NASA CR-73434-III

REPRODUCIBLE COPY

Study of Systems and Cost/Performance Methodologies for Optimization of Vehicle Assignment

FINAL REPORT

Volume 3
Programmer's Manual
Integrated Budgest Smoothing and Vehicle Assignment Machine Model

NASA CR-73434-III

STUDY OF SYSTEMS AND COST/PERFORMANCE METHODOLOGIES FOR OPTIMIZATION OF VEHICLE ASSIGNMENT

FINAL REPORT

VOLUME 3

PROGRAMMER'S MANUAL INTEGRATED BUDGET SMOOTHING AND VEHICLE ASSIGNMENT MACHINE MODEL

8 MAY 1970

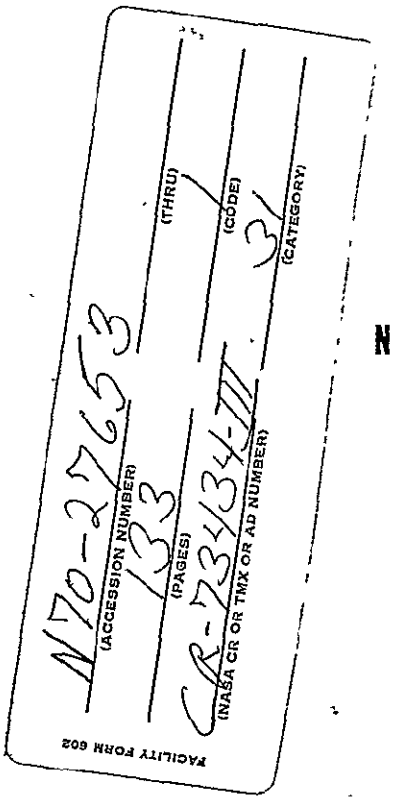
PREPARED UNDER CONTRACT NAS2-5202

FOR

MISSION ANALYSIS DIVISION
OFFICE OF ADVANCED RESEARCH AND TECHNOLOGY
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AMES RESEARCH CENTER
MOFFETT FIELD, CALIFORNIA

BY

LOCKHEED MISSILES & SPACE COMPANY
SUNNYVALE, CALIFORNIA



"Distribution of this report is provided in the interest of information exchange. Responsibility for the contents resides in the authors or organization that prepared it."

NASA CR-73434

STUDY OF SYSTEMS AND COST/PERFORMANCE METHODOLOGIES FOR OPTIMIZATION OF VEHICLE ASSIGNMENT

FINAL REPORT

VOLUME 3

PROGRAMMER'S MANUAL

INTEGRATED BUDGET SMOOTHING AND VEHICLE ASSIGNMENT MACHINE MODEL

8 MAY 1970

Prepared Under Contract NAS2-5202

By

LOCKHEED MISSILES & SPACE COMPANY
SUNNYVALE, CALIFORNIA

For

MISSION ANALYSIS DIVISION
OFFICE OF ADVANCED RESEARCH AND TECHNOLOGY
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AMES RESEARCH CENTER
MOFFETT FIELD, CALIFORNIA

FOREWORD

This report volume provides a programmer's manual for an integrated budget smoothing and vehicle assignment model. The model was developed during a study of cost and performance methologies for optimal assignment of space vehicles to advanced space missions. The study is being performed for the National Aeronautics and Space Administration under Contract NAS2-5202 and is monitored by Mr. Robert Slye and Mr. Harold Hornby of the Mission Analysis Division of the Office of Advanced Research and Technology.

Individuals of Lockheed Missiles & Space Company, Sunnyvale, California, who contributed to this study are L. F. Fox, project leader; C. J. Golden, key technical member; and M. A. Brunet.

CONTENTS

SUMMARY

This document is Volume 3 of a three-volume series comprising a final report of the Study of Systems and Cost/Performance Methodologies for Optimization of Vehicle Assignment. This volume is a programmer's manual for the integrated budget smoothing and vehicle assignment program. Volumes 1 and 2 present a technical description and details on a computer program for optimal vehicle assignment, respectively.

This volume contains appendixes that provide model input requirements, a sample case, flow charts, and a program listing. At the beginning of each appendix, descriptive details and technical comments are provided to indicate any special instructions applicable to the use of that Appendix. In addition, the program listing, Appendix H, includes comment cards that state the purpose of each subroutine in the complete program and also describe operations performed within the subroutine.

Appendix E

INPUT REQUIREMENTS

E.1 GENERAL

A complete glossary of input terms and detailed format requirements are included in this appendix. Variable names are listed by order of input in corresponding sections of use to make the glossary easier to use than an alphabetical listing. Comments are also included which describe either external or internal restrictions associated with the variable.

Figure E-1 illustrates the basic data deck layout for this program. The same restrictions apply as described in Appendix A with the following modification. Constraint and budget level cards are input to the SMOOTH subroutine of this integrated program. The last data card input to SMOOTH is followed by a card containing only an asterisk in the first column. Then the control card for the next set of data appears unless there are no more data cases to follow. In this latter case, a blank card follows the asterisk card in order to terminate the run under normal circumstances.

E.2 INPUT FORM AND DEFINITIONS

Cards Columns	Variable Name	Format	Description and Comments
Control Card			
1 – 5	NOPT	I5	Code for mission/vehicle compatibility screen 1 – ΔV vs. payload weight + availability + A Priori Assignment 2 – (not used) 3 – All criteria

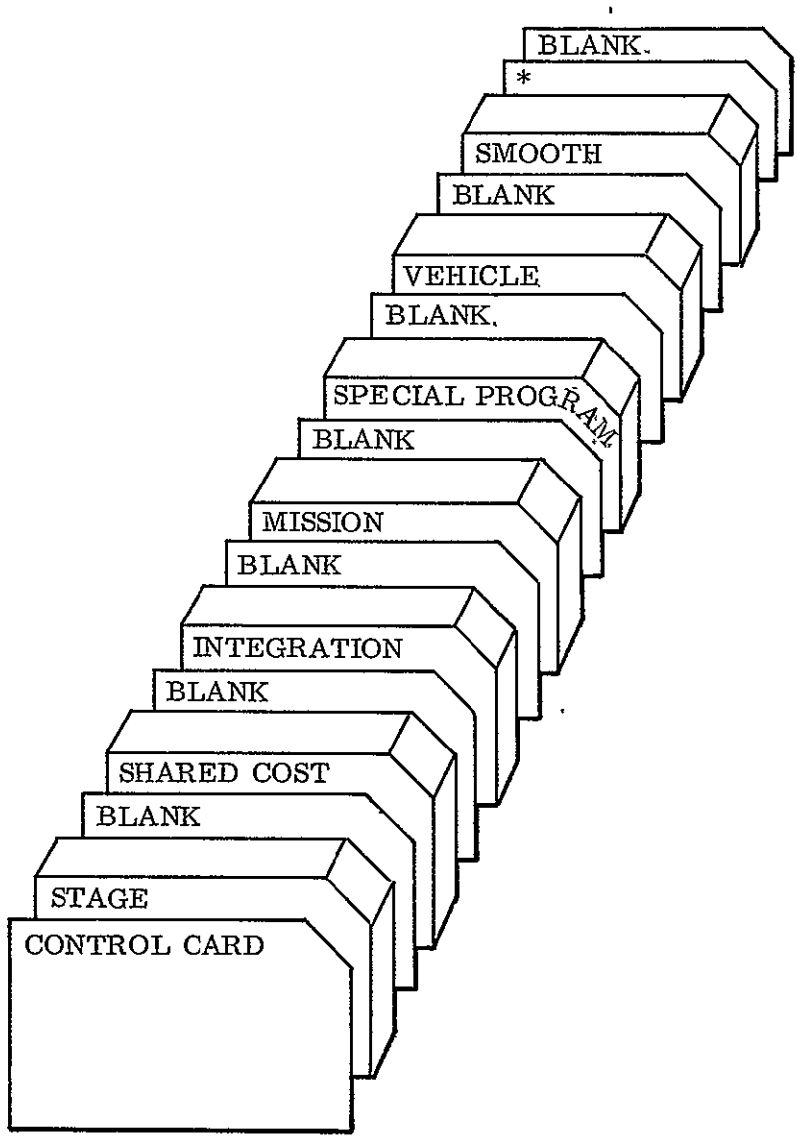


Fig. E-1 Data Deck Layout – Integrated Program

Cards Columns	Variable Name	Format	Description and Comments
6–10	MYRS	I5	Mission model duration in years
11–15	IBY	I5	Last 2 digits of 1st year of mission model
16–27	GUESS	F12.2	Upper bound for total mission cost (SAVES STORAGE SPACE IF REALISTIC VALUE) If GUESS = 0.0, then GUESS is assigned a value 1.0 E10
28–32	MITR	I5	Maximum number of iterations between SMOOTH and ASSIGN
33–37	TREF	F5.1	Last 2 digits of reference year for SMOOTH
38–42	MXITR	I5	Maximum number of iterations between ASSIGN and CHOOZ
68–69	IG	I2	Code for stage input*
70–71	IFM	I2	Code for shared cost group input*
72–73	II	I2	Code for integration cost input*
74–75	IM	I2	Code for mission input*
76–77	ISD	I2	Code for special program data*
78–79	IV	I2	Code for vehicle input*
Stage Information (Input only if IG ≥ 0) I = 1, NSTG < 50 cards			
1–2	KODS(I)	I2	Reference number of stage on card I
4–7	STG(I)	A4	Name of stage on card I
8–13 14–19 20–25	SR(I, J) J = 1, 3	3F6.3	Recurring cost for first unit of stage on card I J = 1 Hardware J = 2 ETR launch support J = 3 WTR launch support
26–30 31–35 36–40	PLC(I, J) J = 1, 3	3F5.3	Recurring cost learning curve percent for stage on card I in decimal form (e.g., .95) J = 1 Hardware J = 2 ETR launch support J = 3 WTR launch support

