT BY PAGE

SELECT AN ITEM WHICH HELD TO BE CHANGED FOR DP-3

Check each display carefully for erroneous inputs. The results computed are based on the values appearing in this table.

Change by page

*****TYPE IN A LIST OF PAGES YOU DESIRE TO CHANGE:*****

EX:1234 WILL ASK FOR NEW DATA ON ALL PAGES; OR, 23 WILL ASK FOR NEW DATA ON PAGES 2 AND 3.

*****WARNING******

YOU MUST ENTER AT LEAST ONE DIGIT EVEN IF IT IS A '0'.

Changing pages 1 and 2
REENTER PAGE 1

*****INPUT PAGE 1 DP-3 INFO*****
CONTRACTOR NUMBER,(1 OR 2)?:
GUIDANCE APPROACH,(1 OR 2)?:
GUIDANCE CONFIGURATION,(1-4; NOT 1 OR 3 AFTER DP-3):?
MAINTAINABILITY ENGINEERING FUNDS IN MS:?
VALUE ENGINEERING FUNDS IN MS:?

Parallel guidance selections 1 or 3 will be rejected after DP-3.

NOTE No query for parallel guidance funds.
If guidance configuration selected is 2 or 4, or after DP-3, parallel guidance funds will not be queried. They are zeroed.

Check of data values will occur at the end of each page.
Necessary corrections will be queried.
Sample *****WARNINGS***** of data errors are demonstrated below.
The value of parallel development funds at $0.5 million is not an acceptable option.
The program now cues the operator to the problem by asking for a correction in the guidance configuration which had prevented the desired change in parallel development funds.

DATA CHECKS...
CORRECTIONS...

*****WARNING*****
PARALLEL DEV FUNDS OF $ 0.50 M ARE NOT REQUIRED AND HAVE BEEN RESET TO $0.0
DO YOU WISH TO CHANGE GUIDANCE CONFIGURATION:
Y"=YES OR "N"=NO?

===>N
******INPUT PAGE 2 DP-3 INFO.******

MOTOR RELIABILITY?
AIRFRAME RELIABILITY?
LAUNCHER/GSE RELIABILITY?
FIRE CONTROL ACCURACY?
GUIDANCE ACCURACY?
MOTOR QUAL TESTS?
AIRFRAME QUAL TESTS?
LAUNCHER/GSE QUAL TESTS?
FIRE CONTROL QUAL TESTS?
GUIDANCE SYSTEM QUAL TESTS?
FLIGHT TESTS?

............D A T A C H E C K S .............

............C O R R E C T I O N S .............

Checks of data values will occur at the end of each page. Necessary corrections will be queried. Sample ******WARNINGS****** of data errors are demonstrated below.
<table>
<thead>
<tr>
<th>NO.</th>
<th>ITEM</th>
<th>CURRENT VALUE @ DP-3</th>
<th>NO.</th>
<th>ITEM</th>
<th>CURRENT VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CONTRACTOR</td>
<td>1</td>
<td>16</td>
<td>GUIDANCE QUAL TESTS</td>
<td>6.00</td>
</tr>
<tr>
<td>2</td>
<td>GUIDANCE APPROACH</td>
<td>1</td>
<td>17</td>
<td>FLIGHT TESTS</td>
<td>17.00</td>
</tr>
<tr>
<td>3</td>
<td>GUIDANCE CONFIGURATION</td>
<td>2</td>
<td>18</td>
<td>MAXIMUM COST</td>
<td>MS: 93.00</td>
</tr>
<tr>
<td>4</td>
<td>MAINTAINABILITY ENG MS</td>
<td>1.75</td>
<td>19</td>
<td>MINIMUM COST</td>
<td>MS: 47.00</td>
</tr>
<tr>
<td>5</td>
<td>VALUE ENG</td>
<td>0.75</td>
<td>20</td>
<td>MAX COST INCENTIVE</td>
<td>%: 5.00</td>
</tr>
<tr>
<td>6</td>
<td>PARALLEL DEVELOP MS</td>
<td>0.0</td>
<td>21</td>
<td>LATEST WEEK</td>
<td>238.00</td>
</tr>
<tr>
<td>7</td>
<td>MOTOR RELIABILITY</td>
<td>92.00</td>
<td>22</td>
<td>EARLIEST WEEK</td>
<td>202.00</td>
</tr>
<tr>
<td>8</td>
<td>AIRFRAME RELIABILITY</td>
<td>97.00</td>
<td>23</td>
<td>MAX DELIVER INCENTIVE</td>
<td>%: 3.00</td>
</tr>
<tr>
<td>9</td>
<td>LAUNCHER RELIABILITY</td>
<td>98.50</td>
<td>24</td>
<td>MAXIMUM RELIABILITY</td>
<td>%: 91.00</td>
</tr>
<tr>
<td>10</td>
<td>FIRE CONTROL ERROR YDS</td>
<td>80.00</td>
<td>25</td>
<td>MINIMUM RELIABILITY</td>
<td>%: 85.00</td>
</tr>
<tr>
<td>11</td>
<td>GUIDANCE ERROR YDS</td>
<td>70.00</td>
<td>26</td>
<td>MAX RELIABILITY INCENTIVE</td>
<td>%: 3.50</td>
</tr>
<tr>
<td>12</td>
<td>MOTOR QUAL TESTS</td>
<td>25.00</td>
<td>27</td>
<td>MAXIMUM ERROR YDS</td>
<td>160.00</td>
</tr>
<tr>
<td>13</td>
<td>AIRFRAME QUAL TESTS</td>
<td>6.00</td>
<td>28</td>
<td>MINIMUM ERROR YDS</td>
<td>140.00</td>
</tr>
<tr>
<td>14</td>
<td>LAUNCHER QUAL TESTS</td>
<td>3.00</td>
<td>29</td>
<td>MAX ERROR INCENTIVE</td>
<td>%: 3.50</td>
</tr>
<tr>
<td>15</td>
<td>FIRE CONTROL QUAL TESTS</td>
<td>3.00</td>
<td>30</td>
<td>WEEK FOR LOT 10</td>
<td>337.00</td>
</tr>
</tbody>
</table>

Select an item which needs to be changed for DP-3.

Zero from the Selection Menu sends the program to the Main Menu.
MAIN MENU

DO YOU WISH TO:

1. RECEIVE THE TABLE OF ACHIEVED VALUES.
   *****WARNING***** AFTER THE TENTH DP=3 RUN
   AN ADDITIONAL DEVELOPMENT COST OF $100,000
   WILL BE INCURRED FOR EACH ADDITIONAL RUN.
   THIS IS RUN 1 OF DP=3.

2. INPUT SELECTION MENU

3. SUBMIT A CONTRACT PROPOSAL

4. EXIT

"1" tells the program to proceed with a check of the total fee percentage. It
insures the total fee percentage, 15% of the development contract, is
allocated. If the fees do not total 15%, the program queries the team
for corrected fee data. If the percentage check is passed, the program
calculates achieved values, counts the run number for computation
costs and displays the Achieved Values Table below.
### Achieved Values

**DP-3 *** Development Contract Summary *** Team 1**

<table>
<thead>
<tr>
<th>Incentive Provisions</th>
<th>Inc. Achievements</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Incentive Area</strong></td>
<td><strong>Worst Value</strong></td>
</tr>
<tr>
<td>Dev. Cost</td>
<td>$53.0M</td>
</tr>
<tr>
<td>Test Compl.</td>
<td>238 wk</td>
</tr>
<tr>
<td>Reliability</td>
<td>85.0%</td>
</tr>
<tr>
<td>Accuracy</td>
<td>160yds</td>
</tr>
<tr>
<td>Totals</td>
<td></td>
</tr>
</tbody>
</table>

**Total Contract Price = $ 54.6M**

The "PAUSE" causes the screen to remain until the operator indicates he is ready to proceed. "<ENTER>" symbolizes pressing the enter key which causes the program to start processing again.
DO YOU WISH TO:
1. RECEIVE THE TABLE OF ACHIEVED VALUES.
   *****WARNING**** AFTER THE TENTH DP=3 RUN AN ADDITIONAL DEVELOPMENT COST OF $100,000 WILL BE INCURRED FOR EACH ADDITIONAL RUN.
   THIS IS RUN 2 OF DP=3.
2. INPUT SELECTION MENU
3. SUBMIT A CONTRACT PROPOSAL
4. EXIT

Sometimes the display will show "MORE" in the lower right margin of the screen. After several seconds the screen will scroll itself. PAUSE prevents this screen scrolling action. When "MORE" is displayed and the operator wishes to prevent the display from automatically scrolling, the display may be held by pressing "<ENTER>". "HOLDING" will be displayed. To proceed from a "HOLDING" or "MORE" display, press the "<ALT>" and "<CLEAR>" keys simultaneously.

NOTE the count of calculation runs has changed. The program counts the occurrences of achieved value calculations. After ten calculations, the program adds a charge of $100,000 for each additional run.
The result of selecting 2 from the Main Menu, menu 2, which a date item may be selected for change.
The procedure for changing any data in the database is accomplished as follows.

To change the parallel development funds, choose item number 6 from the Selection Menu.

Parallel development funds are selected for change.

The current value input for parallel development funds is $2 million.

The value of parallel development funds at $2 million is not an acceptable option.

The program now cues the operator to the prob-
DO YOU WISH TO CHANGE GUIDANCE CONFIGURATION:
"Y"=YES OR "N"=NO?

****=N

**SELECT** MENU

<table>
<thead>
<tr>
<th>NO.<em>ITEM</em>*****</th>
<th>CURRENT VALUE @ DP-3</th>
<th>NO.<em>ITEM</em>*****</th>
<th>CURRENT VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. CONTRACTOR</td>
<td>1</td>
<td>16. GUIDANCE QUAL TESTS</td>
<td>6.00</td>
</tr>
<tr>
<td>2. GUIDANCE APPROACH</td>
<td>1</td>
<td>17. FLIGHT TESTS</td>
<td>17.00</td>
</tr>
<tr>
<td>3. GUIDANCE CONFIGURATION</td>
<td>2</td>
<td>18. MAXIMUM COST</td>
<td>$53.00</td>
</tr>
<tr>
<td>4. MAINTAINABILITY ENG MS</td>
<td>1.75</td>
<td>19. MINIMUM COST</td>
<td>$47.00</td>
</tr>
<tr>
<td>5. VALUE ENG MS</td>
<td>0.75</td>
<td>20. MAX COST INCENTIVE</td>
<td>5.00%</td>
</tr>
<tr>
<td>6. PARALLEL DEVELOP MS</td>
<td>0.0</td>
<td>21. LATEST WEEK</td>
<td>238.00</td>
</tr>
<tr>
<td>7. MOTOR RELIABILITY</td>
<td>92.00</td>
<td>22. EARLIEST WEEK</td>
<td>202.00</td>
</tr>
<tr>
<td>8. AIRFRAME RELIABILITY</td>
<td>97.00</td>
<td>23. MAX DELIVER INCENTIVE</td>
<td>3.00</td>
</tr>
<tr>
<td>9. LAUNCHER RELIABILITY</td>
<td>98.50</td>
<td>24. MAXIMUM RELIABILITY</td>
<td>91.00</td>
</tr>
<tr>
<td>10. FIRE CONTROL ERROR YDS</td>
<td>80.00</td>
<td>25. MINIMUM RELIABILITY</td>
<td>85.00</td>
</tr>
<tr>
<td>11. GUIDANCE ERROR YDS</td>
<td>10.00</td>
<td>26. MAX RELIABILITY INCENTIVE</td>
<td>3.50</td>
</tr>
<tr>
<td>12. MOTOR QUAL TESTS</td>
<td>25.00</td>
<td>27. MAXIMUM ERROR YDS</td>
<td>160.00</td>
</tr>
<tr>
<td>13. AIRFRAME QUAL TESTS</td>
<td>6.00</td>
<td>28. MINIMUM ERROR YDS</td>
<td>140.00</td>
</tr>
<tr>
<td>14. LAUNCHER QUAL TESTS</td>
<td>3.00</td>
<td>29. MAX ERROR INCENTIVE</td>
<td>3.50</td>
</tr>
<tr>
<td>15. FIRE CONTROL QUAL TESTS</td>
<td>3.00</td>
<td>30. WEEK FOR LOT 10</td>
<td>337.00</td>
</tr>
</tbody>
</table>

====>20

PREVIOUS VALUE 5.00; INPUT A VALUE?

====>4

Choose the item to be changed, the maximum fee for development cost incentive.
The previous value was 5%.
The desired new value is 4%.
The value is changed in the Selection Menu, and the menu is reprinted.
### SELECTION MENU

<table>
<thead>
<tr>
<th>NO.<em>ITEM</em>*</th>
<th>CURRENT VALUE @ DP-3</th>
<th>NO.<em>ITEM</em>*</th>
<th>CURRENT VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. CONTRACTOR</td>
<td>1</td>
<td>16. GUIDANCE QUAL TESTS</td>
<td>6.00</td>
</tr>
<tr>
<td>2. GUIDANCE APPROACH</td>
<td>1</td>
<td>17. FLIGHT TESTS</td>
<td>17.00</td>
</tr>
<tr>
<td>3. GUIDANCE CONFIGURATION</td>
<td>2</td>
<td>18. MAXIMUM COST</td>
<td>52.00</td>
</tr>
<tr>
<td>4. MAINTAINABILITY ENG MS</td>
<td>1.75</td>
<td>19. MINIMUM COST</td>
<td>47.00</td>
</tr>
<tr>
<td>5. VALUE ENG MS</td>
<td>0.75</td>
<td>20. MAX COST INCENTIVE %</td>
<td>4.00</td>
</tr>
<tr>
<td>6. PARALLEL DEVELOP MS</td>
<td>0.00</td>
<td>21. LATEST WEEK</td>
<td>238.00</td>
</tr>
<tr>
<td>7. MOTOR RELIABILITY</td>
<td>95.00</td>
<td>22. EARLIEST WEEK</td>
<td>202.00</td>
</tr>
<tr>
<td>8. AIRFRAME RELIABILITY</td>
<td>97.00</td>
<td>23. MAX DELIVER INCENTIVES</td>
<td>3.00</td>
</tr>
<tr>
<td>9. LAUNCHER RELIABILITY</td>
<td>98.50</td>
<td>24. MAXIMUM RELIABILITY %</td>
<td>91.00</td>
</tr>
<tr>
<td>10. FIRE CONTROL ERROR YDS</td>
<td>80.00</td>
<td>25. MINIMUM RELIABILITY %</td>
<td>85.00</td>
</tr>
<tr>
<td>11. GUIDANCE ERROR YDS</td>
<td>70.00</td>
<td>26. MAX RELIABIL INCENTIVE</td>
<td>3.50</td>
</tr>
<tr>
<td>12. MOTOR QUAL TESTS</td>
<td>25.00</td>
<td>27. MAXIMUM ERROR YDS</td>
<td>160.00</td>
</tr>
<tr>
<td>13. AIRFRAME QUAL TESTS</td>
<td>6.00</td>
<td>28. MINIMUM ERROR YDS</td>
<td>140.00</td>
</tr>
<tr>
<td>14. LAUNCHER QUAL TESTS</td>
<td>3.00</td>
<td>29. MAX ERROR INCENTIVE %</td>
<td>3.50</td>
</tr>
<tr>
<td>15. FIRE CONTROL QUAL TESTS</td>
<td>3.00</td>
<td>30. WEEK FOR LOT 10</td>
<td>337.00</td>
</tr>
</tbody>
</table>

**Notice the new value.**

Select an item which needs to be changed for DP-3.
===>0

***WARNING***  TOTAL OF FEE PERCENTAGES IS LESS THAN 15%.  
THE VALUES YOU PREVIOUSLY SUBMITTED ARE:
COST FEE INCENTIVE= 4.00
DELIVERY DATE FEE INCENTIVE= 3.00
RELIABILITY INCENTIVE FEE= 3.50
ACCURACY INCENTIVE FEE= 3.50
REDO YOUR INPUT.

===>4

ENTER THE COST INCENTIVE FEE.
PREVIOUS VALUE  4.00; INPUT A VALUE?

===>3.5

YOU HAVE 11.00% OF THE INCENTIVE FEES REMAINING.
ENTER THE DELIVERY DATE INCENTIVE FEE.
PREVIOUS VALUE  3.00; INPUT A VALUE?

===>3.5

YOU HAVE 7.50% OF THE INCENTIVE FEES REMAINING.
ENTER THE RELIABILITY INCENTIVE FEE.
PREVIOUS VALUE  3.50; INPUT A VALUE?

===>3.5

YOU HAVE 4.00% OF THE INCENTIVE FEES REMAINING.
ENTER THE ACCURACY INCENTIVE FEE.
PREVIOUS VALUE  3.50; INPUT A VALUE?

The result of selecting 0 from the Selection Menu will be to reprint the Main Menu.
Zero from the Selection Menu sends the program to the Main Menu.
Exit from the Selection Menu routine by entering '0'. The program will first evaluate the fee total percentage.
Whenever the fee total does not equal 15%, the program queries for correction.
These are the old values.

The program keeps track of the fee remainder. It tells how much fee remains to be allocated.

If the ACCURACY FEE entered is not equal to this last remainder, the program assumes a data entry error. It re-
loops through the EXIT routine and can return to the percentage routine.

DO YOU WISH TO CONTINUE; "Y"=YES OR "N"=NO?
***WARNING*** TOTAL OF FEE PERCENTAGES IS LESS THAN 15%.
THE VALUES YOU PREVIOUSLY SUBMITTED ARE:
COST INCENTIVE = 8.00
DELIVERY DATE FEE INCENTIVE = 3.00
RELIABILITY INCENTIVE FEE = 3.50
ACCURACY INCENTIVE FEE = 3.50
REDO YOUR INPUT.

ENTER THE COST INCENTIVE FEE.
PREVIOUS VALUE 4.00; INPUT A VALUE?

YOU HAVE 11.00% OF THE INCENTIVE FEES REMAINING.
ENTER THE DELIVERY DATE INCENTIVE FEE.
PREVIOUS VALUE 3.00; INPUT A VALUE?

YOU HAVE 7.50% OF THE INCENTIVE FEES REMAINING.
ENTER THE RELIABILITY INCENTIVE FEE.
PREVIOUS VALUE 3.50; INPUT A VALUE?

YOU HAVE 4.00% OF THE INCENTIVE FEES REMAINING.
ENTER THE ACCURACY INCENTIVE FEE.
PREVIOUS VALUE 3.50; INPUT A VALUE?

Unsuccessful correction of the percentage fees returns the program to the percentage correction routine.
These are the old values.
The program keeps track of the fee remainder.
It tells how much fee remains to be allocated.
Successful completion of the corrections returns the program to the Main Menu.
**MAIN MENU**

DO YOU WISH TO:

1. RECEIVE THE TABLE OF ACHIEVED VALUES.
   *****WARNING***** AFTER THE TENTH DP-3 RUN
   AN ADDITIONAL DEVELOPMENT COST OF $100,000
   WILL BE INCURRED FOR EACH ADDITIONAL RUN.
   THIS IS RUN 2 OF DP-3.

2. INPUT SELECTION MENU
3. SUBMIT A CONTRACT PROPOSAL
4. EXIT

**====>1**
## Achieved Values

**DP-3 *** Development Contract Summary *** Team 1**

### Incentive Provisions

<table>
<thead>
<tr>
<th>INCENTIVE AREA</th>
<th>WORST VALUE</th>
<th>BEST VALUE</th>
<th>MAX FEE ALLOWED</th>
<th>ACHIEVED VALUE</th>
<th>FEE EARNED</th>
<th>FEE % EARNED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dev. Cost</td>
<td>$53.0M</td>
<td>$47.0M</td>
<td>4.0%</td>
<td>$50.81M</td>
<td>$0.730M</td>
<td>1.46%</td>
</tr>
<tr>
<td>Flt Tst Compl</td>
<td>238 wk</td>
<td>202 wk</td>
<td>3.5%</td>
<td>199 wk</td>
<td>$1.750M</td>
<td>3.50%</td>
</tr>
<tr>
<td>Reliability</td>
<td>85.0%</td>
<td>91.0%</td>
<td>3.5%</td>
<td>83.81%</td>
<td>$0.00 M</td>
<td>0.0%</td>
</tr>
<tr>
<td>Accuracy</td>
<td>160 YDS</td>
<td>140 YDS</td>
<td>4.0%</td>
<td>143 YDS</td>
<td>$1.680 M</td>
<td>3.36%</td>
</tr>
<tr>
<td><strong>TOTALS</strong></td>
<td></td>
<td></td>
<td>15.0%</td>
<td><strong>$4.161M</strong></td>
<td><strong>8.32%</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Total Contract Price = $55.0M**

---

Press [Enter] to continue.
**MAIN MENU**

<table>
<thead>
<tr>
<th>DO YOU WISH TO:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. RECEIVE THE TABLE OF ACHIEVED VALUES.</td>
</tr>
<tr>
<td>2. INPUT SELECTION MENU</td>
</tr>
<tr>
<td>3. SUBMIT A CONTRACT PROPOSAL</td>
</tr>
<tr>
<td>4. EXIT</td>
</tr>
</tbody>
</table>

---

From the Main menu, selection of 3 provides a printout of the database as the Report Submission Table.
<table>
<thead>
<tr>
<th>No.</th>
<th>Item</th>
<th>Current Value</th>
<th>No.</th>
<th>Item</th>
<th>Current Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Contractor</td>
<td>1</td>
<td>16</td>
<td>Guidance Qual Tests</td>
<td>6.00</td>
</tr>
<tr>
<td>2</td>
<td>Guidance Approach</td>
<td>1</td>
<td>17</td>
<td>Flight Tests</td>
<td>17.00</td>
</tr>
<tr>
<td>3</td>
<td>Guidance Configuration</td>
<td>2</td>
<td>18</td>
<td>Maximum Cost</td>
<td>53.00</td>
</tr>
<tr>
<td>4</td>
<td>Maintainability ENG MS</td>
<td>1.75</td>
<td>19</td>
<td>Minimum Cost</td>
<td>47.00</td>
</tr>
<tr>
<td>5</td>
<td>Value ENG</td>
<td>0.75</td>
<td>20</td>
<td>Max Cost Incentive %</td>
<td>4.00</td>
</tr>
<tr>
<td>6</td>
<td>Parallel Develop MS</td>
<td>0.0</td>
<td>21</td>
<td>Latest Week</td>
<td>238.00</td>
</tr>
<tr>
<td>7</td>
<td>Motor Reliability</td>
<td>92.00</td>
<td>22</td>
<td>Earliest Week</td>
<td>202.00</td>
</tr>
<tr>
<td>8</td>
<td>Airframe Reliability</td>
<td>97.00</td>
<td>23</td>
<td>Max Deliver Incentive %</td>
<td>3.50</td>
</tr>
<tr>
<td>9</td>
<td>Launcher Reliability</td>
<td>98.50</td>
<td>24</td>
<td>Maximum Reliability %</td>
<td>91.00</td>
</tr>
<tr>
<td>10</td>
<td>Fire Control Error YDS</td>
<td>80.00</td>
<td>25</td>
<td>Minimum Reliability %</td>
<td>85.00</td>
</tr>
<tr>
<td>11</td>
<td>Guidence Error YDS</td>
<td>70.00</td>
<td>26</td>
<td>Max Reliabil Incnetive %</td>
<td>3.50</td>
</tr>
<tr>
<td>12</td>
<td>Motor Qual Tests</td>
<td>25.00</td>
<td>27</td>
<td>Maximum Error YDS</td>
<td>150.00</td>
</tr>
<tr>
<td>13</td>
<td>Airframe Qual Tests</td>
<td>6.00</td>
<td>28</td>
<td>Minimum Error YDS</td>
<td>140.00</td>
</tr>
<tr>
<td>14</td>
<td>Launcher Qual Tests</td>
<td>3.00</td>
<td>29</td>
<td>Max Error Incentive %</td>
<td>4.00</td>
</tr>
<tr>
<td>15</td>
<td>Fire Control Qual Tests</td>
<td>3.00</td>
<td>30</td>
<td>Week For Lot 10</td>
<td>337.00</td>
</tr>
</tbody>
</table>

Do you wish to enter this data as your proposed DP-3 Decision?

Caution: You may only input a decision once.

Enter your team security code if you wish to commit to a decision; or "cont".

The result of entering "continue" is to return and reprint the Main Menu. Continue returns to the Main Menu without submitting a report.
MAIN MENU

DO YOU WISH TO:

1. RECEIVE THE TABLE OF ACHIEVED VALUES.
   *****WARNING***** AFTER THE TENTH DP-3 RUN
   AN ADDITIONAL DEVELOPMENT COST OF $100,000
   WILL BE INCURRED FOR EACH ADDITIONAL RUN.
   THIS IS RUN 3 OF DP-3.

2. INPUT SELECTION MENU

3. SUBMIT A CONTRACT PROPOSAL

4. EXIT

---

Provides a printout of the database to verify the proposed values being submitted.
<table>
<thead>
<tr>
<th>No.*Item</th>
<th>Current Value @ DP-3</th>
<th>No.*Item</th>
<th>Current Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. Maintainability Eng Mfgs</td>
<td>2.75</td>
<td>18. Maximum Cost Mfgs</td>
<td>53.00</td>
</tr>
<tr>
<td>4. Guidance Configuration</td>
<td>1.75</td>
<td>19. Minimum Cost Mfgs</td>
<td>47.00</td>
</tr>
<tr>
<td>5. Value Eng</td>
<td>0.75</td>
<td>20. Max Cost Incentive %</td>
<td>4.00</td>
</tr>
<tr>
<td>6. Parallel Develop Mfgs</td>
<td>0.0</td>
<td>21. Latest Week</td>
<td>238.00</td>
</tr>
<tr>
<td>7. Motor Reliability</td>
<td>92.00</td>
<td>22. Earliest Week</td>
<td>202.00</td>
</tr>
<tr>
<td>8. Airframe Reliability</td>
<td>96.50</td>
<td>23. Max Deliver incentives</td>
<td>3.50</td>
</tr>
<tr>
<td>9. Launcher Reliability</td>
<td>98.00</td>
<td>24. Maximum Reliability %</td>
<td>91.00</td>
</tr>
<tr>
<td>10. Fire Control Error YDS</td>
<td>80.00</td>
<td>25. Minimum Reliability %</td>
<td>85.00</td>
</tr>
<tr>
<td>11. Guidance Error YDS</td>
<td>70.00</td>
<td>26. Max Reliability Incentive %</td>
<td>3.50</td>
</tr>
<tr>
<td>12. Motor Qual Tests</td>
<td>25.00</td>
<td>27. Maximum Error YDS</td>
<td>160.00</td>
</tr>
<tr>
<td>13. Airframe Qual Tests</td>
<td>6.00</td>
<td>28. Minimum Error YDS</td>
<td>140.00</td>
</tr>
<tr>
<td>14. Launcher Qual Tests</td>
<td>3.00</td>
<td>29. Max Error Incentive %</td>
<td>4.00</td>
</tr>
<tr>
<td>15. Fire Control Qual Tests</td>
<td>3.00</td>
<td>30. Week for Lot 10</td>
<td>337.00</td>
</tr>
</tbody>
</table>

**DO YOU WISH TO ENTER THIS DATA AS YOUR PROPOSED DP-3 DECISION?**

**WARNING***

YOU MAY ONLY INPUT A DECISION ONCE.

ENTER YOUR TEAM SECURITY CODE IF YOU WISH TO COMMIT TO A DECISION;

OR "CONT".
REPORT ACCEPTED VALUES TABLE

<table>
<thead>
<tr>
<th>NO.</th>
<th>ITEM DESCRIPTION</th>
<th>DP-3 PROPOSED INPUT</th>
<th>CURRENT VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CONTRACTOR</td>
<td>1</td>
<td>16.00</td>
</tr>
<tr>
<td>2</td>
<td>GUIDANCE APPROACH</td>
<td>1</td>
<td>1.75</td>
</tr>
<tr>
<td>3</td>
<td>MAINTAINABILITY ENG MS</td>
<td>2</td>
<td>18.00</td>
</tr>
<tr>
<td>4</td>
<td>VALUE ENG</td>
<td>0.75</td>
<td>19.00</td>
</tr>
<tr>
<td>5</td>
<td>PARALLEL DEVELOP MS</td>
<td>0.0</td>
<td>20.00</td>
</tr>
<tr>
<td>6</td>
<td>MOTOR RELIABILITY</td>
<td>92.00</td>
<td>21.00</td>
</tr>
<tr>
<td>7</td>
<td>AIRFRAME RELIABILITY</td>
<td>97.00</td>
<td>22.00</td>
</tr>
<tr>
<td>8</td>
<td>LAUNCHER RELIABILITY</td>
<td>98.50</td>
<td>23.00</td>
</tr>
<tr>
<td>9</td>
<td>FIRE CONTROLS ERROR YDS</td>
<td>80.00</td>
<td>24.00</td>
</tr>
<tr>
<td>10</td>
<td>MAX DELIVER INCENTIVE</td>
<td>70.00</td>
<td>25.00</td>
</tr>
<tr>
<td>11</td>
<td>MAX ERROR INCENTIVE</td>
<td>160.00</td>
<td>26.00</td>
</tr>
<tr>
<td>12</td>
<td>FIRE CONTROL QLTS</td>
<td>3.00</td>
<td>27.00</td>
</tr>
<tr>
<td>13</td>
<td>MAX ERROR INCENTIVE</td>
<td>337.00</td>
<td>28.00</td>
</tr>
<tr>
<td>14</td>
<td>FIRE CONTROL QLTS</td>
<td>3.00</td>
<td>29.00</td>
</tr>
<tr>
<td>15</td>
<td>MAX ERROR INCENTIVE</td>
<td>337.00</td>
<td>30.00</td>
</tr>
</tbody>
</table>

*****SEE YOUR MONITOR IF YOU HAVE A PROBLEM.*****

Entering the team's security code will rewrite the file DATABASE TEAMxx. From the Main menu, selection of option 3 directs the program to the report submission routine. This routine determines the DP report last submitted, it then displays the current database selections in a Report Submission Table as the offering for the next report. Team approval of the Submission Table is made by entering the team's security code. The program can sequence the team through
**MAIN MENU**

DO YOU WISH TO:

1. RECEIVE THE TABLE OF ACHIEVED VALUES.
   **WARNING:** AFTER THE TENTH DP-3 RUN AN ADDITIONAL DEVELOPMENT COST OF $100,000 WILL BE INCURRED FOR EACH ADDITIONAL RUN. THIS IS RUN 3 OF DP-3.
2. INPUT SELECTION MENU
3. SUBMIT A CONTRACT PROPOSAL
4. EXIT

File Selection:

- PUN FILE 6092 TO 0543P COPY 001 NOHOLD
- PUN FILE 6094 TO 0543P COPY 001 NOHOLD
- ANOTHER RUN, 'Y'/'N'?

**BEGIN RECORDING OF TERMINAL SESSION**

* NOTE! YOU ARE NOW LINKED TO PROJING ON YOUR 192 DISK MODE C
* NOTE! PRESS PF09 TO BREAK THE LINK
FILE 'DATAINST DISKNUM A' NOT FOUND
FILE 'DATAFILE TEAM01 C' NOT FOUND
FILE 'DATACODE TEAM01 C' NOT FOUND

WAIT FOR "EXECUTION TO BEGIN"
EXECUTION BEGINS...

the stages of contract submission including DP-3 and 4's proposed and final contract reports.
Pressing <ENTER> returns the program to the Main Menu display.

Exits from the program.
The data files are now transmitted to the monitor.
The LINKPROG EXEC program allows the user to loop to the program start and begin again.

another session.
This time DATAFILE TEAM01 A is found and so is DATACODE TEAM01 A.
They now exist on your A disk.
TO TERMINATE THE PROGRAM AT ANY POINT,
TYPE "N=EXIT IN RESPONSE TO ANY "YES/NO" QUERY.
***CAUTION*** IN RESPONSE TO A QUERY, PRESSING <<ENTER>>
WITHOUT PROVIDING DATA WILL DUMP THE PROGRAM.

PLEASE ENTER YOUR TEAM SECURITY CODE.

YOU HAVE ATTEMPTED TO ENTER A TEAM FILE
WITHOUT THE PROPER SECURITY CODE.
DO YOU WISH TO CONTINUE: "Y"=YES OR "N"=NO?

PLEASE ENTER YOUR TEAM SECURITY CODE.

If you misenter your previously selected security code the following message will appear.

In case of an error you may enter 'y' to return to the TEAM query.