*If ≥ 0, new input for this case
If < 0, use data from previous case

Cards Columns	Variable Name	Format	Description and Comments
44–49	SNR(I)	F6.3	Development cost of stage on card I
50–55	STS(I)	F6.3	Sustaining cost of stage on card I
56–58	NYS(I)	I3	First year stage on card I is available*
59–61	LSA(I)	I3	Last year stage on card I is available*+
62–64	NBY(I)	I3	Batching duration in years for stage recurring cost
65–67 68–70 71–73 74–76	NFS(I, J) J = 1, 4	4I3	KODEF of the shared cost groups (up to 4) to which stage belongs
78 79 80	MODE(I, J) J = 1, 3	3I1	Code to indicate type of input for recurring cost of stage on card I** J = 1 Hardware J = 2 ETR launch support J = 3 WTR launch support

Stage Duration Information (2nd Stage Card)

1–3	YDS(I)	F3.1	Duration in years over which β function distributes development cost for stage on card I
			Input necessary if SNR or STS ≠ 0
4–6	IST(I)	I3	Last 2 digits of start date for Stage Development Program
7–9	NSFX(NSDC***)	I3	Duration in years ≤ 12 for any miscellaneous (fixed or development) program associated with stage on card I (e.g. Run out costs). (Standard Development costs are distributed by a Beta Function – any other development distribution may be input under this special category.)

*1 corresponds to year IBY.

+If available through mission model, any number = or > MYRS may be input. If number < MYRS is input then this termination date is maintained through all iterations.

**If = 0, learning curve type input
If ≠ 0, jump type input

***NSDC = Number of special development costs ≤ 50

Card Column	Variable Name	Format	Description and Comments
If MODE(I, J) ≠ 0 for some J, require following Jump Type Input Card for Each such J.			
5–14	SRJ(LX, 1)	F10.3	Total recurring cost for up to POJ number of stages
15–24	SRJ(LX, 2)	F10.3	Slope of line defining total recurring cost for over POJ number of stages
25–34	SRJ(LX, 3)	F10.3	Y-intercept of line defining total recurring cost for over POJ number of stages
35–44	POJ(LX)	F10.3	Number of stages at which function defining total recurring cost changes slope
If NSFX(NSDC) ≠ 0 read in following card.			
1–3	NRFX(NSDC)	I3	Start date for special development cost associated with stage on card I. (Referenced to IST(I))
4–9 10–15 16–21 etc.	RXD(J, NSDC) J = 1, 12	12F6.2	Special development cost to be spent in year 1900 + IST(I) + NRFX(NSDC) - 1 + J (Input distribution)
Last Stage Card must be followed by a blank card.			
Shared Cost Group Cards (Input only if IFM ≥ 0) I = 1, NFAM < 40			
1–2	KODEF(J)=I	I2	Reference Number of group on card J
4–7	FAM(I)	A4	Name of group I
8–17	FMNR(I)	F10.0	Development cost of group I
18–27	FMSUS(I)	F10.0	Sustaining cost of group I
28–31	YDF(I)	F4.1	Duration in years of Development Program cost distribution (β Function)

Card Column	Variable Name	Format	Description and Comments
32–34	JST(I)	I3	Last 2 digits of start date for group Development Program – necessary if FMNR or FMSUS ≠ 0
35–37	NSFX(NSDC)	I3	Duration in years for any miscellaneous fixed or development program distribution associated with group I. (Distribution input on following card.)

If (NSFX(NSDC) ≠ 0) read following card.

1–3	NRFX(NSDC)	I3	Start date for special Development cost associated with group I. (Referenced to JST(I))
4–9 10–15 16–21 etc.	RXD(J,NSDC) J = 1,12	12F6.2	Special Development cost to be spent in year 1900 + JST(I) + NRFX(NSIC) – 1 + J (Input distribution)

Last Group card must be followed by a blank card.

Integration Cost Cards (Input only if II ≥ 0) I = 1, NCI < 30

3–5	NFML(I)	I3	KODEF of shared cost group which is lower member of integration pair I
6–8	NFMU(I)	I3	KODEF of shared cost group which is upper member of integration pair I
9–18	RINT(I)	F10.0	Recurring cost for first unit of integration I
19–28	PLCINT(I)	F10.0	Recurring cost learning curve percent for integration I
29–38	DINT(I)	F10.0	Development cost of integration I
39–48	SINT(I)	F10.0	Sustaining cost of integration I
49–52	YDI(I)	F4.1	Development duration in years for β distribution
53–55	KST(I)	I3	Last 2 digits of start date for integration development program – input necessary if DINT or SINT ≠ 0

Card Column	Variable Name	Format	Description and Comments
56-58	NSFX(NSDC)	I3	Duration in years for any miscellaneous fixed or development program associated with integration I (Distribution input on following card)

If NSFX(NSDC) ≠ 0 read following card.

1-3	NRFX	I3	Start date for Special Development cost associated with integration I (Referenced to KST(I))
4-9 10-15 16-21 etc.	RXD(J,NSDC) J = 1,12	12F6.2	Special Development cost to be spent in year 1900 + KST(I) + NRFX(NSDC) - 1 + J (Input distribution)

Last Integration card must be followed by a blank card.

Mission Data Card – (Input only if IM ≥ 0) I = 1, NMIS < 65

1-2	KODEM(I)	I2	Reference number of MISSION on card I
3-8	NAME(I)	A6	Name of MISSION on card I
9-12	PB(I)	F4.2	Priority of MISSION on card I
15-16	NSYR(I)	I2	Number of sustaining years required for PLS(I) after last launch year
17-18	NYRSFX(I)	I2	Duration in years of any fixed or special development cost distribution associated with mission KODEM(I)
19-25	VLR(I)	F7.0	Characteristic velocity required in fps to accomplish mission on card I after attaining 100 n.m orbit
32-38	WPR(I)	F7.0	Payload weight in lb required for mission on card I
41-80	MISN(I, J) J = 1, MYRS	20I2	Number of launches for mission on card I in year J + 1900 + IBY -1

2nd Card needed for each mission.

Card Column	Variable Name	Format	Description and Comments
3–12	PLR(I)	F10.2	Payload recurring cost for mission KODEM(I)
13–22	PLS(I)	F10.2	Payload sustaining cost
23–32	PLD(I)	F10.2	Payload development cost
33–37	YDPL(I)	I5	Duration in years over which development cost is to be distributed by Beta Function
38–42 43–47 48–52 53–57	RDIST(I, L) L = 1,4	4F5.3	Input recurring cost distribution for PLR in decimal form (e.g. RDIST(I, 1) = 05)
58–67	PLMD(I)	F10 2	Maximum diameter of payload for MISSION on card I
68–69	NPLS(I)	I2	Code for payload stabilization requirement 0 – No requirement 1 – Must be spin stabilized 2 – Must not be spin stabilized
70–71	MR(I)	I2	Code for man-rating requirement for Mission on card I 0 – No requirement 1 – Must be man-rated
72–73	LTR(I)	I2	Code for launch site of Mission 1 – ETR 2 – WTR
74–75	NRR(I)	I2	Number of restarts required for Mission
76–77	IS(I)	I2	Last 2 digits of start year for development cost PLD(I)
78–80	IVEHA(I)	I3	A priori vehicle assignment for Mission on card I If no vehicle assigned – 0 input; KODEV of vehicle input otherwise

Card Column	Variable Name	Format	Description and Comments
If NYRSFX(I) ≠ 0 read following card.			
1–3	NSTRFX(I)	I3	Start date for special development cost associated with mission KODEM(I) referenced to IS(I)
4–9 10–15 16–21 etc.	RFIXD(J,I) J = 1,12	12F6.2	Special Development cost to be spent in year 1900 + IS(I) + NSTRFX(I) − 1 + J (Input distribution)
Last Mission card must be followed by a blank card.			
Special Program Data Card (No launch associated with program) – Input only if ISD ≥ 0, I = 1, NSPR ≤ 6			
1–2	KODESP(I)	I2	Code number for Special Program (must be larger than 100)
4–9	NAME(I)	A6	Name of Special Program on card I
10–19	PLD(I)	F10.2	Development cost associated with program (distributed by β Function)
20–24	YDPL(I)	I5	Duration in years of Development program
25–26	IS(I)	I2	Last 2 digits of start year for development cost PLD(I)
27–36	PLS(I)	F10.2	Annual sustaining cost associated with program
37–38	NYRSST(I)	I2	Duration in years of sustaining program
39–40	NYRSFX(I)	I2	Duration in years of any fixed cost which does not have a β distribution
If NYRSFX(I) ≠ 0 read following card.			
1–3	NSTRFX	I3	Start date for fixed cost referenced to IS(I)
4–9 10–15 16–21 etc.	RFIXD(J,I) J = 1,12	12F6.2	Fixed Cost to be spent in year 1900 + IS(I) + NSTRFX(I) - 1 + J
Last Special Program Data card must be followed by a blank card.			

Card Column	Variable Name	Format	Description and Comments
Vehicle Data Card (Input only if IV ≥ 0) J = 1, NV ≤ 60			
1–8	VEH(I,J) I = 1,4	4I2	KODS of stage in Ith position, where I = 1 corresponds to booster, for vehicle on card J
9–21	B1(J)	E13.6	Payload vs. characteristic velocity curve constants for performance evaluation of vehicle on card J PL = EXP(B1 - B2*V-B3/(B4-V)) and V = Excess Velocity = Total Characteristic Velocity-Circular Velocity at 100 n.m.
22–34	B2(J)	E13.6	
35–47	B3(J)	E13.6	
48–60	B4(J)	E13.6	
79–80	KODEV(J)	I2	Reference number of vehicle on card J
2nd Card needed for each vehicle.			
4–5	NVS(J)	I2	Code for stabilization of vehicle on card J 1 – Is Spin stabilized 2 – Is not spin stabilized
6–7	MRV(J)	I2	Code for man-rating of vehicle on card J 0 – Is not man-rated 1 – Is man-rated
8–9	NRP(J)	I2	Number of restarts possible for vehicle on card J
80	JKEY	I1	Code for recurring cost distribution for vehicle on card J JKEY = 0 – Standard distribution is used 1st year of distribution = 05 Recurring cost 2nd year of distribution = 20 Recurring cost 3rd year of distribution = 50 Recurring cost 4th year of distribution = .25 Recurring cost = Launch year generating this recurring cost
Input only if JKEY ≠ 0			JKEY = 1 - Distribution is to be input on following card

Card Column	Variable Name	Format	Description and Comments
4–8 9–13 14–18 19–23	ALPI(I,J) I = 1,4	4F5.2	Input Recurring cost distribution for vehicle on card J in year I where I = 4 corresponds to year of launch

Last Vehicle Data card must be followed by a blank card.

Budget Smoothing Data is input in subroutine SMOOTH using a CALL INPUT statement. The following variables may be input at this time.

Variable Name	Description and Comments
TITLE(I)	Output page HEADING – if no input, blanks are output. 40 characters are allocated for storage, e.g. TITLE = 'LUNAR OPTION'
LEVEL(J)	YEARLY DESIRED FUNDING LEVEL (20 year maximum) e.g. LEVEL = 300., 375., 18 × 300
ISTRT	FIRST YEAR of smoothing interval – referenced to TREF = 1
IFIN	Last year of smoothing interval – referenced to TREF
MAXITR	Maximum number of iterations allowed per case in SMOOTH subroutine
NCSTR	Number of constraints on mission programs ≤ 60
NPROG(K)	The reference number (KODEM or KODESP) of the mission being constrained
KPROG(K)	The reference number (KODEM or KODESP) of the constraining program or mission
KODE(K)	Code number for type of constraint ≤ 11
CS(K)	Constant associated with each constraint
FIXED(I)	Yearly total fixed overhead costs (I = 1,20) If no input, is set to zero
PMAX PMIN	Constants associated with PLOT2 – if no input they are set to 5000. and 1500. respectively

Variable Name	Description and Comments
ACCL	Code for use of acceleration option — if no input it is set = .TRUE
EXT	Code for use of extension option — if no input it is set = .TRUE. If FALSE is input these options will not be used.

The next card contains an * in the first column.

The next card is either a new control card for the next case of data or a blank card so that the run is terminated under normal circumstances.

Appendix F
SAMPLE CASE

F.1 DESCRIPTION

The output from one of the more interesting sample cases is presented in this section. Data is not realistic and no significance should be attached to the values used. The listing includes a module map so that storage requirements are defined for each subroutine and common block.

Data input to the program which is used in the ASSIGN algorithm is output in the same format as for the general assignment program described in Appendix B. After the optimum assignment has been listed, any input to subroutine SMOOTH is automatically output as it appears on the data cards. "Average" recurring cost data for each of the vehicles in the optimum assignment is calculated in VEHRC and output on the following page. Each vehicle is assigned a key number used internally which is output with the associated stage component names defining the vehicle.

The breakdown of costs by program and type, and by program and year on the following pages are essentially the same as for the original budget smoothing model (Ref. 4, Vol. 1). For example, the Mars 71 mission (PN = Program Name) has development start date in 1969, has no development (DEVL) costs and hence no development duration (YRS). Sustaining costs (SUST) start in year 1969 (= START + SS - 1.) They are spent for 3 (SD) years. Recurring costs start in year 1969 (= START + RS - 1.) and last for 3 (RD) years. The distribution follows on the same line (e.g., $12.00 in year 1969, $25.00 in year 1970 and $12.00 in year 1971). On the following line fixed miscellaneous costs are similarly listed if any have been input for that program. (e.g., Fixed costs start in year 1971 (= START + RS - 1.) and last for 2 years (RD). The distribution follows on the same line of output. More complete data on these entries are provided in the reference indicated.

Programs associated with missions are output first. For the selected sample case Programs 1 thru 14 are mission related. Programs 15 and 16 are miscellaneous programs having no associated launches and the remaining program are development or sustaining costs associated with launch vehicles. These last programs are identified by the decision number used in the ASSIGN algorithm. A list of decision numbers, their associated values and types of expenditure has been output previously for reference.

The section "Total Program Costs and Launch Vehicle Schedule" is output as in the original smoothing program with the following modification. Instead of printing the launch vehicle key name under its associated program and year of launch, the key number already output with each corresponding vehicle name is substituted for simplicity.

A plot follows this tabulated data showing actual yearly totals (*) and desired yearly level of spending (0). The smoothed data is then output using the same formats. Only data input to SMOOTH directly from ASSIGN and the final smoothed data are output. Intermediate output is suppressed.

No output appears if all input data meet input constraints. If any constraint is violated, however, the program number and type of constraint is output. The program continues to smooth the data anyway since the violation may be removed by internal shifting. However, the warning that the input data does not meet all constraints will alert the user to possible discrepancies in his input data.

Current dimensional constraints on the program are all listed in Appendix E for input variables. There is one additional internal dimensional constraint. MASTER automatically constrains development costs, corresponding to the optimum assignment, to their associated launch dates. Ninety constraints are provided at present. However, in general it is not possible to anticipate the exact number of such automatic constraints to be generated. To alert the user if a problem does exist for some input data, the

program prints out a notice when ninety constraints have been generated in MASTER. Then the SMOOTH subroutine is entered. If no more constraints are added to the set from MASTER, the program smoothes the data under the ninety constraints computed in MASTER. Otherwise the program terminates the case with an output explanation and goes back to ASSIGN where it looks for new data. This dimensional restriction can be relieved whenever necessary by increasing the associated dimensional variables.

The sample case presented in this section is particularly interesting because it demonstrates the interrelationship between the 2 programs ASSIGN and SMOOTH. The program required 1.60 mins. on the 360/67 computer at Ames Laboratory, Moffett Field, California. A description of the over-all output is provided below in outline form.

F.1.1 First Iteration

Input data was purposely constrained so that a key stage (B2S) is not available until the 3rd year after the reference year (1971).

The optimum assignment is determined based on 4 main stages, T3D, TRAN, CENT and B2S. The total program cost is $521.92 (M).

SMOOTH shifts B2S stage (Dev. Cost 7) so that its development start date is 1969 instead of the previous 1973. Other mission dates are also shifted and costs modified. (Each change is marked with an *).

F.1.2 Second Iteration

Using these shifted dates and costs ASSIGN determines a new optimum assignment based on 3 main stages, (T3B, T3D, T3M), (AGD, AGLT) and B2S. The total program cost is $418.99. (More than $100.00 less than first iteration solution).

SMOOTH shifts the new program data such that Dev. Cost 7 (B2S stage) has its development start date moved to 1971. Hence it is not available until 1973.

F.1.3 Third Iteration

Using the shifted data from SMOOTH, ASSIGN determines that the new optimum assignment is the same as the optimum assignment determined in iteration two. The total cost is $484.54 based on modified data.

SMOOTH changes some dates in the new input, but since these changes are small and the final RMS value equals the input data RMS value (within 0.4) the program terminates.

The assignment input to SMOOTH with its corresponding distribution dates represents the optimum solution within the budget smoothing constraints.

F.2 SAMPLE CASE PRINTOUT

The sample case printout follows.

```
{1
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,MAP
          VARIABLE OPTIONS USED - SIZE=(126976,24576)              DEFAULT OPTION(S) USED
IEW0000       INCLUDE DECKS(MOX01CL,MOX02MS)
IEW0000       INCLUDE DECKS(MOX02BS,MOX02RU,MOX02CR,ALINPT,MOX01UP,MOX02AT)
IEW0000       INCLUDE DECKS(MOX02TT,MOX02ST)
IEW0000       INCLUDE DECKS(MOX02AN,MOX02SM,MOX02VC,MOX02DC)
IEW0000       INCLUDE DECKS(MOX02CZ,MOX01PK,MOX02AL,MOX02CL,MOX02LD)
IEW0000       ENTRY MAIN
```

MODULE MAP

CONTROL SECTION NAME	ORIGIN	LENGTH
CLEAR	00	38
MAIN	38	1528
SMOOTH	1560	213C
REVALU	36A0	474
CONSTR	3B18	3EC
ALINPT	3F08	B1A
UMPLOT	4A28	F68
AFRMT	5990	40
TCOST	59D0	784
SHIFT	6188	7D8
ASSIGN	6960	213D
STGNUM	8A90	F98
VEHRC	9A28	2D8
DECISN	9D00	ED8
CHOOZ	ABD8	1544
PACK	C120	E8
AVAIL	C208	710
CAPABL	C918	652
LBOUND	CF70	952
IHCSLOG *	D8C8	1BA
IHCSEXP *	DA88	1B0
IHCFRXPR*	DC38	183
IHCECOMH*	DDC0	F31
IHCCOMH2*	ECF8	545
IHCFMAXI*	F240	C9
IHCSSQRT*	F310	149
IHCFCVTH*	F460	1175

ENTRY NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
INPUT	3F08						
PLOT1	4A5A	PLOT2	4C76	PLOT3	4E3A	PLOT4	4F8A
OMIT	51EE	PLTAPE	5224				
UNPACK	C172	ITEM	C1B6				
ALOG10	D8C8	ALOG					
EXP	DA88						
FRXPR*	DC38						
IBCOM*	DDC0	FDIOCS*	DE7C	INTSWTCH	ECDE		
SEQDASD	EF58						
MAXO	F240	MINO	F256	AMAXO	F26C	AMINO	F282
SQRT	F310						