The program will now proceed to check the database for previous information.
It will display the data in a SELECTION MENU.
<table>
<thead>
<tr>
<th>NO.*ITEM********CURRENT VALUE @ DP-3</th>
<th>NO.*ITEM********CURRENT VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. CONTRACTOR: 1</td>
<td>16. GUIDANCE QUAL TESTS: 6.00</td>
</tr>
<tr>
<td>2. GUIDANCE APPROACH: 1</td>
<td>17. FLIGHT TESTS: 17.00</td>
</tr>
<tr>
<td>3. GUIDANCE CONFIGURATION: 2</td>
<td>18. MAXIMUM COST $: 53.00</td>
</tr>
<tr>
<td>4. MAINTAINABILITY ENG $: 1.75</td>
<td>19. MINIMUM COST $: 47.00</td>
</tr>
<tr>
<td>5. VALUE ENG $: 0.75</td>
<td>20. MAX COST INCENTIVE %: 4.00</td>
</tr>
<tr>
<td>6. PARALLEL DEVELOP $: 0.0</td>
<td>21. LATEST WEEK: 238.00</td>
</tr>
<tr>
<td>7. MOTOR RELIABILITY %: 92.00</td>
<td>22. EARLIEST WEEK: 202.00</td>
</tr>
<tr>
<td>8. AIRFRAME RELIABILITY %: 97.00</td>
<td>23. MAX DELIVER INCENTIVE %: 3.50</td>
</tr>
<tr>
<td>9. LAUNCHER RELIABILITY %: 98.50</td>
<td>24. MAXIMUM RELIABILITY %: 91.00</td>
</tr>
<tr>
<td>10. FIRE CONTROL ERROR YDS: 80.00</td>
<td>25. MINIMUM RELIABILITY %: 89.00</td>
</tr>
<tr>
<td>11. GUIDANCE ERROR YDS: 70.00</td>
<td>26. MAX RELIABIL INCENTIVE %: 3.50</td>
</tr>
<tr>
<td>12. MOTOR QUAL TESTS: 29.00</td>
<td>27. MAXIMUM ERROR YDS: 160.00</td>
</tr>
<tr>
<td>13. AIRFRAME QUAL TESTS: 6.00</td>
<td>28. MINIMUM ERROR YDS: 140.00</td>
</tr>
<tr>
<td>14. LAUNCHER QUAL TESTS: 3.00</td>
<td>29. MAX ERROR INCENTIVE %: 4.00</td>
</tr>
<tr>
<td>15. FIRE CONTROL QUAL TSTS: 3.00</td>
<td>30. WEEK FOR LOT 10: 337.00</td>
</tr>
</tbody>
</table>

****0. NO**********31. CHANGE BY PAGE****
SELECT AN ITEM WHICH NEEDS TO BE CHANGED FOR DP-3.
MAIN MENU

DO YOU WISH TO:
1. RECEIVE THE TABLE OF ACHIEVED VALUES AFTER THE TENTH DP 3 RUN
2. INPUT A SELECTION FOR PROPOSAL
3. EXIT

AN ADJUSTABLE FOR EACH ADDITIONAL RUN
THIS IS RUN 3 OF DP 3.
### FINAL SUBMISSION TABLE FOR DP-3

<table>
<thead>
<tr>
<th>NO.</th>
<th><em>ITEM</em>***CURRENT VALUE</th>
<th>NO.</th>
<th>*ITEM********CURRENT VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>CONTRACTOR: 1</td>
<td>16.</td>
<td>GUIDANCE QUAL TESTS: 6.00</td>
</tr>
<tr>
<td>2.</td>
<td>GUIDANCE APPROACH: 1</td>
<td>17.</td>
<td>FLIGHT TESTS: 17.00</td>
</tr>
<tr>
<td>3.</td>
<td>GUIDANCE CONFIGURATION: 2</td>
<td>18.</td>
<td>MAXIMUM COST: 53.00</td>
</tr>
<tr>
<td>4.</td>
<td>MAINTAINABILITY ENG MS: 1.75</td>
<td>19.</td>
<td>MINIMUM COST: 47.00</td>
</tr>
<tr>
<td>5.</td>
<td>VALUE ENG MS: 0.75</td>
<td>20.</td>
<td>MAX COST INCENTIVE: 4.00</td>
</tr>
<tr>
<td>6.</td>
<td>PARALLEL DEVELOP MS: 0.0</td>
<td>21.</td>
<td>LATEST WEEK: 236.00</td>
</tr>
<tr>
<td>7.</td>
<td>MOTOR RELIABILITY MS: 92.00</td>
<td>22.</td>
<td>EARLIEST WEEK: 202.00</td>
</tr>
<tr>
<td>8.</td>
<td>AIRFRAME RELIABILITY MS: 95.00</td>
<td>23.</td>
<td>MAX DELIVER INCENTIVE: 3.50</td>
</tr>
<tr>
<td>9.</td>
<td>LAUNCHER RELIABILITY MS: 98.50</td>
<td>24.</td>
<td>MAXIMUM RELIABILITY: 91.00</td>
</tr>
<tr>
<td>10.</td>
<td>FIRE CONTROL ERROR YDS: 40.00</td>
<td>25.</td>
<td>MINIMUM RELIABILITY: 82.00</td>
</tr>
<tr>
<td>11.</td>
<td>GUIDANCE ERROR YDS: 70.00</td>
<td>26.</td>
<td>MAX RELIABLE INCENTIVE: 3.50</td>
</tr>
<tr>
<td>12.</td>
<td>MOTOR QUAL TESTS: 25.00</td>
<td>27.</td>
<td>MAXIMUM ERROR YDS: 160.00</td>
</tr>
<tr>
<td>13.</td>
<td>AIRFRAME QUAL TESTS: 6.00</td>
<td>28.</td>
<td>MINIMUM ERROR YDS: 140.00</td>
</tr>
<tr>
<td>14.</td>
<td>LAUNCHER QUAL TESTS: 3.00</td>
<td>29.</td>
<td>MAX ERROR INCENTIVE: 4.00</td>
</tr>
<tr>
<td>15.</td>
<td>FIRE CONTROL QUAL TEST: 3.00</td>
<td>30.</td>
<td>WEEK FOR LOT 10: 337.00</td>
</tr>
</tbody>
</table>

**IMPORTANT:****You MUST CHECK THE ABOVE ENTRIES WITH THE APPROVED FINAL DP-3.****

**ENTER YOUR TEAM SECURITY CODE IF YOU WISH TO COMMIT TO A DECISION; OR "CONT".**

The following sequence will demonstrate *
- first the filing of a final DP-3 contract proposal;*
- second we will exit from the program to reenter the program at DP-4;*
- fourth we will review the DP-4 Selection Menu as developed from the current DP-3 data base;*
- next the changing of the paired sets of data for expected minimum and maximum reliability;*
- sixth DP-4 will be evaluated with
REPORT ACCEPTED VALUES TABLE

<table>
<thead>
<tr>
<th>NO.</th>
<th>ITEM</th>
<th>CURRENT VALUE @ DP-3</th>
<th>NO.</th>
<th>ITEM</th>
<th>CURRENT VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CONTRACTOR</td>
<td>1</td>
<td>16</td>
<td>GUIDANCE QUAL TESTS</td>
<td>6.00</td>
</tr>
<tr>
<td>2</td>
<td>GUIDANCE APPROACH</td>
<td>1</td>
<td>17</td>
<td>FLIGHT TESTS</td>
<td>17.00</td>
</tr>
<tr>
<td>3</td>
<td>GUIDANCE CONFIGURATION</td>
<td>2</td>
<td>18</td>
<td>MAXIMUM COST</td>
<td>53.00</td>
</tr>
<tr>
<td>4</td>
<td>MAINTAINABILITY ENG MS</td>
<td>1.75</td>
<td>19</td>
<td>MINIMUM COST</td>
<td>47.00</td>
</tr>
<tr>
<td>5</td>
<td>VALUE ENC</td>
<td>0.75</td>
<td>20</td>
<td>MAX COST INCENTIVE %</td>
<td>4.00</td>
</tr>
<tr>
<td>6</td>
<td>PARALLEL DEVELOP MS</td>
<td>0.00</td>
<td>21</td>
<td>LATEST WEEK</td>
<td>238.00</td>
</tr>
<tr>
<td>7</td>
<td>MOTOR RELIABILITY</td>
<td>92.00</td>
<td>22</td>
<td>EARLIEST WEEK</td>
<td>202.00</td>
</tr>
<tr>
<td>8</td>
<td>AIRFRAME RELIABILITY $</td>
<td>97.00</td>
<td>23</td>
<td>MAX DELIVER INCENTIVE $</td>
<td>3.50</td>
</tr>
<tr>
<td>9</td>
<td>LAUNCHER RELIABILITY $</td>
<td>96.50</td>
<td>24</td>
<td>MAXIMUM RELIABILITY $</td>
<td>91.00</td>
</tr>
<tr>
<td>10</td>
<td>FIRE CONTROL ERROR YDS</td>
<td>80.00</td>
<td>25</td>
<td>MINIMUM RELIABILITY $</td>
<td>65.00</td>
</tr>
<tr>
<td>11</td>
<td>GUIDANCE ERROR YDS</td>
<td>70.00</td>
<td>26</td>
<td>MAX RELIABLE INCENTIVE $</td>
<td>3.50</td>
</tr>
<tr>
<td>12</td>
<td>MOTOR QUAL TESTS</td>
<td>25.00</td>
<td>27</td>
<td>MAXIMUM ERROR YDS</td>
<td>160.00</td>
</tr>
<tr>
<td>13</td>
<td>AIRFRAME QUAL TESTS</td>
<td>6.00</td>
<td>28</td>
<td>MINIMUM ERROR YDS</td>
<td>140.00</td>
</tr>
<tr>
<td>14</td>
<td>LAUNCHER QUAL TESTS</td>
<td>3.00</td>
<td>29</td>
<td>MAX ERROR INCENTIVE %</td>
<td>4.00</td>
</tr>
<tr>
<td>15</td>
<td>FIRE CONTROL QUAL TESTS</td>
<td>3.00</td>
<td>30</td>
<td>WEEK FOR LOT TO</td>
<td>337.00</td>
</tr>
</tbody>
</table>

***SEE YOUR MONITOR IF YOU HAVE A PROBLEM.***

1H0001A PAUSE ; PRESS <<ENTER>> TO CONTINUE.
MAIN MENU

DO YOU WISH TO:
1. RECEIVE THE TABLE OF ACHIEVED VALUES.
   WARNING: AFTER THE TENTH DP-4 RUN
   AN ADDITIONAL DEVELOPMENT COST OF $100,000
   WILL BE INCURRED FOR EACH ADDITIONAL RUN.
   THIS IS RUN 1 OF DP-4.
   <****
   NOTE the change in DP- level in this block.
2. INPUT SELECTION MENU
3. SUBMIT A CONTRACT PROPOSAL
4. EXIT

====>4

PUN FILE 8092 TO 0543P COPY 001 NOHOLD
PUN FILE 8094 TO 0543P COPY 001 NOHOLD
ANOTHER RUN, 'Y'/'N'?

====>Y

This step will stop the program
and return control to the EXEC,
PROJMG EXEC.
The data files are now transmitted
to the monitor.
The LINKPROJ EXEC program allows the user to
reloop to the program start and
begin again.
The following are DP-4 procedures. Logon as you did for DP-3.

BEGIN RECORDING OF TERMINAL SESSION
* NOTE1: YOU ARE NOW LINKED TO PROJHNG ON YOUR 192 DISK
MODE C
* NOTE1: PRESS PF9 TO BREAK THE LINK
FILE 'DATAINST DISKNUM A' NOT FOUND.
FILE 'DATAFILE TEAMO1 C' NOT FOUND.
FILE 'DATACODE TEAMO1 C' NOT FOUND.
PRESS <ENTER> WHEN YOU ARE READY TO CONTINUE

WAIT FOR "EXECUTION TO BEGIN"
EXECUTION BEGINS...

***************
TO TERMINATE THE PROGRAM AT ANY POINT,
TYPE "e"=EXIT IN RESPONSE TO ANY "YES/NO" QUERY.
***************
** CAUTION ** IN RESPONSE TO A QUERY, PRESSING <ENTER> WITHOUT PROVIDING DATA WILL DUMP THE PROGRAM.
***************

PLEASE ENTER YOUR TEAM SECURITY CODE.

Another session. This time DATAFILE TEAMO1 A is found; and, so is DATACODE TEAMO1 A. They now exist on your A disk.

The program will now proceed to check the database for previous information. It will display the data in a SELECTION MENU.
## SELECTION MENU

<table>
<thead>
<tr>
<th>NO.*ITEM</th>
<th>CURRENT VALUE</th>
<th>NO.*ITEM</th>
<th>CURRENT VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. CONTRACTOR</td>
<td>1</td>
<td>16. GUIDANCE QUAL TESTS</td>
<td>6.00</td>
</tr>
<tr>
<td>2. GUIDANCE APPROACH</td>
<td>1</td>
<td>17. FLIGHT TESTS</td>
<td>17.00</td>
</tr>
<tr>
<td>3. GUIDANCE CONFIGURATION</td>
<td>2</td>
<td>18. MAXIMUM COST</td>
<td>93.00</td>
</tr>
<tr>
<td>4. MAINTAINABILITY ENG MS</td>
<td>1.75</td>
<td>19. MINIMUM COST</td>
<td>47.00</td>
</tr>
<tr>
<td>5. VALUE ENG MS</td>
<td>0.75</td>
<td>20. MAX COST INCENTIVE</td>
<td>4.00</td>
</tr>
<tr>
<td>6. PARALLEL DEVELOP MS</td>
<td>0.0</td>
<td>21. LATEST WEEK</td>
<td>238.00</td>
</tr>
<tr>
<td>7. MOTOR RELIABILITY</td>
<td>292.00</td>
<td>22. EARLIEST WEEK</td>
<td>202.00</td>
</tr>
<tr>
<td>8. AIRFRAME RELIABILITY</td>
<td>97.00</td>
<td>23. MAX DELIVER INCENTIVE</td>
<td>3.50</td>
</tr>
<tr>
<td>9. LAUNCHER RELIABILITY</td>
<td>98.50</td>
<td>24. MAXIMUM RELIABILITY</td>
<td>91.00</td>
</tr>
<tr>
<td>10. FIRE CONTROL ERROR YDS</td>
<td>80.00</td>
<td>25. MINIMUM RELIABILITY</td>
<td>85.00</td>
</tr>
<tr>
<td>11. GUIDANCE ERROR YDS</td>
<td>70.00</td>
<td>26. MAX RELIABLE INCENTIVE</td>
<td>3.50</td>
</tr>
<tr>
<td>12. MOTOR QAL TESTS</td>
<td>25.00</td>
<td>27. MAXIMUM ERROR YDS</td>
<td>160.00</td>
</tr>
<tr>
<td>13. AIRFRAME QAL TESTS</td>
<td>6.00</td>
<td>28. MINIMUM ERROR YDS</td>
<td>140.00</td>
</tr>
<tr>
<td>14. LAUNCHER QAL TESTS</td>
<td>3.00</td>
<td>29. MAX ERROR INCENTIVE</td>
<td>4.00</td>
</tr>
<tr>
<td>15. FIRE CONTROL QAL TESTS</td>
<td>3.00</td>
<td>30. WEEK FOR LOT 10</td>
<td>337.00</td>
</tr>
</tbody>
</table>

Notice the value.

We are changing the maximum reliability.

The new value for maximum reliability is 85.

The program pairs the maximums and minimums for queries. If one is changed the program will ask if you wish to change the other.
<table>
<thead>
<tr>
<th>No.</th>
<th>Item</th>
<th>Current Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Contractor</td>
<td>1</td>
</tr>
<tr>
<td>2.</td>
<td>Guidance Approach</td>
<td>1</td>
</tr>
<tr>
<td>3.</td>
<td>Guidance Configuration</td>
<td>2</td>
</tr>
<tr>
<td>4.</td>
<td>Maintainability ENG MS</td>
<td>1.75</td>
</tr>
<tr>
<td>5.</td>
<td>Value ENG</td>
<td>0.75</td>
</tr>
<tr>
<td>6.</td>
<td>Parallel Develop MS</td>
<td>0.0</td>
</tr>
<tr>
<td>7.</td>
<td>Motor Reliability</td>
<td>92.00</td>
</tr>
<tr>
<td>8.</td>
<td>Airframe Reliability</td>
<td>97.00</td>
</tr>
<tr>
<td>9.</td>
<td>Launcher Reliability</td>
<td>98.50</td>
</tr>
<tr>
<td>10.</td>
<td>Fire Control Error YDS</td>
<td>80.00</td>
</tr>
<tr>
<td>11.</td>
<td>Guidance Error YDS</td>
<td>70.00</td>
</tr>
<tr>
<td>12.</td>
<td>Motor Qual Tests</td>
<td>25.00</td>
</tr>
<tr>
<td>13.</td>
<td>Airframe Qual Tests</td>
<td>6.00</td>
</tr>
<tr>
<td>14.</td>
<td>Launcher Qual Tests</td>
<td>3.00</td>
</tr>
<tr>
<td>15.</td>
<td>Fire Control Qual Tests</td>
<td>3.00</td>
</tr>
<tr>
<td>16.</td>
<td>Guidance Qual Tests</td>
<td>6.00</td>
</tr>
<tr>
<td>17.</td>
<td>Flight Tests</td>
<td>17.00</td>
</tr>
<tr>
<td>18.</td>
<td>Maximum Cost MS</td>
<td>53.00</td>
</tr>
<tr>
<td>19.</td>
<td>Minimum Cost MS</td>
<td>47.00</td>
</tr>
<tr>
<td>20.</td>
<td>Max Cost Incentive %</td>
<td>4.00</td>
</tr>
<tr>
<td>21.</td>
<td>Latest Week</td>
<td>238.00</td>
</tr>
<tr>
<td>22.</td>
<td>Earliest Week</td>
<td>202.00</td>
</tr>
<tr>
<td>23.</td>
<td>Max Deliver Incentive %</td>
<td>3.50</td>
</tr>
<tr>
<td>24.</td>
<td>Maximum Reliability</td>
<td>85.00</td>
</tr>
<tr>
<td>25.</td>
<td>Minimum Reliability</td>
<td>78.00</td>
</tr>
<tr>
<td>26.</td>
<td>Max Reliability Incentive %</td>
<td>3.50</td>
</tr>
<tr>
<td>27.</td>
<td>Maximum Error YDS</td>
<td>160.00</td>
</tr>
<tr>
<td>28.</td>
<td>Minimum Error YDS</td>
<td>160.00</td>
</tr>
<tr>
<td>29.</td>
<td>Max Error Incentive %</td>
<td>4.00</td>
</tr>
<tr>
<td>30.</td>
<td>Week for Lot 10</td>
<td>337.00</td>
</tr>
</tbody>
</table>

Notice the new value. Select an item which needs to be changed for DP-4.
MAIN MENU

DO YOU WISH TO:

1. RECEIVE THE TABLE OF ACHIEVED VALUES.
   WARNING: AFTER THE TENTH DP-4 RUN
   AN ADDITIONAL DEVELOPMENT COST OF $100,000
   WILL BE INCURRED FOR EACH ADDITIONAL RUN.
   THIS IS RUN 1 OF DP-4.
2. INPUT SELECTION MENU.
3. SUBMIT A CONTRACT PROPOSAL
4. EXIT

=====>1
### Achieved Values

**DP-4 *** DEVELOPMENT CONTRACT SUMMARY *** TEAM 1**

<table>
<thead>
<tr>
<th>INCENTIVE AREAS</th>
<th>INCENTIVE PROVISIONS</th>
<th>INCENTIVE ACHIEVEMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WORST VALUE</td>
<td>BEST VALUE</td>
</tr>
<tr>
<td>DEV. COST</td>
<td>$ 53.0M</td>
<td>$ 47.0M</td>
</tr>
<tr>
<td>TEST COMPL.</td>
<td>238 WK</td>
<td>202 WK</td>
</tr>
<tr>
<td>RELIABILITY</td>
<td>78.0%</td>
<td>85.0%</td>
</tr>
<tr>
<td>ACCURACY</td>
<td>160 YDS</td>
<td>140 YDS</td>
</tr>
<tr>
<td>TOTALS</td>
<td>15.0%</td>
<td></td>
</tr>
<tr>
<td>TOTAL CONTRACT PRICE = $ 56.5M</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

100000A PAUSE ; PRESS <ENTER> TO CONTINUE.

The "PAUSE" causes the screen to retain the display until the operator is ready to continue.
MAIN MENU

DO YOU WISH TO:
1. RECEIVE THE TABLE OF ACHIEVED VALUES.
   WARNING: AFTER THE TENTH DP-4 RUN
   AN ADDITIONAL DEVELOPMENT COST OF $100,000
   WILL BE INCURRED FOR EACH ADDITIONAL RUN.
   THIS IS RUN 2 OF DP-4.
2. INPUT SELECTION MENU.
3. SUBMIT A CONTRACT PROPOSAL
4. EXIT

=====>3
### Proposal Submission Table

<table>
<thead>
<tr>
<th>No.</th>
<th>Item</th>
<th>Current Value @ DP-1</th>
<th>No.</th>
<th>Item</th>
<th>Current Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Contractor</td>
<td>1</td>
<td>16</td>
<td>Guidance Qual Tests</td>
<td>6.00</td>
</tr>
<tr>
<td>2</td>
<td>Guidance Approach</td>
<td>1</td>
<td>17</td>
<td>Flight Tests</td>
<td>17.00</td>
</tr>
<tr>
<td>3</td>
<td>Guidance Configuration</td>
<td>2</td>
<td>18</td>
<td>Maximum Cost</td>
<td>$53.00</td>
</tr>
<tr>
<td>4</td>
<td>Maintainability Eng M$</td>
<td>1.75</td>
<td>19</td>
<td>Minimum Cost</td>
<td>$47.00</td>
</tr>
<tr>
<td>5</td>
<td>Value Eng M$</td>
<td>0.75</td>
<td>20</td>
<td>Max Cost Incentive</td>
<td>4.00</td>
</tr>
<tr>
<td>6</td>
<td>Parallel Develop M$</td>
<td>0.0</td>
<td>21</td>
<td>Latest Week</td>
<td>238.00</td>
</tr>
<tr>
<td>7</td>
<td>Motor Reliability</td>
<td>92.00</td>
<td>22</td>
<td>Earliest Week</td>
<td>202.00</td>
</tr>
<tr>
<td>8</td>
<td>Airframe Reliability %</td>
<td>97.00</td>
<td>23</td>
<td>Max Deliver Incentive</td>
<td>3.50</td>
</tr>
<tr>
<td>9</td>
<td>Launcher Reliability %</td>
<td>98.50</td>
<td>24</td>
<td>Maximum Reliability %</td>
<td>85.00</td>
</tr>
<tr>
<td>10</td>
<td>Fire Control Error YDS</td>
<td>80.00</td>
<td>25</td>
<td>Minimum Reliability %</td>
<td>76.00</td>
</tr>
<tr>
<td>11</td>
<td>Guidance Error YDS</td>
<td>70.00</td>
<td>26</td>
<td>Max Reliability Incentive</td>
<td>3.50</td>
</tr>
<tr>
<td>12</td>
<td>Motor Qual Tests</td>
<td>25.00</td>
<td>27</td>
<td>Maximum Error YDS</td>
<td>160.00</td>
</tr>
<tr>
<td>13</td>
<td>Airframe Qual Tests</td>
<td>6.00</td>
<td>28</td>
<td>Minimum Error YDS</td>
<td>140.00</td>
</tr>
<tr>
<td>14</td>
<td>Launcher Qual Tests</td>
<td>3.00</td>
<td>29</td>
<td>Max Error Incentive</td>
<td>4.00</td>
</tr>
<tr>
<td>15</td>
<td>Fire Control Qual Tests</td>
<td>3.00</td>
<td>30</td>
<td>Week for Lot 10</td>
<td>337.00</td>
</tr>
</tbody>
</table>

*Do you wish to enter this data as your proposed DP-4 decision?***

****CAUTION****

You may only input a decision once. Enter your team security code if you wish to commit to a decision; or "CONT".

Entering the team's security code completes the filing of the new report to the monitor.
### ACCEPTED FINAL DP-4

---

******THE FOLLOWING PARAMETERS WERE ACCEPTED******

AS TEAM 1 DP-4 PROPOSED INPUT TO THE CONTRACTOR.

<table>
<thead>
<tr>
<th>NO.</th>
<th>ITEM</th>
<th>CURRENT VALUE @ DP-4</th>
<th>NO.</th>
<th>ITEM</th>
<th>CURRENT VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>CONTRACTOR</td>
<td>6.00</td>
<td>16.</td>
<td>GUIDANCE QUAL TESTS</td>
<td>6.00</td>
</tr>
<tr>
<td>2.</td>
<td>GUIDANCE APPROACH</td>
<td>1</td>
<td>17.</td>
<td>FLIGHT TESTS</td>
<td>17.00</td>
</tr>
<tr>
<td>3.</td>
<td>GUIDANCE CONFIGURATION</td>
<td>2.00</td>
<td>18.</td>
<td>MAXIMUM COST</td>
<td>53.00</td>
</tr>
<tr>
<td>4.</td>
<td>MAINTAINABILITY ENG MS</td>
<td>1.75</td>
<td>19.</td>
<td>MINIMUM COST</td>
<td>47.00</td>
</tr>
<tr>
<td>5.</td>
<td>VALUE ENG</td>
<td>0.75</td>
<td>20.</td>
<td>MAXIMUM INCENTIVE</td>
<td>4.00</td>
</tr>
<tr>
<td>6.</td>
<td>PARALLEL DEVELOP</td>
<td>0.00</td>
<td>21.</td>
<td>LATEST WEEK</td>
<td>238.00</td>
</tr>
<tr>
<td>7.</td>
<td>MOTOR RELIABILITY</td>
<td>92.00</td>
<td>22.</td>
<td>EARLIEST WEEK</td>
<td>202.00</td>
</tr>
<tr>
<td>8.</td>
<td>AIRFRAME RELIABILITY</td>
<td>97.00</td>
<td>23.</td>
<td>MAX DELIVER INCENTIVE</td>
<td>3.50</td>
</tr>
<tr>
<td>9.</td>
<td>LAUNCHER RELIABILITY</td>
<td>96.50</td>
<td>24.</td>
<td>MAXIMUM RELIABILITY</td>
<td>85.00</td>
</tr>
<tr>
<td>10.</td>
<td>FIRE CONTROL ERROR YDS</td>
<td>80.00</td>
<td>25.</td>
<td>MINIMUM RELIABILITY</td>
<td>78.00</td>
</tr>
<tr>
<td>11.</td>
<td>GUIDANCE ERROR YDS</td>
<td>70.00</td>
<td>26.</td>
<td>MAX RELIABILITY INCENTIVE</td>
<td>3.50</td>
</tr>
<tr>
<td>12.</td>
<td>MOTOR QL TESTS</td>
<td>25.00</td>
<td>27.</td>
<td>MAXIMUM ERROR YDS</td>
<td>160.00</td>
</tr>
<tr>
<td>13.</td>
<td>AIRFRAME QL TESTS</td>
<td>6.00</td>
<td>28.</td>
<td>MINIMUM ERROR YDS</td>
<td>140.00</td>
</tr>
<tr>
<td>14.</td>
<td>LAUNCHER QL TESTS</td>
<td>3.00</td>
<td>29.</td>
<td>MAX ERROR INCENTIVE</td>
<td>4.00</td>
</tr>
<tr>
<td>15.</td>
<td>FIRE CONTROL QL TESTS</td>
<td>3.00</td>
<td>30.</td>
<td>WEEK FOR LOT 10</td>
<td>337.00</td>
</tr>
</tbody>
</table>

******SEE YOUR MONITOR IF YOU HAVE A PROBLEM.******

1H0001A PAUSE; PRESS <ENTER> TO CONTINUE.
PROJMNG FORTRAN: AN INTERACTIVE COMPUTER PROGRAM FOR USE WITH THE DEFENSE MANAGEMENT SIMULATION EXERCISE(U)
NAVAL POSTGRADUATE SCHOOL MONTEREY CA  G W SCHULTZ
UNCLASSIFIED  MAR 84
<table>
<thead>
<tr>
<th>User Selection</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Request to receive the table of achieved values.</td>
</tr>
<tr>
<td>2.</td>
<td>Input the required additional runs.</td>
</tr>
<tr>
<td>3.</td>
<td>Exit the program.</td>
</tr>
</tbody>
</table>

**Notes:**
- Do not proceed with the request if the table is not complete.
- Additional runs will be necessary for each additional run requested.
- Select a contract proposal accordingly.
Defense Management Simulation

Simulation and Computer Directorate
Industrial College of the Armed Forces
Port Lesley J. McNair, Washington, D.C.

1970 April 1

The following Fortran code listing represents the reformatted and
aligned version of PROJING FORTRAN for inclusion as Appendix G
in a thesis of the same name. The thesis PROJING FORTRAN was under-
taken in June 1983 to upgrade the computer software support for the
DNS exercises. The main effort of the thesis was to make the program
user-friendly.

Requests for this document should be referred to the Superintendent,
Naval Postgraduate School, Monterey, California 93943.

For a list of the variables definitions, see the Glossary,

Appendix B

ccnow, rest(48), tab(20,25,3), ctest(3), cton(3), xris(2), isubs(48)
& : ifac(48), xi(48), irc(49), idbt(48)
& . ifd(48), item(48), qty(48), cha(48)
ccnow, res(48), rres(48), rresb(48), itype(48)
& : perf(48), tde(48), dec(3), y(8)
& : iprc(48), resi(48), if(48), resi(48)
*2X. **CAUTION**! IN RESPONSE TO A QUERY, PRESSING <<ENTER>> WITHOUT PROVIDING DATA WILL DUMP THE PROGRAM. /*1X.57('**')
*<<X.57('**') PAUSE: PRESS <<ENTER>> WHEN YOU ARE READY TO CONTINUE. */

---

**MAIN PROGRAM LOOP**

20 CONTINUE
CALL FERSSET(216, 1, -1, 1, 1)

(ESTABLISH INPUT VALUES)

ITEAM=0

CALL TEAMIN(ITEME, MR, DATFIL)

READ(DATFIL, 35, ERR=20, END=20) (MDUMP(JA, K), K=1, 5)

IF(STAT(3) .EQ. FINAL) KDP=4
IF(STAT(3) .EQ. FINAL) KDP=5
IF(STAT(3) .EQ. FINAL) KDP=6

BAYDF=KDP

(Initialize the team database)

(Monitor select the DP-level)

IF(MR.EQ.0) GOTO 33

PRINT 32, ITAM, KEP, STAT(KDF)

*1X, 'WHAT DECISION POINT DO YOU WISH TO ANALYZE?'

READ(IN, *, ERR=20, END=20) KDF

IF(KDP.GT.5 .OR. KEP .LT. 3) GOTO 30

IF(MR.EQ.3 OR MR.EQ.2) GOTO 33

GOTO 410

(STUDENT DP-level)

IF(MDUMP(3, 2) .NE. 0) GOTO 410

CALL PAGEIN
THE FOLLOWING LINE IMPedes STUDENT USE OF INSTRUCTOR OPTIONS; MB=1 OR MB=2.

C

MB=0

DO 300 I=1,4

IPG(I)=I

DATPI1=9

KDP=3

C

THE FOLLOWING LINE IMPedes STUDENT USE OF INSTRUCTOR OPTIONS; MB=1 OR MB=2.

C

MB=0

DO 300 I=1,4

IPG(I)=I

DATPI1=9

KDP=3
CALL SELECT(KDF)
CALL STORE
GOTO 207

CALL SETUP(KDP)

CONTINUE

(UPDATE DATABASE BY 'SELECTION' MENU)
(START FROM LAST COMPLETE DATABASE)

JDF=KDF
CONTINUE

IF(MDUP(JDF,2).NE.0.OR.JDF.LE.3)GOTO 76

JDF=KDF-1
GOTO 75
CONTINUE

CALL SELECT(JDF)

PRECALCULATION DATA PROCESSING

(CHECK THAT FEE TOTAL PERCENTAGE IS 15%)

CALL FRCTCK

CALL SETUP(KDP)

(CONTINUE

(DETERMINE DP, DECISION POINT; AND, PROMPT REPORT SUBMISSIONS)

IF(MR.NE.0)GOTO 231

KDP=3
IF(STAT(3).EQ.FINAL)KDP=4
IF(STAT(4).EQ.FINAL)KDP=5
IF(STAT(5).EQ.FINAL.AND.BR.EQ.0)STOP
IF(STAT(5).EQ.FINAL.AND.BR.NE.0)GOTO 859
NEWDP=KDP

CALL SETUP(KDP)
CALL GET(KDP)

IF(KDP.NE.3) CALL INPUT2

C

DPCOST=0.0

DC 876 IKDP=3,KDP

876  

DPCOST=DPCOST+(ADUMP(IKDP,28)-10.)*10000.

IDUMP=ADUMP(KDP,28)+1.

C

(BUFFER DATA ORGANIZATION FOR CALCULATIONS)

IF(MR.NE.0) GOTO 415

CALL STORE

C

875

CALL PROCTHS('CLSCHR

IQUESN=999

PRINT 211,KDP,IDUMP,KDP

211 FORMAT(' DO YOU WISH TO: /11/ RECEIVE THE TABLE OF ACHIEVED V

VALUES: /10X,'HARMING'-4 AFTER THE TENTH DE-11/.' RUN'./

*4,10X,'AN ADDITIONAL DEVELOPMENT COST OF $100,000'.

*2,10X,'10 X WILL BE INCURRED FOR EACH ADDITIONAL RUN'./11,10X,

*4,10X,'THIS IS RUN '12, OF DE-11/.'

IF(DPCOSGT.0.) PRINT 225,DPCTST

PRINT 225

225 FORMAT(10X,'DP RUNS HAVE COST AN ADDITIONAL $',P8.0)

PRINT 212

212 FORMAT(1X,'2. INPUT SELECTION MENU'

PRINT 213

213 FORMAT(1X,'3. SUBMIT A CONTRACT PROPOSAL')

IF(MR.EQ.0) GOTO 216

IQUESN=9

216

PRINT 214,IQUESN

214 FORMAT(1X,'1.. MONITOR'S MENU')

PRINT 217,IQUESN

217 FORMAT(1X,'1. EXIT')

IF(MR.EQ.0) CALL EXIT

208

IF(MR.NE.1) GOTO 226

IF(STORE.EQ.999.) GOTO 415

(ADD THE COST FOR THIS PRINTOUT)
IF(EQ.0) ADUMP(KDP,28)=ADUMP(KDP,28) +1.
DPCOST=DPCOST +100000.