NAME	ORIGIN	LENGTH
IHCEFNTH*	105D8	512
FIOCS* *	10AF0	160
IHCEFIOS*	10C50	111C
IHCERRM *	11D70	5AC
IHCUOPT *	12320	398
IHCETRCH*	12688	28E
IHCUATBL*	12948	638
SAVES	12F80	1660
SAVEA	145E0	2160
SAVEA1	16740	33C
SAVEB	16A80	1BD0
SAVEB1	18650	410
PLSAVE	18A60	138
SAVECZ	18B98	3654
ASGN	1C1F0	438
SAVEDC	1C628	5A8
SAVESM	1CBD0	30F8
SAVECL	1FCC8	8F8
SAVED	205C0	690
SAVELZ	20C50	F0
SAVENV	20D40	8
BATCH	20D48	2C
SAVESG	20D78	4
SCRACH	20D80	47A4
SAVELB	25528	108
SAVENR	25630	8

ENTRY ADDRESS 38
TOTAL LENGTH 25638

NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
ADCON=	F460	FCVAOUTP	F50A	FCVLOUTP	F59A	FCVZOUTP	F6EA
FCVIOUTP	FA76	FCVEOUTP	FF78	FCVCOUTP	10192	INT6SWCH	10478
ARITH*	105D8	ADJSWTCH	10944				
SET899	1087C	RESB99	10B96				
FIOCSBEP	10C56						
ERRMON	11D70	IHCERRE	11D88				
IHCTRCH	12688	ERRTRA	126C0				

****MAIN DOES NOT EXIST BUT HAS BEEN ADDED TO DATA SET

STAGE COST DATA

TITLE	RECURRING LC (HARDWARE)		RECURRING LC (ETR ONLY)		RECURRING LC (NTR ONLY)		DEVELOPMENT	SUSTAINING	AVAILABLE FROM	AVAILABLE TO	SHARED COST GROUPS				BATCH FACT
SV3A	6.89	0.950	6.23	0.950	3.27	0 950	0.0	3.37	1	5	1	0	0	0	5
AG D	1.87	0.900	1.40	0.900	2.50	0.900	0.0	2.60	1	20	2	7	0	0	6
RECURRING COST TYPE 1 FOR X LESS THAN OR = 2.00, TOTAL COST = 2.30.FOR X GREATER THAN 2.00, TOTAL COST = 0.40 X + 1.40															
SV3C	7.09	0.850	6.43	0.850	3.47	0.850	0.0	0.0	1	20	1	0	0	0	4
CENT	11.85	0.900	4.62	0.900	4.62	0.900	25.30	16.00	1	20	4	0	0	0	5
T3B	4.91	1.000	1.44	1.000	1.44	1.000	0.0	0.0	1	20	3	8	0	0	3
T3D	8.14	0.950	5.01	0.950	3.45	0.950	0.0	0.0	1	20	3	6	0	0	3
TRAN	6.09	1.000	0.0	1.000	0.0	1.000	0.0	0.0	1	20	3	0	0	0	4
AGLT	3.50	1.000	0.0	1.000	0.0	1.000	17.50	1.20	3	20	2	9	0	0	5
RECURRING COST TYPE 2 FOR X LESS THAN OR = 2.00, TOTAL COST = 2.30.FOR X GREATER THAN 2.00, TOTAL COST = 0.40 X + 1.40															
SV3X	6.66	0.850	5.76	0.850	2.58	0.850	13.00	0.0	3	20	1	10	0	0	4
T3M	22.00	0.950	5.01	0.950	3.45	0.950	25.00	0.0	3	20	3	11	0	0	3
B2S	0.87	1.000	0.10	1.000	0.10	1.000	0.25	0.10	3	20	5	0	0	0	5

SHARED COST DATA

NO.	TITLE	DEVELOPMENT	SUSTAINING
1	ATLS	0.0	4.66
3	TITN	0.0	6.00
5	B2S	0.0	0.0
7	AG D	0.0	0.0
9	AGLT	0.0	0.0
11	T3M	0.0	0.0
2	AGEN	0.0	2.50
4	CENT	0.0	0.0
6	T3D	0.0	0.0
8	T3B	0.0	0.0
10	SV3X	0.0	0.0

INTEGRATION COST DATA

LOWER GROUP	UPPER GROUP	RECURRING	LC	DEVELOPMENT	SUSTAINING
TITN	AGEN	0.0	1.000	14.00	0.0
T3D	AG D	0.0	1.000	2.50	0.0
T3D	AGLT	0.0	1.000	5.50	0.0
TITN	CENT	0.0	1.000	80.00	0.0

MISSION MODEL

	MISSION	VELOCITY	PAYLOAD	PRIORITY	TR	LAUNCH SCHEDULE					
						71	72	73	74	75	76
1	MARS71	41500.	1100.	1.00	1	2	0	0	0	0	0
2	PIONER	49000.	460.	1.00	1	0	1	1	0	0	0
3	MARS73	40000.	6000.	1.00	1	0	0	2	0	0	0
4	MERCRY	38900.	1000.	0.50	1	0	0	0	0	1	0
5	GRDTRT	49500.	800.	0.50	1	0	0	0	1	0	0
6	MARS75	39400.	3000.	0.50	1	0	0	0	0	2	0
7	COMET	37200.	2000.	0.50	1	0	0	0	0	0	2
8	ASTRA	26300.	7000.	0.50	2	0	0	0	0	0	1
9	RELTIV	14200.	2000.	0.50	2	0	0	0	1	0	0
10	PIONOE	51400.	1000.	0.50	2	0	0	0	0	1	0
11	ATS	33600.	4000.	1.00	2	0	1	1	0	0	0
12	DRELAY	14200.	2000.	0.50	2	0	0	0	0	0	1
13	USAMSC	33600.	4100.	0.50	2	0	0	0	0	1	0
14	USAMSM	30000.	6800.	0.50	2	0	0	0	0	0	2

NUMBER OF STAGES	11
NUMBER OF VEHICLES	20
NUMBER OF FAMILIES	11
NUMBER OF INTEGRATION COSTS	4
NUMBER OF MISSIONS	14
NUMBER OF YEARS	6
LAUNCH BASE YEAR	71
TOTAL COST ESTIMATE	600.00

QUANTITIES BRANCHED UPON

	DEVELOPMENT	SUSTAINING		YEAR AVAIL	LAST YEAR	DEV START	DEV DURATION
1	0.0	3.37	SV3A STAGE	1	5	1970	0.
2	30.00	2.60	AG D STAGE	1	6	1971	0.
3	25.30	16.00	CENT STAGE	1	6	1969	3.
4	17.50	1.20	AGLT STAGE	3	6	1970	4.
5	13.00	0.0	SV3X STAGE	3	6	1970	4.
6	25.00	0.0	T3M STAGE	3	6	1970	4.
7	0.25	0.10	B2S STAGE	3	6	1973	1.
8	10.00	4.66	ATLS SHARED	1	6	1969	3.
9	0.0	6.00	TITN SHARED	1	6	1970	0.
10	0.0	2.50	AGEN SHARED	1	6	1970	0.
11	14.00	0.0	INTEGRATION OF TITN AND AGEN	1	6	1969	3.
12	2.50	0.0	INTEGRATION OF T3D AND AG D	1	6	1969	1.
13	5.50	0.0	INTEGRATION OF T3D AND AGLT	3	6	1969	5.
14	81.00	0.0	INTEGRATION OF TITN AND CENT	1	6	1968	2.

VEHICLE/MISSION CAPABILITY
(1 = POSSIBLE, 0 = IMPOSSIBLE)

VEHICLE / MISSION	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
MISSION NUMBER	1	2	3	4	5	6	7	8	9	10	11	11	12	13	14																				
1 T3B AG D	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0																			
2 T3B AG D B2S	0	0	0	0	1	0	0	0	1	1	0	0	0	1	0	0																			
3 T3B CENT	0	0	0	0	1	0	0	1	1	1	0	1	1	1	1	1																			
4 T3B CENT B2S	0	0	1	0	1	0	0	1	1	1	0	0	1	1	1	1																			
5 T3D TRAN	0	0	0	0	1	0	0	1	1	1	0	1	1	1	1	1																			
6 T3D TRAN B2S	0	0	0	0	1	0	1	1	1	1	0	0	1	1	1	1																			
7 T3D AG D	1	0	0	0	1	0	1	1	1	1	0	1	1	1	1	1																			
8 T3D AG D B2S	0	0	1	0	1	1	1	1	1	1	0	0	1	1	1	1																			
9 T3D AGLT	0	0	1	0	1	0	1	1	1	1	0	0	1	1	1	1																			
10 T3D AGLT B2S	0	0	1	0	1	1	1	1	1	1	0	0	1	1	1	1																			
11 T3D CENT	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1																			
12 T3D CENT B2S	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1																			
13 SV3A AG D	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0																			
14 SV3A AG D B2S	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0																			
15 SV3C CENT	0	0	0	0	1	0	0	1	1	1	0	1	1	1	0	0																			
16 SV3C CENT B2S	0	0	0	0	1	0	0	1	1	1	0	0	1	1	1	0																			
17 SV3X CENT	0	0	0	0	1	0	0	1	1	1	0	0	1	1	1	1																			
18 SV3X CENT B2S	0	0	0	0	1	0	0	1	1	1	0	0	1	1	1	1																			
19 T3M AGLT	0	0	1	1	1	1	1	1	1	1	0	0	1	1	1	1																			
20 T3M AGLT B2S	0	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1																			

********** SOLUTION **********

165	293.56	222.85	516.41

MISSION TITLE	CHARACTERISTIC VELOCITY(FT/SEC)	PAYLOAD (LBS)	LAUNCH YEAR	NUMBER OF LAUNCHES	OPTIMUM LAUNCH VEHICLE
MARS71	41500.	1100	1971	2.00	T3D CENT
PIONER	49000.	460.	1972	1.00	T3D CENT
			1973	1.00	T3B CENT B2S
MARS73	40000.	6000.	1973	2.00	T3D CENT
MERCRY	38900.	1000.	1975	0.50	T3D TRAN
GRDTRT	49500.	800.	1974	0.50	T3D CENT
MARS75	39400.	3000.	1975	1.00	T3D TRAN B2S
COMET	37200.	2000.	1976	1.00	T3D TRAN
ASTRA	26300.	7000.	1976	0.50	T3D TRAN
RELTIV	14200.	2000.	1974	0.50	T3D TRAN
PIONOE	51400.	1000.	1975	0.50	T3D CENT B2S
ATS	33600.	4000.	1972	1.00	T3D TRAN
			1973	1.00	T3D TRAN
DRELAY	14200.	2000.	1976	0.50	T3D TRAN
USAMSC	33600.	4100.	1975	0.50	T3D TRAN
USAMSM	30000.	6800.	1976	1.00	T3D TRAN

********** SOLUTION **********

164	299.07	222.85	521.92

MISSION TITLE	CHARACTERISTIC VELOCITY(FT/SEC)	PAYLOAD (LBS)	LAUNCH YEAR	NUMBER OF LAUNCHES	OPTIMUM LAUNCH VEHICLE
MARS71	41500	1100.	1971	2.00	T3D CENT
PIONER	49000.	460.	1972	1.00	T3D CENT
			1973	1.00	T3B CENT B2S
MARST3	40000.	6000.	1973	2.00	T3D CENT
MERCRY	38900.	1000.	1975	0.50	T3D TRAN
GRDTRT	49500.	800.	1974	0.50	T3D CENT
MARS75	39400.	3000.	1975	1.00	T3D TRAN B2S
COMET	37200.	2000.	1976	1.00	T3D TRAN
ASTRA	26300.	7000.	1976	0.50	T3D TRAN
RELTIV	14200.	2000.	1974	0.50	T3D CENT B2S
PIONOE	51400.	1000.	1975	0.50	T3D TRAN
ATS	33600.	4000.	1972	1.00	T3D TRAN
			1973	1.00	T3D TRAN
DRELAY	14200.	2000.	1976	0.50	T3D TRAN
USAMSC	33600.	4100.	1975	0.50	T3D TRAN
USAMSM	30000.	6800.	1976	1.00	T3D TRAN

THE OPTIMUM SOLUTION HAS BEEN DETERMINED

```
        TITLE='TEST CASE',
LEVEL=20X100.,
ISTRT=1,  IFIN=6,       MAXITR=10,
NCSTR=2,      KODE=6,9,
NPROG=1,101,            XPROG=3.0,              CS=-1.,1970.,
PMAX=700.,    PMIN=0.,
*
```

RECURRING COST DATA

KEY	NAME	UNIT COST
4	T3B CENTB2S	20.05
5	T3D TRAN	16.92
6	T3D TRANB2S	19.01
11	T3D CENT	24.96
12	T3D CENTB2S	25.46

REFERENCE YEAR 1969. TEST CASE

PN	NAME	START	DEVL	YRS	SUST	SS	SD	RS	RD	RECURRING OR FIXED ITEMS				
1	MARS71	1969.	0.	0.	4.	1	3	1	3	12.	25.	12.		
								3	2	1.	2.			
2	PIONER	1900.	0.	0.	0.	1	0	70	5	1.	4.	16.	16.	5.
3	MARS73	1900.	0.	0.	0.	1	0	71	4	2.	10.	25.	12.	
4	MERCRY	1900.	0.	0.	0.	1	0	73	4	0.	2.	5.	3.	
5	GRDTRT	1900.	0.	0.	0.	1	0	72	4	1.	2.	6.	3.	
6	MARS75	1900.	0.	0.	0.	1	0	73	4	1.	4.	10.	5.	
7	COMET	1900.	0.	0.	0.	1	0	74	4	1.	3.	8.	4.	
8	ASTRA	1900.	0.	0.	0.	1	0	74	4	0.	2.	4.	2.	
9	RELTIV	1900.	0.	0.	0.	1	0	72	4	0.	2.	4.	2.	
10	PIONOE	1900.	0.	0.	0.	1	0	73	4	1.	3.	6.	3.	
11	ATS	1900.	0.	0.	0.	1	0	70	5	1	4.	12.	13.	4.
12	DRELAY	1900.	0.	0.	0.	1	0	74	4	0.	2.	4.	2.	
13	USAMSC	1900.	0.	0.	0.	1	0	73	4	0.	2.	4.	2.	
14	USAMSM	1900.	0.	0.	0.	1	0	74	4	1.	3.	8.	4.	
15	IMAGE	1969.	50.	5.	40.	4	2	0	0	0.				
16	PRETEN	1972.	2.	1.	10.	1	3	0	0	0.				
								3	1	15.				
17	DEV 3	1969.	25.	3.	16.	2	6	0	0					
18	DEV 7	1973.	0.	1.	0.	1	3	0	0					
19	DEV 9	1970.	0.	0.	6.	1	5	0	0					
20	DEV 14	1968.	80.	2.	0.	2	0	0	0					
								1	1	1.				
TOTAL			158.		247.									

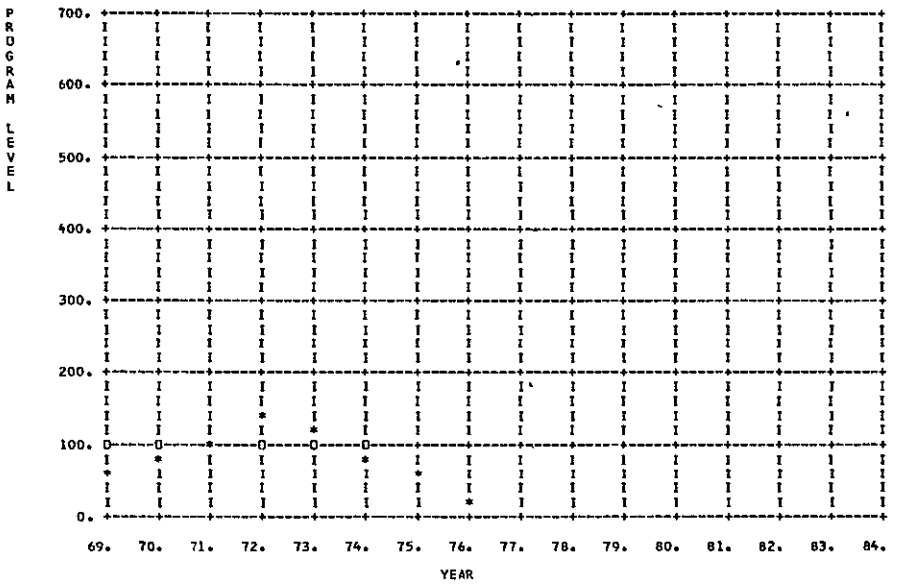
TOTAL PROGRAM COSTS AND LAUNCH VEHICLE SCHEDULE

YEAR	1969.	1970.	1971.	1972.	1973.	1974.	1975.	1976.	1977.	1978.	1979.	1980.	1981.	1982.	1983.	1984.	1985.	1986.	1987.	1988.
PROGRAM																				
1 MARS71	16.	28.	17. 11*	2.																
2 PIONER	1.	6.	16.	16. 11*	5. 4*															
3 MARS73	0.	2.	10.	25.	12. 11*															
4 MERCRY	0.	0.	0.	0.	2.	5.	3 5*													
5 GRDTRT	0.	0.	1.	2.	6.	3 11*														
6 MARS75	0.	0.	0.	1.	4.	10.	5. 6*													
7 COMET	0.	0.	0.	0.	1.	3.	8	4 5*												
8 ASTRA	0.	0.	0.	0.	0.	2.	4.	2 5*												
9 RELTIV	0.	0.	0.	2.	4.	2. 5*														
10 PIONDE	0.	0.	0.	1.	3.	6.	3. 12*													
11 ATS	1.	4.	12.	13. 5*	4. 5*															
12 DRELAY	0.	0.	0.	0.	0.	2.	4.	2. 5*												
13 USAMSC	0.	0.	0.	0.	2.	4.	2. 5*													
14 USAMSH	0.	0.	0.	0.	1.	3.	8.	4. 5*												
15 IMAGE	5.	12.	16.	52.	45.															
16 PRETEN	0.	0.	0.	12.	10.	25.														
17 DEV 3	7.	28.	23.	16.	16.	16.	16.													
18 DEV 7	0.	0.	0.	0.	0.	0.	0.													
19 DEV 9	0.	6.	6.	6.	6.	6.														
20 DEV 14	40.																			
SUM	69.	87.	101.	149.	122.	88.	54.	13.												
FIXED	0.	0.	0.	0.	0.	0.	0.	0.												
TOTAL	69.	87.	101.	149.	122.	88.	54.	13.												
	*	*	*	*	*	*														

LEVEL 100. 100. 100. 100. 100. 100. 100. 100.

RMS = 26. SMOOTHING INTERVAL 1969. THRU 1974.