GOTO 415
IF(ANT.EQ.IOUERSN) GOTO 859
IF(ANT.EQ.IOUERSE) STOP
IF(ANT.EQ.3) CALL FINISH
IF(ANT.EQ.2) GOTO 25
GOTO 230
CONTINUE
CALL EXITs
GOTO 230

FA IN PROGRAM CALCULATIONS

CALL ZNW
MTMP(1)=KDP
CALL GET(IVER)
DO 850 IVER=3,KDP
CALL IMULT
CALL IMPRINT
CALL TRADE
CALL DESRES
GOTO(90,100),KKPP
CALL DESRES
CALL SE
IF(MTMP(1),EQ.3,OR.MTMP(1),EQ.IDP) GOTO 130
ITCT=0
DO 110 M=1,INA
IF (ITEM(M)=0,1) 110,110,115
ITCT=ITCT+1
CONTINUE
GO TO 140
110
IDP=7
ITCT=-1
ITAB(25)=IDP
CALL M0D9
CALL REBUN
CALL REPCGT
CALL PERI
CALL GET(KDP)
CALL PRGSUM
CALL GET(KDP)
IF(MR.NE.1)G0T0 850
IF(MR.NE.0)G0T0 859
CALL FRBCMS('CLRSCRM ')
CALL GET(KDP)
PRINT 860
FORMAT(': MAIN MENU: 1. CHANGE TEAM NUMBER',/2X,
**2. THE DP SELECTION CHER',/2X,
**3. THE INPUT SELECTION MEND',/2X,
**4. REBUN THE DATA CALCULATION',/2X,
IF(DPCOST.GT.0.)PRINT 225, DPCOST
PRINT 861
FORMAT(' 2X,'5. DO A SENSITIVITY ANALYSIS',/2X,'6. PRINTCULL CCS
*1+FACTORS')
IQUES=7
IF(MR.NE.1)G0T0 862
IQUESM=IQUES
IQUES=IQUES+1
PRINT 863, IQUESM2
FORMAT (1X, I2, ' FILE TEAM PROPOSAL')

CALL PLOT (CKPLT)

IF (CKPLT .EQ. 0) GOTO 885
PCTPLG = 1.0
IQUEST = IQUEST
IQUEST = IQUEST + 1
IQUEST = IQUEST + 1
IQUEST = IQUEST + 1
IQUEST = IQUEST + 1
IQUEST = IQUEST + 1

PRINT 885, IQUEST

FORMAT (1X, I2, ' PRINT LAST SENSITIVITY TABLE : ' / I2,
* I2, ' PLOT LAST SENSITIVITY ANALYSIS ACHIEVED VALUES : ' / I2,
* I2, ' PLOT LAST SENSITIVITY ANALYSIS PEE PERCENTAGES : ' / I2,
* I2, ' USE OPTIMUM EI VALUE FROM LAST SENSITIVITY ANALYSS FOR DP- ' / I2,
* I2, ' DATA: ' )

PRINT 890, IQUEST

FORMAT (1X, I2, ' EXIT')
READ (IN, *, ERR=870, END=870) ANS
REWIND IN
IF (HR .NE. 2) GOTO 877
IF (ANS .NE. IQUEST .AND. ANS .NE. 1) GOTO 877
PRINT 27, ITEN
* CHNG DISK: Y/N ?
READ (IN, 28, ERR=870, END=870) ANS
FORMAT (A1)
IF (ANS .NE. YES) GOTO 877

CALL SETUP (KDP)
CALL STORE
CALL PLOTS ('CF', 'SPOOL', 'PUN', 'CONT', 'CL',
* 'TC', '076X', 'DATAPLIE', 'ITEM')
CALL PLTCHS ('CF', 'SPOOL', 'PUN', 'CLOSE', 'NOCONT',
* )
IF (ANS .EQ. 1) HR = 1

IF (ANS .EQ. IQUEST) STCP
IF (ANS.EQ. 1. OR. ANS.EQ. 2. OR. ANS.EQ. 3.) CKPLT=0.0
IF (ANS.EQ. 1. OR. ANS.EQ. 2. OR. ANS.EQ. 3.)
*I CALL ESEIX(CKPLT)
IF (ANS.EQ. 1.) STCPB=-0.0
IF (ANS.EQ. 1.) GOTO 20
IF (ANS.EQ. 2.) GOTO 30
IF (ANS.EQ. 3.) GOTO 75
IF (ANS.EQ. 4.) GOTO 415
IF (ANS.EQ. 6.) CALL PROSBL
IF (ANS.EQ. 1.) QUESB) CALL SNSPRT
IF (ANS.EQ. 1.) QUESB) CALL FLSCH
IF (ANS.EQ. 1.) QUESB) CALL FLIPIT(E2CTPLG)
IF (ANS.EQ. 1.) QUESB) GOTO 891
PRINT 894
FORMAT(I6, 'LOWEST VALUE FOR THE Y-AXIS')
READ(IN, *, ERR=859, END=859) YMN
PRINT 892
FORMAT(I6, 'HIGHEST VALUE FOR THE Y-AXIS')
READ(IN, *, ERR=859, END=859) YMX
CALL FLTSC(ymn, ymx)
IF (ANS.EQ. 1.) QUESB) CALL OPTIM
IF (ANS.EQ. 1.) QUESB) GOTO 75
IF (ANS.EQ. 1.) QUESB) CALL FINISH
IF (ANS.EQ. 1.) QUESB) GOTO 20
IF (ANS.EQ. 1.) QUESB) CALL STORE
IF (ANS.EQ. 1.) QUESB) GOTO 20
CALL EXIT1(KDP)
GOTO 859
CALL EXITS
GOTO 20
END

**************************************************************************************************************
SUBROUTINE EXITs

THIS SUBROUTINE PROVIDES A COMMON EXITING ROUTINE FROM ANY YES/NO QUERY IN THE PROGRAM. IT CONSIDERS THE NEED TO CLOSE ANY RECORD SESSIONS IN EFFECT. ALL DATABASE VARIABLES MUST BE ACCESSIBLE TO 'FINISH' THROUGH THIS SUBROUTINE.

COMMON REST(48), TAB(20, 25, 3), CTEST(3), CT0H(3), XMIS(2), ISUBS(48)
& IFAC(48), MT(48), IRC(49), IDT(48)
rewind

print 15

format(1x,'do you wish to continue;"y"=yes cr "n"=no')

read(in,25,err=10,end=10) ans

format(a1)

rewind

if(mr.ne.2)go to 877

if(ans.eq.yes)go to 877

print 27,team

format(1x,'should the rewritten team *","z"," pile ee sent to the pr

*ghng disk; y/n ???')

read(in,28,err=10,end=10) ans

format(a1)

call setup(kdp)

rewind

print 15

format(1x,'do you wish to continue;"y"=yes cr "n"=no')

read(in,25,err=10,end=10) ans

format(a1)

rewind

if(mr.ne.2)go to 877

if(ans.eq.yes)go to 877

print 27,team

format(1x,'should the rewritten team *","z"," pile ee sent to the pr

*ghng disk; y/n ???')

read(in,28,err=10,end=10) ans

format(a1)

call setup(kdp)
CALL STORE
CALL IFCHS('CF', SPOOL ; 'PUN ', 'COM ', 'CL '
** '.
CALL IFCHS('CF', SPOOL ; 'PUN ' ; 'DATAFILE ', 'ITEM')
CALL IFCHS(CF)
877 CONTINUE
IF (ANS. 'EQ. 'E') GOTO 999
IF (ANS. 'EQ. 'YES') RETURN
IF (ANS. 'EQ. 'ABC') GOTO 999
IF (ANS. 'EQ. 'S') CALL FINISH
IF (ANS. 'EQ. 'YES') RETURN
GOTO 10

C PROGRAM TERMINATION BRANCHES

C CONTINUE (DETERMINE IF STUDENT OR INSTRUCTOR)
C IF (NR. 'EQ. '1') GOTO 55
C CONTINUE (STUDENT TERMINATION)
C (DETERMINE IF FILES WERE GENERATED)
C CALL STORE (STORED NEW OR REVISED DATA FOR STUDENTS)
C STCP (INSTRUCTOR TERMINATION)
C CONTINUE
C STCP
C END
C
C SUBROUTINE GET (JDF)
C IN ORDER TO FORMAT THE DATABASE BUFFER, THIS ROUTINE PUTS THE INPUT
C VARIABLES INTO AN ARRAY.
C
COMMON REST(48), TAB(20, 25, 3), CTEST (3), CTOH (3), XMIS(2), ISUBS(48)
& IPAC(48), HT(48), IRB(49), TDRT(48)
& IPD(48), ITERH(48), QTY(48), CHAT(48)
& RBR(48), RESS(48), RESS(48), ITYPE(48)
& PRED(48), TDUE(48), DEC(3), Y(8)
& IPAC(48), RESC(48), YF(48), RESA(48)
C                                                                                          **********
C SUBROUTINE SELECT (JDP)
C THE MENU GENERATES PROVIDES BOTH A TABLE TO VERIFY THE CURRENT VALUES
C OF INPUT VARIABLES, AND ENABLES CHANGE OF THE DATA BY EITHER TOTAL
C LIST OF DATA, BY PAGE OF DATA, OR BY INDIVIDUAL VARIABLE. THE DATA
C INPUTS ARE ALSO CHECKED FOR VALIDITY, WITHIN DESIGN RANGE; AND, ASSISTS
C UPDATES BY PROMPTING RELATED GROUPS OF DATA AND BY SPECIFYING GAME
C RANGES.
C
C COMMON REST(48) ,TAB(20,25,3) ,CTEST(3),CTOH(3) ,XNIS(2) ,ISUBS(48)
C          IFAC(48) , MT48 , IBC(49) , IDDT(48)
C          IFD(48) , ITERM(48) , QTY(48) , CHAT(48)
C          RES(48) , RESU(48) , RESB(48) , ITYPE(48)
C          PEC(48) , TCRE(48) , DEC(3) , Y(8)
C          IPIC(48) , RESC(48) , V(48) , RESA(48)
C          TSTAR(48) , CL(6,5) , ALN(48) , CI(48)
C          TFIM(48) , THIN(48) , IQ(48) , CUN(30)
C C BM CN RESTNM(30) , RESMAX(30) , AK(30) , ISW(3)
C          TAB2(2) , MN(8) , N(8) , INDEX(8)
C          RNTO(6) , AN(9) , ADJ(2,8) , PKR(6)
C          PRR(6) , PVR(6) , CD , ICB , INA , INF , IDE
C C BM CN CPUMAX , CPUMIN , FCPIX , X , ITEMP(25) , COV
C          C10 , CQ(18)
C          TACT(46,2) , IPC(48,5)
C          BDS(48,2) , JINN , IC , IFIG , KDP , M6
C C BM CN MC , IC , FIG , AP , KNP , ITERM
C C COMMON RAM , PRSN , MAX , MIN , EMIN , FRSIM , CRNAX , CMDIN ,
C EFCDAX , TDHAX , TEMIN , TDHIN , QT(25) , YC(6) , CTG(8) , DD(32) , HIS(9)
C
C COMMON STORED / STAT(5) , NDUMP (5,5) , ADUMP (5,28) , DFCOST , MAXDE
C COMMON /PAGES / IPG(4)
REAL*8 STAT
INTEGER ANS, IN/5/, E/1",E/", YES/1",ANO/1"

CALL GET(JDP)
CALL FETCH(CLRSCRN *)
CALL SETUP(KDF)

PRINT 14, ITEM, MAXLP, STAT(MAXDP)

PRINT 15, "**********ITEM"’, ’I2,’ IS IN DP: ’’I1,’ I1. I8,’. "**********

* KDF. 14 *, NDUMP(KDP. 5); ADUMP(KDP, 1),
* ADUMP(KDP, 16),
* PRINT 30, ADUMP(KDP, 3); ADUMP(KDP, 17),
* ADUMP(KDP, 20). ADUMP(KDP, 1),
* ADUMP(KDP, 6), ADUMP(KDP, 21),
* ADUMP(KDP, 8), ADUMP(KDP, 22),
* ADUMP(KDP, 9), ADUMP(KDP, 23),
* ADUMP(KDP, 10), ADUMP(KDP, 24),
* ADUMP(KDP, 11), ADUMP(KDP, 25),
* ADUMP(KDP, 12), ADUMP(KDP, 27),
* IF (FLAG, EO, 2), RETURN

FORMAT( 1 NO. *ITEM**: "CURRENT VALUE @ DP: ’’I1, 4X,’ NO.. *ITEM********
********CURRENT VALUE: ’’I1,16.
* GUIDANCE QUAL TESTS: '”F6. 2/’’ 2. GUIDANCE APPROACH: ’’I1.
* F6. 2, ’’ 3. GUIDANCE CONFIG: ’’F6. 2. ’’ 4. MAINT
* MAXIMUM COST: ’’F6. 2. ’’ 5. MINIMUM COST: ’’F6. 2.
* MAX COST INCENTIVE: ’’F6. 2.
* 6. PARALLEL DEVELOP M$: ’’F6. 2, 4X,
* 7. LATEST WEEK: ’”F6. 2.
* 8. AIRFRAME RELIABILITY: ’”F6. 2, 4X,
* 11. "F6. 2, 4X,
* 12. "F6. 2, 4X,
* 13. "F6. 2, 4X,
* 14. "F6. 2, 4X,
CHANGE FORMAT SELECTION

PRINT 40

PRODUCT(15X,

SELECT AN ITEM WHICH NEEDS TO BE CHANGED FOR DP I1

GOTO 130

CONTINUE

DO 4025 I=1,4

CALL FRTCM('CLRSCRN')

PRINT 4015

FORMAT('*****TYPE IN A LIST OF PAGES YOU DESIRE TO CHANGE:*****/ 1M EX: 1234 WILL ASK FOR NEW DATA ON ALL PAGES; CR, 23 WILL ASK FOR 2M NEW DATA ON PAGES 2 AND 3. **WARNING**** YOU MUST ENTER 3\ AT LEAST ONE LIGHT EVEN IF IT IS A 0.*/

READ(IN 4020, ERR=41, END=41) (IPG(I), I=1,4)

FORMAT(411)

CALL PAGEIN

GOTO 10

RETURN TO MAIN PROGRAM PROCESSING

CONTINUE
DO A READ IN,I, ERR=133, END=133

IF (ANS.EQ.0) RETURN
IF (ANS.GE.1.OR.ANS.LT.32) GO TO 160
CALL EXITS

C DATA INPUT PROCESSING

CONTINUE

IANS=ANS
IF (KDP.GE.4.AND. (ANS.EQ.1.OR.ANS.EQ.2))
CALL PRINTS ('CLRSRN')
IF (KDP.GE.4.AND. (ANS.EQ.1.OR.ANS.EQ.2))
PRINT 162

FORMAT(' AT DP-4, CONTRACTOR AND GUIDANCE APPROACH ARE FROZEN.'
*,24(/))
IF (KDP.GE.4.AND. (ANS.EQ.1.OR.ANS.EQ.2))
GOTO 10
IF (ANS.LT.7.AND.KDP.GT.4) GOTO (160, 160, 401, 500, 600, 701), IANS
GOTO (200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200
*,1300, 1400, 1500, 1600, 1700, 1800, 1900, 2000, 2100, 2200, 2300, 2400, 2500
*,2600, 2700, 2800, 2900, 3000, 3100, 3200), IANS
CONTINUE

GOTO 10

CALL CHANGI(NC)
IF ((NC.EQ.1).OR.(NC.EQ.2)) GOTO 10
PRINT 230, NC
FORMAT (' CONTRACTOR = ' , I2, ' IS INVALID; PLEASE RETYPE CONTRACTOR'
* NUMBER; EITHER '1' OR '2'.')
GO TO 200

CALL CHANGI (ICFG)
IF ((ICFG.EQ.1).OR.(ICFG.EQ.2)) GOTO 10
PRINT 310, ICFG
FORMAT (' GUIDANCE APPROACH = ' , I2, ' IS INVALID; PLEASE RETYPE GUI'
* DANCE APPROACH; EITHER '1' OR '2'.')
GO TO 300
C 400 CALL CHANGE(HIS(6))
C 401 IF (HIS(EQ. 3).EQ. CTC 405
IF (ADDO.KRP.5).GT.0.) CTC(5) = 0.
IF (HIS(6).EQ. 2).CR. HIS(6).EQ. 4) GOTO 405
PRINT 131
131 FORMAT(' AT DP=4 FINAL GUIDANCE CONFIGURATION MUST BE SELECTED.
* / *PLEASE SELECT EITHER 2 OR 4 FOR GUIDANCE CONFIGURATION.*)
GOTO 400
405 IF ( (HIS(6).GE.1).AND. (HIS(6).LE.4) ) GOTO 701
PRINT 410, HIS(6)
410 FORMAT (' GUIDANCE CONFIGURATION =',I2,' IS INVALID; RETYPE GUIDANCE CONFIGURATION =', '1','2','3','4',')
GOTO 400
C 500 CALL CHANGE(CTC(8))
505 IF (CTC(8).GE.XC8) GOTO 10
PRINT 510
510 FORMAT ( 'H0. 'MAINT').
PRINT 515, CTC(8), XC8
515 FORMAT ('H+', 'ENG FUNDS OF $', P5.2, ' HAVE BEEN RESET TO THE $', P5.2', MINIMUM.')
CTC(8) = XC8
PRINT 525
525 FORMAT ('H+', 'MAINT ENG FUNDS = $', P5.2)
GOTO 500
C 600 CALL CHANGE(CTC(7))
605 IF (CTC(7).GE.XC7) GOTO 10
PRINT 610
610 FORMAT ( 'H0. 'VALUE').
PRINT 515, CTC(7), XC7
CTC(7) = XC7
PRINT 620, CTC(7)
620 FORMAT ('H+', VALUE ENG FUNDS = $', P5.2)
GOTO 600
C 700 CALL CHANGE(CTC(5))
701 IF (REF. EQ. 3) GOTO 704
CTC(5) = 0
PRINT 703
703 FORMAT(' PARALLEL DEVELOPMENT FUNDS ARE NOT AVAILABLE AFTER D2-3',
* ' DO NOT ATTEMPT TO ADD PARALLEL DEVELOPMENT FUNDS; AND/',
* ' DO NOT ATTEMPT TO CHANGE GUIDANCE CONFIGURATION TO PARALLEL CONFIGURATIONS. ')
PRINT 702
702 FORMAT(20(A))
IF (HIS(6).EQ.1) .OR. (HIS(6).EQ.3) GOTO 401
GOTO 10
704 IF ((HIS(6).EQ.1) .OR. (HIS(6).EQ.3)) GOTO 750
IF (CTC(5).LE.0.0) GOTO 10
PRINT 705
705 FORMAT (1HO, 'PARALLEL DEV FUNDS')
PRINT 710, CTC(5)
710 FORMAT (1G5.2, F5.2, 'M')
PRINT 720
720 FORMAT (1H+, ' ARE NOT REQUIRED AND HAVE BEEN RESET TO $0.0')
CTC(5) = 0.0
PRINT 725
725 FORMAT(1H, 'DO YOU WISH TO CHANGE GUIDANCE CONFIGURATION: "Y"=YES O
*B "N"=NO? ')
READ (IN, 730, ERR=704, END=704) ANS
730 FORMAT(A1)
IF (ANS.EQ.1) CALL EXITS
IF (ANS.EQ.A) GOTO 10
GO TO 400
C
750 IF (CTC(5).GT.0.0) GOTO 10
PRINT 705
PRINT 755
755 FORMAT (1H+, ' ARE REQUIRED. ')
PRINT 760
760 FORMAT(1H, 'DO YOU WISH TO CHANGE GUIDANCE CONFIGURATION: "Y"=YES C
*B "N"=NO? ')
READ (IN, 765, ERR=750, END=750) ANS
765 FORMAT(A1)
IF (ANS.EQ.E) CALL EXITS
IF (ANS.EQ.ANC) GO TO 700
  GO TO 400

800  CALL CHANGE(YC(1))
     IF(YC(1).GE.0. AND.YC (1).LE.100.0) GO TO 10
     PRINT 805,YC(1)
805  FORMAT(6 RELIABILITY EXCEEDED 100%; RELIABILITY='F3.1,'.'.'/' PLEASE ENTER RELIABILITY.')
     GO TO 800

900  CALL CHANGE(YC(2))
     IF(YC(2).GE.0. AND.YC (2).LE.100.0) GO TO 10
     PRINT 905,YC(2)
905  GO TO 900

1000 CALL CHANGE(YC(3))
     IF(YC(3).GE.0. AND.YC (3).LE.100.0) GO TO 10
     PRINT 1005,YC(3)
1005  GO TO 1000

1100 CALL CHANGE(YC(4))
     IF(YC(4).GE.0. AND.YC (4).LE.1000.0) GO TO 10
     PRINT 1105,YC(4)
1105  FORMAT(6 ERROR EXCEEDED 1000YDS; ERROR='F3.1,'.'.'/' PLEASE ENTER ERROR.')
     GO TO 1100

1200 CALL CHANGE(YC(5))
     IF(YC(5).GE.0. AND.YC (5).LE.1000.0) GO TO 10
     PRINT 1205,YC(5)
1205  GO TO 1200

1300 CALL CHANGE(QT(2))
     IF(QT(2).GE.20. AND.QT (2).LE.40.) GO TO 10
     PRINT 1305,QT(2)
1305  FORMAT('X: NOTICE QUALIFICATION TESTS MUST BE GREATER THAN 20 AND LESS THAN 40. YCOR QT(2) =','F2.0')
     GO TO 1300

1400 CALL CHANGE(QT(3))
IF QT (3) .GE. 3 .AND. QT (3) .LE. 9.) GO TO 10
PRINT 1405 QT (3)
1405 FORMAT (X, 'TIMEFRAME QUALIFICATION TESTS MUST BE GREATER THAN 3 AND
* LESS THAN 9. YOUR TESTS =',F1.0)
GO TO 1400
C
CALL CHANGE (QT (6))
IF QT (6) .GE. 2 .AND. QT (6) .LE. 6.) GO TO 10
PRINT 1505 QT (6)
1505 FORMAT (X, 'LAUNCHER QUALIFICATION TESTS MUST BE GREATER THAN 2 AND
* LESS THAN 6. YOUR TESTS =',F1.0)
GO TO 1500
C
CALL CHANGE (QT (4))
IF QT (4) .GE. 2 .AND. QT (4) .LE. 4.) GO TO 10
PRINT 1615 QT (4)
1615 FORMAT (X, 'FIRE CONTROL QUALIFICATION TESTS MUST BE GREATER THAN 2
* AND LESS THAN 4. YOUR TESTS =',F1.0)
GO TO 1600
C
CALL CHANGE (QT (5))
IF QT (5) .GE. 3 .AND. QT (5) .LE. 9.) GO TO 10
PRINT 1725 QT (5)
1725 FORMAT (X, 'GUIDANCE QUALIFICATION TESTS MUST BE GREATER THAN 3 AND
* LESS THAN 9. YOUR TESTS =',F1.0)
GO TO 1700
C
CALL CHANGE (QT (1))
IF QT (1) .GE. 10 .AND. QT (1) .LE. 25.) GO TO 10
PRINT 1835 QT (1)
1835 FORMAT (X, 'FLIGHT TESTS MUST BE GREATER THAN 10 AND LESS THAN 25.
* YOUR TESTS =',F2.0)
GO TO 1800
C
CALL CHANGE (CDMAX)
PRINT 1910
1910 FORMAT ('DO YOU DESIRE TO CHANGE MINIMUM DEVELOPMENT COST, ALSO.')
READ (IN 2520, ERR=1900 IND=1900) ANS
IF (ANS .EQ. 'A') CALL EXIT5
IF (ANS .EQ. 'A') GO TO 10
CALL CHANGE(CD2MIN)
IF(CD2MAX.GE.CD2MIN)G0TC 10
PRINT 1995.CD2MAX.CD2MIN
FORMAT(1X'HAP IS LESS THAN MIN COST. MAX COST = ',F4.1,
* ',MIN COST = ',F4.1)
GO TO 1900
C
CALL CHANGE(FC2MIN)
G20 10
G
CALL CHANGE(TD2MAX)
PRINT 2210
FORMAT(1X'DO YOU DESIRE TO CHANGE MINIMUM FLIGHT TEST COMPLETION, A
*LSC.?)
READ(IN,2520,ERR=2200)AN
IF(ANS.EQ.E)CALL EXITS
IF(ANS.EQ.ANO)G0TC 10
C
CALL CHANGE(TD2MIN)
PRINT 2216
FORMAT(1X'FLIGHT TEST LATEST WEEK IS LESS THAN EARLIEST WEEK:/
* LATEST DATE = ',F6.0,': EARLIEST DATE = ',F6.0)
GO TO 2200
C
CALL CHANGE(PTD2MIN)
G0TC 10
C
CALL CHANGE(RS2MAX)
PRINT 2510
FORMAT(1X'DO YOU DESIRE TO CHANGE MINIMUM RELIABILITY, ALSO.?)
READ(IN,2520,ERR=2500,END=2500)AN
2520 FORMAT(1X)
IF(ANS.EQ.E)CALL EXITS
IF(ANS.EQ.ANO)G0TC 10
C
CALL CHANGE(RS2MIN)
PRINT 2505
FORMAT(1X'MAXIMUM SYSTEM RELIABILITY IS LESS THAN /,5X,
*MINIMUM RELIABILITY: RS2MAX = ',F6.0,': RS2MIN = ',F6.0)
G0TC 2500
**January 18, 1978**

```
C 2700 CALL CHANGE(FRESHMAX)
    GOTO 10
C 2800 CALL CHANGE(ESMAX)
    GOTO 10
C 2900 CALL CHANGE(ESMIN)
    GOTO 10
C 3000 CALL CHANGE(FESMIN)
    GOTO 10
C 3100 CALL CHANGE(DD(13))
    GOTO 10
C 3200 RETURN
END
```

************************************************************
SUBROUTINE SETUP (ODP)
COMMON REST(48),TAB(20,25,3),CTEST(3),CTOR(3),XMIS(2),ISUBS(48)
    IFAC(48),HT(48),IRC(49),IDDT(48)
COMMON RES(48),RES(48),RESB(48),ITIP(48)
    IPBC(48),TDUE(48),DEC(3),Y(8)
 COMMON TSTAR(48), CL(6,5),ALN(48), CT(48)
    TPIN(48),TIM(48),IQ(48),CON(30)
COMMRCN RESMIN(30), RESMAX(30), A(30), ISW(3)
    ITAB(25), MN(8), N(8), IND(8)
    RATIO(6), AM(9), ADJ(2,8), PDB(6)
COMMON CPUMAX,CPUMIN,FCCHAIN,X,ITEMP(25),COV
    CT(10), CO(18)
COMMON IACT(48,2), IPC(48),5
    TDS(48,2), JHIN, IC, IFIG, KDP, MR
COMMON HC,IFIG, KP, KEP, KPP, ITEAM
COMMON RMAX(8),ESMIN,ESMAX,ESMIN,FESMIN,FEMAX,FEMIN,
    FCCHAIN,TMAX,TMIN,PTDMIN,QT(25),YC(6),CCT(8),DD(32),MIS(9)
C
COMMON/STORED/ STAT(5), NDUMP(5,5), ADUMP(5,28), DPCGS1, MAXDP

REAL*8 STAT
INTEGER DAFP(5), QDP
NDUMP(ODP,1) = ITIM
NDUMP(ODP,2) = OCP
NDUMP(ODP,3) = NC
NDUMP(ODP,4) = ICFG
NDUMP(ODP,5) = MIS(6)
ADUMP(ODP,1) = TIC(9)
ADUMP(ODP,2) = TIC(5)
ADUMP(ODP,3) = TIC(1)
ADUMP(ODP,4) = TIC(2)
ADUMP(ODP,5) = TIC(3)
ADUMP(ODP,6) = TIC(2)
ADUMP(ODP,7) = TIC(3)
ADUMP(ODP,8) = TIC(3)
ADUMP(ODP,9) = TIC(2)
ADUMP(ODP,10) = TIC(1)
ADUMP(ODP,11) = TIC(6)
ADUMP(ODP,12) = TIC(5)
ADUMP(ODP,13) = TIC(5)
ADUMP(ODP,14) = TIC(1)
ADUMP(ODP,15) = CMAX
ADUMP(ODP,16) = CMIN
ADUMP(ODP,17) = ECMIN
ADUMP(ODP,18) = TMAX
ADUMP(ODP,19) = TMIN
ADUMP(ODP,20) = TMIN
ADUMP(ODP,21) = ESMA
ADUMP(ODP,22) = ESMA
ADUMP(ODP,23) = ESMA
ADUMP(ODP,24) = ESMA
ADUMP(ODP,25) = ESMA
ADUMP(ODP,26) = ESMA
ADUMP(ODP,27) = TD(13)
RETURN
END

C ******************************************************
C SUBROUTINE CHANGE(XXX)
C
INTEGER IN/5/
CALL ERSET(218,1,-1,1,1)

120 FORMAT(IX,'PREVIOUS VALUE',I6,2,'INPUT A VALUE')
READ(IN,*,ERR=150,END=150) IX
RETURN
150 CALL EXITS
    GO TO 120
END

******************************************************************************

C SUBROUTINE CHANGE(IXX)

C INTEGER IN/5/
CALL ERSET(218,1,-1,1,1)

120 FORMAT(IX,'PREVIOUS VALUE',I3,'INPUT A VALUE')
READ(IN,*,ERR=150,END=150) IX
RETURN
150 CALL EXITS
    GO TO 120
END

******************************************************************************

C SUBROUTINE TEAMIN(ITEM,MR,DATFIL)

C INTEGER ANS,ANC/'N'/YES,'Y'/,E/'E'/,IN/5/,T/'M'/,DATFIL,IFAILT
INTEGER DATC0D,11/1,'P'/,ANT
COMMON/STOED/ STAT(5),NDUMP(5,5),ADUMP(5,28),DPCCSI,MAXDP
COMMON/ICDE,FLAG,IEM,STORFL
REAL*8 STAT,USERID,ITEM,FLAG,PL21,'TEAM21,/
REAL*8 ICDE,INCOD,BLK/
REAL*8 TEAM01,'TEAM02,'TEAM03,'TEAM04,
TEAM10,'TEAM11,'TEAM12,'TEAM13,'TEAM14,'TEAM15,'TEAM16,'TEAM17
* TEAM18,'TEAM19,'TEAM20,'TEAM21,
CALL ERSET(218,1,-1,1,1)
ISEN=0
CALL PLOTST(ISRN)
C
C
C
ITEAMB=999
IF(FLAG.EQ.FL21.OR.NR.NE.0)GOTO 21
READ(5,*1ITEM
READ(5,*95)FLAG
REWRITE 9
READ(9,16,ERR=17,END=17)ITEAMB
CONTINUE
16
17
IF(ITEM.EQ.21)GOTO 475
IF(FLAG.EQ.FL21)GOTO 475
GOTO 32
475
476
ITEM=BLK
PRINT 476
REWRITE IN
READ(IN,95,ERR=450,END=450)INCOD
REWRITE DATCOD
READ(DATCOD,95,ERR=450,END=450)USERID
READ(DATCOD,95,ERR=450,END=450)ICODE
IF(INCOD.NE.ICODE)GOTO 450
PRINT 28
21
20
FORMAT('SELECT WHICH INSTRUCTORS MODE YOU DESIRE...1. RUN TEAM
* SCENARIOS WHICH ARE ON THE MONITOR'S DISK, 2. CHANGE TEAM F
ILES...8X,THERE MODE WILL ALTER THE STUDENT'S FILES...')
PRINT 23
23
FORMAT('3. RUN THE MONITOR'S TEAM FILE "TEAM21", 4. EXIT
* T')
READ(IN,*,ERR=450,END=450)ANS
IF(ANS.EQ.1)NR=1
IF(ANS.EQ.2)NR=2
IF(ANS.NE.3)GOTO 22
NR=3
ITEAMB=21
CALL FTCMS('FILEDEF',9,'DISK',DATAFILE,FL21
RETURN
IF(ANS.NE.4)GOTO 877
IF(MR.NE.2)STOP
PRINT 27,ITEM
FORMAT(1X, 'SHOULD THE REWRITTEN TEAM? ',I2, ' FILE BE SENT TO THE PR
*ING DISK: Y/N ?')
READ(IN,28,ERR=22,END=22)ANT
FORMAT(A1)
IF(ANT.NE.YES)GOTO 877
CALL SETUP(KDP)
CALL STORE
CALL FRTCHS('CF 'SPOOL ', 'PUN ', 'CONT ', 'CL '
CALL FRTCHS('RC 'O276P 'DATAPILE ', 'ITEM)
CALL FRTCHS('PUN ', 'SPOOL ', 'PUN ', 'CLOSE ', 'NOCONT '
)STOP
CONTINUE
IF(ITEM.EQ.BLK)GOTOC 32
CONTINUE
IF(ITEM.EQ.21)GOTO 31
IF(MR.EQ.1.AND.ITEMB.EQ.ITEM)RETURN
IF(MR.EQ.2.AND.ITEMB.EQ.ITEM)RETURN
CONTINUE
IF(ITEMB.NE.999.AND.ITEMB.NE.ITEM)PRINT 24,ITEM
FORMAT(1X, 'RECORDS DO NOT EXIST FOR TEAM ',I2)
PRINT 25
FORMAT(1X, 'WHAT TEAM NUMBER?')
READ(IN,*)ERR=495,END=495)ITEM
CONTINUE
IF(ITEM.LE.0.OR.ITEM.GT.21)GOTO 485
IF(ITEM.EQ.1)ITEM=TEAM01
IF(ITEM.EQ.2)ITEM=TEAM02
IF(ITEM.EQ.3)ITEM=TEAM03
IF(ITEM.EQ.4)ITEM=TEAM04
IF(ITEM.EQ.5)ITEM=TEAM05
IF(ITEM.EQ.6)ITEM=TEAM06
IF(ITEM.EQ.7)ITEM=TEAM07
IF(ITEM.EQ.8)ITEM=TEAM08
IF(ITEM.EQ.9)ITEM=TEAM09
IF(ITEM.EQ.10)ITEM=TEAM10
IF (ITEM='EQ.11) ITEM=ITEM+1
IF (ITEM='EQ.12) ITEM=ITEM+2
IF (ITEM='EQ.13) ITEM=ITEM+3
IF (ITEM='EQ.14) ITEM=ITEM+4
IF (ITEM='EQ.15) ITEM=ITEM+5
IF (ITEM='EQ.16) ITEM=ITEM+6
IF (ITEM='EQ.17) ITEM=ITEM+7
IF (ITEM='EQ.18) ITEM=ITEM+8
IF (ITEM='EQ.19) ITEM=ITEM+9
IF (ITEM='EQ.20) ITEM=ITEM+10
IF (ITEM='EQ.21) ITEM=ITEM+11

CALL FRMCMS('FILEDEF',9,'DISK','DATAFILE',ITEM)
CALL FRMCMS('FILEDEF',11,'DISK','DATAFILE',ITEM)