ITERATION 1



REFERENCE YEAR 1969. TEST CASE

PN	NAME	START		DEVL		YRS		SUST	SS	SD	RS	RD	RECURRING OR FIXED ITEMS				
1	MARS71	1969.		0.		0.		4.	1	3	1	3	12.	25.	12.		
											3	2	1.	2.			
2	PIONER	1900.		0.		0.		0.	1	0	70	5	1.	6.	16.	16.	5.
3	MARS73	1902.	*	0.		0.		0.	1	0	71	4	2.	10.	25.	12.	
4	MERCRY	1903.	*	0.		0.		0.	1	0	73	4	0	2.	5.	3.	
5	GRDTRT	1903.	*	0.		0.		0.	1	0	72	4	1.	2.	6.	3.	
6	MARS75	1903.	*	0.		0.		0.	1	0	73	4	1.	4.	10.	5.	
7	COMET	1902.	*	0.		0.		0.	1	0	74	4	1.	3	8.	4.	
8	ASTRA	1900.		0.		0.		0.	1	0	74	4	0.	2.	4.	2.	
9	RELTIV	1901.	*	0.		0.		0.	1	0	72	4	0.	2.	4.	2.	
10	PIONDE	1901.	*	0.		0.		0.	1	0	73	4	1.	3.	6.	3.	
11	ATS	1900.		0.		0.		0.	1	0	70	5	1.	4.	12.	13.	4.
12	DRELAY	1900.		0.		0.		0.	1	0	74	4	0.	2.	4.	2.	
13	USAMSC	1901.	*	0.		0.		0.	1	0	73	4	0.	2.	4.	2.	
14	USAMSH	1900.		0.		0.		0.	1	0	74	4	1.	3.	8.	4.	
15	IMAGE	1969.		52.		6.	*	28.	4	3	1	0	0.				
16	PRETEN	1972.		2.		2.	*	12.	2	3	1	0	0.				
											4	1	15.				
17	DEV 3	1969.		25.		3.		16.	2	6	0	0					
18	DEV 7	1969.	*	0.		4.	*	0.	3	3	3	0					
19	DEV 9	1969.	*	0.		0.		6.	1	5	0	0					
20	DEV 14	1968.		88.		3.	*	0.	2	0	1	0					
											2	1	1.				
TOTAL				168.				256.									

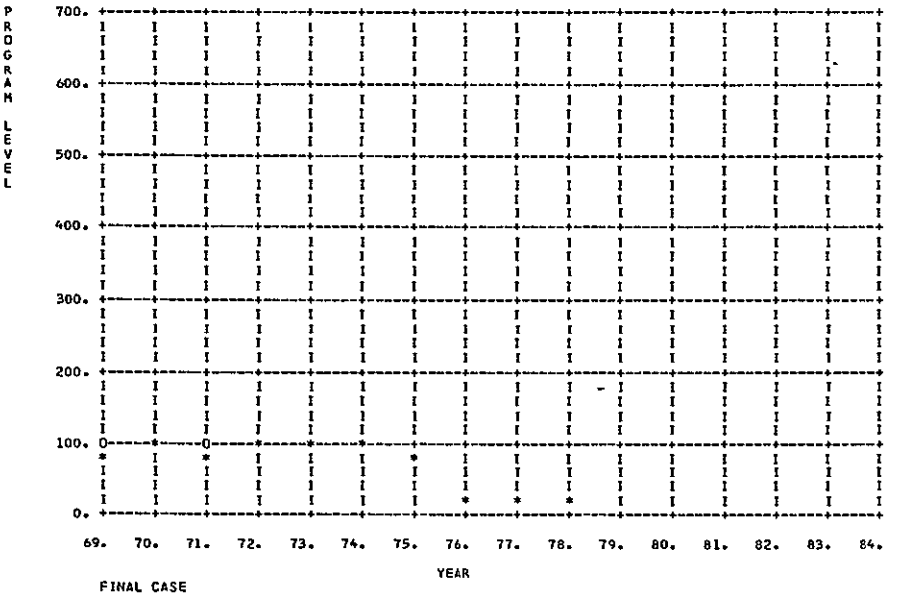
* INDICATES CHANGE FROM INPUT DATA

TOTAL PROGRAM COSTS AND LAUNCH VEHICLE SCHEDULE

YEAR	1969.	1970.	1971.	1972.	1973.	1974.	1975.	1976	1977.	1978.	1979.	1980.	1981.	1982.	1983.	1984.	1985.	1986.	1987.	1988.
PROGRAM																				
1 MARS71	16.	28.	17. 11*	2.																
2 PIONER	1.	6.	16.	16. 11*	5. 4*															
3 MARS73	0.	0.	0.	2.	10.	25.	12. 11*													
4 MERCRY	0.	0.	0.	0.	0.	0.	0.	2.	5.	3. 5*										
5 GRDTRT	0.	0.	0	0.	0.	1.	2.	6.	3. 11*											
6 MARS75	0.	0.	0.	0.	0.	0.	1.	4.	10.	5. 6*										
7 COMET	0.	0.	0.	0.	0.	0.	1.	3.	8.	4. 5*										
8 ASTRA	0.	0.	0.	0.	0.	2.	4.	2. 5*												
9 RELTIV	0.	0.	0.	0.	2.	4.	2. 5*													
10 PIONOE	0.	0.	0.	0.	1.	3.	6.	3. 12*												
11 ATS	1.	4.	12.	13. 5*	4. 5*															
12 DRELAY	0.	0.	0.	0.	0.	2.	4.	2. 5*												
13 USAMSC	0.	0.	0.	0.	0.	2.	4.	2. 5*												
14 USAMSM	0.	0.	0.	0.	1.	3.	8.	4. 5*												
15 IMAGE	3.	9.	13.	41.	37.	31.														
16 PRETEN	0.	0.	0.	1.	13.	12.	27.													
17 DEV 3	7.	28.	23.	16.	16.	16.	16.													
18 DEV 7	0.	0.	0.	0.	0.															
19 DEV 9	6.	6.	6.	6.	6.															
20 DEV 14	42.	23.																		
SUM	76.	105.	88.	98.	96.	100.	90.	29.	26.	12.										
FIXED	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.										
TOTAL	76. *	105. *	88. *	98. *	96. *	100. *	90.	29.	26.	12.										
LEVEL	100.	100.	100.	100.	100.	100.	100.	100.	100.	100.										

RMS = 11. SMOOTHING INTERVAL 1969. THRU 1974.

ITERATION 9



FINAL CASE

LAUNCH VEHICLE REQUIREMENTS BY YEAR

YEAR		1969.	1970.	1971.	1972.	1973.	1974.	1975.	1976.	1977.	1978.
LV	TOTAL										
4	1.00	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
5	6.50	0.0	0.0	0.0	1.0	1.0	0.0	0.5	2.5	0.0	1.5
6	1.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
11	5.50	0.0	0.0	2.0	1.0	0.0	0.0	2.0	0.0	0.5	0.0
12	8.50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.0

QUANTITIES BRANCHED UPON

	DEVELOPMENT	SUSTAINING		YEAR AVAIL	LAST YEAR	DEV START	DEV DURATION
1	0.0	3.37	SV3A STAGE	1	5	1970	0.
2	30.00	2.60	AG D STAGE	1	8	1971	0.
3	25.30	16.00	CENT STAGE	1	8	1969	3.
4	17.50	1.20	AGLT STAGE	3	8	1970	4.
5	13.00	0.0	SV3X STAGE	3	8	1970	4.
6	25.00	0.0	T3M STAGE	3	8	1970	4.
7	0.40	0.16	B2S STAGE	2	8	1969	4.
8	10.00	4.66	ATLS SHARED	1	8	1969	3.
9	0.0	6.00	TITN SHARED	1	8	1969	0.
10	0.0	2.50	AGEN SHARED	1	8	1970	0.
11	14.00	0.0	INTEGRATION OF TITN AND AGEN	1	8	1969	3.
12	2.50	0.0	INTEGRATION OF T3D AND AG D	1	8	1969	1.
13	5.50	0.0	INTEGRATION OF T3D AND AGLT	3	8	1969	5.
14	89.00	0.0	INTEGRATION OF TITN AND CENT	1	8	1968	3.

VEHICLE/MISSION CAPABILITY
(1 = POSSIBLE, 0 = IMPOSSIBLE)

VEHICLE / MISSION	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45
MISSION NUMBER	1 2 3 4 5 6 7 8 9 10 11 11 12 13 14
1 T3B AG D	0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0
2 T3B AG D B2S	0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0
3 T3B CENT	0 0 0 0 1 0 0 1 1 1 0 1 1 1 1 1
4 T3B CENT B2S	0 1 1 0 1 0 0 1 1 1 0 1 1 1 1 1
5 T3D TRAN	0 0 0 0 1 0 0 1 1 1 0 1 1 1 1 1
6 T3D TRAN B2S	0 0 0 0 1 0 1 1 1 1 0 1 1 1 1 1
7 T3D AG D	1 0 0 0 1 0 1 1 1 1 0 1 1 1 1 1
8 T3D AG D B2S	0 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1
9 T3D AGLT	0 0 1 0 1 0 1 1 1 1 0 0 1 1 1 1
10 T3D AGLT B2S	0 0 1 0 1 1 1 1 1 1 0 0 1 1 1 1
11 T3D CENT	1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1
12 T3D CENT B2S	0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
13 SV3A AG D	0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
14 SV3A AG D B2S	0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
15 SV3C CENT	0 0 0 0 1 0 0 1 1 1 0 1 1 1 0 0
16 SV3C CENT B2S	0 0 0 0 1 0 0 1 1 1 0 1 1 1 1 0
17 SV3X CENT	0 0 0 0 1 0 0 1 1 1 0 0 1 1 1 1
18 SV3X CENT B2S	0 0 0 0 1 0 0 1 1 1 0 0 1 1 1 1
19 T3M AGLT	0 0 1 1 1 1 1 1 1 1 0 0 1 1 1 1
20 T3M AGLT B2S	0 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1

********** S O L U T I O N **********

53	230.36	184.12	414.48

MISSION TITLE	CHARACTERISTIC VELOCITY(FT/SEC)	PAYLOAD (LBS)	LAUNCH YEAR	NUMBER OF LAUNCHES	OPTIMUM LAUNCH VEHICLE
MARS71	41500.	1100.	1971	2.00	T3D AG D
PIONER	49000.	460.	1972	1.00	T3D AG D B2S
			1973	1.00	T3D AG D B2S
MARS73	40000.	6000.	1975	2.00	T3M AGLT
MERCRY	38900.	1000.	1978	0.50	T3B AG D B2S
GRDTRT	49500.	800.	1977	0.50	T3D AG D B2S
MARS75	39400.	3000.	1978	1.00	T3D AG D
COMET	37200.	2000.	1978	1.00	T3D AG D
ASTRA	26300.	7000.	1976	0.50	T3B AG D
RELTIV	14200.	2000.	1975	0.50	T3B AG D
PIONDE	51400.	1000.	1976	0.50	T3M AGLT B2S
ATS	33600.	4000.	1972	1.00	T3D AG D
			1973	1.00	T3D AG D
DRELAY	14200.	2000.	1976	0.50	T3B AG D
USAMSC	33600.	4100.	1976	0.50	T3D AG D
USAMSM	30000.	6800.	1976	1.00	T3D AG D
********** S O L U T I O N **********					
53		234.87		184.12	418.99

MISSION TITLE	CHARACTERISTIC VELOCITY(FT/SEC)	PAYLOAD (LBS)	LAUNCH YEAR	NUMBER OF LAUNCHES	OPTIMUM LAUNCH VEHICLE
MARS71	41500.	1100.	1971	2.00	T3D AG D
PIONER	49000.	460.	1972	1.00	T3D AG D B2S
			1973	1.00	T3D AG D B2S
MARS73	40000.	6000.	1975	2.00	T3M AGLT
MERCRY	38900.	1000.	1978	0.50	T3B AG D B2S
GRDTRT	49500.	800.	1977	0.50	T3D AG D B2S
MARS75	39400.	3000.	1978	1.00	T3D AG D
COMET	37200.	2000.	1978	1.00	T3D AG D
ASTRA	26300.	7000.	1976	0.50	T3B AG D
RELTIV	14200.	2000.	1975	0.50	T3B AG D
PIONDE	51400.	1000.	1976	0.50	T3M AGLT B2S
ATS	33600.	4000.	1972	1.00	T3D AG D
			1973	1.00	T3D AG D
DRELAY	14200.	2000.	1976	0.50	T3B AG D
USAMSC	33600.	4100.	1976	0.50	T3D AG D
USAMSM	30000.	6800.	1976	1.00	T3D AG D

THE OPTIMUM SOLUTION HAS BEEN DETERMINED

RECURRING COST DATA

KEY	NAME	UNIT COST
1	T3B AG D	9.28
2	T3B AG DB2S	9.40
7	T3D AG D	13.83
8	T3D AG DB2S	15.31
19	T3M AGLT	29.97
20	T3M AGLTR2S	28.66

— REFERENCE YEAR 1969. TEST CASE

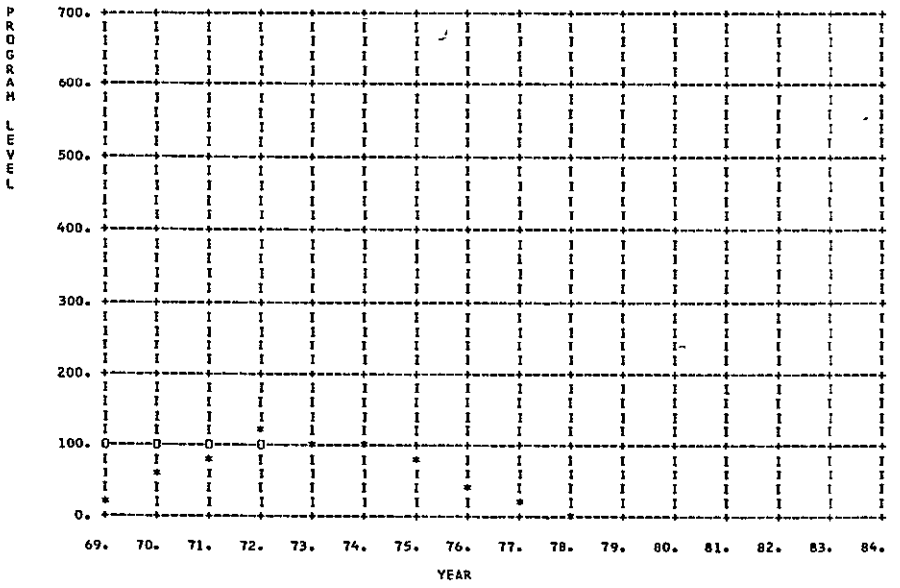
PN	NAME	START	DEVL	YRS	SUST	SS	SD	RS	RD	RECURRING OR FIXED ITEMS				
1	MARS71	1969.	0.	0.	4.	1	3	1	3	7.	14.	7.		
								3	2	1.	2.			
2	PIONER	1900.	0.	0.	0.	1	0	70	5	1.	4.	11.	11.	4.
3	MARS73	1902.	0.	0.	0.	1	0	71	4	0.	15.	30.	15.	
4	MERCRY	1903.	0.	0.	0.	1	0	73	4	0.	1.	3.	2.	
5	GRDTRT	1903.	0.	0.	0.	1	0	72	4	0.	2.	4.	2.	
6	MARS75	1903.	0.	0.	0.	1	0	73	4	1.	3.	7.	3.	
7	COMET	1902.	0.	0.	0.	1	0	74	4	1.	3.	7.	3.	
8	ASTRA	1900.	0.	0.	0.	1	0	74	4	0.	1.	1	2.	
9	RELTIV	1901.	0.	0.	0.	1	0	72	4	0.	1.	1.	2.	
10	PIONOE	1901.	0.	0.	0.	1	0	73	4	1	3.	7.	4.	
11	ATS	1900.	0.	0.	0.	1	0	70	5	1.	3.	10.	10.	3.
12	DRELAY	1900.	0.	0.	0.	1	0	74	4	0.	1.	1.	2.	
13	USAMSC	1901.	0.	0.	0.	1	0	73	4	0.	1.	3.	2.	
14	USAMSH	1900.	0.	0.	0.	1	0	74	4	1.	3.	7.	3.	
15	IMAGE	1969.	52.	6.	28.	4	3	0	0	0.				
16	PRETEN	1972.	2.	2.	12.	2	3	0	0	0.				
								4	1	15.				
17	DEV 2	1971.	0.	0.	3.	1	7	0	0					
								1	2	10	20.			
18	DEV 4	1970.	18.	4.	1.	3	5	0	0					
19	DEV 6	1970.	25.	4.	0.	3	0	0	0					
20	DEV 7	1969.	0.	4.	0.	3	8	0	0					
21	DEV 9	1969.	0.	0.	6.	1	7	0	0					
22	DEV 10	1970.	0.	0.	3.	1	7	0	0					
23	DEV 11	1969.	14.	3.	0.	2	0	0	0					
24	DEV 12	1969.	3.	1.	0.	1	0	0	0					
TOTAL			114.		215.									

TOTAL PROGRAM COSTS AND LAUNCH VEHICLE SCHEDULE

YEAR	1969.	1970.	1971.	1972.	1973.	1974.	1975.	1976.	1977.	1978.	1979.	1980.	1981.	1982.	1983.	1984.	1985.	1986.	1987.	1988.
PROGRAM																				
1 MARS71	10.	17.	11. 7*	2.																
2 PIONER	1.	4.	11.	11. 8*	4. 8*															
3 MARS73	0.	0.	0.	0.	15.	30.	15. 19*													
4 MERCRY	0.	0.	0.	0.	0.	0.	0.	1.	3.	2. 2*										
5 GRDTRT	0.	0.	0.	0.	0.	0.	2.	4.	2. 8*											
6 MARS75	0.	0.	0.	0.	0.	0.	1.	3.	7.	3. 7*										
7 COMET	0.	0.	0.	0.	0.	0.	1.	3.	7.	3. 7*										
8 ASTRA	0.	0.	0.	0.	0.	1.	1.	2. 1*												
9 RELTIV	0.	0.	0.	0.	1.	1.	2. 1*													
10 PIONOE	0.	0.	0.	0.	1.	3.	7.	4. 20*												
11 ATS	1.	3.	10.	10. 7*	3. 7*															
12 DRELAY	0.	0.	0.	0.	0.	1.	1.	2. 1*												
13 USAMSC	0.	0.	0.	0.	0.	1.	3.	2. 7*												
14 USAMSM	0.	0.	0.	0.	1.	3.	7.	3 7*												
15 IMAGE	3.	9.	13.	41.	37.	31.														
16 PRETEN	0.	0.	0.	1.	13.	12.	27.													
17 DEV 2	0.	0.	13.	23.	3.	3.	3.	3.	3.											
18 DEV 4	0.	3.	6.	7.	4.	1.	1.	1.												
19 DEV 6	0.	4.	9.	9.	4.															
20 DEV 7	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.										
21 DEV 9	6.	6.	6.	6.	6.	6.	6.													
22 DEV 10	0.	3.	3.	3.	3.	3.	3.	3.												
23 DEV 11	4.	7.	4.																	
24 DEV 12	2.																			
SUM	27.	56.	85.	113.	94.	96.	80.	31.	22.	9.										
FIXED	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.										
TOTAL	27.	56.	85.	113.	94.	96.	80.	31.	22.	9.										
LEVEL	* 100.	* 100.	* 100.	* 100.	* 100.	* 100.	100.	100.	100.	100.										

RMS = 36. SMOOTHING INTERVAL 1969. THRU 1974.