IF (MR='EQ.0) GOTO 15
READ(9,16,ERR=485,END=485) ITEM
GOTO 485

CONTINUE

READ(9,16,ERR=30,END=30) ITEM
READ DATCOD
REIND 9
READ(DATCOD,95,ERR=450,END=450) ICODE
READ(DATCOD,230,ERR=450,END=450) IFAULT
IF (ICODE='EQ. BLK.OR.ICOD=EQ.ZEO) GOTO 100
IF (FAULT.GE.5) GOTO 65
GOTO 70
PRINT 110

**YOU ARE ENCOURAGED TO CHANGE YOUR SECURITY CODE?***

PRINT 130
PRINT NEW CODE

READ(IN,95,ERR=120,END=120) ICODE
IF (ICODE='EQ.BLK.OR.ICOD=EQ.ZERO) GOTO 120

CALL EXITS
GOTO 100

PRINT 60
FORMAT('PLEASE ENTER YOUR TEAM SECURITY CODE.')
READ(IN, 95, ERR=15, END=15) INCOD
IF(INCOD.NE.ICODE) GO TO 200
RETURN
95 IF(FLAG.NE.FL21) WRITE(DATCOD, 95) ICODE
FORMAT (A8)
RETURN
99 IF(FLAG.NE.FL21) WRITE(DATCOD, 230) IFAULT
IPROOF=FAULT+1
210 FORMAT('YOU HAVE ATTEMPTED TO ENTER A TEAM FILE WITHOUT THE PROPER
SECURITY CODE.')
RE WIND DATCOD
IF(FLAG.NE.FL21) WRITE(DATCOD, 95) ICODE
IF(FLAG.NE.FL21) WRITE(DATCOD, 230) IFAULT
230 FORMAT(T3)
IF(IF AAULT.LT.5) CALL EXITS
IF(IFVAULT.LT.5) GOTO 70
CONTINUE
310 FORMAT('THE USER ID CODE FOR TEAM:*12:*HAS BEEN TRIED INCORRECTLY
MORE THAN '*13:' TIMES.'/"'PLEASE LOGOFF AND CONTACT YOUR MC
ITOR IN ORDER TO REGAIN ACCESS TO YOUR TEAM*'S FILES.')
CONTINUE
450 CONTINUE
PRINT 455
455 FORMAT('YOUR INPUT WAS READ INCORRECTLY.'/
"PLEASE REINITIATE THE PROGRAM TO CORRECT THE ERROR.')
IF(FLAG.NE.FL21) PRINT 456
456 FORMAT('IF YOU HAVE FURTHER QUESTIONS PLEASE SEE YOUR CLASS*'S
ITOR.'/;PRESS <ENTER>
CONTINUE
STOP
END
C**********************************************************************
C SUBLTUTINE PAGEIN
C THIS SUBLTUTINE EMALES MASS PAGE DATA INPUT.
CALL ERRSET(216,1,-1,1,1)
CALL FBTCMS('CLESCBN')
DO 11 INV=1,6
10 IF(IFG(INV).EQ.1) GOTO 15
11 CONTINUE
GOTO 360
600 CALL EXITS
CONTINUE
20 PRINT 20,KDP
20 FORMAT(1X,'*****INPUT PAGE 1 DP-.II.' INFO*****')
55 PRINT 55
55 FORMAT('CONTRACTOR NUMBER, (1 OR 2):')
READ (IN),ERR=600,END=600) NC
PRINT 700
700 FORMAT('GUIDANCE APPROACH, (1 OR 2):')
READ (IN,*,ERR=600,END=600) ICOFG
PRINT 705
705 FORMAT(' GUIDANCE CONFIGURATION 1-4, NOT 1 OR 3 AFTER DE-3:')
READ (IN,*,ERR=600,END=600) MIS(6)
PRINT 710
710 FORMAT(' MAINTAINABILITY ENGINEERING FUNDS IN $M:')
READ (IN,*,ERR=600,END=600) CIC(8)
715 FORMAT(' VALUE ENGINEERING FUNDS IN $M:')
IF (MIS(6) .EQ. 2) OR (MIS(6) .EQ. 4) GO TO 80
PRINT 715
720 FORMAT(' PARALLEL GUIDANCE FUNDS IN $M:')
READ (IN,*,ERR=600,END=600) CIC(7)
80 IF ((NC.EQ.1) .OR. (NC.EQ.2)) GO TO 140
PRINT 491
90 PRINT 100, NC
100 FORMAT('/',' CONTRACTOR=','I2',' IS INVALID.', '/',' PLEAS REENTER CONTRACTOR NUMBER; EITHER "1" OR "2".')
READ (IN,*,ERR=130,END=130) NC
GO TO 80
CALL EXITS
130 GO TO 80
C
140 IF ((ICOGF.EQ.1) .OR. (ICOGF.EQ.2)) GOTO 160
PRINT 491
145 PRINT 145, ICOFG
C
145 FORMAT('/',' GUIDANCE APPROACH=','I2',' IS INVALID.', '/',' PLEAS REENTER GUIDANCE APPROACH; "1" OR "2".')
READ (IN,*,ERR=150,END=150) ICOFG
GO TO 160
150 CALL EXITS
GOTO 140
C
160 IF ((MIS(6).GE.1) .AND. (MIS(6).LE.4)) GOTO 180
PRINT 491
165 PRINT 165, MIS(6)
165 FORMAT('/',' GUIDANCE CONFIGURATION=','I2',' IS INVALID.', '/',' RETYPE GUIDANCE CONFIGURATION; "1","2","3"," OR "4".')
170 READ (IN,*,ERR=175,END=175) MIS(6)
GO TO 160

CALL ENTS

GO TO 160

C

180 IC8 = 1.5
IF (CTC(8) .GE. IC8) GOTO 205
PRINT 491
PRINT 490
185 FCNAM (1H0, 'MAINT')
PRINT 190, CTC(8), IC8
190 FCNAM (1H+, 'ENG FUNDS OF $', P5,2, ' MINIMUM.')
CTC(8) = IC8
CALL CHANGE(CTC(8))
GO TO 180

C

205 IC7 = 0.5
IF (CTC(7) .GE. IC7) GOTO 230
PRINT 491
PRINT 490
210 FCNAM (1H0, 'VALUE')
PRINT 190, CTC(7), IC7
CTC(7) = IC7
PRINT 220, CTC(7)
220 FCNAM (1H+, 'VALUE ENG FUNDS = $', P5,2)
CALL CHANGE(CTC(7))
GO TO 205

C

230 IF (H1C =(E), EQ.1) .OR. (NIS(6) .EQ. 3) GOTO 300
235 IF (CTC(7) .LE. 0.0) GOTO 360
PRINT 491
255 PRINT 240
240 FCNAM (1H0, 'PARALLEL DEV FUNDS')
PRINT 245, CTC(5)
245 FCNAM (1H+, 'OF $', P5,2)
PRINT 250
250 FCNAM (1H+, 'ARE NOT REQUIRED, AND HAVE BEEN RESET TO $0.0' )
CTC(5) = 0.0
PRINT 260
260 FCNAM (1X, 'DO YOU WISH TO CHANGE GUIDANCE CONFIGURATION: "Y"=YES G
*R "N=NO?"
READ(IN,54,ERR=255,END=255)ANS
54 FORMAT(A1)
   IF(ANS.EQ.E)CALL EXITS
   IF(ANS.EQ.ANC)GO TO 360
267 PRINT 270
270 FORMAT('PLEASE RETYPE GUIDANCE CONFIGURATION; "1","2","3" OR "4"
*D")
   CALL CHANGE(MIS(6))
   GOTO 160
C
300 IF (CTC(5).GT.0.0) GO TO 360
310 PRINT 491
320 PRINT 320
320 FORMAT('1H+, ' ABE REQUIRED.
)')
335 PRINT 335
335 FORMAT('DO YOU WISH TO CHANGE GUIDANCE CONFIGURATION: "Y"=YES O
*D "N=NO?"
READ(IN,54,ERR=300,END=300)ANS
   IF(ANS.EQ.E)CALL EXITS
   IF(ANS.EQ.ANC)GOTO 280
   CALL CHANGE(MIS(6))
   GOTO 160
280 PRINT 290
290 FORMAT('INPUT PARALLEL DEVELOPMENT FUNDS; EX: $5,000,000- 5 ')
READ(IN,*,ERR=300,END=300)CTC(5)
GC 1C 160
C
360 CALL ERTCMS('CLSCRN ')
DO 361 INV=1,4
   IF (IPG(INV).EQ.2) GOTO 362
361 CONTINUE
   GOTO 445
610 CONTINUE
   CALL EXITS
610 CALL CONTINUE
610 PRINT 365,KDP
365 FORMAT('*****INPUT PAGE 2 DP-','I1',' INFO.*****')
365 PRINT 725
725 FORMAT('MOTOR RELIABILITY.')
<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>730</td>
<td>READ (IN *, ERR=610, END=610) YC (1)</td>
</tr>
<tr>
<td>730</td>
<td>PRINT 730</td>
</tr>
<tr>
<td>735</td>
<td>FORMAT (' AIRFRAME RELIABILITY.' )</td>
</tr>
<tr>
<td>735</td>
<td>READ (IN *, ERR=610, END=610) YC (2)</td>
</tr>
<tr>
<td>735</td>
<td>PRINT 735</td>
</tr>
<tr>
<td>740</td>
<td>FORMAT (' LAUNCHER/GSE RELIABILITY.' )</td>
</tr>
<tr>
<td>740</td>
<td>READ (IN *, ERR=610, END=610) YC (3)</td>
</tr>
<tr>
<td>740</td>
<td>PRINT 740</td>
</tr>
<tr>
<td>745</td>
<td>FORMAT (' FIRE CONTROL ACCURACY.' )</td>
</tr>
<tr>
<td>745</td>
<td>READ (IN *, ERR=610, END=610) YC (4)</td>
</tr>
<tr>
<td>745</td>
<td>PRINT 745</td>
</tr>
<tr>
<td>750</td>
<td>FORMAT (' GUIDANCE ACCURACY.' )</td>
</tr>
<tr>
<td>750</td>
<td>READ (IN *, ERR=610, END=610) YC (5)</td>
</tr>
<tr>
<td>750</td>
<td>PRINT 750</td>
</tr>
<tr>
<td>755</td>
<td>FORMAT (' MOTOR QUAL TESTS: 20-40.' )</td>
</tr>
<tr>
<td>755</td>
<td>READ (IN *, ERR=610, END=610) QT (2)</td>
</tr>
<tr>
<td>755</td>
<td>PRINT 755</td>
</tr>
<tr>
<td>760</td>
<td>FORMAT (' AIRFRAME QUAL TESTS: 3-9.' )</td>
</tr>
<tr>
<td>760</td>
<td>READ (IN *, ERR=610, END=610) QT (3)</td>
</tr>
<tr>
<td>760</td>
<td>PRINT 760</td>
</tr>
<tr>
<td>765</td>
<td>FORMAT (' LAUNCHER/GSE QUAL TESTS: 2-6.' )</td>
</tr>
<tr>
<td>765</td>
<td>READ (IN *, ERR=610, END=610) QT (4)</td>
</tr>
<tr>
<td>765</td>
<td>PRINT 765</td>
</tr>
<tr>
<td>770</td>
<td>FORMAT (' FIRE CONTROL QUAL TESTS: 2-4.' )</td>
</tr>
<tr>
<td>770</td>
<td>READ (IN *, ERR=610, END=610) QT (4)</td>
</tr>
<tr>
<td>770</td>
<td>PRINT 770</td>
</tr>
<tr>
<td>775</td>
<td>FORMAT (' GUIDANCE SYSTEM QUAL TESTS: 3-9.' )</td>
</tr>
<tr>
<td>775</td>
<td>READ (IN *, ERR=610, END=610) QT (5)</td>
</tr>
<tr>
<td>775</td>
<td>PRINT 775</td>
</tr>
<tr>
<td>782</td>
<td>IF (QT (2) GE 20. AND. QT (2) LE 40.) GO TO 385</td>
</tr>
<tr>
<td>782</td>
<td>PRINT 782</td>
</tr>
<tr>
<td>783</td>
<td>FORMAT (IX ' NOTICE QUALIFICATION TESTS MUST BE GREATER THAN 20 AND LESS THAN 40.' )</td>
</tr>
<tr>
<td>783</td>
<td>* YOUR QUAL TESTS = *P2.0)</td>
</tr>
<tr>
<td>783</td>
<td>CALL CHANGE (QT (2))</td>
</tr>
<tr>
<td>783</td>
<td>GO TO 382</td>
</tr>
<tr>
<td>382</td>
<td>IF (QT (3) GE 3. AND. QT (3) LE 9.) GO TO 400</td>
</tr>
<tr>
<td>385</td>
<td>PRINT 385</td>
</tr>
</tbody>
</table>
PRINT 390, QT(3)
FORMAT(1X,'AIREFRAME QUALIFICATION TESTS MUST BE GREATER THAN 3 AND
* LESS THAN 9. ',/,' YOUR QUAL TESTS= ',F2.0)
    CALL CHANGE(QT(3))
GO TO 385
400 IF QT(6).GE.2..AND.QT(6).LE.6.) GO TO 410
PRINT 405, QT(6)
405 FORMAT(1X,'LAUNCHER QUALIFICATION TESTS MUST BE GREATER THAN 2 AND
* LESS THAN 6. ',/,' YOUR QUAL TESTS= ',F2.0)
    CALL CHANGE(QT(6))
GO TO 400
410 IF QT(4).GE.2..AND.QT(4).LE.4.) GO TO 420
PRINT 415, QT(4)
415 FORMAT(1X,'FIRE CONTROL QUALIFICATION TESTS MUST BE GREATER THAN 2
* AND LESS THAN 4. ',/,' YOUR QUAL TESTS= ',F2.0)
    CALL CHANGE(QT(4))
GO TO 410
420 IF QT(5).GE.3..AND.QT(5).LE.9.) GO TO 430
PRINT 425, QT(5)
425 FORMAT(1X,'GUIDANCE QUALIFICATION TESTS MUST BE GREATER THAN 3 AND
* LESS THAN 9. ',/,' YOUR QUAL TESTS= ',F2.0)
    CALL CHANGE(QT(5))
GO TO 420
430 IF QT(1).GE.10..AND.QT(1).LE.25.) GO TO 440
PRINT 435, QT(1)
435 FORMAT(1X,'FLIGHT TESTS MUST BE GREATER THAN 10 AND LESS THAN 25. '
* ',/,' YOUR FLIGHT TESTS= ',F2.0)
    CALL CHANGE(QT(1))
GO TO 430
440 CONTINUE
445 CALL CRTCMS('CIESCRN ')
DC 446 INV=1.4
IF(1PG(INV).EQ.3) GOTO 447
446 CONTINUE
GOTO 510
620 CALL EXITS
CALL CHANGE(CDMAX)
497 PCHMAT(IIX,'FOR MINIMUM COST,\n')
CALL CHANGE(CDRIN)
GO TO 480
490 IF(TDMAX .GE. TDMIN)GOTC 500
PRINT 491
491 PCHMAT('*****WARNING******')
PRINT 496 TDMAX, TDMIN
496 FORMAT(IIX,'FLIGHT TEST LATEST WEEK IS LESS THAN EARLIEST WEEK; LATEST=',F6.0,':EARLIEST=',F6.0)
CALL CHANGE(TD MAX)
PRINT 498
498 PCHMAT(IIX,'FOR MINIMUM TIME,\n')
CALL CHANGE(TDIN)
GO TO 490
500 IF(RS MAX .GE. RS MIN)GOTC 510
PRINT 505 RS MAX, RS MIN
505 FORMAT(IIX,'MAXIMUM SYSTEM RELIABILITY IS LESS THAN MINIMUM RELIABILITY\nMAX=',F6.0,':MIN=',F6.0)
PRINT 506
506 FORMAT(IIX,'FOR MAXIMUM RELIABILITY,\n')
CALL CHANGE(RS MAX)
PRINT 507
507 FORMAT(IIX,'FOR MINIMUM RELIABILITY,\n')
CALL CHANGE(RS MIN)
GO TO 500
510 CALL FRTCMS('CINSCREEN')
DO 511 INV=1,4
511 IF(IPG(INV).EQ.4)GOTO 512
CONTINUE
RETURN
630 CALL EXITS
631 CONTINUE
PRINT 520,KDP
520 FORMAT('*****INPUT PAGE 4 DP-,I1, INFOM.*****//)
PRINT 860
860 FORMAT('MAXIMUM IMPACT ERROR EXPECTED.\n')
READ(IN,*,ERR=630,END=630)ESMAX
CONTINUE
PRINT 450 'KDP
450 FORMAT (1.******INPUT PAGE 3 DP=",.1", INFO.******//)
PRINT 800
800 FORMAT ('MAXIMUM COST IN $M.')
READ (IN, * , ERR=620, END=620) CMDMAX
PRINT 805
805 FORMAT ('MINIMUM COST IN $M.')
READ (IN, * , ERR=620, END=620) CMDMIN
810 FORMAT ('INCENTIVE FEE % FOR DEVELOPMENT COST.')
READ (IN, * , ERR=620, END=620) FCDMIN
IF (FCDMIN .LT. 0. OR. FCDMIN .GT. 15.) GOTO 806
PRINT 815
815 FORMAT ('LATEST WEEK FOR FLIGHT TEST COMPLETION.')
READ (IN, * , ERR=620, END=620) TDMAX
PRINT 820
820 FORMAT ('EARLIEST WEEK FOR FLIGHT TEST COMPLETION.')
READ (IN, * , ERR=620, END=620) TDMIN
EFFECT=15.-FCDMIN
825 PRINT 825
825 FORMAT ('INCENTIVE FEE % FOR FLIGHT TEST COMPLETION')
READ (IN, * , ERR=620, END=620) FTDMIN
IF (FTDMIN .LT. 0. OR. FTDMIN .GT. PERTOT) GOTO 821
PRINT 830
830 FORMAT ('MAXIMUM RELIABILITY % EXPECTED.')
READ (IN, * , ERR=620, END=620) RSMAX
PRINT 835
835 FORMAT ('MINIMUM RELIABILITY % EXPECTED.')
READ (IN, * , ERR=620, END=620) RSMIN
EFFECT=15.-CMDMIN-FTDMIN
840 PRINT 840
840 FORMAT ('INCENTIVE FEE % FOR RELIABILITY.')
READ (IN, * , ERR=620, END=620) FRSMAX
IF (FRSMAX .LT. 0. OR. FRSMAX .GT. PERTOT) GOTO 836
480 IF (CMDMAX .GE. CMDMIN) GOTO 490
PRINT 491
491 PRINT 495 'CDMAX, CMDMIN
495 FORMAT (1X,'MAXIMUM COST IS LESS THAN MINIMUM COST: MAX =", F4.1,
*", 'MIN =", F4.1)
PRINT 865
FORMAT('MINIMUM IMPACT ERROR EXPECTED: ',)
READ(IN,*,ERR=630,END=630)ESMIN
EESTOT=1:PCDATA-FTDMIN-YESMAX
PRINT 870
FORMAT('INCENTIVE FEE % FOR IMPACT ACCURACY: ',)
C FORMAT('*****REMAINING FEE PERCENTAGE FROM 15% TOTAL IS ',F4.1)
READ(IN,*,ERR=512,END=512)PESMIN
PRINT 875
FORMAT('INPUT DESIRED WEEK FOR COMPLETION OF DEPLOYMENT: '
IF(ESMAX.GE.ESMIN(GOTO 553)
PRINT 550,ESMAX,ESMIN
FORMAT('*****WARNING*****MAXIMUM IMPACT ERROR IS LESS THAN MINI
*IMUM IMPACT ERROR: ',ESMAX,F6.0),MIN ERROR =',F6.0)
PRINT 552
FORMAT('FOR MINIMUM ERROR, ')
CALL CHANGE(ESMAX)
PRINT 551
FORMAT('FOR MINIMUM ERROR, '
CALL CHANGE(ESMIN)
GO TO 545
553 CALL PRCTCK
RETURN
END
C*****************************************************************************
C SUBROUTINE INPUT2
C
C COMMON REST(48),TAB(20,25,3),CTEST(3),CTOH(3),XNIS(2),ISUBS(48)
C COMMON IPAC(48),INT4G(49),IDDT(49)
C COMMON RESD(48),RESO(48),RESB(48),ITYPE(48)
C COMMON PERC(48),TDUE(48),DEC(3),Y(8)
C COMMON IPAC(48),RESC(48),YF(48),RESA(48)
C COMMON TSPR(48),CL(6,5),ALN(48),CT(48)
C COMMON TPIN(48),TMIN(48),IQ(48),CUB(30)
C COMMON RESMIN(30),ESMAX(30),AK(30),ISW(3)
C COMMON ITAB(25),NN(8),N(8),INDX(8)
C COMMON RATIO(8),AN(9),ADJ(2,9),FDR(6)
& COMM  CPHMA, CPBMA, CPHMIN, CPMAX, I TEMP(25), COV, C10, TRADE, TRADER.
& TRADE, DAA1, DAA2, DEF, SUB, COST, COSTD, COSTP1
& COMON COSTP2, COSTO1, IACT, ALPHA, PRICR
& : TRADE?, TF, WA, IACT(48,2), IPC(48,5)
& : TDS(48,2), JMIN, IC, IPEG, KDP, NA
& : NC ICOFG, KP, KDP, ITEM
& CCOMM KSHAI, KSHMIN, PKSMAI, PKSMIN, KSHMAX, PKSHMIN, CDMAV, CDMAH, CDMAI,
& ECDMAI, TKMAI, TKMAH, FTCMAI, FTMDMIN, OT(25), VC(6), CTC(8), DD(32), HIS(9)
& COMON STORED/ STAT(5), ADUMP(5,5), ADUMF(5,28), DPCOST, MAXDP
REAL*8 STAT
REAL*8 FINA1, FINT, PROPOSED
INTEGER E/1, C/0, C/C, IN/5, ANO/"N", ANS/YES/
DIMENSION MEMO(48,2)
------ 972 MAR 23
JDE=KDP-1
IF(JDP.LT.3)JDE=3
NTF = 17
NTF = 5
------ => DP-4 EDIT* -- --
231
930 IF(CTC(8), GE, ADUMP(JDP,1)) GOTO 940
CTC(8)=ADUMP(JDP,1)
PRINT 935, JDP, JDE, ADUMP(JDP,1), KDP
935 FCMAT(13), MAINTENANCE ENG FUNDS ARE LESS THAN THOSE AT DP-1, I1,
& DO YOU WISH TO RESET DP-1, I1, MAINTENANCE ENGINEERING FUNDS? Y/N
*/*"N?"
READ (IN, 2896, ERR=2888, END=2888) ANS
2896 FORMAT (A1)
IF (ANS.EQ.E) CALL EXITS
IF (ANS.EQ.NO) GOTO 940
IF (ANS.EQ.YES) GOTO 2889
2888 CALL EXITS
GO TO 930
2889 CALL CHANGE(CTC(8))
ADUMP(KDP,1)=CTC(8)
GO TO 930
2889
940 IF (CTC(7), GE. ALUMP(JDP, 2)) GOTO 950
   CTC(7) = ALUMP(JDP, KDP)
   PRINT 1851, KDP, JDP, CTC(7), KDP
1851 FORMAT (5X, 'VALUE ENGINEERING FUNDS ARE LESS THAN DP-1, I1
* 1, I1' 'VALUE ENGINEERING FUNDS HAVE BEEN RESET TO: $ ', F4
* 2, 1X 'DO YOU WISH TO CHANGE DP-1, I1' 'VALUE ENGINEERING FUNDS:
* Y/N' )
   READ (IN, 3896, ERR=3888, END=3888) ANS
3896 FORMAT (1X)
   IF (ANS, 'EQ', 'E') CALL EXITS
   IF (ANS, 'EQ', 'NO') GO TO 950
   IF (ANS, 'EQ', 'YES') GO TO 3889
3888 CALL EXIT5
   GO TO 540
3889 CALL CHANGE(CTC(7))
   GO TO 940

C 550 CALL SETUP(KDP)
   RETURN
   END

C **********************************************************************

C SUBROUTINE INPUT3

C COMMON REST(48), TAB(20, 25, 3), CTEST(3), CTGH(3), XMIS(2), ISUBS(48)
   & CTM(3), CTOH(3), XMIS(2), ISUBS(48)
   & TPE(48), N1(48), IRC(49), CIDT(48)
C COMMON RESH(48), RESU(48), RESR(48), ITYPE(48)
   & TPE(48), TUE(48), DEC(3), Y(8)
C COMMON RESH(48), RESR(48), YF(48), RESA(48)
   & TPE(48), TUE(48), DEC(3), Y(8)
C COMMON RESH(48), RESA(48), RESA(48), CUM(30)
   & RESH(30), RESH(30), AK(30), ISW(3)
C COMMON RTAB(25), NN(8), N(8), INDI(8)
   & RATIO(8), AN(9), ADJ(2, 8), PDR(6)
C COMMON PER(6), PV(6), CD, IC, NMA, INF, IDP
C COMMON CPUMAX, CPUMIN, PCEMIN, I, ITEMP(25), COV, C10, TRACET, TRADER,
   & TRADER, DHA, DHA, DEP SUB, COST, COSTD, COSTP1
C COMMON C0STP2, COSTP3, IACT, ALPHA, PRICR
   & TRADER, TF, IF, IACT(48, 2), IP7(48, 5)
C   & TDS(48, 2), JMIN, IC, IFIG, KDP, N4
& ICOFG, KP, KPP, ITMA
CCMGN RSMH, RSHM, RSHM, RESM, RESM, RESM, CDMAX, CDMIN, 
& FCDMH, TMAX, TDMIN, TTDHM, QT(25), YC(6), CTC(8), DD(32), MTS(9)
DIMENSION MENO(48,2)
MTF = 5
1002 IC = NC
IF (MTMP .EQ. 1) GOTO 1919
IF (MTMP .NE. 0) GOTO 1920
1919 MTMP (2) = 3
IDP = 3
GOTO (21, 24), NC
21 GOTO (22, 23), ICOFG
22 NSTP = 17
23 NSTP = 18
GOTO 27
24 GOTO (25, 26), ICOFG
25 NSTP = 19
26 NSTP = 20
27 CCNITUE
REIND NSTP
DO 20 K = 1, 13
READ (NSTP, 100), L (TAB(K, I, 2) I = 1, 25)
100 FORMAT (5X, I2, 4X, 7F6.2; 1X, 9F6.2; 1X, 9F6.2)
20 CONTINUE
READ (NSTP, 102), (DS(K, I), K = 1, 13)
102 FORMAT (5X, 13F4.0)
READ (NSTP, 103), (ISUB(S(I), I = 1, 48)
103 FORMAT (5X, 7F12/1X, 2612)
READ (NSTP, 105), (ISUB(A(I), I = 1, 48)
105 FORMAT (5X, 14I3 /1X, 17I3 /1X, 17I3)
READ (NSTP, 103), (MT(I), I = 1, 48)
READ (NSTP, 108), (YF(I), I = 1, 48)
108 FORMAT (5X, 9F5.0 /1X, 13F5.0 /1X, 13F5.0)
READ (NSTP, 100), (MC(I), I = 1, 48)
READ (NSTP, 103), (ID(I), I = 1, 48)
READ (NSTP, 105), (IQ(I), I = 1, 48)
READ (NSTT P, 105) (IF8(I), I=1.48)
READ (NSTT P, 108) (TDW(I), I=1.48)
READ (NSTT P, 111) (CMALL(I), I=1.48)
110 FORMAT (3X, 10F6.0)
READ (NSTT P, 108) (RESB(I), I=1.48)
READ (NSTT P, 108) (RESB(I), I=1.48)
DO 500 J=1,48
READ (NSTT P, 143) (IPC(I,J), I=1.48)
113 FORMAT (1X, 14J3 / 1X, 17I3 / 1X, 17I3)
500 CONTINUE
READ (NSTT P, 103) (ITYPE(I), I=1.48)
READ (NSTT P, 116) (PERC(I), I=1.48)
115 FORMAT (7X, 8F6.4 / 1X, 10F6.4 / 1X, 10F6.4 / 1X, 10F6.4 / 1X, 10F6.4)
READ (NSTT P, 108) (TDUE(I), I=1.48)
READ (NSTT P, 117) (ALN(I), I=1.48)
117 FORMAT (9X, 14F3.0 / 1X, 17F3.0 / 1X, 17F3.0)
READ (NSTT P, 105) (IPRC(I), I=1.48)
READ (NSTT P, 120) (CUM(I), I=1.29)
120 FORMAT (9X, 9F5.0 / 1X, 10F5.0 / 1X, 10F5.0)
READ (NSTT P, 120) (RESMIN(I), I=1.29)
READ (NSTT P, 120) (RESMAX(I), I=1.29)
DO 9039 I=1,48
IF (IFAC(I) .GT. 30) IFAC(I)=30
9039 CONTINUE
CUM(30) = .0
RESMIN(30) = .1
RESMAX(30) = .1.

... CALCULATE COL 1 FROM COL 2 OF TABLES ...

TAE (1, 1, 1) =300
TAE (2, 1, 1) =400
TAE (3, 1, 1) =500
TAE (4, 1, 1) =600
TAE (5, 1, 1) =700
TAE(0,1) = 0
TAE(1,1) = 0.2
TAE(2,1) = 0.1
TAE(3,1) = 0.1
TAE(4,1) = 0.1
TAE(5,1) = 0.1
TAE(6,1) = 0.1
TAE(7,1) = 0.1
TAE(8,1) = 0.1
TAE(9,1) = 0.1
TAE(10,1) = 0.1
TAE(11,1) = 0.1
TAE(12,1) = 0.1
TAE(13,1) = 0.1
DO 30 K=1,2,5
DO 30 K=1,2,5
30 TAE(K,1) = TAB(K,1-1,1) + 10
DO 31 K=1,2,5
31 TAE(K,1) = TAB(K,1-1,1) + 1
DO 40 K=1,2,5
40 TAE(K,1) = TAB(K,1-1,1) + 2
DO 41 K=1,2,5
41 TAE(K,1) = TAB(K,1-1,1) + 2
CONTINUE

C

TDS(1,2) = 2.
INA = 48
INF = 30
C
DO 3 I=1,2,5
3 TAE(I,K,2) = TAB(I,K,1) + 10.0 * TDS(I,1) - 5.0
C
CONTINUE
TDS(I,1) = 0.0
TDS(I,2) = 0.0
C
CONTINUE
IF (MC. ME. 2) GOTO 1920
C
--- APEX PERFORMANCE BUMP ---
H = 100.0
P = 0.05
TAE(14, 25, 3) = F

DO 8001 I = 1, 25
T1 = TAB(1, I, 4)
T2 = TAB(2, I, 1)
T3 = TAB(3, I, 1)
T4 = TAB(4, I, 2)
T5 = TAB(5, I, 2)
T6 = TAB(6, I, 2)
T9 = TAB(9, I, 2)
T10 = TAB(10, I, 2)
T11 = TAB(11, I, 2)

TAE(1, I, 2) = T1*(F*(1.0 - T1))
TAE(2, I, 2) = T2*(1.0 - T2)
TAE(4, I, 2) = T4*((1.0 - T4)/H)
TAE(5, I, 2) = T5 - (F*T5)/H
TAE(6, I, 2) = T6 - (F*T6)/H
TAE(9, I, 2) = T9 - (F*T9)/H
TAE(10, I, 2) = (T10 - (F*T10))/H
TAE(11, I, 2) = (T11 - (F*T11))/H

8001 CONTINUE

1920 CONTINUE
RSMAX = RSMAX/100.0
RSMIN = RSMIN/100.0
FRMAX = FRMAX/100.0
FRMIN = FRMIN/100.0
CDBMAX = CDBMAX/1000000.0
CDBMIN = CDBMIN/1000000.0
FCERMIN = FCERMIN/100
CPUMAX = CPUMAX/1000
CPUMIN = CPUMIN/1000
FCPMMIN = FCPMMIN/100
FPMIN = FPMIN/100

DO 13 I = 1, 5
13 YC(I) = YC(I)/100.
** Tribunal ZRB **

** This subroutine zeros the entire common. **

```plaintext
Common IRR(3600)
DO 4 I = 1, 3506
  IRR(I) = 0
END
```

** Subroutine PESF. **

```plaintext
CCHM COM RESH(48), TAB(20, 25, 3), CTEST(3), CTOH(3), IMIS(2), ISUBS(48)
& IP(48), M(48), IRC(48), IDT(48)
& IP(48), ITERM(48), QTY(48), CHAI(48)
CC HN RES(48), RES(48), RSL(48), TYPE(48)
& PERC(48), TORG(48), DEC(3), Y(8)
& TSTC(48), CL(6, 5), ALW(48), CT(48)
& TPIN(48), MIM(48), IQ(48), CON(30)
CCHN RESH(30), RESHMAX(30), AK(30)
& ITAB(30), X(30), Y(8), INDEX(8)
& RATIO(6), AM(9), ADJ(2, 8), PDR(6)
& PER(6), PVR(6), CD, IC, IAW, INF, IDP
COMN CPUR, CWER, FCPUR, FCWER, ICTEMP(25), COX, C10, TRACET, TRADER,
6THATA, DMT, DMTB, DEF, SUB, COSTL, COSTR, COSTP,
CCHN COST2, COTCL, ITCT, ALPHA, PRICR
& TRADER, TP, RA, IACT(48), IPC(48, 5)
& TDS(48, 2), JMIN, IC, IFIG, KD, NE
& IC, ICOFG, KP, KPP, ITEAM
CCHN RESH, RESIN, RESMAX, RESMAX, RESMIN, RESMIN, CDHAX, CDHIN,
6PCDHIN TDDM, TDDH, TDDAT, CT2(25), TC(8), CT(6), DD(32), M(9)
ITAB(25) = 1DP
```

** Total Flight Test FAB. Costs **

```plaintext
DO 980 M = 1, INA
  I = IFD(M) - 100
```
901 IF (I-6) 950, 950, 901
902 J=IFAC(N)
   CL(I,J)=CL(I,J)*CUN(J)
   CL(I,2)=CL(I,2)*CMA(N)
   CL(I,5)=ALM(A)
950 CONTINUE
949 IF (INC(N), 980, 980, 949
951 IF (ISUBS(N)=6) 951, 951, 980
952 IF (TYPE(N)=1) 980, 953, 980
953 I=INC(N)
   CL(I,4)=CL(I,4)*RESB(M)
980 CONTINUE
C
400 QT(I+1)=200.0
402 QT(I)=QT(I)/100.*5.
C
404 DO 109 M=1,INA
   I=IQ(B)
   IF (TYPE(M), 109, 109, 500
500 IF (I-1), 109, 103, 101
101 IF (I-12), 103, 103, 109
103 X=QT(I)*QT(M)
   IF (I-1), 105, 107, 106
105 IF (PERC(N), 105, 107, 106
106 IF (I-7), 97, 107, 107
97 X=1.0
107 TAE(I+3)=X
   QT(I)=QT(M)*X
109 CONTINUE
C
110 DO 119 M=1,INA
   I=IQ(B)-100
   IF (I-12), 119, 113, 111
111 QT(M)=QT(M)*TAB(I,I,3)
I=QTY (M) + 0.1
QTY (M) = 1
CONTINUE

C

119 IF (IP 10.0) 120, 120, 250
120 IF (IP(M) = 1) 129, 121, 122
121 IF (HIS (M) = 4) 124, 123, 129
122 QTY (M) = 0
123 GO TO 129
124 QTY (M) = 1
125 GO TO 129
126 IF (IP(M) = 2) 129, 125, 129
127 IF (HIS (M) = 2) 124, 123, 124
128 CONTINUE

C

SET END DATES
250 DO 269 M = 1, 10
251 IF (PERC (M) = 1) 257, 269, 269
252 TDUE (M) = 900.0
253 IF (IG (M) = 1) 258, 260, 258
254 IF (IG (M) = 1) 269, 262, 262
255 IF (TDUE (M) = 256, 265, 256
256 IF (TDUE (M) = 256, 251, 256, 269
257 TDUE (M) = 251, 251, 269
258 GO TO 269

C

260 DO 268 I = 1, 10
261 A = I
262 dd = IFAC (M)
263 IF (dd = 261, 261, 262
264 A = RESU (M) * 20.0 * RESMIN (M)
265 dd (M) = dd (M) - (10.0 - A) * A
266 II = dd (M) - (RESU (M) * (20.0 * A - 1.0) **
267 II = II - ALN (M) ** RESMIN (M))
268 CONTINUE
**SUBROUTINE TRADE**

```fortran
C CCONTINUE
RETURN
END
C******************************************************************************
C COMMON REST(48), TAB(20, 25, 3), CTEST(3), CTOH(3), XNIS(2), ISUBS(48)
C & IFAC(48), NT(48), IRC(48), IDGR(48)
C & IPD(48), ITEM(48), QTY(48), CHAT(48)
C COMMON RES(48), RESU(48), RESB(48), ITYPE(48)
C & IPBC(48), TDUE(48), DEC(3), Y(8)
C & TSTAR(48), CI(6, 5), YF(48), RESA(48)
C & TPIN(48), TMIN(48), IQ(48), CT(48)
C & RESMN(30), RESMA(30), AK(30), ISW(3)
C & ITAB(28), NH(8), N8, INDX(8)
C & PE(8), VTR(8), AN(9), ADJ(2, 8), PDR(6)
C & PER(6), PV(6), CD, IC, INA, INF, IDP
C COMMON CPU, CPUIN, IPC, X, M, TEMP(25), CQD, C10, TRADR, TRADER,
C & TRADER, DADM, DMB, DEF, SUR, CSTH, CSTO, COSTP1
C COMMON COSTP2, COSTO, IACT, ALPHA, PBIC
C & TRADEP, TF, GM, IPC(48, 45)
C & TDS(48, 2), JMIN, IC, IFIC, KDP, ME
C & NC, ICOF, KP, KDP, ITHAM
C COMMON RESMN, RESMN, ESMAK, ESMAK, ESMIN, FESMIN, CDMAK, CDMAK,
C & CDIN, TDMAK, THIN, THIN, QT(25), IC(6), CTC(8), DD(32), XNIS(9)
DO 40 M=1, INA
   IF (IO(N) .LT. 20) 40, 20, 10
10   IF (IO(N) .LT. 20) 20, 20, 40
20   K=10(N)-200
    JJ=IFAC (8)
    M(K)=M
    K=MT(N)
    DO 30 I=1, 24
       AAA=ABS(TAB(K, J, 2)-TAB(K, I, 2))
       TAB(K, I, 3)=CUN(33)*((TAB(K, J, 1)-TAB(K, I, 1))/YF(M)*AAA)
    DO 30
30   CCONTINUE
40   CCONTINUE
   IF (CDMAX .GT. 0) GOTO 100
```
GO TERM1=0.0
100  TERM1=(CDMAX-CDMIN)/FCDMIN
110  IF (TDMIN.GT.0) GOTO 120
    TRADET=0.0
    GO TO 130
C ZERNUL WAS ADDED IN MAR 84 TO PREVENT DIVIDE BY ZERO ERRORS.
120  ZERNUL=TDMIN-TDMIN
       IF (ZERNUL.EQ.0.0) ZERNUL=0.00000001
130  IF (RSMAX.GT.0) GOTO 140
    TRADER=0.0
140  IF (ZERNUL.EQ.0.0) ZERNUL=0.00000001
    TRADER=TERM1*FRDMIN/(ABS(ZERNUL))
150  IF (ESMAX.GT.0) GOTO 160
    TRADEA=0.0
160  IF (ZERNUL.EQ.0.0) ZERNUL=0.00000001
    TRADEA=TERM1*FSMIN/(ABS(ZERNUL))
170  IF (CPUMAX.GT.0) GOTO 180
    TRADEP=0.0
180  IF (ZERNUL.EQ.0.0) ZERNUL=0.00000001
    TRADEP=TERM1*FCPUMIN/(ABS(ZERNUL))
C 190 CONTINUE
    RETURN
END
C**************************************************************************
  SUBROUTINE DESRES
C    73MAR07 REVISIED TO DEGRADE TEST VALUES TO
    - EQUIVALENT TABLE VALUES.
C COMMON REST(48), TAB(20,25,3), CTEST(3), CTOH(3), XMIS(2), ISUBS(48)
C    IPAC(48), NTG(48), IDTC(49), IDDT(49)
C DIMENSION YDC(6) 5/83; TO SATISFY WATFIV COMPILATION.
C DELETE KEP=ITAB(25) 2/84; IT WAS RESETING KDP TO 3 ALL THE TIME.

DO 95 M=1,INA
   IF (TMIN(N),LE.0) TMIN(M)=50.0
   KIQ=IQ(M)
   IF (KIQ.LT.201).OR.(KIQ.GT.208) GOTO 95
C DESIGN DUBSYSTEM
   K=NIC-200
   N(K)=N
   IF ((K.NE.5).AND.(K.NE.6))GOTO 11
C GUIDANCE
   IRC(M)=0
   IF (K.EQ.5).AND.(N6.LE.2))GOTO 10
   IF (K.EQ.6).AND.(N6.GE.3))GOTO 10
GOTO 11
C
   IRC(M)=5
   KP=K
JMIN = K
MT(36) = K

11 IF (KDP GT 4) GOTO 95
XCTC = XCTC (K)
IF (XCTC LE . C.D) GOTO 95
I = IPAC (K)
XREST = XCTC/CUN (I)
REST (K) = IREST-HESS (K)
REST (K) = IREST
MN (K) = 1
IF (KDP GT 4) GOTO 95
IF (K NE. 5) AND (K NE. 6) GOTO 95
C BACKUP GUIDANCE
IF (K = EQ -5) AND (ME. EQ -3) GOTO 408
IF (K EQ .6) AND (M6. EQ -1) GOTO 408
GOTC 95
C
-408
L = MT (K)
LL = IPAC (K)
X3 = 0. 3333
DO 410 J = 1, 25
410 TAB (L, J, I) = X3*TAB (L, J, 1)
RESMAX (LL) = X3*RESMAX (LL)
RESHN (LL) = X3*RESHN (LL)
C
95 CONTINUE
IF (J GE. 5) GOTO 100
C
DEGRADE INPUT NR. ACCEPT TO EQUIL TAB VALUES
- FOR INIT SEARCH. DEGRADE INCENT LIMITERS
- SAME FOR ENDS OF INCENT SEARCHES.
- THROUGH STAT 103.
C
HEITABILITY FACTORS
DQT1 = 1. 0 - (40. 0 - QT(2)) * . 00162
DQT2 = 1. 0 - (9. 0 - QT (3)) * . 002
DQT3 = 1. 0 - (6. 0 - QT (6)) * . 003
DPTR = 1. 0 - (25. 0 - QT (1)) * . 0065
C ERRCR FACTORS
DQT4 = 1. 0 - (14. 0 - QT(4)) * . 019
DQTS = 1.0 + ((5.0 - QT(5)) * .0062)
DQT5 = DQTS
DFTE = 1.0 + ((25.0 - QT(1)) * .005)

C RELIABILITY TEST VALUES
YDC(1) = YC(1) * DQT1 * DFTE
YDC(2) = YC(2) * DQT2 * DFTE
YDC(3) = YC(3) * DQT3 * DFTE
YDC(4) = YC(4) * DQT4 * DFTE
YDC(5) = YC(5) * DQT5 * DFTE
YDC(6) = YC(6) * DQT6 * DFTE
DRSMAX = DRSMAX * DQT1 * DQT2 * DQT3 * DFTE

C ERROR TEST VALUES
YDC(7) = YC(7) * DQT4 * DFTE
YDC(8) = YC(8) * DQT5 * DFTE
YDC(9) = YC(9) * DQT6 * DFTE

C THRES
WQ = 0
KA = 1
KZ = 3

C NOTCH AIRFRAME, GSE/LAUNCHER
ASSIGN 102 TO KGO
99 DO 101 K=KA,KZ
   M=M(K)
   J=MT(M)
   IF(MM(J),NE.1) GOTO 9
   IF(K.LE.3) NR = NR+1
   XRESU=RESU(M)
   DO 120 I=1,25
      IF(XRESU.LE.TAB(J,I,1)) GOTO 130
   120   CONTINUE
   I=25
   GOTO 130
   130   INDEX(K)=I
101 CONTINUE
GOTO KGO, (102, 56)
C
102 IF(MR.GE.3) GOTO 39
KA=1
KZ=3
ASSIGN 31 TO KGCA
150 DO 30 K=KA,KZ
      N=N(K)
      J=NT(N)
      I=INDI(K)
      RATIO(K)=TAB(J,I,3)
      Y(K)=TAB(J,I,2)
      30 CONTINUE
GOTO KGOA, (31, 38)
C
37 XY1=Y(1)
XY2=Y(2)
XY3=Y(3)
IF (XY1*XY2+XY3.GE. DRSMAX) GOTO 159
DEC(1)=RATIO(1)/(XY2*XY3)
DEC(2)=RATIO(2)/(XY1*XY3)
DEC(3)=RATIO(3)/(XY1*XY2)
GOTO 33
C
159 KA=1
KZ=3
ASSIGN 44 TO KGCB
160 DO 35 K=KA,KZ
      N=N(K)
      J=NT(N)
      TAB(20,K,1)=NT(N)
      IF(IDP.LE.3) GOTO 1450
C TO PREVENT PERFORMANCE SLIP
 I=TAB(20,K,2)
 IF (INDX(K).GE.I) GOTO 1450
 INDX(K)=I
 GOTO 1470
C
1450 I=INDX(K)
TAB(20,K,2)=I
C  FINE SMALLEST COST/PERFORMANCE FACTOR
33  I=0
    DO 34  K=1,3
    IF (MM(K).EQ.1) GOTO 34
C MAX PERFORMANCE NOT REACHED
    XDECK=DEC(K)
    IF (I.GT.0) GOTO 22
    I=1
    DECMIN=XDECK
    MK=K
    CONTINUE
34  CC
    IF (XDECK.LT.DECMIN) MK=K
    IF (XDECK.LT.DECMIN) DECMIN=XDECK
    IF (DEC(MK).GE.DELEAD) GOTO 39
C THE "SMALLEST" IS LESS THAN RELIABILITY TRADE-OFF FACTOR
    K=MK
    M=K(K)
    J=M(K)
    INDEX(K)=INDEX(K)+1
    IF (INDEX(K).GT.25) INDEX(K)=25
    I=INDEX(K)
    RATIO(K)=TAB(J, I, 3)
    Y(K)=TAB(J, I, 2)
    IF (I.LT.25) GOTO 31
C MAX PERFORMANCE REACHED
    RATIO(K)=9.55010**9
    NR(K)=1
    NR=NR+1
    IF (NR.LT.3) GOTO 31
C 39  KA=1
    KB=3
    ASSIGN 44 TO KGCB
    GOTO 160
C 44 IF(TRADEA.LE.0) ME=2
C NO INCENTIVE FOR ACCURACY
KA=4
KZ=6
C FIRE CONTROL, GUIDANCE A, GUIDANCE B
ASSIGN 56 TC KGO
GOTO 99
56 ME=ME+MN(4)+MN(KP)
IF(ME.GE.2) GOTO 61
KA=4
KZ=6
ASSIGN 38 TC KGOA
GOTO 150
C 38 IF((Y(4)+Y(KP)).GT.(DESMIN /YF(M))) GO TO 62
61 KA=4
KZ=6
ASSIGN 80 TC KGOB
GOTO 160
C 62 IF(RATIO(4).LE.RATIO(KP)) GOTO 66
64 MK=KP
GOTO 70
66 IF(MN(KP).EQ.1) GOTO 64
68 MK=4
C 70 IF(RATIO(MK).GE.TRADEA) GOTO 61
C K=MK
M=M(K)
J=MT(K)
INDX(K)=INDX(K)+1
IF(INDX(K).GT.25) INDX(K)=25
I=INDX(K)
RATIO(K)=TAB(J,I,3)
Y(K)=TAB(J,I,2)
IF(I.LT.25) GOTO 38
C MAX PERFORMANCE REACHED

```
RATIO(K)=9.59*10**9
IF(WE.LT.2) GOTO 38
80  KKFP=1
500  GOTO 9999
9999  RETURN
END

*******************************************************************************
SUBROUTINE DESRET
*******************************************************************************
COMMON REST(48), TAB(20,25,3), CTEST (3), CTOH (3), XMIS(2), ISUBS(48)
COMMON IPAC(48), ITEM(48), IRC(49), IBCD(48)
COMMON RESS(48), RES0(48), RESB(48), ITYPE(48)
COMMON PRC(48), TDUE(48), DEC(3), Y(8)
COMMON TSTAR(48), CL(6,5), ALN(48), CT(48)
COMMON TFIN(48), TFINI(48), IQ(48), CUN(30)
COMMON RESMIN(30), RESMAX(30), AK(30), ISW(3)
COMMON ITAB(25), WM(8), WM(8), INDEX(8)
COMMON RATIO(8), AN(3), ADJ(5,8), PDR(6)
COMMON PER(6), PVR(6), CD, ICD, IMA, IMP, IDE
COMMON CPUMAX, CFMIN, FCPMIN, IWT(32), COV, CTO, TRADTR, TRADER,
COMMON DAM1, DAM2, DEF, SUR, COST, COSTD, COSTP1
COMMON COSTP2, COSTP4, ITCT, ALPHA, PRIOR
COMMON TRADEP, TF, WA, IACT(48,2), IPC(48,5)
COMMON IPC(48,5), IPC(48,2), IPC(48,2), IPC(48,2)
COMMON TRES, TRES, TRES, TRES, TRES, TRES, TRES, TRES, TRES
COMMON TRES, TRES, TRES, TRES, TRES, TRES, TRES, TRES, TRES

50  N=7
51  M=K(N)
52  J=I(T(M)
53  IF (ILP.GE.5) GOTO 90
54  IF (ZRSU.LT.17,11,3)) GOTO 2643
55  DO 89 I=1,25
IF (ZBSU1.LE.TAE(J,I,1)) GOTO 960

C

I=25
GOTO 965

C

960 IF (J.EQ.1) GOTO 962
ZTB1=TAB(J,I,1)
ZTB2=TAB(J,I,1)
ZTB3=TAB(J,I,1)
ZTB4=TAB(J,I,2)
ALPHA=ZTB3-((ZBSU1-ZTB1)/(ZTB2-ZTB1))*(ZTB3-ZTB4))
GOTO 970

C

962 ZBSU1=TAB(J,I,1)
965 ALPHA=TAB(J,I,2)
970 TAE(17,12,3)=ZBSU1
TAE(20,7,1)=I
TAE(20,7,2)=1
GOTO 76

C

2643 RESU(N)=TAE(17,12,3)
ALPHA=TAB(17,12,3)

C

76 CI0 = 0
DO 850 K=1,2
M=M(K)
J=INDX(K)
I=INDY(K)
RESU(N)=TAE(J,I,1)
JJ=INC(M)
CL=CI(JJ,5)
AL=2*I+1
CTCT = CL(JJ,2) + AL * CL(JJ,1) * CL(JJ,3) * CL(JJ,4) * CI0 + CTOT
850 CONTINUE

C

K=K
M=M(K)
J=1T(N)
I=1D0(K)
RESU(M)=TAB(J,I,1)
JJ=IRC(M)
AL=CL(JJ,5)
AL=.1*AL
CTCT = CL(JJ,2) + AL*CL(JJ,1) * CL(JJ,3) *

(1.0 + 0.5) * RESU(M) / CL(JJ,4)
C10 = C10 + CTOT
AL=CL(JJ,5)
AL=.1*AL
C10=C10+CL(6,2)+AL*CL(6,1) * CL(6,3)
TAE(20,25) =C10
C10 = C10*ALPHA

C
K=7
N=K(K)
CALL IN VALUE ENGINEERING RESOURCES FROM TABLE
90 XRESU=TAB(17,11,3)
RESU(M)=XRESU
RESU(M)=XRESU+RESM
ALPHA=TAB(17,12,3)
C
COMPUTE UNIT COST OF PRODUCTION
C10=TAB(20,25,1)*ALPHA
TAE(20,25) =C10
C
K=8
N=K(K)
J=M1(N)
ZBU2=RESU(M)
IF(I*IP.GE.5) GOTO 86
IF(ZBU2.LE.TAE(17,13,3)) GOTO 86
C
IF CURRENT MAINT. ENG. =LE. PREVIOUS, DO NOT SEARCH
85 CC0NTINUE
I=25
GOTO 975
C
84 IF(I.LE.1) GOTC 972
C INTERPOLATE
ZTE5 = TAB(J, I, 1)
ZTE6 = TAB(J, I, 2)
ZTBE7 = TAB(J, I, 1.2)
ZTBE8 = TAB(J, I, 2)
ZULU = ZTB7 - (((ZSU2 - ZTB5) / (ZTB6 - ZTB5)) * (ZTB7 - ZTB8))
GOTO 980
C STORE MAINT: ENG., ZUIU, + TABLE ID
972 ZRSU2 = TAB(J, I, 1)
975 ZUIU = TAB(J, I, 2)
980 TAB(17, 13, 3) = ZRSU2
TAB(17, 14, 3) = ZUIU
TAE(20, 8, 1) = I
TAE(20, 8, 2) = I
GOTO 985
C 986 REST(H) = TAB(17, 13, 3)
985 REST(H) = REST(M) + REST(H)
C
100 CONTINUE
RETURN
END

SUBROUTINE SE

C

COMMON REST(4), TAB(20, 25, 3), CTEST(3), CTOH(3), XMIS(2), ISUBS(48)
& ITAC(48), NT48(48), INC49(48), IDT48(48)
& IPD(48), ITER48(48), QTY4(48), CMAT4(48)
& RESS(48), RES0(48), RES4(48), IPIPE4(48)
& PERC(48), TDB(48), DEC3(48), Y(8)
& TPC(48), BESC(48), YF(48), RES48(48)
& TSTAB(48), CL(6, 5), ALN(48), CT(48)
& TIN4(48), TIN48(48), IQ4(48), CUN(30)
& RES(30), RESHAR(30), AK(30), ISW(3)
& RATIO(6), AM(9), ADJ(2, 8), PDR(6)
& PEB(6), PV(6), CD, IC, TIM, INF, IDE
& CPMAL, CPMIN, ECPHIN, ITEMP(25), COV, CTO, TRALET, TRADER,
& TRALE, DAB1, DAB2, DEF, SUR, COST(A, COSTD, COSTP)
& COMMON COSTP, COST01, ITCT, ALPHA, PHICR

253
& ; TRADEP, TF, WM, IACT(48,2), IFC(48,5) 
& ; TDS(48,2), JMIN, IC, IFIG, KDF, ME
& IM, ICMG, KEE, KFFP, ITEAM
& COMMON RESM, RESMIN, FM, ESMAX, RESMIN, ESFIN, CMAX, CDMIN.
& ECURLTDMA, TDHIN, FTDTIN, QT(25), LC6, CTC(8), DD(32), HIS(9)
DATA IN/5/
DO 290 L=1, ITHF
AN(I)=0.5*CNW(I)+RESMIN(I)-COV
IAC(I)=RESMIN(I)
290 CONTINUE
C C
SET ECP
C C
MODULE 4
C C
ADD DESIGN RESOURCES
DO 390 H=1, JNA
IF (ITYPE(H)-1) 390, 310, 390
310 I=IRF(H) 
320 ADV(I)=ADV(I)+RESU(H) 
390 CONTINUE
C C
MODULE 2
C C
SETS PBC COUNTERS FOR DUE DATE CALCULATION,ADJUSTS PAB
C C
RESOURCES ABE SETS PBC COUNTERS
DO 3901 H=1, JNA
3901 IPBC(H)=O
C
DO 490 M=1, JNA
DO 440 I=1, 5
IF (IFC(M, I)) 440, 440, 410
410 J=IPBC(H, I)
IF (IURL(H)-8) 420, 420, 420
420 IF (IURL(H)=IURL(J)) 440, 430, 440
430 IPBC(J)=IPBC(J)+1
440 CONTINUE
C
450 IF (IRF(M)) 470, 470, 460
460 IF (ITYPE(H)-5) 470, 465, 470
465 J=IRF(M)
RESB(M) = (RESB(M) + (ADJ(1, J) + ADJ(2, J)) * 0.5 * RESB(M)) * ALPHA

REST(M) = 0.0
PEBC(M) = 0.0

IF (CNY(M) - 1.0 = 0, 1000, 490)
PEBC(M) = 1.0
REST(M) = RESA(M)
IF (TPIN(M) = 1000, 1000, 490)
GO TO 490

C
475 IF (RESA(M) * 0.01 - REST(M)) = 477, 471, 471
476 TERM1 = ((RESA(M) - REST(M)) / REST(M)) **
8 (1.0 - (1.0 - ALN(M))) / QTY(M)

C
IF (TERM1) = 490, 490, 490
480 IF (TERM1 - 1.0) = 481, 481, 471
481 PEBC(M) = TERM1

C
500 CONTINUE

C
MODULE 3
SET DUE DATES

510 IF (I.THR0 = 0)
DO 585 MILA = 1, IMA
H = (IMA - MILA) + 1
IF (PEBC(H)) = 585, 510, 585
520 IF (PEBC(H) = 0.0) = 520, 585, 585
510 IF (I.THR0 = K) = 510, 510 + 1
L = L + 1
IF (L.GT. 30) = 30
IPEC(H) = 0
AA = DUE(M) + (REST(M) - REST(M) - REST(M)) / RESMIN(L)
TERM3 = DUE(A) + (REST(M) - REST(M)) / RESMIN(L)
TERM4 = AA - DUE(A)

C
DO 570 I = 1, 5
IF (ppeb(M, I)) = 570, 580, 530
530 IF (ISUBS(A) = 8) = 540, 535, 535
535 IF (ISUBS(A) = 8) = 570, 540, 570
PRINT 1001, M, J, A, I
1001 FORMAT(4I3)
GO TO 2
C
540 K=IFAC(J)
   IF(K.GT.30)K=30
   IF(J.GT.48)GC TO 1
   2 CONTINUE
      TERM1=REST(J)-RESU(J)-RESU(J))/RESMIN(K)
      IF(TERM1-TERM4) 545, 545, 550
      GC TO 555
C
550 DDD=TERM3-(TERM1-TERM4)
      555 IF(TDUE(J)-DDD) 565, 565, 560
      560 TDUE(J)=DDD
      565 IPRC(J)=IPRC(J)-1
C
      570 CONTINUE
      580 TDUE(N)=AA
C
      585 CONTINUE
      IF(KTRM1.GT.0) GOTO 590
C
      FIX DATA REDUCTION, FLIGHT TEST
      DO 850 M=1, INA
      IF(TDUE(M)-998.0) 850, 810, 810
      810 I=PRC(M)
      IF(I) 850, 850, 820
      820 L=IFAC(M)
      TDUE(M)=TDUE(I)+REST(M)/RESMIN(L)
      850 CONTINUE
C
      MODULE 5
C
      SET ACTIVITY POINTERS BY FACILITY
      DO 690 M=1, INA
      IF(PERC(M)-1.0) 610, 690, 690
      610 I=IFAC(M)
      611 IF(ISUB5(M)-8) 620, 640, 640
      620 IF(IAC(T(I))-1) 641, 630, 641
      630 AK(I)=AK(I)-TRAD
      GO TO 641
C
540 AK(I)=5000.0
(Image content not readable)
& IPSC(48), RESC(48), YF(48), RESA(48)
& TSP(48), CT(63), ALN(48), CT(48)
& TSPIN(48), THIN(48), IQ(48), CT(48)
& COMN(30), RESMIN(30), RESMAX(30), AK(30), ISW(3)
& ITAB(35), M(98), N(98), INDX(9)
& RATIO(8), AN(9), ADJ(28), PDR(6)
& PER(6), PV(6), CD, IC(6), IMA, INF, IDP
& COMMON CPUMACH, CFMIN, FCMIN, X, Y, XTEMP(25), COV, C10, TRADET, TRADER,
& TRADEX, DAM1, DAM2, DETR, SUR, COSTL, COSTD, COSTP1
& CCMCOST, COST01, ITCT, ALPHA, PRICER
& TRADER, TE, MA, IACT(48,2), IPC(48,5)
& TDS(48,2), JMIN, IC, IPG, KDP, HB
& MC, ICG, KP, KPP, TRAPM
& COMMON RSMAX, RSMIN, PRSMA, ESMA, ESMAX, ESMIN, PESMIN, CDMAX, CDMIN,
& EDMIN, TDMIN, TDMIN, QT(25), YC(6), CTC(8), DD(32), MIS(9)

67 JJJ=JJJ+1
65 CONTINUE
61 CONTINUE
60 ISW(I)=0
CD=CD+1.0
ICI=ICI+1

FROM 25 TO 22 IS EXTRACTED FROM
MOD7 IN DES PROGRAM

DO 64 M=1, INM
L = IACT(M)
MIANT = IACT(L,1)
IF (MIACT.LT.0) GOTO 64
IF (MIACT.NE.M) GOTO 64
4 IF (TSPAR(M)-CL) .LT.5, 5, 0
5 PRIOR=1.0
X=0.0

DO 7 I=1, 5
J=IPC(M,1)
IF (J.LT.0) GO TO 7
KK = 0
IF (I.LT.5) KK= IPC(I, I+1)
6 IF (ITERM(J)-TDUE-1) N.B.10
10 IF (PERC(J)-PRICR) N.B.11, 12, 12
11 PRIOR=PERC(J)
12 IF (TSTAR(J)-X) 13, 13, 14
14 X=TSTAR(J)
13 IF (KK) 15, 7, 15
15 IF (PERC(J)-1.0) 16, 7, 16
16 IF (PERC(H)) 7, 7, 17
17 JJ=IFAC(J)
XX=((PERC(J)-PERC(H))*REST(J))/RESC(JJ)
18 THIN(J)=XX
7 CONTINUE

C
IF (X-ISTAR(M)) 15, 19, 20
19 ISTAR(M)=X
10 IF (PERC(M)-PRIOR) 21, 8, 8
21 WA=RESMIN(I)
11 IF (AK(L)) 22, 27, 27
22 CONTINUE

C
FROM HERE TO 26 LESS ONE IS EXTRACTED
FROM MOD1 IN DMS PROGRAM...

C
T=((REST(M)-RESA(M))/RESMIN(L))*CD
IF TDUE(M) .LE. CD) GO TO 83
IF TDUE(M) .GE. 1) GO TO 82
TERM1=(REST(M)-RESA(M))/(TDUE(M)-CD)

C
RESTMAX(L) (A TYPO) CHANGED TO CD 1970 MARCH 25...
83 RESC(I)=RESMAX(L)
GO TO 82
84 RESC(L) = TERM1
82 CONTINUE
IF (RESC(L) .GE. RESMIN(I)) GO TO 26
RESC(I) = RESMIN(I)
C
FROM HERE TO 100 IS EXTRACTED FROM MOD7

IN DM'S PROGRAM . . . . . . . . . . . . .

26 WA=RESQ(L)
27 X=0.0
28 IF (QTY(M)+0.1 > 31, 50, 23
29 I=1.0
30 WA=0.0
31 REST(M)=0.0
32 GO TO 24

50 IF (FEBOR-1.0) 6, 23, 23
51 IF (TERM1=REST(M)-RESQ(M)) 25, 25, 32
52 REST(M)=REST(M)
53 WA=TERM1
54 PEC(M)=1.0
55 GO TO 100

FROM HERE TO 93 PLUS TWO IS EXTRACTED FROM

MOD8 IN DM'S PROGRAM . . . . . . . . . . .

I=0
75 IF (ISUBS(M)-8) 75, 76, 77
76 I=I+1
77 I=I+1
78 I=I+1
79 IF (RESQ(M) .NE. WA) GO TO 71
80 CTEST(I)=CTEST(I)+(CMAT(M)*QTY(M))**ALPHA
81 CT(M)=CT(M)+(CMAT(M)*QTY(M))**ALPHA
C
ISW(I) = ISW(I) + 1
X = 0
IJ = IFAC(N)
IF(WA .LE. RESMIN(IJ)) GO TO 81
X = WA - RESMIN(IJ)

C
XX = (WA*CUN(IJ) + .5*X*CUN(IJ))
CT(N) = CT(N) + XX
CTEST(I) = CTEST(I) + XX
IF(RESA(N) + 1 .EQ. REST(N)) GO TO 64
PESC(N) = 1
RESA(N) = REST(N)
TPTB(N) = CD

C
IA1 = IACT(1, 1)
IA2 = IACT(1, 2)
IF(M .NE. IA1) GOTO 101
IF(M .NE. IA2) GOTO 102
PRINT 1002

1002 FORMAT(' 8**MOD* IACT ERROR ****')

STCP
101 IACT(1, 1) = IA2
102 IACT(1, 2) = 0

C
J = ITERM(M)
IF(J .LT. IDP .NE. 1) GO TO 64
ITCT = ITCT - 1

C
.. FROM HERE TO 64 IS EXTRACTED FROM MOD7
IN DMSPROGRAM.........

8 CONTINUE

64 BESC(I) = RESMIN(I)
.. 64 CONTINUE CHANGED TO ABOVE 1970 MARCH 25...

C
.. FROM HERE TO END IS EXTRACTED FROM MOD9
IN DMSPROGRAM.........

IF(ISW(1) .EQ. 0) GO TO 66
CTOH (1) = CTOH (1) + COV

CONTINUE
IF (ICT - GT. 0) GO TO 67
IF (IDP . NE. 0) GO TO 60
DC 52 L = 1
IF (IACT (1, 1) .GT. 0) GO TO 67
CONTINUE
CONTINUE
RETURN

C **********************************************************************

SUBROUTINE REPORT

C

COMMON RST (48), TAB (20, 25, 3), CTEST (3), CTOH (3), XMIS (2), ISUBS (48)

$ IF I = 48$, MT (48), IRC (49), IDDT (48)
$ END COMMON RES (48), RESD (48), RESB (48), IYPE (48)

$ IF IPRC (48), RESC (48), IF (48), RESA (48)
$ IF (48), CL (48), ALN (48), CT (48)

$ TPIN (48), TMIN (48), IQ (48), CUN (48)

CCHCN RESMIN (30), RESMAK (30), AK (30), ISW (3)

C ITAB (23), N (8), N (8), INDX (8)

C RATIO (6), AN (9), ADJ (2, 8), PDR (6)

C PSEQU (6), PV2 (6), CDL (1), ICNT, INP, IDP

CCHCN CPUMAX, CEUMIN, FCJOIN, XTEMP (25), COV, C10, TRACT, TRADER,

CTRADE, DAPR1, DAPR2, DAP2, SUR, COSTA, COSTD, COSTP1

CCHCN COSTP2, COSTOL, IACT, ALPHA, PRIGR

C TRADE, TF, IACT (48, 2), IPC (48, 5)

C NC, ICOFG, KP, KPP, ITEAM

CCHCN RMSA, RESMIN, PRSMAX, ESMA, ESMIN, MESMIN, CMSMAX, CMSMIN,

CFRMIN, TDMIN, TDHMIN, TDMIN, QT (25), FC (6), CTC (8), DD (32), MIS (9)

T1 = 0
T2 = 0
T3 = 0

ITAB (24) = IDP
TAB (17, 25, 3) = CTEST (1) + CTOH (1)
DO 11 I = 1, 3
IF (CTST (1)) 12, 12, 11
12 CTEST(1)=1.0
14 CCONTINUE
C
X1= CTEST(1) + CICH(1) / CTEST(1)
X2= CTEST(2) + CICH(2) / CTEST(2)
X3= CTEST(3) + CICH(3) / CTEST(3)
DO 1 M=1, IM
1 IF (ISUBS(M)-8) 2, 1, 1
2 X=M
GO TO 5
5 IF (QTY(M)-1.0) 6, 6, 9
9 TDS(M,1)= PERC(M)
GO TO 10
20 TDS(M,1)= 1.
GO TO 10
C
6 IF (REST(M) EQ 0) GO TO 20
10 TDS(M,2)= RESA(M)*REST(M)
1 CCONTINUE
C
IDF=7
ITAB(25)=3
CCONTINUE
RETURN
END
C
******************************************************************************
SUBROUTINE PERF
C
COMMON REST(48), TAB(20,25,3), CTEST(3), CTOH(3), XMIS(2), ISUBS(48)
C
*** IFAC(48), H1(48), IRC(49), IDDT(48)
C
*** IPD(48), ITERM(48), QTY(48), CMAT(48)
C
*** PEBC(48), TDUF(48), RESB(48), IYPE(48)
C
*** IFBC(48), REBC(48), YP(48), RESA(48)
C
*** TSTAR(48), CL(6,5), ALN(48), CT(48)
C
*** TPIN(48), TMN(48), IQ(48), CUN(30)
C
*** RESMIN(30), RESMAX(30), AK(30), ISW(3)
C
*** ITAB(25), N(8), N(8), INDX(8), PDR(6)
100 DO 118 M = 1, INA
   TDS (M, 1) = 1.
   IF IT (M) .EQ. 6) 116, 116, 118
116 IF IT (M) = 116, 400, 117
117 I = ISUBS (M)
   AN (I) = 1.0 + QTY (I) * TDS (M, 1)
   AK (I) = 1.0 + QTY (B)
   GO TO 118
400 FTEST = QTY (I) * TDS (M, 1)
   TEST = QTY (B)
118 CONTINUE

119 DO 130 K = 1, 6
   R = K (R)
   I = ISUBS (M)
   L = TAB (20, K - 1)
   J = TAB (20, K - 2)
   PRICE = 100.0 * TAB (L, 2) * (1.0 - TAB (L, J, 2)) - 1.0
   IF K (R) = TAB (L, 1)
106 AN (I) = AN (I) + PRICE + FTEST
   AK (I) = AK (I) + PRICE + FTEST
   GO TO 130
C 107 IF (K-KP) 110, 106, 110  
110 AN(6)=AN(I)+PRIKR*TTEST  
AK(6)=AK(I)+PRIKR*TTEST  
SS=AK(6)-PRIOR-1.0  
PFA=TA[N(I)-0.2]  
EV=EXP(AN(6)-PEA)/(AN(6)+1.0)  
PDA=EXP[(AN(6)+1.0)-PEA]  
TF=EXP(0.1*SS)  
FP=0.915+0.314/TEM  
C  IF (PP-II) 510, 510, 500  
500 PP=1.0  
510 PEA=PEA*PP  
520 FAMIN=PEA-PDA  
IF (PAHMIN-0.1) 128, 129, 129  
128 FAMIN=0.1  
129 FAMAX=PEA+PDA  
C  IF (PAHMAX-0.99) 109, 109, 430  
430 PAHMAX=0.99  
PDA= (PEA*AN(6)-1.0)/ (AN(6)-2.0)  
IF (ELA-PAMAX) 109, 109, 515  
515 EMA=PEA+0.33*(FAMAX-PEA)  
C 109 IF (ELAMIN) 68, 70, 70  
68 EMA=PEA-0.33*(FAMAX-PEMIN)  
70 SPEM=EMA  
SPEF=FAMAX  
SPEF=FAMAX  
C 130 CONTINUE  
C DO 200 I=1,5  
200 IS CLOSE TO THE END ...  
GO TO 171, 173, 175, 177, 178  
C 171 A=1.067  
B=-1.34  
C=0.1
J3 = 15

THROUGH STATEMENT 191 REVISED 720123 TO CONFORM
WITH DNS PERFOR CHANGE OF 720530.
CAUSES FLIGHT TEST RESULTS TO ALWAYS BE COMPUTED
ON THE EXPECTED NUMBER OF FLIGHT TESTS.

180 L = TAB(20, K + 1)
LL = TAB(10, 1) + E-4
XPER = TAB(L, LL, 2)
TABLL = XPER
PRIOR = 100.0 * TABLL * (1.0 - TABLL) - 1.0
ANIII = AN(I)

C BLOCK II MOST LIKELY
SS = AR(I) - PRIOR - 1.0
TEM = EXP(C * SS)
FP = (A + B / TEM)
IF (K. GE. 4) GOTO 535
IF (FP. LT. 0.90) FP = 0.90
GOTO 550
535 IF ((FP * XPER) .GT. 0.99) GOTO 91
550 XPER = XPER * FP
IF (XPER .LE. 0.99) GOTO 195
91 XPER = 0.99
195 TAB(14, J3, 2) = XPER

C BLOCK II DEVIATION
PV = XPER * (1.0 - XPER) / (ANIII + 1.0)
PD = SORT(PV)
TAB(14, J3, 2) = FD

C BLOCK II EXPECTED
PM = (XPER * ANIII - 1.0) / (ANIII - 2.0)
PMA = XPER + E-4
IF (PMA. GT. 0.99) PMA = 0.99
IF (PM. LE. PMA) GOTO 197
FM = XPER + 0.33 * (PMA - XPER)
197 FMIN = XPER - E-4
IF (FMIN. LT. 0.1) FMIN = 0.1
IF (FM. GE. FMIN) GOTO 191
191  PM = XPER - 0.33 * (XPER - PMIN)

C TAE(14, 32.2) = PM

C --- ABOVE 10 STATEMENTS INSERTED 1970 APRIL 1 ---

C 200 CONTINUE

C TAE(14, 15, 2) = TAE(14, 1, 2) * TAE(14, 4, 2) * TAE(14, 7, 2)
C TAE(14, 17, 2) = TAE(14, 2, 2) * TAE(14, 3, 2) * TAE(14, 8, 2)
C TAE(14, 21, 2) = TAE(14, 3, 2) * TAE(14, 5, 2)

C --- ABOVE STATEMENT INSERTED 1970 APRIL 1 ---

C CONTINUE

C RETURN

C ***************************************************

C SUBROUTINE PROSUM

C COMMON REST(48), TAB(20, 25, 3), CTEST(3), CTOH(3), XMIS(2), ISUBS(48)
C 
C COMMON IPAC(48), NT(48), IRC(49), IDDT(48)
C COMMON RESS(48), RESO(48), RESB(48), ITYPE(48)
C COMMON PESC(48), PDUE(48), DEC(3), Y(8)
C COMMON IPRC(48), RESC(48), YP(48), RESA(48)
C COMMON TSTAR(48), CL(6, 5), LNL(48), CT(48)
C COMMON TSPIN(48), TMIN(48), IQ(48), CUN(30)
C COMMON BESMIN(30), RESMAX(30), AK(30), ISW(3)
C COMMON ITAB(25), N(8), X(8), INDEX(8)
C COMMON RATIO(8), A(9), ADJ(5, 8), PDR(6)
C COMMON PEP(6), PVR(6), CD, ICD, IMA, INF, IDE
C COMMON CDMAX, CDMIN, FCEMN, YTEM(25), COT, CIO, TRADET, TRADER,
C COMMON COSTP2, COSTOT, ITCT, ALPHA, PRCR
C COMMON TEADEP, TP, WA, TACT(48, 2), ICP(48, 5)
C COMMON TDS(48), JMIN, IC, IFIG, KDF, MB
C COMMON RSMAX, BSMIN, RSMAX, ESMAK, BSMIN, PSEXMIN, CMAX, CDMIN,
C FCEM, TMAK, TMIN, FTDMIN, QT(25), YC(6), CTC(8), DD(32), MIS(32)
C COMMON STORED, STAT(5), NDUMP(5, 5), ADUMP(5, 28), DPCGST, MAHP
C REAL YTEP(48), PRATIO(8), YPC(5)
C REAL *8 STAT
REAL*8 FINAL/"FINAL"/ INTEN/"PROPOSED"/
INTEGER ANS, ANZ, 1, N, YES/"Y", E/"E", IN/5/
TM1TAB=0.
TM2TAB=0.
CTESTA=CTEST (1) - (CT (19) + CT (20))
FCTOH=CTESTA/CTEST (1)
CTOH (1)=CTOH (1) * FCTOH
Y (1)=CTEST (1) + CTOH (1)
DO 5 J=1, 10
ITAB (J)=TAB (19, J, 1)
5 CONTINUE
DO 10 M=1, INA
IF (IN (M) = 1) 10, 15, 10
10 CONTINUE
C
M=INA
15 Y (2)=TPIN (M)
25 Y (3)=TAB (14, 16, 2)
Y (4)=TAB (14, 20, 2)
C
30 K=4
M=N (K)
Y (4)=Y (4) * YF (M)
Y (5)=CT0
C
PER (1)=CDMAX
AN (1)=CDMIN
FCYC (1)=PCDMIN
PER (2)=PDH MAX
AN (2)=PTDMIN
FCYC (2)=PTDMAX
AN (3)=ESMAX
PER (3)=EMIN
FCYC (3)=EFSMAX
PER (4)=EFSMAX
AN (4)=FSMIN
FCYC (4)=PFSMIN
PER (5)=CPUMAX
AN (5)=CPUMIN
FCYC (5)=PCPMIN
SEVA=0.0
FRAIO(5)=0,0
Y(3)=Y(3)/1000000.0
Y(3)=Y(3)*100.0

C THIS SET OF STEPS TO TABLE 50 CHECKS THE MOST DESIREABLE VALUES,
C AN(I), VS ACHIEVED VALUES, Y(I), AND LEAST DESIREABLE VALUE, PER(I).
C VS ACHIEVED VALUE IN ORDER TO PLACE THE FEE WITHIN THE LINEAR RATIO
C BOUNDS OF THE DESIRE FEE.

DO 50 I=1,5
   IF(FLNO,I3)GOTO 48
   IF(Y(I)-AN(I)-GE.0.-AND.PER(I)-Y(I)-GE.0.-AND.PER(I)-NE.0.-AN(I))
      *FRACTO(I)=Y(I)*ABS((PER(I)-Y(I))/(PER(I)-AN(I)))
      IF(Y(I)-PER(I)-GT.0.)FRATIO(I)=0.0
      IF(AN(I)-Y(I)-GE.0.)FRATIO(I)=Y(I)
   GOTO 48

48 IF(Y(3)-PER(3).GE.0.-AND.AN(3).NE.PER(3))
   *FRACTO(3)=Y(3)*ABS((Y(3)-PER(3))/(AN(3)-PER(3)))
   IF(PER(3)-Y(3)-GT.0.)FRATIO(3)=0.0
   IF(Y(3)-AN(3).GE.0.)FRATIO(3)=Y(3)

49 YFEE=SEVA*YFEE(I)
   FRATIO(8)=FRATIO(8)+FRATIO(I)
   GOTO 48

50 CONTINUE

C TOC=Y(I)+SPVA+DFCOST/1000000.0

C CALL FRTCMS('CISSCRN')

IF(MR.NE.0)PRINT 14,ITEM,MAXDP,STAT(MAXDP)
PRINT 1000,ITEM,13)
PRINT 1002
14 FORMAT('12X',**TEAM',12, 'IS IN DP',11, 'X',A8,'**'

1000 FORMAT('1DEX',11,'** DEVELOPMENT CONTRACT',

1010 FORMAT('1X','area',9X,'value',5X,'value',3X,'allowed',6X)
C
6 'VALUE*X,'EARNED’, 3X,’EARNED’/)
C
PRINT 1020, CDMAX, CDMIN, PCDMEX, Y(1), YFEE(1), FRATIO(1)
1020 FORMAT ('DEV. COST', 6X, $1, P5.1, $1, P5.1, 41
   & P7.1, %, 5X, $1, P6.2, M$, P6.3, M$, P7.2, %)
C
   ITA = TDMAX
   ITI = TDMIN
   IV = Y(2)
   PRINT 1030, ITA, ITI, PTDMIN, IV, YFEE(2), FRATIC(2)
1030 FCMEM(TH, 'PLT TST COMPL', 16, 'WK', 16, 'WK', P7.1,
C
   Y(3) = TAB(14, 16, 2) * 100.
   PRINT 1040, BSMIN, BSMAX, FRSMAX, Y(3), YFEE(3), FRATIO(3)
C
   ITA = ESMAX
   ITI = ESMIN
   IV = TAB(14, 20, 2) * 100.
   CPUMAX = CPUMAX / 1000.
   CPDMIN = CPDMIN / 1000.
   PRINT 1050, ITA, ITI, ESMIN, IV, YFEE(4), FRATIO(4)
1050 FORMAT ('ACCURACY', 'RUN', 'YD$, 'YDS', 'P7.1, 'X',
   & 110, 'YDS', 3X, $1, P6.3, M$, P7.2, 'X')
C
   FEET01 = PCDMEX + PTDMIN + FRSMAX + ESMIN + PCDMIN
   PRINT 1070, FEET01, SPY, FRATIO(6)
1070 FORMAT (1H0, 'TOTALS',
   & 28X, 'X',
   & P4.1, 'X',
   & 15X, '
   IF (ABS(FEET01 - 15.0) LT .0001) GOTO 1090
   PRINT 1080
1080 FORMAT (1H0, 'TOT MAX FEE ALLOWED DOES NOT EQUAL
   & 15.0% ***)
C
1090 DPCST=DPCST/10000000.0
CONTINUE
2001 IF(DPCST GT 0.00) PRINT 2001,DPCST
PRINT 2000, 'ADDED COST FOR CONTRACT ANALYSES = $',F5.1,'M'
2000 FORMAT(1H0,'TOTAL CONTRACT PRICE = $',F5.1,'M',/)
PAUSE ';PRESS <ENTER> TO CONTINUE.'
RETURN

REPORT TEST COMPONENT COSTS TO THE MONITOR.

ENTRY PROSM2
CALL GET(KDP)
ITA = TMAX
CALL FRTGMS('CIBSCRN ')
111 FORMAT (1X,25(''-'))
210 FORMAT (1H0,'COSTS OF TESTS PER UNIT (IN THOUSANDS OF DOLLARS)'
220 FORMAT (1H0, 3X, 'QUALIFICATION TESTS')
PRINT 220
23 = 1000.0

MOTOR
ZXY = ((CT(29) +CT(34)) / QT(2)) / 23
PRINT 261, ZXY
261 FORMAT (1H+, 5X, 'MOTOR :',T23,F8.3)

AIRCRAFT
ZXY = ((CT(30) +CT(35)) / QT(3)) / 23
PRINT 262, ZXY
262 FORMAT (1H+, 5X, 'AIRCRAFT :',T23,F8.3)

GUIDANCE
ZXY = ((CT(31) +CT(36)) / QT(5)) / 23
PRINT 263, ZXY
263 FORMAT (1H+, 6X, 'GUIDANCE :',T23,F8.3)

FIRE CONTROL
ZXY = ((CT(32) +CT(37)) / QT(4)) / 23
IF (MIS(6) .LE. 2) TMTABA=TAB(14,14,2)
PRINT 402, KZTE, KZRW, TAB(KZTB, KZRW, 2) TMTABA
402 FORMAT(7X, 'GUIDANCE A': 'T23, 12, T29, 12, T34, P6.3, T44, P7.3)

C
GUIDANCE B
KZTB = TAB(20, 6, 1)
KZRW = TAB(20, 6, 2)
IF (MIS(6) .GE. 3) TMTABB=TAB(14, 14, 2)
PRINT 267, KZTB, KZRW, TAB(KZTB, KZRW, 2) TMTABB
267 FCNMAT(7X, 'GUIDANCE B': 'T23, 12, T29, 12, T34, P6.3, T44, P7.3)

C
FIRE CONTROL
KZTB = TAB(20, 4, 1)
KZRW = TAB(20, 4, 2)
PRINT 268, KZTB, KZRW, TAB(KZTB, KZRW, 2), TAB(14, 11, 2)
268 FORMAT(5X, 'FIRE CONTROL': 723, 12, T29, 12, T34, P6.3, T44, P7.3)

C
LAUNCHER
KZTB = TAB(20, 3, 1)
KZRW = TAB(20, 3, 2)
PRINT 269, KZTB, KZRW, TAB(KZTB, KZRW, 2), TAB(14, 9, 2)
269 FORMAT(9X, 'LAUNCHER': 'T23, 12, T29, 12, T34, P8.3, T44, P7.3)

C
9800 PAUSE 1; PRESS <<ENTER>> TO CONTINUE. RETURN

C
END

C******************************************************************************

C******************************************************************************

C
SUBROUTINE RERUN
C
C("CMCN RE3T(48), TAB(20, 25, 3), CTEST(3), CTOH(3), XMIS(2), ISUBS(48)
C; IBAC(48), IDC(48), IPAC(48), ITAB(48), ITYP(48), ITY(48), ITYP(48)
C; IESS(48), IRES(48), IRESB(48), IRES(48), IRESB(48)
C; IPRC(48), IRES(48), IRE(48), IRES(48)
C; ISTAR(48), ISTAR(48), ISTAR(48), ISTAR(48)
C; ITPN(48), ITPN(48), ITPN(48), ITPN(48)
C; IMCN(30), IMCN(30), IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C; IMCN(30), IMCN(30)
C
ALEHA=QQ(14)
IDPE=4
JMIN = 0
WTEMP(2)=4

C
DO 1 I=1,48
IF (I.GT. 3) GC TO 4
DEC(I)=0.
JSW(I)=0.
4 IF (I.GT. 6) GC TO 2
FVE(I)=0.
DOR(I)=0.
PEE(I)=0.
DO 6 J=1,5
6 CL(I,J)=0.

C
2 IF (I.GT. 8) GC TO 3
CTC(I)=0.
INDEX(I)=0.
RATIO(I)=0.
ADJ(I,1)=0.
ADJ(I,2)=0.
NM(I)=0.

C
3 IF (I.GT. 9) GC TO 5
MIS(I)=0.
AN(I)=0.
5 IF (I.GT. 18) GC TO 15
QQ(I)=0.

C
15 IF (I.GT. 25) GC TO 7
Q1(I) = 0
7 IF (I .GT. 32) GO TO 8
   DD(I) = 0.
C 8 RESC(I) = 0.
C DO 10 J = 1, 2
  10 TDS(I, J) = 0
      CONTINUE
C DO 20 I = 1, 25
   20 TAE(I, J, 3) = 0.
C QC(14) = ALPHA
RETURN
END
C*****************************************************************************
C SUBROUTINE STORE
C*****************************************************************************
INTEGER IN/, DATFIL/9/
COMMON/STORED/ STAT(5), NCUMP(5, 5), ADUMP(5, 28), DPCCST, MAXDP
REAL*8 STAT
REAL*8 FINAL, 'FINAL'/'INTER/'PROPOSED'/'BLK'/'
REAL*8 NULL'
REWIND DATFIL
DO 25 JA = 3, 5
   IF (STAT(JA), EQ, NULL) STAT(JA) = BLK
   WRITE (DATFIL, 35) (NDUMP(JA, K), K = 1, 5), (ADUMP(JA, K), K = 1, 28), STAT(JA)
25 CONTINUE
35 FORMAT (2I2, 1X, 3I1, 2X, 8F7.2, /, 3X, 9F7.2, /, 3X, 6F7.2, /, 3X, 5F7.2, 2X, A8)
RETURN
END
C*****************************************************************************
COMMON REST(48), TAB(20, 25, 3), CTEST(3), CTOH(3), XMIS(2), ISUBS(48)
&  IFAC(48), MT(48), IRC(49), IDDT(48)
&  IFD(48), ITERM(48), QTY(48), CHAT(48)
&  PESC(48), RPRC(48), SPE(48), RESB(48), ITYPE(48)
&  PESC(48), TREC(48), RESB(48), LREC(48), Y(8)
&  TSTA(48), CL(6, 5), ALN(48), CT(48)
&  TPIN(48), THIN(48), IQ(48), CON(30)
&  ITAB(25), N(8), N(8), INDEX(8)
&  RATIO(8), AN(9), ADJ(2, 8), PDR(6)
&  PER(6), PVR(6), CD, ICB, LNA, INF, IDE
&  CPUMA, CPUMIN, FEHIN, FEHIN, INIEN, TEMP(25), COV, C10, TRADER, TRADER,
&  TRADER, TRADER, TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
&  TRADER, TRADER, TRADER,
* *WHAT DP AND WHAT STATUS DO YOU WANT: EX: 3 FINAL; 4 PROPOSED?*

READ IN, 416, ERR=420, END=420) KDP, PROP

FCNMAT(111, 1X, 1A8)

IF(PROP.EQ.INTEN) STAT(KDP) = BLK
IF(PROP.EQ.FINAL) STAT(KDP) = INTEN
IF(KDP.EQ.EASY) RETURN
IF(KDP.LT.3 OR KDP.GT.5) GOTO 417
IF(PROP.EQ.INTEN OR PROP.EQ.FINAL) CALL FRICHS(* CLASCRN *)
IF(PROP.EQ.INTEN OR PROP.EQ.FINAL) GOTO 410

GO TO 417
RETURN

CONTINUE

PRINT 85, KDP, NDUMP(KDP, 3)
ADUMP(KDP, 13), NDUMP(KDP, 4), ADUMP(KDP, 14)
NDUMP(KDP, 5), ADUMP(KDP, 15), ADUMP(KDP, 1)
ADUMP(KDP, 16)
PRINT 90, ADUMP(KDP, 2), ADUMP(KDP, 17),
ADUMP(KDP, 3), ADUMP(KDP, 18),
ADUMP(KDP, 4), ADUMP(KDP, 19),
ADUMP(KDP, 5), ADUMP(KDP, 20),
ADUMP(KDP, 6), ADUMP(KDP, 21),
ADUMP(KDP, 7), ADUMP(KDP, 22),
PRINT 110, ADUMP(KDP, 8), ADUMP(KDP, 23),
ADUMP(KDP, 9), ADUMP(KDP, 24),
ADUMP(KDP, 10), ADUMP(KDP, 25),
ADUMP(KDP, 11), ADUMP(KDP, 26),
ADUMP(KDP, 12), ADUMP(KDP, 27),

IF(STAT(KDP).EQ.BLK OR STAT(KDP).EQ.WULI) PRINT 36, INTEN, KDF
FORMAT(* " DC YOU WISH TO ENTER THIS DATA AS YOUR *. A8. * DP-1",
*12." DECISION?")
IF(STAT(KDP).EQ.INTEN) PRINT 37, KDP
FORMAT(1X, 32 (*. *)), "IMPORTANT: 32 (*. 1X, 5 (*"),
*.. 5 (*")
PRINT 38
FORMAT(4X, " ENTER YOUR TEAM SECURITY CODE IF YOU WISH TO COMMIT
to a decision; ", 30X, " OR " CONT", ")
READ IN, 125, ERR=10, END=10) INCOD
FORMAT(A8)

IF(INCCL.EQ.CMNT) RETURN
IF (INCCR.EQ.EASY) RETURN
IF (INCCR.NE.CODE) GOTO 300
       STAT(KDP) = PROP
       CALL STORE
       CALL FRDCMS('CLASCN ')
       PRINT 105, ITEAN, KDP, EBCP
       PRINT 85, KDP, ADUMP(KDP, 3)
       ADUMP(KDP, 13), ADUMP(KDP, 4), ADUMP(KDP, 14)
       ADUMP(KDP, 9), ADUMP(KDP, 15), ADUMP(KDP, 1)
       ADUMP(KDP, 16)
       PRINT 90, ADUMP(KDP, 2), ADUMP(KDP, 17)
       ADUMP(KDP, 3), ADUMP(KDP, 18)
       ADUMP(KDP, 4), ADUMP(KDP, 19)
       ADUMP(KDP, 5), ADUMP(KDP, 20)
       ADUMP(KDP, 6), ADUMP(KDP, 21)
       ADUMP(KDP, 7), ADUMP(KDP, 22)
       PRINT 110, ADUMP(KDP, 8), ADUMP(KDP, 23)
       ADUMP(KDP, 9), ADUMP(KDP, 24)
       ADUMP(KDP, 10), ADUMP(KDP, 25)
       ADUMP(KDP, 11), ADUMP(KDP, 26)
       ADUMP(KDP, 12), ADUMP(KDP, 27)
       FORMAT(* NO.*ITEM******CURRENT VALUE & DP='I',*ITEM******
       *****CURRENT VALUE */ 1. CONTRACTOR: 2x, 17, 8x, 16
       GUALTE TESTS */ $F6.2, 3. GUIDANCE CONFIGUR
       ATION: 2x, 17, 8x, 18, MAXIMUM COST $F6.2, 19, MINIMUM COST
       ATABILITY ENG $F6.2, 4X
       FORMATT(* 5. VALUE ENG $F6.2, 4X
       20. MAX COST INCENTIVE $F6.2,
       21. LATEST WEEK $F6.2,
       22. EARLIEST WEEK $F6.2,
       23. MAX DELIVER INCENTIVE $F6.2,
       24. MAXIMUM RELIABILITY $F6.2,
       25. MINIMUM RELIABILITY $F6.2,
       FORMATT(* 11. GUIDANCE ERROR YDS: $F6.2, 4X,
       FORMATT(* 12. GUIDANCE ERROR YDS: $F6.2, 4X,
** 26. MAX RELIABILITY INCENTIVE: 'P6.2
** 27. MAXIMUM ERROR YDS: 'P6.2
** 28. AIRFRAME QUAL TESTS: 'P6.2
** 29. LAUNCHER QUAL TESTS: 'P6.2
** 30. WEEK FOR IOT 10: 'P6.2

FCMAT('14X, "THE FOLLOWING PARAMETERS WERE ACCEPTED****")
"AS TEAM '12, DP= '12, 1X, A8, INPUT TO THE CONTRACTOR.")
IF(NR.EQ.0)PRINT 425
PAUSE 'PRESS <ENTER> TO CONTINUE.'
C
IF(STAT(KDF).EQ.FINAL)KDF=KDF+1
C
FORMAT(' "WARNING*****YOUR CODE WAS NOT ENTERED CORRECTLY."')
PAUSE 'PRESS <ENTER>.'
RETURN
C
SUBROUTINE PITCH
C
COMMON/PLOTA/XBASE,YBASE,YCURVE
DIMENSION XBASEx(12), YCURVE(12, 12)
REAL RBLK, T
INTEGER HORIZ, CHARR(12), J, K, M, N
INTEGER QUES(12), I, J, BLK, T
LINE(50, 22), VERT, I,
PLUS, DASH, QUOTE, CENT, DOL, $,
LESS, GREAT, COLON, SEMICO,
INTEGER SPRT, H, R/
REAL*8 CHARDP(20)
REAL*8 DVCOST, DEV COST,
REAL*8 TSTCMP, TESTCOMP,
REAL*8 RLIBIL, RELIABILITY
REAL*8 ACRACY, ACCURACY,
REAL*8 TALPEE, TLPEE,
REAL*8 TLCOST, TLCOST,
REAL*8 EFINDX, EFFINDX,
REAL*8 MULPTS, MULPTS,
CHARDP(4) = TCOST
CHARDP(6) = TSCHMP
CHARDP(8) = TIBEL
CHARDP(10) = ACRACY
CHARDP(12) = TALPEE
CHARDP(14) = TLCOST
CHARDP(16) = EFINDX
CHARDP(18) = MULPTS

C ZERCIZE 'ENTRY ROUTINE' FLAGS AND PARAMETERS
PCTFLG=0.0
YNX=99999.9
YMN=0.0

C ALL ENTRY ROUTINES & THE END OF PLTSCM JUMP IN HERE
CONTINUE

10 C INITIALIZE ALL X AND Y ARRAYS THAT ARE NOT DEFINED BY THE INPUT DATA
C PROVIDE GRAPH FORMAT FOR GRID SYSTEM
DO 40 INCRE=1,11
   DC 40 JNCRE=1,11
   DO 40 KNCRE=1,10
      DO 40 LNCRE=1,2
         JNCRE=(JNCRE-1)*5+LNCRE
         KNCRE=(KNCRE-1)*2+LNCRE
         LINE(KNCRE,JNCRE)=BLK
         IF (LNCRE.EQ.1) LINE(KNCRE,JNCRE)=PRD
         IF (JNCRE.EQ.1) LINE(KNCRE,JNCRE)=PRD
         IF (LNCRE.EQ.1) LINE(KNCRE,JNCRE)=VERT
         IF (JNCRE.EQ.1) LINE(KNCRE,JNCRE)=PLUS
      CONTINUE
   DO 40 JNCRE=1,11
      DO 40 LNCRE=1,5
         JNCRE=(JNCRE-1)*5+LNCRE
         LINE(JNCRE,KNCRE)=HORIZ
         IF (LNCRE.EQ.1) LINE(JNCRE,KNCRE)=VERT
   CONTINUE
   XMAX=0.0
C

IF(YMNX.EQ.99999.9)GOTO 62
C DETERMINE X AND Y SIT SIZE, MIN, AND MAX FOR SCALE GENERATION
YMAX=0.0
YMIN=0.0

IF(PCTFLG.NE.0.0)GOTO 63
DO 60 JNCRE=1
DO 60 INCRE=1
60 CONTINUE
GOTO 61

C THE NEXT TWO LINES SET THE Y-AXIS VALUES FOR THE SCALED ROUTINE
61

C APPROXIMATE THE AXIS SCALES.

YSIZE=ABS(YMAX-YMIN)
IF(YSIZE.EQ.0.0)YSIZE=1.0
YSCALE=(YSIZE/20.0)
XSCALE=ABS(YMAX-YMIN)
IF(XSIZE.EQ.0.0)XSIZE=1.0
GOTO 63

C ESTABLISH THE X AND Y SCALES AS STANDARD MODULUS.

C64 YLOG=ALOG10(YSCALE)
C

C CONTINUE

C74
C76  YSCALE=.10**(1YLOG10(LOG10(BIT))*SIG)
C     IF(YSCALE.LE.0.0005) YSCALE=0.0005
C     MAKE THE AXIS MIN AND MAX INTEGER MULTIPLES OF THE SCALE
C     YMIN=SIGN(INT(ABS(YMIN/YSCALE)))+1.0,YMIN
C     YMAX=SIGN(INT(ABS(YMAX/YSCALE)))+1.0,YMAX
C63  IF(PCTFLG.NE.0.0) YSCALE=0.5
C     DEVELOP LINE AND COLUMN MATRIX OF CHARACTERS
C     *CHARA* CHARACTERS REPRESENT DATA POINTS:
C     T = FLIGHT TEST COMPLETION DATE DATA
C     R = RELIABILITY DATA
C     A = ACCURACY DATA
C     % = TOTAL FEE PERCENTAGE
C     $ = TOTAL CST
C     * = EFFECTIVENESS INDEX
C     & = MULTIPLE CURVE INTERSECTION
C     CHARA(1)=CENT
C     CHARA(2)=CENT
C     CHARA(3)=T
C     CHARA(4)=R
C     CHARA(5)=R
C     CHARA(6)=R
C     CHARA(7)=AT
C     CHARA(8)=AT
C     CHARA(9)=PCT
C     CHARA(10)=DCL
C     CHARA(11)=AST
C     CHARA(12)=AMPER
C
C     DO 70 IYMM=1,12
C     DO 75 IXMM=1,11
C     DO 70 INNERW=1,21
C     YFAC=(21-INNERW)*YSCALE+YMIN
C     INNER=INNERW+1
C     DO 70 LINECL=1,51
C     IF(AES(YCURVE(IYMM,IXMM)-YFAC).LE.(0.5*YScale).AND.((IYMM-1)*5.EQ.(1
C     *LINECL-1).AND.(LINE(LINECL,INROW).NE.BLK.OR.LINE(LINECL,INROW).NE.
C     *HORIZ. OR.LINE(LINECL,INROW).NE. VERT))
C     *LINE(LINECL,INROW)=AMPER
C     IF(AES(YCURVE(IYMM,IXMM)-YFAC).LE.0.5*YScale. AND.
C     *(IYMM-1)*5.EQ.(LINECL-1).AND.LINE(LINECL,INROW).NE.AMPER)
*LINE (LINECL, INFCW) = CHARA (IYMM)
 *CONTINUE
CHARKY (4) = CENT
CHARKY (6) = T
CHARKY (8) = R
CHARKY (10) = AT
CHARKY (12) = PRTPCT
CHARKY (14) = DOI
CHARKY (16) = AST
CHARKY (18) = AMPER
CALL FRTCMS ('CLESER')
DO 60 JNROW = 1, 10
JROW = (JNROW - 1) * 2
KROW = JROW + 1
YFACT = (21 - KROW) * YSCALE + YMIN
PRINT 85, YFACT, (LINE (INCREM, KROW), INCREM = 1, 51)
85 FCMAT (P8, 2, T10, 51A1)
KROW = KROW + 1
IF (KRCW.GE.4. AND. KROW.LE.18) PRINT 89, (LINE (INCREM, KROW), INCREM = 1, 51
* 1) CHARKY (KROW), CHARDF (KROW)
89 FCMAT (T10, 51A1, 2X, A1 = ' A9')
IF (KRCW.LT.4. OR. KROW.GT.10) PRINT 90, (LINE (INCREM, KRCW), INCREM = 1, 51
* 1)
90 FCMAT (T10, 51A1)
95 CONTINUE
PRINT 95, YMIN, (LINE (KNCREM, 21), KNCREM = 1, 51)
IF (XBASE(11).GE.100.0) GOTO 147
IF (XBASE(11).GE.10.0) GOTO 148
140 PRINT 145, (XBASE (INCRE), INCRE = 1, 11)
145 FORMAT (T8, 11F5.2, '/)
C PAUSE ' ; PRESS <ENTER> TO CONTINUE.'
RETURN
147 PRINT 149, (XBASE (INCRE), INCRE = 1, 11)
149 FCMAT (T6, 11F5.0, '/)
C PAUSE ' ; PRESS <ENTER> TO CONTINUE.'
RETURN
148 PRINT 150, (XBASE (INCRE), INCRE = 1, 11)
150 FCMAT (T7, 11F5.1, '/)
C PAUSE ' ; PRESS <ENTER> TO CONTINUE.'
C...............GENERATE ENTRY CONDITIONS FOR THE FOLLOWING 'ENTRY ROUTINES'
ENTRY PLLSQL(YNN,YNH)
   PCTFLG=0.0
GOTO 10
ENTRY PLLFCT(PCITFLG)
   YNN=0.0
   YNH=99999.9
GOTO 10
END
C

SUBROUTINE FSEFF(TOC, EINDEX)
C
COMMON REST(48), TAB(20,25,3), CTEST(3), ROT(31), YNIST(2), TUSBS(48)
C
COMMON IFAC(48), MTF(48), IRC(48), IDDT(48)
COMMON RES(48), RESD(48), RESE(48), CMAT(48)
COMMON PER(48), IREV(48), YP(48), YF(48)
COMMON TSAR(48), CL(6,5), ALN(48), CT(48)
COMMON TFIN(48), THIN(48), IQ(48), CUN(30)
COMMON RESMIN(30), RESMAX(30), AK(30), I5W(3)
COMMON ITAB(25), N(8), N(8), INDX(5)
COMMON RATIO(E), AN(9), ADJ(3,8), PDR(6)
COMMON PER(6), PVR(6), CDL, IPC, INA, IMF, IDP
COMMON CPFMAX, CPFMIN, CPFMAX, CPFMIN, IITEMP(25), COV, CT, TRIAD, TRADER,
ETRADEA, DMIN1, DMIN2, DMIN, DEF, COST, COSTD, COSTP1
COMMON COSTP2, COSTOT, ITCT, ALPHA, PRICR
COMMON TRADER, TP, WA, IACT(48,2), IPC(48,5)
COMMON TDS(48,2), JMIN, IC, IFIG, KDF, M6
COMMON NC, ICO, K, KPP, IITEM
COMMON RSMAX, RSMIN, RESMAX, ESMAX, ESSENT, FESMIN, CMAX, CDMIN
COMMON RDMIN, RDMIN, RDMIN, RTDMIN, OT(25), YC(6), CT(8), DD(32), MIS(9)
REAL TRAY(20), FEERA(20), ASE/65./, THAVL/85./
C
EXPECTED AVAILABILITY= THEORETICAL AVAIL/(1+AVAILABILITY FACTOR)
AVAILABILITY FACTOR (MAINTAINABILITY FUNDS) = .29*EXP(-.44*FUNDS)
THE AVICENT FORMULA IS FROM AN EXPONENTIAL STATISTICAL CURVE FITTING SOLUTION. THE AVICENT FORMULA IS: 

\[ A = \frac{1}{e^{C_2 + m} \times 2 \\ \times \Biggl( \frac{1}{e^{C_2 + m} + 1} \Biggr)} \]

IF \( I = 0.5 \), THEN \( I = 0.5 \)

IF \( I = 0.5 \), THEN \( I = 0.5 \)

CINDEX = \( 0.99 - A \times 10 \)

END

DO 50 INDEX = 1, 28100

CONTINUE

C C

C C

C C

286
C

SUMFSE = 0.0
DO 80 I=1,20
   IF (XRAY(I).LT.ZEBRA(I))
      BIGGER = XRAY(I)
   ELSE
      BIGGER = ZEBRA(I)
   END IF
80  CONTINUE
AFSE = SUMFSE / 20.0
RETURN
END

******************************************************************************

C SUBROUTINE SNSIV

C

COMMON REST(48),TAB(20,25,3),CTEST(3),CTOH(3),XMIS(2),ISUBS(48)
E IFAC(48),N1(48),IDTT(48)
E IPD(48),ITEM(48),QTY(48),CMAT(48)
E RRES(48),RESB(48),ITYPE(48)
E PERC(48),TDUE(48),DEC(3),Y(8)
E IPRC(48),RESC(48),YF(48),RESK(48)
E TSTAR(48),CL(6,5),ALN(48),CT(48)
E TFIM(48),THIN(48),IQ(48),CUN(30)
E RESMIN(30),RESMAX(30),AK(30),ISW(3)
E ITAB(25),NN(8),N(8),INDX(6)
E RATIO(8),AN(9),ADJ(2,8),PDR(6)
E PRR(6),PVR(6),CD,ICD,IND,INF,IDP
E CPUA,CPUA1,CPUA2,CUM1,CUM2,CUM3,CUM4,CUM5
E C103,C104
E ITACT(48,2),IFAC(48,5)
E JMIN,JIC,IFIG,KDP,MI
E NC ICON,IK,ITP,ISW
E NC RMAX,RMIN,FSMAX,FSMIN,RMAX,RMIN,CDMAX,CDMIN,
E CDMAX,CDMIN,TFM1,TFM2,QT(25),Y(6),CTC(8),DD(32),MIS(9)
C COMMON/PLTV/A1SAVEM,SAVDC
C COMMON/STO/STAT(5),NDUMP(5,5),ADUMP(5,28),DPCCST,MAXDP
C REAL XFACT(11),RBLK(7),VFY(8),FRATIO(8),FYC(5)
C THE FOLLOWING SETS OF REAL DATA CREATE LABELS FOR THE TABLES.
C REAL*8 AREAS1, AREAS2, AREAS3
PROJMNG FORTRAN: AN INTERACTIVE COMPUTER PROGRAM FOR USE WITH THE DEFENSE MANAGEMENT SIMULATION EXERCISE (U)
NAVYAL POSTGRADUATE SCHOOL MONTEREY CA  G W SCHULTZ
UNCLASSIFIED MAR 84  F/G 9/2  NL
REAL*9 LATE, HIGHT, HIGH, LATE, LATE
REAL*9 LESS, LESS, BLK
REAL*9 GUES, GUES, DELAY, DEVEP, FLIGHT, FLIGHT
REAL*9 TEST, TEST, ACCUR, ACCUR, ENG, ENG
REAL*9 RELI, RELI, MOTOR, MOTOR, AIRFR, AIRFR
REAL*9 MAINT, MAINT, PARAL, PARAL, COST, COST
REAL*9 DELI, DELI, FUNDS, FUNDS, VALUE, VALUE
REAL TOWER, SAVEM, SAVEM, Y, r, s, r, s,
INTEGER TS, AM, AM, IOU, IMU

LABEL=BLK
CALL ERRSET (218, 1, -1, 1, 1)
SHE=0
BIT=0.
DO 11 ITIG=1, 11
SAVEM (ITIG) = 0.
DO 11 JITG=1, 11
SAVEM (ITIG, JITG) = 0.
CONTINUE
CALL PRINTS ('CLESCHN ')
10 PRINT 35
20 FORMAT (' *****SELECT THE ITEM WHICH IS TO BE SENSITIVITY ANALYZED.******')
35 PRINT 15
15 FORMAT (' 1. MAINTAINABILITY ENG COST 13. GUIDANCE GUES TESTS', /
  2. MAINTAINABILITY ENG COST 14. FLIGHT TESTS', /
  3. PARALLEL GUIDANCE COST 15. DEVEL COST RANGE', /
  4. PARALLEL GUIDANCE COST 16. DEVEL COST INCENTIVE %', /
  5. AIRFRAME RELIABLE RANGE 17. END WK OF FLIGHT TEST RANGE', /
  6. LAUNCHER/GSE RELIABILITY 18. FLIGHT TEST CCMP INCENTIVE', /
  7. FIRE CONTROL IMPACT ERROR 19. SYSTEM RELIABILITY RANGE', /
  8. GUIDANCE IMPACT ERROR 20. SYS RELIABILITY INCENTIVE %', /
  9. MOTOR GUES TESTS 21. IMPACT ERROR RANGE')
* 10: AIRFRAME QUAL TESTS
* 11: LAUNCHER/GSE QUAL TESTS
* 12: FIRE CONTROL QUAL TESTS
* 201, 202, 203, 204, 205, 206

READ(IN, * , EBR=20, EN=20) SENS

IF(SENS .EQ. 0) RETURN

C

DO 2600 ISEN=1,11

WHEN(1)=1

DO 150 IVER=3,15

CALL GET(IVER)

105 GOTO (200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200, 1300, 1400,
*SENS
GOTO 20

200 CONTINUE

IF(ISEN .GT. 1) GOTO 220

AREA1=MAINT

AREA2=ENGINE

AREA3=FUELS

PRINT 210

210 FORMAT(' INPUT THE VALUES TO VARY MAINT ENG FUNDS BETWEEN: ',
*'/', ' WHAT IS THE LOWER VALUE: >1.5 ??')

READ(IN, * , EBR=20, END=200) LOWER

IF(LOWER .LE. 1.5) LOWER=1.5

PRINT 215

215 FORMAT(' WHAT IS THE UPPER VALUE: <4.0 ??')

READ(IN, * , EBR=200, END=200) UPPER

IF(UPPER .GT. 4.0)

RANGE=ABS(UPPER-LOWER) * .1

220 CTC(0)=LOWER+RANGE*(ISEN-1)

IF(CTC(0) .GT. UPPER) GOTO 2620

SENVAR=CTC(0)

GOTO 2440

300 CONTINUE

IF(ISEN .GT. 1) GOTO 320
AREA1-VALUE
AREA2-VALUE
AREA3-VALUE
PRINT 310
FORMAT(' INPUT THE VALUES TO VARY VALUE ENG FUNDS BETWEEN.*
* \* WHAT IS THE LOWER VALUE. MUST BE 1.75 TO 2.2 MAX ?')
READ (IN * ERR=300, END=300) LOWER
IF (LOWER.LE.0.5) LOWER=0.5
PRINT 315
FORMAT(' WHAT IS THE UPPER VALUE. MUST BE LESS THAN 2.2?*)
READ (IN * ERR=300, END=300) UPPER
IF (UPPER.GT.3.0) UPPER=3.0
RANGE=(ABS(UPPER-LOWER)) *.1
CIC(7)=LOWER+RANGE*(ISGN-1)
IF (CIC(7).GT.UPPER) GOTO 2620
SERNVAR=CIC(7)
GOTO 2400
CONTINUE
IF (KDP.GT.3.0 OR. NIS(6).EQ.2.0) NIS(6).EQ.4) GOTO 20
IF (ISGN.GT.1) GOTO 420
AREA1-VALUE
AREA2-VALUE
AREA3-VALUE
PRINT 410
FORMAT(' INPUT THE VALUES TO VARY PARALLEL DEVELOPMENT FUNDS.*
* \* WHAT IS THE LOWER VALUE?*)
READ (IN * ERR=400, END=400) LOWER
IF (LOWER.LE.0) LOWER=0.1
PRINT 415
FORMAT(' WHAT IS THE UPPER VALUE?*)
READ (IN * ERR=400, END=400) UPPER
IF (UPPER.GT.3.0) UPPER=3.0
RANGE=(ABS(UPPER-LOWER)) *.1
CIC(5)=LOWER+RANGE*(ISGN-1)
IF (CIC(5).GT.UPPER) GOTO 2620
SERNVAR=CIC(5)
GOTO 2640
CONTINUE
IF (ISGN.GT.1) GOTO 520
IF(ISEN.GT.1) GOTO 720

AREA3=L 0 0
AREA3=M 0 0
AREA3=O 0 0
PRINT 710

710 FORMAT(' INPUT THE VALUES TO VARY LAUNCHER RELIABILITY BETWEEN."
*/* "WHAT IS THE LOWER VALUE?')
READ(IN,*)ERR=700,END=700)LOWER
IF(LOWER.LE.0)LOWER=0.01
PRINT 715

715 FORMAT(' WHAT IS THE UPPER VALUE?"
READ(IN,*)ERR=700,END=700)UPPER
IF(UPPER.GT.100)UPPER=100.
RANGE=ABS(UPPER-LOWER)GOTO 700
IF(LOWER.GT.UPPER)GOTO 700

720 YC(3)=LOWER+RANGE*.1*ISPN-1
IF(YC(3).GT.UPPER)GOTO 2620
IF(YC(3).GT.100.0)GOTO 2620
SENMAR=YC(3)
GOTO 2440

800 CONTINUE
IF(ISEN.GT.1) GOTO 820
AREA1=FIRE
AREA2=ACCUR
AREA3=UND
PRINT 810

810 FORMAT(' INPUT THE VALUES TO VARY FIRE CONTROL IMPACT BETWEEN."
*/* "WHAT IS THE LOWER VALUE. MUST BE GREATER THAN 0?"
READ(IN,*)ERR=800,END=800)LOWER
IF(LOWER.LE.0)LOWER=0.01
PRINT 815

815 FORMAT(' WHAT IS THE UPPER VALUE. MUST BE LESS THAN 100?"
READ(IN,*)ERR=800,END=800)UPPER
IF(UPPER.GT.250.0)UPPER=250.
RANGE=ABS(UPPER-LOWER)GOTO 800
IF(LOWER.GT.UPPER)GOTO 800
YC(4)=LOWER+(RANGE/ISPN-1)
GOTO 820
IF(YC(4).GT.UPPER)GOTO 2620
SENMAR=YC(4)

820 YC(4)=LOWER+(RANGE/ISPN-1)
GOTO 2620
SENMAR=YC(4)
900  GO TO 2640
   CONTINUE
   IF (ISEN.GT.1) GO TO 920
   AREA1=MOTOR
   AREA2=QUAL
   AREA3=UNDELI
   PRINT 910
910  FORMAT('INPUT THE VALUES TO VARY GUIDANCE IMPACT ERROR BETWEEN.'
         , 'WHAT IS THE LOWER VALUE?'
         , 'READ(IN,*HBR=900,END=900) LOWER
         IF (LOWER.LE.0) LOWER=0.1
         PRINT 915
   FORMA7('WHAT IS THE UPPER VALUE?'
         , 'READ(IN,*HBR=900,END=900) UPPER
         IF (UPPER.GT.250) UPPER=250.
         IF (LOWER.GT.UPPER) GO TO 500
         RANGE=(ABS(UPPER-LOWER)+1)
         IF (Y(5).GT.UPPER) GO TO 2620
         SEVFAR=Y(5)
1000  GO TO 2640
   CONTINUE
   IF (ISEN.GT.1) GO TO 1020
   AREA1=MOTOR
   AREA2=QUAL
   AREA3=UNDELI
   PRINT 1010
1010  FORMAT('INPUT THE VALUES TO VARY MOTOR QUAL TESTS BETWEEN.'
         , 'WHAT IS THE LOWER NUMBER OF TESTS? >20 ?'
         , 'READ(IN,*HBR=1000,END=1000) LOWER
         IF (LOWER.LE.20.) LOWER=20.
         PRINT 1015
1015  FORMAT('WHAT IS THE UPPER NUMBER? <40 ?'
9         'READ(IN,*HBR=1000,END=1000) UPPER
         IF (UPPER.GT.40.) UPPER=40.
         IF (LOWER.GT.UPPER) GO TO 1000
         RANGE=INT((UPPER-LOWER)+1)
         IF (RANGE.LE.1) RANGE=1.
920  QU(2)=LOWER+INT(RANGE*(ISEN-1))
         IF (QU(2).GT.UPPER) GO TO 2620
SENVAR=QT(2)
GOTO 2440
CONTINUE
IF (ISEN.GT.1) GOTO 1120
AREA1=AIRFR.
AREA2=UAL.
AREA3=TEST
PRINT 1110

1110 FORMAT(1X,'INPUT THE VALUES TO VARY AIRFRAME QUAL TESTS BETWEEN."
*./",' WHAT IS THE LOWER NUMBER OF TESTS; >3 ?")
READ(EN *. ERR=1100, END=1100) LOWER
IF (LOWER.GT.3.) LOWER=3.
PRINT 1115

1115 FORMAT(1X,'WHAT IS THE UPPER NUMBER; <9 ?")
READ(EN *. ERR=1100, END=1100) UPPER
IF (UPPER.GT.9.) UPPER=9.
IF (LOWER.GT.UPPER) GOTO 1100
RANGE=INT((UPPER-LOWER) *. 1))
IF (RANGE.LT.1.) RANGE=1.

1120 GT(3)=LOWER+INT(RANGE*(ISEN-1))
IF (GT(3).GT.UPPER) GOTO 2820
SENVAR=QT(3)

1200 CONTINUE
GOTO 2440
CONTINUE
IF (ISEN.GT.1) GOTO 1220
AREA1=LAUNCH.
AREA2=UAL.
AREA3=TEST
PRINT 1210

1210 FORMAT(1X,'INPUT THE VALUES TO VARY LAUNCHER QUAL TESTS BETWEEN."
*./",' WHAT IS THE LOWER NUMBER OF TESTS; >2 ?")
READ(EN *. ERR=1200, END=1200) LOWER
IF (LOWER.GT.2.) LOWER=2.
PRINT 1215

1215 FORMAT(1X,'WHAT IS THE UPPER NUMBER; <6 ?")
READ(EN *. ERR=1200, END=1200) UPPER
IF (UPPER.GT.6.) UPPER=6.
IF (LOWER.GT.UPPER) GOTO 1200
RANGE=INT((UPPER-LOWER) *. 1))
IF (RANGE.LT.1.) RANGE=1.
1220   QT (6) = LOWER+INT (RANGE*(ISEN-1))
        IF (QT (6) .GT. UPPER) GOTO 2620
        SENVAR=QT (6)
        GOTO 2640
1300   CONTINUE
        IF (ISEN .GT. 1) GOTO 1320
        AREA1=PIRE
        AREA2=QUAL
        AREA3=TEST
        PRINT 1310
        FORMAT (' INPUT THE VALUES TO VARY FIRE CONTROL QUAL TESTS BETWEEN
* * *
* WHAT IS THE LOWER NUMBER OF TESTS: >2 ?')
        READ (IN . *, ERR=1300, END=1300) LOWER
        IF (LOWER .LT. 2.) LOWER=2.
        PRINT 1315
        FORMAT (' WHAT IS THE UPPER NUMBER: <4 ?')
        READ (IN . *, ERR=1300, END=1300) UPPER
        IF (UPPER .GT. 4.) UPPER=4.
        RANGE=INT ((UPPER-LOWER) + 1)
        IF (RANGE .LT. 1.) RANGE=1.
1320   QT (4) = LOWER+INT (RANGE*(ISEN-1))
        IF (QT (4) .GT. UPPER) GOTO 2620
        SENVAR=QT (4)
        GOTO 2640
1400   CONTINUE
        IF (ISEN .GT. 1) GOTO 1420
        AREA1=GUIDE
        AREA2=QUAL
        AREA3=TEST
        PRINT 1410
        FORMAT (' INPUT THE VALUES TO VARY GUIDANCE QUAL TESTS BETWEEN
* * *
* WHAT IS THE LOWER NUMBER OF TESTS: >3 ?')
        READ (IN . *, ERR=1400, END=1400) LOWER
        IF (LOWER .LT. 3.) LOWER=3.
        PRINT 1415
        FORMAT (' WHAT IS THE UPPER NUMBER: <9 ?')
        READ (IN . *, ERR=1400, END=1400) UPPER
        IF (UPPER .LT. 9.) UPPER=9.
        IF (LOWER .GT. UPPER) GOTO 1400
RANGE = INT((UPPER - LOWER) * 1)

1420  QT (5) = LOWER + RANGE * (ISEN - 1)
          IF (QT (5) < GT. UPPER) GOTO 2620
          SEND = QT (5)
1500  GOTO 2440
      CONTINUE
          IF (ISEN . GT. 1) GOTO 1520
          AREA1 = FLIGHT
          AREA2 = TEST
          AREA3 = UNDEP
          PRINT 1510
1510  FORMAT(' INPUT THE VALUES TO VARY FLIGHT TESTS BETWEEN:
      * / . WHAT IS THE LOWER NUMBER OF TESTS; >10 ?')
      READ (IN, 1500, END=1500) LOWER
      IF (LOWER . LT. 10) LOWER = 10.
      PRINT 1515
1515  FORMAT(' WHAT IS THE UPPER NUMBER OF TESTS; <25 ?')
      READ (IN, 1500, END=1500) UPPER
      IF (UPPER . GT. 25) UPPER = 25.
      IF (LOWER .GT. UPPER) GOTO 1500
      RANGE = (ABS(UPPER - LOWER)) * 1
      IF (RANGE . LT. 1.) RANGE = 1.
1520  QT (1) = LOWER + INT(RANGE * (ISEN - 1))
          IF (QT (1) < GT. UPPER) GOTO 2620
          SEND = QT (1)
1600  GOTO 2440
      CONTINUE
          IF (ISEN . GT. 1) GOTO 1620
          AREA1 = DEVEL
          AREA2 = COST
          PRINT 1625
1625  FORMAT(' DO YOU WANT TO VARY THE MAXIMUM OR MINIMUM DEVELOPMENT CC
      ST? / . ENTER EITHER MAXIMUM OR MINIMUM .')
      READ (IN, 1630, ERR=1600, END=1600) LABEL
          AREA3 = LABEL
1630  FORMAT(A5)
      PRINT 1610, LABEL
1610  FORMAT(' INPUT THE LOWER VALUE FOR "A", DEVELOPMENT COST. ')
      READ (IN, "", ERR=1600, END=1600) LOWER
IF (LOWER.LE.0) LOWER=.01
PRINT 153, LABEL
1615 FORMAT('WHAT IS THE UPPER COST FOR "A8",?')
READ IN, E8, ERR=1600, END=1600, UPPER
IF (UPPER.GT.125.) UPPER=125.
IF (LOWER.GT.UPPER) GOTO 1600
RANGE=ABS(UPPER-LOWER)
1620 SENVAR=LOWER+RANGE*1*(ISEN-1)
IF (ISEN.EQ.1.AND.SENVAR.EQ.0.) PRINT 2535
IF (SENVAR.EQ.0.) GOTO 2600
IF (LABEL.EQ.N) CDH1=SENVAR
IF (LABEL.EQ.NMIN) CDH2=SENVAR
GOTO 2540
1700 CONTINUE
IF (ISEN.EQ.1) GOTO 1720
AREA1=DEVEL
AREA2=COST
AREA3=INCEN
PRINT 1710
1710 FORMAT('INPUT THE RANGE TO VARY INCENTIVE % OF DEVELOPMENT COST?"
*,/" WHAT IS THE LOWER VALUE?')
READ IN, E9, ERR=1700, END=1700, LOWER
IF (LOWER.LE.0) LOWER=.01
PRINT 1715
1715 FORMAT('WHAT IS THE UPPER VALUE?')
READ IN, E9, ERR=1700, END=1700, UPPER
IF (UPPER.GT.100.) UPPER=100.
IF (LOWER.GT.UPPER) GOTO 1700
1720 FCBMIN=LOWER+((ABS(UPPER-LOWER))*1*(ISEN-1))
IF (FCBMIN.LE.0.) PRINT 2535
IF (FCBMIN.LE.0.) GOTO 2600
DELPER=FCDMIN+FTDMIN+FRSHAX+FESMIN
SENVAR=FCDMIN
IF (DELPER.LE.15.) GOTO 2440
BATO2=(15.0-FCDMIN)/(FTDMIN+FRSHAX+FESMIN)
FTDMIN=FTDMIN+BATO2
FRSHAX=FRSHAX+RATIC2
FESMIN=FESMIN+RATIC2
GOTO 2440
1800 CONTINUE
IF (ISN. GT. 1) GOTO 1820
AREA1=FLIGHT
AREA2=TESTC
AREA3=LABEL
PRINT 1910
FORMAT(' DO YOU WANT TO VARY THE EARLIEST OR LATEST FLIGHT TEST?'
       'COMPLETION DATE? ENTER EITHER LATE OR EARLY.')
READ(IN, 1630, ERR=1800, END=1800) LABEL
AREA3=LABEL
PRINT 1910
FORMAT(' WHAT IS THE RANGE TO CHANGE 'A8,' COMPLETION DATE?'
       'WHAT IS THE LOWER VALUE?')
READ(IN, 1630, ERR=1800, END=1800) LOWER
IF (LOWER.LE. 0) LOWER=.01
PRINT 1915
FORMAT(' WHAT IS THE UPPER VALUE FOR 'A8,' COMPLETION?')
READ(IN, 1630, ERR=1800, END=1800) UPPER
IF (LOWER.GT.UPPER) GOTO 1800
RANGE=(ABS(UPPER-LOWER))*.1
IF (RANGE.LT.1.) RANGE=1.
SENVAR=LOWER+RANGE*(ISEN-1)
IF (SENVAR.GT.UPPER) SENVAR=UPPER
IF (SENVAR.LT.LATE) TDMIN=SENVAR
IF (LABEL.EQ. EARLY) TDMIN=SENVAR
GOTO 2440
1900 CCNTINUE
IF (ISEN.NE.1) GOTO 1920
AREA1=FLIGHT
AREA2=TESTC
AREA3=INCEN
PRINT 1910
FORMAT(' INPUT THE UPPER VALUE FOR DELIVERY INCENTIVE.')
READ(IN, 1630, ERR=1900, END=1900) UPPER
IF (UPPER.GT.100.) UPPER=100.
PRINT 1915
FORMAT(' WHAT IS THE LOWER VALUE FOR DELIVERY INCENTIVE?')
READ(IN, 1630, ERR=1900, END=1900) LOWER
IF (LOWER.LE. 0) LOWER=.01
IF (LOWER.GT.UPPER) GOTO 1900
1920 TDMIN=LOWER+(ABS(UPPER-LOWER))*.1 *(ISEN-1)
IF (ISEM.EQ.1.AND.FTDMIN.LE.0.) PRINT 2535
IF (FTDMIN.LE.0.) GOTO 2600
DELPFR=FTDMIN+FTDMIN+FRMAX+PESMIN
SENVAR=FTDMIN
IF (DELPFR.LE.15.) GOTO 2440
RATIO2=(15.-FTDMIN)/(FCDMIN+FRMAX+PESMIN)
FCDMIN=FCDMIN*RATIO2
FRMAX=FRMAX*RATIO2
PESMIN=PESMIN*RATIO2
GOTO 2440
2000 CONTINUE
IF (ISEM.NE.1) GOTO 2020
AREA1=RELI
PRINT 2005
2005 FORMAT(" DO YOU WANT TO VARY THE UPPER OR LOWER LEVEL OF RELIABILITY?
\*Y?",/*/ INPUT HIGH OR LOW")
READ (IN, 1630, ERR=2000, END=2000) LABEL
AREA2=LABEL
AREA3=UNDEL
PRINT 2010, LABEL
2010 FORMAT(" WHAT IS THE UPPER VALUE FOR 'AB', RELIABILITY?")
READ (IN, *) ERR=2000, END=2000) UPPER
IF (UPPER.GT.100.) UPPER=100.
PRINT 2015, LABEL
2015 FORMAT(" WHAT IS THE LOWER VALUE FOR 'AB', RELIABILITY?")
READ (IN, *) ERR=2000, END=2000) LOWER
IF (LOWER.LE.0.) LOWER=0.1
IF (LOWER.GT.UPPER) GOTO 2000
2020 SENVAR=LOWER+(ABS(UPPER-LOWER)) *.1* (ISEM-1)
IF (ISEM.EQ.1.AND. (SENVAR.GT.100..OR.SENVAR.LT.0.) PRINT 2535
IF (SENVAR.GT.100..OR.SENVAR.LT.0.) GOTO 2440
IF (LABEL.EQ.HIGH) RSMAX=SENVAR
IF (LABEL.EQ.LOW) RSHMIN=SENVAR
GOTO 2440
2100 CONTINUE
IF (ISEM.NE.1) GOTO 2120
AREA1=RELI
AREA2=NUMEN
AREA3=UNDEL
PRINT 2110
2110 FORMAT('INPUT THE UPPER VALUE OF INCENTIVE % FOR RELIABILITY.')
    READ (IN, * ERR=2, END=2100) UPPER
    IF (UPPER.GT.100.) UPPER=100.
    PRINT 2115
2115 FC5MA1('WHAT IS THE LOWER VALUE?')
    READ (IN, * ERR=2, END=2100) LOWER
    IF (LOWER.LE.0.) LOWER=0.
    IF (LOWER.GT.UPPER) GOTO 2100
    FRMAX=LOWER+(ABS(UPPER-LOWER))*1*(ISEN-1)
    DELPER=FCDHM1+FTDHM1+FRMAX+FEMIN
    SENVAR=FRMAX
    IF (FRMAX.LE.0.) PRINT 2535
    IF (FRMAX.LE.1.) GOTO 2600
    IF (DELPER.LE.15.) GOTO 2440
    RATIO2=(15.-FRMAX)/(FCDHM1+FRMAX+FEMIN)
    FCHM1=FCDHM1+RATIO2
    FTDHM1=FTDHM1*RATIO2
    FEMIN=FEMIN*RATIO2
    GOTO 2440
2200 CONTINUE
    IF (ISEN.NE.1) GOTO 2220
    AREA1=PBEC
    PRINT 2210
2210 FC5MA1('DO WANT TO CHANGE THE GREATER OR LESSER ACCURACY?'/
    * 'INPUT EITHER GREATER OR LESSER.' )
    READ (IN, 1630, ERR=2, END=2200)
    AREA2=LABEL
    AREA3=UNDL
    PRINT 2215,LABEL
2215 FORMAT('WHAT IS THE UPPER VALUE FOR 'A8,' ACCURACY?')
    READ (IN, * ERR=2, END=2200) UPPER
    IF (UPPER.GT.300.) UPPER=300.
    PRINT 2216,LABEL
2216 FORMAT('WHAT IS THE LOWER VALUE FOR 'A8,' ACCURACY?')
    READ (IN, * ERR=2, END=2200) LOWER
    IF (LOWER.LE.0.) LOWER=.01
    IF (LOWER.GT.UPPER) GOTO 2200
    SENVAR=LOWER+(ABS(UPPER-LOWER))*1*(ISEN-1)
    IF (SENVAR.LE.0.) GOTO 2620
    IF (LABEL.EQ.GREAT) ESMAK=SENVAR
IF (LABEL-1) THEN EQ-10
ELSE SHIM=SEVAR

CTR=4601
AREA=PRICE
AREA2=PRICE
AREA3=PRICE

2315 READ IN P R I C E (O O ) (P R I C E ) (L O W E R ) (U P P E R )
2320 IF (UPPER .GT. 15) THEN
2400 CONTINUE
2535 FORMAT (1X,'WHAT IS THE FORMER DELIVERY DATE?')
2600 FORMAT (1X,'WHAT IS THE LATER DATE?')
READ (IN, * ERR=2400, END=2400) UPPER
  IF (LOWER .GT. UPPER) GOTO 2400
  RANGE = (ABS (UPPER - LOWER)) .GT. 1.
  IF (RANGE .LT. 1.) RANGE = 1.
  DD (13) = LOWER + INT (RANGE * (ISEN - 1))
  IF (DD (13) .GT. UPPER) GOTO 2620
  IF (DD (13) .LE. 0. AND. ISEN .EQ. 1) PRINT 2535
  IF (DD (13) .LT. 0.) GOTO 2600
  SENVAR = DD (13)
  CALL INPUT3
  CALL PRESET
  CALL TRADE
  CALL DESRES
  GOTO (2445, 2450), KKPP
2445 CALL DESRES
2450 CALL SET
  IF (NTEMP (1) .EQ. 3 .OR. NTEMP (1) .EQ. IDP) GOTO 130
  ITCT = 0
  DO 110 N = 1, INA
      IF (CTY (N) .LE. 0.1) 110, 110, 115
      IF (ITEM (B) - IDP - 1) 110, 120, 110
  110 CONTINUE
  GO TO 140
  IDP = 7
  ITCT = 1
  ITAB (25) = IDP
240 C 140 CALL MOD9
  IF (IVER2 .EQ. KDP) GOTO 150
  IF (IVER2 .NE. KDP - 1) GOTO 155
245 CALL REGEN
  CONTINUE
255 CALL REPORT
  CALL PERF
  CALL GET (KDP)
  IF (SENSYT .EQ. 1.0) CTC (0) = SENVAR
  IF (SENSYT .EQ. 2.0) CTC (7) = SENVAR
  IF (SENSYT .EQ. 3.0) CTC (5) = SENVAR
C H = H M

3315 \{1\}(N) = \{FIN\}(N) \{1\}(N) = \{FIN\}(N)
Y \{1\} = \{TAB\}(14) \{1\}(N) = \{TAB\}(14) \{1\}(N) = \{TAB\}(14)
C 30 K = 4
Y \{2\} = \{1\}(N) * \{YP\}(N)
Y \{2\} = \{1\}(N) * \{YP\}(N)
C

PER \{1\} = \{CDMAX\}
AN \{1\} = \{CDMAX\}
FYC \{1\} = \{PCDMIN\}
PER \{2\} = \{CDMAX\}
AN \{2\} = \{CDMAX\}
FYC \{2\} = \{PCDMIN\}
PER \{3\} = \{BRSMAX\}
AN \{3\} = \{BRSMAX\}
FYC \{3\} = \{BRSMAX\}
PER \{4\} = \{BRSMAX\}
AN \{4\} = \{BRSMAX\}
FYC \{4\} = \{BRSMAX\}
PER \{5\} = \{CPUMAX\}
AN \{5\} = \{CPUMAX\}
FYC \{5\} = \{PCPH\}
SET \{4\} = 0.0
FRATIO \{4\} = 0.0
Y \{1\} = \{1\}(N) / 1000000.0
Y \{3\} = \{1\}(N) / 100.0
C
C THIS SET OF STEPS IN TABLE 50 CHECKS THE MOST DESIRABLE VALUES.
C AN \{1\} VS ACHIEVED VALUES, \{Y\}(N), AND LEAST DESIRABLE VALUE, PER \{1\},
C VS ACHIEVED VALUE IN ORDER TO PLACE THE PER WITHIN THE LINEAR RATIO
C BOUNDS OF THE DESIRABLE PER.
DO 50 I = 1, 5
IF (I.EQ.3) GOTO 48
IF (Y \{I\} = AN \{I\} .GE. 0 .AND. PER \{I\} = Y \{I\} .GE. 0 .AND. PER \{I\} .GE. AN \{I\})
IF (Y \{I\} = \{PER\}(I) = \{ABS\} (\{PER\}(I) - \{Y\}(I)) / (\{PER\}(I) - \{AN\}(I)))
IF (Y \{I\} = \{PER\}(I) .GE. 0.) FRATIO \{I\} = 0.0
IF (AN (1) - Y (1) * .61) .FRATIO (1) = FVC (1)
GOTO 49

48 IF (Y (3) - PER (3) * .61) .AND. (X (3) - Y (3) * .61) .AND. (AN (3) - M.E. PER (3))
  IF ((Y (3) - X (3)) * .61) .AND. (AN (3) - PER (3))
    IF (Y (3) - X (3))  * .61) .FRATIO (3) = .00
    IF (Y (3) - AN (3)) .AND. (AN (3) - PER (3))
      YPER (1) = FRATIO (3) * (CNDX + CNDXN) / 200.0
      SFVA = SFVA + YPER (1)
      FRATIO (3) = FRATIO (3) + 1
      CONTINUE

T0C = Y (1) * SFVA + DECOST / 1000000.0
C
C CHANGED Y TO Y2 TO DIFFERENTIATE FROM Y4 BELOW. 2/84
C
Y2 = Y (2)
C
C CHANGED Y TO Y4 TO DIFFERENTIATE FROM Y2 ABOVE. 2/84

Y4 = TAB (14, 16, 2) * 100.

C
C CPU1A1 = CPUH3 / 1000.
C CPUH3 = CPUH3 / 1000.

IF (ISEN = EQ. 1) CALL FETCHS (*CLRSCRN *)
IF (ISEN = EQ. 1) PRINT 2535, AREA1
IF (ISEN = EQ. 1) PRINT 2536, AREA2, AREA3
CALL FREEPP (T0C, EI)
EI = ABS (EI)

2535 FORMAT (1X, 69*(1 / 1X)
* SENSITIV\DEVELOPMENT\FLIGHTTEST\RELIABILITY\ACCURACY\FEE\CONTRACT
* 1 / 1X \PRE\COST\FEE\DATE\FEE\%\FEE\YDS\%\COST
* 1 / 1X
2536 FORMAT (1X, 1 / 1X
* 1%\COMPLT\%\\%\TGT\$\M
* 1 / 1X
J = ISEN - 1

PRINT 2525, J, SENVAR, FRATIO (1), Y (1), FRATIO (2), Y2, FRATIO (3),
ENTRY OPTIM
IF(SAVESEN(SELI).LE.0.0) RETURN
CALL GET(HDB)
4005 GOTO(4010,4020,4030,4040,4050,4060,4070,4080,4090,4100,4110,4120,
*4130,4140,4150,4160,4170,4180,4190,4200,4210,4220,4230)
*SENSTI
4010 CONTINUE
   CIC(8)=SAVESEN(SELI)
   GOTO 4400
4020 CONTINUE
   CIC(7)=SAVESEN(SELI)
   GOTO 4400
4030 CONTINUE
   CIC(5)=SAVESEN(SELI)
   GOTO 4400
4040 CONTINUE
   IC(1)=SAVESEN(SELI)
   GOTO 4400
4050 CONTINUE
   IC(2)=SAVESEN(SELI)
   GOTO 4400
4060 CONTINUE
   IC(3)=SAVESEN(SELI)
   GOTO 4400
4070 CONTINUE
   IC(4)=SAVESEN(SELI)
   GOTO 4400
4080 CONTINUE
   IC(5)=SAVESEN(SELI)
   GOTO 4400
4090 CONTINUE
   CI(2)=SAVESEN(SELI)
   GOTO 4400
4100 CONTINUE
   CI(3)=SAVESEN(SELI)
   GOTO 4400
CONTINUE
CT(6) = SAVSEN(SELX)
GOTO 4400

CONTINUE
CT(4) = SAVSEN(SELX)
GOTO 4400

CONTINUE
CT(5) = SAVSEN(SELX)
GOTO 4400

CONTINUE
CT(1) = SAVSEN(SELX)
GOTO 4400

CONTINUE
IF (LABEL = EQ. MAX) CMAX = SAVSEN(SELX)
IF (LABEL = EQ. ERLY) CMIN = SAVSEN(SELX)
GOTO 4400

CONTINUE
FCCHIN = SAVSEN(SELX)
GOTO 4400

CONTINUE
IF (LABEL = EQ. LATE) TDHIN = SAVSEN(SELX)
IF (LABEL = EQ. ERLY) TDHIN = SAVSEN(SELX)
GOTO 4400

CONTINUE
FCHIN = SAVSEN(SELX)
GOTO 4400

CONTINUE
IF (LABEL = EQ. HIG) RSHIN = SAVSEN(SELX)
IF (LABEL = EQ. LOW) RSHIN = SAVSEN(SELX)
GOTO 4400

CONTINUE
FRESH = SAVSEN(SELX)
GOTO 4400

CONTINUE
IF (LABEL = EQ. GREAT) RSH = SAVSEN(SELX)
IF (LABEL = EQ. LOW) RSH = SAVSEN(SELX)
GOTO 4400

CONTINUE
FRESH = SAVSEN(SELX)
GOTO 4400
CONTINUE  DE(13) = SAVSEN (SELX)
CONTINUE  CALL SETUP (KDP)
ENTRY PLOTST (ISEN)
RETURN
END

C************************************************************
C
C SUBROUTINE PRCICK
C
COMMON REST(48), TAB(20, 25, 3), CTEST(3), CTOH(3), IWIN(2), ISUBS(48)
& : IFAC(48), MT(48), IRC(49), IDDT(48)
& : IPD(48), ITEM(48), QTY(48), CHT(48)
& COMMON RESS(48), RESD(48), RESB(48), ITYPE(48)
& : PERC(48), TABR(48), LED(3), Y(3)
& : TXT(48), CL(6, 5), ALN(48), CT(48)
& : TFIN(48), THIN(48), IQ(48), CUN(30)
COMMON RESMIN(30), RESMAX(30), AK(30), ISW(3)
& : ITAB(25), MN(8), MN(8), INDX(8)
& : RATIO(8), AN(9), ADJ(2, 8), PDR(6)
& : PER(6), VCR(6), CD, ICD, IIN, INP, IDP
COMMON CPUMAX, CEMIN, CCPMIN, Y, ATMP(25), COV, C10, TRADET, TRADER,
& : DAM1, DAM2, DFR, SOR, CSTH, COSTD, COSTP
COMMON COSTH, COSTD, IACT, ALPHA, PRIOR
& : TRADEP, AT, IC, NIN, IFIG, KDP, NE
& NC ICONF, KP, KK1, IEIN
COMMON RSMIN, RSMAX, PRSMAX, ESMAK, ESMIN, FESMIN, CDMAX, CDMIN,
& : CDMIN, TDMIN, TDMIN, QT(25), IC(6), CTC(6), DD(32), H1S(9)

INTEGER IN, J
REAL AWS, CHG, 'CHG', /,, Y,, /, YES/'Y' '/,
REAL*4 MORE, 'MORE', 'MORE', MORE, /,
REAL*4 GREAT, 'GREATER', 'LESS', 'LESS', ',', SIZE

PRTOT = FCDMIN + TDIN + PRSMAX + FESMIN
IF (PRTOT .GT. 14.99999 .AND. PRTOT .LT. 15.00001) RETURN
CALL FRTCMS ('C15ESCRR '
IF (PRTOT .GT. 15.) SIZE = MORE
IF (PTOT < 15) SIZE = LESS
5 PRINT 510 SIZE, PTOT, PTOT, PTOT, PSMAX, FESMIN
5 FORMAT(' ** B A R R N I N G ** TOTAL OF FEE PERCENTAGES IS ' , 48, ' THAN
5 .15%', ') /
5 * YOU PREVIOUSLY SUBMITTED ARE: '/ . COST FEE INCENTIVE=', '18.1', ';
5 ** DELIVERY DATE FEE INCENTIVE=', '12.2', ' RELIABILITY INCENTIVE FEE
5 * 15.2', ' ACCURACY INCENTIVE FEE=', '15.2', ' REDUCE YOUR INPUT. ' / )
500 PRINT 510
510 FORMAT(' ENTER THE COST INCENTIVE FEE. ')
5 CALL CHANGE (PCTAIN)
5 PCTAIN = PCDTIN
5 IF (PTOT < 15.) GOTO 600
5 PRINT 605 PCTR DC
605 FORMAT(' TOTAL OF INCENTIVE FEES (', '18.1', ') EXCEEDS 15%. ')
5 CALL EXIT5
5 GOTO 400
600 DIFF = 15. - PERTOT
5 PRINT 615, DIFF
615 FORMAT(' YOU HAVE ', '18.1', '% OF THE INCENTIVE FEES REMAINING. ')
5 PRINT 515
515 FORMAT(' ENTER THE DELIVERY DATE INCENTIVE FEE. ')
5 CALL CHANGE (PTRAIN)
5 PCTAIN = PCDTIN + ETAIN
5 IF (PTOT < 15.) GOTO 620
5 PRINT 605 PCTR EC
5 CALL EXIT5
5 GOTO 500
620 DIFF = 15. - PERTOT
5 PRINT 615, DIFF
615 FORMAT(' ENTER THE RELIABILITY INCENTIVE FEE. ')
5 CALL CHANGE (PRTMAX)
5 PRTMAX = PCDTIN + ETAIN + FRMAX
5 IF (PTOT < 15.) GOTO 625
5 PRINT 605 PCTR IC
5 CALL EXIT5
5 GOTO 620
625 DIFF = 15. - PERTOT
5 PRINT 615, DIFF
5 PRINT 525
### APPENDIX H

#### PROGRAM VARIABLES GLOSSARY

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAA</td>
<td>Character string for 'accuracy'</td>
</tr>
<tr>
<td>ACCUR</td>
<td>Recursiv TVU sum</td>
</tr>
<tr>
<td>ADJ</td>
<td>Array variable for data file fields 6 through 33</td>
</tr>
<tr>
<td>ADUMP</td>
<td>Average Force Structure Effectiveness</td>
</tr>
<tr>
<td>APSE</td>
<td>Character string for 'airframe'</td>
</tr>
<tr>
<td>AIRFR</td>
<td>Character string for 'airframe'</td>
</tr>
<tr>
<td>AN</td>
<td>Character for curve plot;</td>
</tr>
<tr>
<td>ANO</td>
<td>Variable for answer to Yes/No questions</td>
</tr>
<tr>
<td>ANS</td>
<td>Negative answer to Yes/No questions</td>
</tr>
<tr>
<td>ANT</td>
<td>Variable for answer to Yes/No questions</td>
</tr>
<tr>
<td>AREA1</td>
<td>Character buffer for creating labels in Sensitivity Table</td>
</tr>
<tr>
<td>AREA2</td>
<td>Character buffer for creating labels in Sensitivity Table</td>
</tr>
<tr>
<td>AREA3</td>
<td>Character buffer for creating labels in Sensitivity Table</td>
</tr>
<tr>
<td>AT</td>
<td>Character a</td>
</tr>
<tr>
<td>BIGGER</td>
<td>Character string used in Sensitivity Table label</td>
</tr>
<tr>
<td>BLK</td>
<td>Character string for '000000000'</td>
</tr>
<tr>
<td>CD</td>
<td>Development cost</td>
</tr>
<tr>
<td>CDMAX</td>
<td>Maximum development cost</td>
</tr>
<tr>
<td>CDMIN</td>
<td>Minimum development cost</td>
</tr>
<tr>
<td>CENT</td>
<td>Character for curve plot;</td>
</tr>
<tr>
<td>CHANGE</td>
<td>Subprogram for changing real variables</td>
</tr>
<tr>
<td>CHANGI</td>
<td>Subprogram for changing imaginary variables</td>
</tr>
<tr>
<td>CHARA</td>
<td>Character string buffer to put characters in plot lines</td>
</tr>
<tr>
<td>CHARKY</td>
<td>Array for character key characters</td>
</tr>
<tr>
<td>CHARKDF</td>
<td>Array for character key definitions</td>
</tr>
<tr>
<td>CKPIIT</td>
<td>Subroutine to check the availability of plot data</td>
</tr>
<tr>
<td>--------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>CL</td>
<td>Common buffer label</td>
</tr>
<tr>
<td>CMATE</td>
<td>Character for plot curve label:</td>
</tr>
<tr>
<td>CMATE</td>
<td>Character for plot curve label:</td>
</tr>
<tr>
<td>CON</td>
<td>Cost of development</td>
</tr>
<tr>
<td>COST1</td>
<td>Cost of manufacture</td>
</tr>
<tr>
<td>C0ST2</td>
<td>Cost total</td>
</tr>
<tr>
<td>COST3</td>
<td>Cost parallel contractor a</td>
</tr>
<tr>
<td>COST4</td>
<td>Cost parallel contractor b</td>
</tr>
<tr>
<td>CPU1MAX</td>
<td>Cost of production maximum</td>
</tr>
<tr>
<td>CPU2MIN</td>
<td>Cost of production minimum</td>
</tr>
<tr>
<td>CTC</td>
<td>Cost of tests total</td>
</tr>
<tr>
<td>CT0ST</td>
<td>Input desired engineering costs</td>
</tr>
<tr>
<td>CV0U</td>
<td>Cost of system tests</td>
</tr>
<tr>
<td>CV0U0</td>
<td>Cost of tests overhead</td>
</tr>
<tr>
<td>D1</td>
<td>Character for curve plot:</td>
</tr>
<tr>
<td>D2</td>
<td>File for LATACODE TEAMxx</td>
</tr>
<tr>
<td>DASH</td>
<td>File for DATAMEX TEAMxx</td>
</tr>
<tr>
<td>DATCC0D</td>
<td>Input desired Delivery Date</td>
</tr>
<tr>
<td>DATFIL</td>
<td>Decrease for smallest performance factor impact</td>
</tr>
<tr>
<td>DC0</td>
<td>Decrease for smallest performance factor impact</td>
</tr>
<tr>
<td>DECK</td>
<td>Decrease for smallest performance factor impact</td>
</tr>
<tr>
<td>DECK0M</td>
<td>Decrease for smallest performance factor impact</td>
</tr>
<tr>
<td>DE0F</td>
<td>Character string for 'delivery' in sensitivity Table</td>
</tr>
<tr>
<td>DELAY</td>
<td>Subroutine to decrease resources</td>
</tr>
<tr>
<td>DELIV</td>
<td>Subroutine to decrease resources because of tests</td>
</tr>
<tr>
<td>DELPER</td>
<td>Character string for 'development' in Sensitivity Table</td>
</tr>
<tr>
<td>DESRES</td>
<td>Decision Point cost</td>
</tr>
<tr>
<td>DESRET</td>
<td>Differential value of flight tests</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>DOT2</td>
<td>Differential value of motor quality tests</td>
</tr>
<tr>
<td>DOT3</td>
<td>Differential value of airframe quality testing</td>
</tr>
<tr>
<td>DOT4</td>
<td>Differential value of fire control quality testing</td>
</tr>
<tr>
<td>DOT5</td>
<td>Differential value of guidance quality testing</td>
</tr>
<tr>
<td>DOT6</td>
<td>Differential value of launcher quality testing</td>
</tr>
<tr>
<td>DMAX</td>
<td>Differential value applied to maximum reliability</td>
</tr>
<tr>
<td>EEARLY</td>
<td>Character string for determining sensitivity analysis selection</td>
</tr>
<tr>
<td>E</td>
<td>Effectiveness index</td>
</tr>
<tr>
<td>EIOPT</td>
<td>Optimum effectiveness index</td>
</tr>
<tr>
<td>ENGIN</td>
<td>Character string for Sensitivity Table 'engineering'</td>
</tr>
<tr>
<td>ERRSET</td>
<td>Macro subroutine for resetting error functions on the IBM</td>
</tr>
<tr>
<td>ESMAX</td>
<td>Accuracy error maximum</td>
</tr>
<tr>
<td>ESMIN</td>
<td>Accuracy error minimum</td>
</tr>
<tr>
<td>EXIT</td>
<td>Subroutine to choose to stop or reloop back into the program</td>
</tr>
<tr>
<td>EXP</td>
<td>Fee for development cost incentive</td>
</tr>
<tr>
<td>FCDBMIN</td>
<td>Fee for production cost incentive</td>
</tr>
<tr>
<td>FCPRIN</td>
<td>Total of fees</td>
</tr>
<tr>
<td>PDTBN</td>
<td>Fee for accuracy incentive</td>
</tr>
<tr>
<td>PDIFFN</td>
<td>Label for team data files to identify final DF proposal</td>
</tr>
<tr>
<td>FINISH</td>
<td>Subroutine for submitting final and proposal contract proposal</td>
</tr>
<tr>
<td>FIREC</td>
<td>Character string for Sensitivity Table 'fire control'</td>
</tr>
<tr>
<td>FLAG</td>
<td>Flag for generating variation to the Selection Table</td>
</tr>
<tr>
<td>FLIGHT</td>
<td>Character string for the Sensitivity Table 'flight'</td>
</tr>
<tr>
<td>PL21</td>
<td>Flag for the monitors sessions</td>
</tr>
<tr>
<td>FRESHX</td>
<td>Fee for reliability incentive</td>
</tr>
<tr>
<td>FRESHS</td>
<td>Macro subroutine for clearing the screen in the IBM</td>
</tr>
<tr>
<td>FSE</td>
<td>Force structure effectiveness</td>
</tr>
<tr>
<td>FSEFF</td>
<td>Subroutine for force structure effectiveness</td>
</tr>
<tr>
<td>FDPMIN</td>
<td>Fee for test completion date incentive</td>
</tr>
<tr>
<td>FUNDS</td>
<td>Character string for Sensitivity Table label 'funds'</td>
</tr>
<tr>
<td>GET</td>
<td>Subroutine to place data file buffer data in array buffer</td>
</tr>
<tr>
<td>GREAT</td>
<td>Sensitivity selection label for parameter choice</td>
</tr>
<tr>
<td>IRC</td>
<td>save sensitivity data for flight test</td>
</tr>
<tr>
<td>-----</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>ISAVE2</td>
<td>date</td>
</tr>
<tr>
<td>ISAVE4</td>
<td>save sensitivity data for accuracy</td>
</tr>
<tr>
<td>ISBN</td>
<td>for do loop of 10 iterations in</td>
</tr>
<tr>
<td></td>
<td>Sensitivity Table</td>
</tr>
<tr>
<td>ISUBS</td>
<td>Team number</td>
</tr>
<tr>
<td>ISW</td>
<td>Team number</td>
</tr>
<tr>
<td>ITAB</td>
<td>Team number</td>
</tr>
<tr>
<td>ITCT</td>
<td>Team number</td>
</tr>
<tr>
<td>ITEM</td>
<td>Team number</td>
</tr>
<tr>
<td>ITEAMB</td>
<td>Team number</td>
</tr>
<tr>
<td>ITEAM</td>
<td>Team number</td>
</tr>
<tr>
<td>ITERM</td>
<td>Team number</td>
</tr>
<tr>
<td>ITPF</td>
<td>Team number</td>
</tr>
<tr>
<td>IVER</td>
<td>DP loops of the main program</td>
</tr>
<tr>
<td>IVER2</td>
<td>Do loops of the Sensitivity achieved</td>
</tr>
<tr>
<td>IXBASE</td>
<td>value loop</td>
</tr>
<tr>
<td>IXM2</td>
<td>Integer value for plot x-axis</td>
</tr>
<tr>
<td>IXMM</td>
<td>Integer for flight test completion date</td>
</tr>
<tr>
<td>IY2</td>
<td>Integer for accuracy</td>
</tr>
<tr>
<td>IY4</td>
<td>Integer for do loops</td>
</tr>
<tr>
<td>J1</td>
<td>Integer for do loops</td>
</tr>
<tr>
<td>J2</td>
<td>INTEGER PCN DP</td>
</tr>
<tr>
<td>J3</td>
<td>Integer for do loops</td>
</tr>
<tr>
<td>JA</td>
<td>Integer for do loops</td>
</tr>
<tr>
<td>JDGP</td>
<td>Integer for do loops</td>
</tr>
<tr>
<td>JGITIG</td>
<td>Integer for do loops</td>
</tr>
<tr>
<td>JGITIGA</td>
<td>Integer for do loops</td>
</tr>
<tr>
<td>JJ</td>
<td>Integer for do loop finding minimum axis</td>
</tr>
<tr>
<td>JMINT</td>
<td>values</td>
</tr>
<tr>
<td>K</td>
<td>Integer dc loop variable</td>
</tr>
<tr>
<td>KA</td>
<td>Current DF</td>
</tr>
<tr>
<td>KB</td>
<td>Assigned goto</td>
</tr>
<tr>
<td>KDP</td>
<td>Assigned goto</td>
</tr>
<tr>
<td>KGO</td>
<td>Assigned goto</td>
</tr>
<tr>
<td>KGOA</td>
<td>Assigned goto</td>
</tr>
<tr>
<td>KGOB</td>
<td>Assigned goto</td>
</tr>
<tr>
<td>KTO</td>
<td>Assigned goto</td>
</tr>
<tr>
<td>KRP</td>
<td>Flag for data input at current dp</td>
</tr>
<tr>
<td>KP</td>
<td>Flag for data input at DP-3</td>
</tr>
<tr>
<td>KTB</td>
<td>Flag for data input at DP-3</td>
</tr>
<tr>
<td>KTR101</td>
<td>Flag for data input at DP-3</td>
</tr>
<tr>
<td>KZ</td>
<td>Sensitivity selection parameter choice</td>
</tr>
<tr>
<td>KZAW</td>
<td></td>
</tr>
<tr>
<td>Character string for Sensitivity Table</td>
<td>Character string for Sensitivity Table</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>Character &lt;</td>
<td>Character string for Sensitivity Table</td>
</tr>
<tr>
<td>Plot line array</td>
<td>Plot line array</td>
</tr>
<tr>
<td>Integer increment</td>
<td>Integer increment</td>
</tr>
<tr>
<td>Sensitivity selection parameter choice</td>
<td>Sensitivity selection parameter choice</td>
</tr>
<tr>
<td>Character string for Sensitivity Table</td>
<td>Character string for Sensitivity Table</td>
</tr>
<tr>
<td>MINIMUM FLIGHT TEST COST</td>
<td>MINIMUM FLIGHT TEST COST</td>
</tr>
<tr>
<td>Subroutine for achieved value calculation</td>
<td>Subroutine for achieved value calculation</td>
</tr>
<tr>
<td>Sensitivity selection parameter choice</td>
<td>Sensitivity selection parameter choice</td>
</tr>
<tr>
<td>Character string for Sensitivity Table</td>
<td>Character string for Sensitivity Table</td>
</tr>
<tr>
<td>Monitor mode selected</td>
<td>Monitor mode selected</td>
</tr>
<tr>
<td>Contractor choice</td>
<td>Contractor choice</td>
</tr>
<tr>
<td>Team data file first five fields</td>
<td>Team data file first five fields</td>
</tr>
<tr>
<td>Newly selected DP level</td>
<td>Newly selected DP level</td>
</tr>
<tr>
<td>Table number from which to read data</td>
<td>Table number from which to read data</td>
</tr>
<tr>
<td>Temporary buffer for DP</td>
<td>Temporary buffer for DP</td>
</tr>
<tr>
<td>Character string for Sensitivity Table</td>
<td>Character string for Sensitivity Table</td>
</tr>
<tr>
<td>'optimum'</td>
<td>'optimum'</td>
</tr>
<tr>
<td>Subroutine for page input</td>
<td>Subroutine for page input</td>
</tr>
<tr>
<td>Variable for pages to be input</td>
<td>Variable for pages to be input</td>
</tr>
<tr>
<td>Flag for per plotting percentage plot</td>
<td>Flag for per plotting percentage plot</td>
</tr>
<tr>
<td>Total percentage</td>
<td>Total percentage</td>
</tr>
<tr>
<td><strong>GUIDE</strong></td>
<td>Character string for Sensitivity Table 'guidance'</td>
</tr>
<tr>
<td><strong>HIGH</strong></td>
<td>Sensitivity selection label for parameter choice</td>
</tr>
<tr>
<td><strong>HORIZ</strong></td>
<td>Character for plot lines:</td>
</tr>
<tr>
<td><strong>IAC</strong></td>
<td>Integer variable for incrementing do loops</td>
</tr>
<tr>
<td><strong>IC</strong></td>
<td>Incremental costs overall</td>
</tr>
<tr>
<td><strong>ICD</strong></td>
<td>Incremental development costs</td>
</tr>
<tr>
<td><strong>ICODE</strong></td>
<td>Security code buffer</td>
</tr>
<tr>
<td><strong>ICODE</strong></td>
<td>Integer cost of guidance</td>
</tr>
<tr>
<td><strong>ID</strong></td>
<td>Integer delivery date</td>
</tr>
<tr>
<td><strong>IDIR</strong></td>
<td>Initial DP computed for</td>
</tr>
<tr>
<td><strong>IDUMF</strong></td>
<td>Integer variable array for first five fields of team data file</td>
</tr>
<tr>
<td><strong>IPAC</strong></td>
<td>Count for security code violations</td>
</tr>
<tr>
<td><strong>IPD</strong></td>
<td>Integer for do loop variable referring to DP as stored KDP</td>
</tr>
<tr>
<td><strong>IP</strong></td>
<td>Integer for do loop incrementing</td>
</tr>
<tr>
<td><strong>IN</strong></td>
<td>Initial accuracy from tables</td>
</tr>
<tr>
<td><strong>INCEN</strong></td>
<td>Character string for Sensitivity Table 'incentive'</td>
</tr>
<tr>
<td><strong>INCOD</strong></td>
<td>Security code input from the team violations</td>
</tr>
<tr>
<td><strong>INDEX</strong></td>
<td>Index for selecting table variables</td>
</tr>
<tr>
<td><strong>INP</strong></td>
<td>Subroutine for input of data to current DP</td>
</tr>
<tr>
<td><strong>INPUT12</strong></td>
<td>Subroutine for input of data from table files</td>
</tr>
<tr>
<td><strong>INTEN</strong></td>
<td>Proposal status</td>
</tr>
<tr>
<td><strong>TREN</strong></td>
<td></td>
</tr>
<tr>
<td><strong>TPG</strong></td>
<td></td>
</tr>
<tr>
<td><strong>TPBC</strong></td>
<td></td>
</tr>
<tr>
<td><strong>IQ</strong></td>
<td>Integer for question response for percentage plot</td>
</tr>
<tr>
<td><strong>IQUEST</strong></td>
<td>Question response: exit</td>
</tr>
<tr>
<td><strong>IQUESM</strong></td>
<td>Question response: monitor's main menu</td>
</tr>
<tr>
<td><strong>IQUESTO</strong></td>
<td>Question response: Use optimum el for parameter value</td>
</tr>
<tr>
<td><strong>IQUIESP</strong></td>
<td>Question response: sensitivity table plot</td>
</tr>
<tr>
<td><strong>IQUEST</strong></td>
<td>Question response: sensitivity table print</td>
</tr>
<tr>
<td><strong>IQUESTS</strong></td>
<td>Question response: variable axis plot</td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>PLOT</td>
<td>Flag to set the plot capability on</td>
</tr>
<tr>
<td>PLOST</td>
<td>Common buffer for plot data</td>
</tr>
<tr>
<td>PLOIVA</td>
<td>Flag to cancel plot when database is changed</td>
</tr>
<tr>
<td>PLTNOT</td>
<td>Entry routine for percent plot</td>
</tr>
<tr>
<td>PLTPCT</td>
<td>Subroutine for plotting</td>
</tr>
<tr>
<td>PLTSCL</td>
<td>Entry routine for scaled plot</td>
</tr>
<tr>
<td>PM</td>
<td>Subroutine for checking the percentage total</td>
</tr>
<tr>
<td>PMA</td>
<td>Character &quot;-&quot;</td>
</tr>
<tr>
<td>PMAK</td>
<td>Subroutine for Proposal printout</td>
</tr>
<tr>
<td>PMSKM</td>
<td>Subroutine to display achieved values</td>
</tr>
<tr>
<td>PM</td>
<td>Character &quot;%&quot;</td>
</tr>
<tr>
<td>PRCICK</td>
<td>Quantity of tests</td>
</tr>
<tr>
<td>PRCET</td>
<td>Adjusted quantity of tests</td>
</tr>
<tr>
<td>PRCL</td>
<td>answer to selection of menu</td>
</tr>
<tr>
<td>PRATC2</td>
<td>Achieved value extrapolation ratio</td>
</tr>
<tr>
<td>RATI</td>
<td>Achieved value extrapolation ratio</td>
</tr>
<tr>
<td>RBLK</td>
<td>Character string of blanks (real)</td>
</tr>
<tr>
<td>REPCT</td>
<td>Subroutine for achieved value table calculations</td>
</tr>
<tr>
<td>RERUN</td>
<td>Subroutine for zeroizing recursive values in DP runs</td>
</tr>
<tr>
<td>RESA</td>
<td>Resource calculation</td>
</tr>
<tr>
<td>RESB</td>
<td>Resource calculation</td>
</tr>
<tr>
<td>RESC</td>
<td>Resource calculation</td>
</tr>
<tr>
<td>RESM</td>
<td>Resource maximum input</td>
</tr>
<tr>
<td>RESH</td>
<td>Resource minimum input</td>
</tr>
<tr>
<td>RESS</td>
<td>Resource calculation</td>
</tr>
<tr>
<td>RESU</td>
<td>Resource calculation</td>
</tr>
<tr>
<td>RESU</td>
<td>Resource calculation</td>
</tr>
<tr>
<td>RSMAX</td>
<td>Reliability maximum</td>
</tr>
<tr>
<td>RSMIN</td>
<td>Reliability minimum</td>
</tr>
<tr>
<td>S</td>
<td>Sensitivity table data storing array</td>
</tr>
<tr>
<td>SAVDC</td>
<td>Sensitivity parameter saved array</td>
</tr>
<tr>
<td>SAVSIN</td>
<td>Subroutine for determining schedule impact on II</td>
</tr>
<tr>
<td>SCHEDU</td>
<td>Subroutine for determining schedule impact on II</td>
</tr>
<tr>
<td><strong>SE</strong></td>
<td>Subroutine to change Selection Menu.</td>
</tr>
<tr>
<td>-------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td><strong>SELICL</strong></td>
<td>Buffer for sensitivity parameter level w cpl ei</td>
</tr>
<tr>
<td><strong>SELIC</strong></td>
<td>Character string for curve plot;</td>
</tr>
<tr>
<td><strong>SEMC</strong></td>
<td>Choice from Sensitivity Menu</td>
</tr>
<tr>
<td><strong>SENSIT</strong></td>
<td>Sensitivity Table parameter value</td>
</tr>
<tr>
<td><strong>SENS</strong></td>
<td>Subroutine to return data values to buffer</td>
</tr>
<tr>
<td><strong>SIZE</strong></td>
<td>Plot y-axis choice</td>
</tr>
<tr>
<td><strong>SENSIV</strong></td>
<td>Subroutine for sensitivity analysis</td>
</tr>
<tr>
<td><strong>SNPSET</strong></td>
<td>Entry routine to reprint Sensitivity Table</td>
</tr>
</tbody>
</table>

| **SPEA** | Date for start of block 1 production |
| **SPEB** | Status of the team DP level |
| **SPRI** | Subroutine to write data files onto the disk |
| **STRT** | Common buffer for database array |
| **STORE** | Stored value of flag |
| **STOR** | Sum of force structure effectiveness |

| **TAB** | Array of data from the data tables |
| **TMAX** | Flight test completion date desired max |
| **TMIN** | Flight test completion date desired min |
| **TDUE** | Subroutine to input team number and verify security code |

<p>| <strong>TEAM01</strong> | Team 1 |
| <strong>TEAM02</strong> | Team 2 |
| <strong>TEAM03</strong> | Team 3 |
| <strong>TEAM04</strong> | Team 4 |
| <strong>TEAM05</strong> | Team 5 |
| <strong>TEAM06</strong> | Team 6 |
| <strong>TEAM07</strong> | Team 7 |
| <strong>TEAM08</strong> | Team 8 |
| <strong>TEAM09</strong> | Team 9 |</p>
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>YC</td>
<td>Curve data arrays</td>
</tr>
<tr>
<td>YDC</td>
<td>Answer to Yes/No questions</td>
</tr>
<tr>
<td>YF</td>
<td>Y-axis scale</td>
</tr>
<tr>
<td>YFACT</td>
<td>Y-axis scale</td>
</tr>
<tr>
<td>YMAX</td>
<td>Y-axis maximum value</td>
</tr>
<tr>
<td>YMIN</td>
<td>Y-axis minimum value</td>
</tr>
<tr>
<td>YFEE</td>
<td>Achieved value fee in dollars</td>
</tr>
<tr>
<td>YMN</td>
<td>Y-axis minimum</td>
</tr>
<tr>
<td>YMH</td>
<td>Y-axis maximum</td>
</tr>
<tr>
<td>YSCALE</td>
<td>Y-axis scale</td>
</tr>
<tr>
<td>ZEBRA</td>
<td>Contractor name</td>
</tr>
<tr>
<td>ZERHUL</td>
<td>Character string for zero or null (blank)</td>
</tr>
<tr>
<td>ZERC</td>
<td>Subroutine to zeroize all non-data base equivalent values</td>
</tr>
<tr>
<td>ZRSU1</td>
<td></td>
</tr>
<tr>
<td>ZRSU2</td>
<td></td>
</tr>
<tr>
<td>ZRW</td>
<td></td>
</tr>
<tr>
<td>ZTB1</td>
<td></td>
</tr>
<tr>
<td>ZTB2</td>
<td></td>
</tr>
<tr>
<td>ZTB3</td>
<td></td>
</tr>
<tr>
<td>ZTB4</td>
<td></td>
</tr>
<tr>
<td>ZTB5</td>
<td></td>
</tr>
<tr>
<td>ZTB6</td>
<td></td>
</tr>
<tr>
<td>ZTB7</td>
<td></td>
</tr>
<tr>
<td>ZTB8</td>
<td></td>
</tr>
<tr>
<td>ZULU</td>
<td></td>
</tr>
<tr>
<td>ZWEK</td>
<td>Final lot deployment week</td>
</tr>
<tr>
<td>No.</td>
<td>Copies</td>
</tr>
<tr>
<td>-----</td>
<td>--------</td>
</tr>
<tr>
<td>1.</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
11. Mr. Robert Butler
   The Assessment Group
   710 Wilshire Boulevard
   Santa Monica, California 90401

12. Library, Code 0142
    Naval Postgraduate School
    Monterey, California 93943

13. LCDR G. Schultz, CSN
    1153 Sandwood Place
    San Pedro, CA 90731