ITERATION 1



REPERENCE YEAR 1969. TEST CASE

PN	NAME	START		DEVL		YRS		SUST	SS	SD	RS	RD	RECURRING OR FIXED ITEMS				
1	MARS71	1969.		0.		0.		4.	1	3	1	3	7.	14.	7.		
											3	2	1.	2.			
2	PIONER	1902.	*	0.		0.		0.	1	0	70	5	1.	4.	11.	11.	4.
3	MARS73	1902.		0.		0		0.	1	0	71	4	0	15.	30.	15.	
4	MERCRY	1903.		0.		0.		0.	1	0	73	4	0.	1.	3.	2.	
5	GRDTRT	1904.	*	0.		0.		0.	1	0	72	4	0	2.	4.	2.	
6	MARS75	1903.		0.		0.		0.	1	0	73	4	1.	3.	7.	3.	
7	COMET	1902.		0.		0.		0.	1	0	74	4	1.	3.	7.	3.	
8	ASTRA	1901.	*	0.		0.		0.	1	0	74	4	0	1.	1.	2.	
9	RELTIV	1903.	*	0.		0.		0.	1	0	72	4	0.	1.	1.	2.	
10	PIONDE	1903.	*	0.		0.		0.	1	0	73	4	1	3.	7.	4.	
11	ATS	1900.		0.		0.		0.	1	0	70	5	1.	3.	10.	10.	3.
12	ORELAY	1901.	*	0.		0.		0.	1	0	74	4	0.	1.	1.	2.	
13	USAMSC	1903.	*	0.		0.		0.	1	0	73	4	0.	1.	3.	2.	
14	USAMSM	1902.	*	0.		0.		0.	1	0	74	4	1.	3.	7.	3.	
15	IMAGE	1969.		76.		5.	*	31.	4	4	-1	0	0.				
16	PRETEN	1969.	*	3.		4.	*	14.	3	3	2	0	0.			*	
											6	1	15.				
17	DEV 2	1969.	*	0.		0.		3.	1	7	0	0					
											1	2	10.	20.			
18	DEV 4	1969.	*	36.		3.	*	2.	2	7	-1	0					
19	DEV 6	1969.	*	27.		6.	*	0.	4	0	2	0					
20	DEV 7	1971.	*	1.		3.	*	0.	2	10	-1	0					
21	DEV 9	1969.		0.		0.		6.	1	7	0	0					
22	DEV 10	1969.	*	0.		0.		3.	1	7	0	0					
23	DEV 11	1969.		49.		2.	*	0.	2	0	-1	0					
24	DEV 12	1969.		3.		1.		0.	1	0	0	0					
TOTAL				195.				268.									

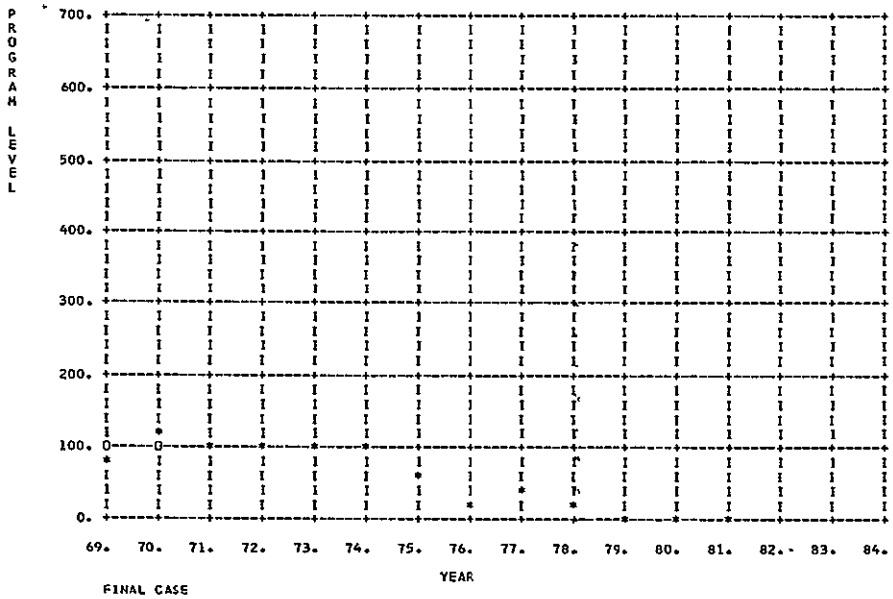
* INDICATES CHANGE FROM INPUT DATA

TOTAL PROGRAM COSTS AND LAUNCH VEHICLE SCHEDULE

YEAR	1969.	1970.	1971.	1972.	1973.	1974.	1975.	1976.	1977.	1978.	1979.	1980.	1981.	1982.	1983.	1984.	1985.	1986.	1987.	1988.
PROGRAM																				
1 MARS71	10.	17.	11. 7*	2.																
2 PIONER	0.	0.	1.	4.	11.	11. 8*	4. 8*													
3 MARS73	0.	0.	0.	0.	15.	30.	15. 19*													
4 MERCRY	0.	0.	0.	0.	0.	0.	0.	1.	3.	2. 2*										
5 GRDTRT	0.	0.	0.	0.	0.	0.	0.	2.	4.	2. 8*										
6 MARS75	0.	0.	0.	0.	0.	0.	1.	3.	7.	3. 7*										
7 COMET	0.	0.	0.	0.	0.	0.	1.	3.	7.	3. 7*										
8 ASTRA	0.	0.	0.	0.	0.	0.	1.	1.	2. 1*											
9 RELTIV	0.	0.	0.	0.	0.	0.	1.	1.	2. 1*											
10 PIONDE	0.	0.	0.	0.	0.	0.	1.	3.	7.	4. 20*										
11 ATS	1.	3.	10.	10. 7*	3. 7*															
12 ORELAY	0.	0.	0.	0.	0.	0.	1.	1.	2. 1*											
13 USAMSC	0.	0.	0	0.	0.	0.	0.	1.	3.	2. 7*										
14 USAMSM	0.	0.	0.	0.	0.	0.	1.	3.	7.	3. 7*										
15 IMAGE	7.	19.	24.	49.	38.	31.	31.													
16 PRETEN	0.	1.	15.	15.	14.	15.														
17 DEV 2	13.	23.	3.	3.	3.	3.	3.													
18 DEV 4	9.	19.	11.	2.	2.	2.	2.	2.												
19 DEV 6	2.	5.	7.	7.	5.	2.														
20 DEV 7	0.	0.	0.	1.	0.	0.	0.	0.	0	0.	0.	0.	0.							
21 DEV 9	6.	6.	6	6.	6.	6.	6.													
22 DEV 10	3.	3.	3.	3.	3.	3.	3.													
23 DEV 11	24.	24.																		
24 DEV 12	2.																			
SUM	78.	119.	91.	101.	100.	102.	70.	21.	46.	20.	0.	0.	0.							
FIXED	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.							
TOTAL	78.	119.	91.	101.	100.	102.	70.	21.	46.	20.	0.	0.	0.							
	*	*	*	*	*	*														
LEVEL	100.	100.	100.	100.	100.	100.	100.	100.	100.	100.	100.	100.	100.							

RHS = 13. SMOOTHING INTERVAL 1969. THRU 1974.

ITERATION 7



LAUNCH VEHICLE REQUIREMENTS BY YEAR

YEAR		1969.	1970.	1971.	1972.	1973.	1974.	1975.	1976.	1977.	1978.	1979.	1980.	1981.
LV	TOTAL													
1	1.50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.5	0.0	0.0	0.0	0.0
2	0.50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.0	0.0
7	7.50	0.0	0.0	2.0	1.0	1.0	0.0	0.0	0.0	0.0	3.5	0.0	0.0	0.0
8	2.50	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0 5	0.0	0.0	0.0
19	2.00	0.0	0.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0	.0.0	0.0	0.0	0.0
20	0.50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.0	0.0

QUANTITIES BRANCHED UPON

	DEVELOPMENT	SUSTAINING		YEAR AVAIL	LAST YEAR	DEV START	DEV DURATION
1	0.0	3.37	SV3A STAGE	1	5	1970	0.
2	30.00	2.60	AG D STAGE	1	8	1969	0.
3	25.30	16.00	CENT STAGE	1	8	1969	3.
4	35.75	1.75	AGLT STAGE	1	8	1969	3.
5	13.00	0.0	SV3X STAGE	3	8	1970	4.
6	27.50	0.0	T3M STAGE	4	8	1969	6.
7	0.82	0.26	B2S STAGE	3	8	1971	3.
8	10.00	4.66	ATLS SHARED	1	8	1969	3.
9	0.0	6.00	TITN SHARED	1	8	1969	0.
10	0.0	2.50	AGEN SHARED	1	8	1969	0.
11	48.86	0.0	INTEGRATION OF TITN AND AGEN	1	8	1969	2.
12	2.50	0.0	INTEGRATION OF T3D AND AG D	1	8	1969	1.
13	5.50	0.0	INTEGRATION OF T3D AND AGLT	3	8	1969	5.
14	89.00	0.0	INTEGRATION OF TITN AND CENT	1	8	1968	3.

VEHICLE/MISSION CAPABILITY
(1 = POSSIBLE, 0 = IMPOSSIBLE)

VEHICLE / MISSION	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45
MISSION NUMBER	1 2 2 3 4 5 6 7 8 9 10 11 11 12 13 14
1 T3B AG D	0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0
2 T3B AG D B2S	0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0
3 T3B CENT	0 0 0 0 1 0 0 1 1 1 0 1 1 1 1 1
4 T3B CENT B2S	0 1 1 0 1 0 0 1 1 1 0 0 1 1 1 1
5 T3D TRAN	0 0 0 0 1 0 0 1 1 1 0 1 1 1 1 1
6 T3D TRAN B2S	0 0 0 0 1 0 1 1 1 1 0 0 1 1 1 1
7 T3D AG D	1 0 0 0 1 0 1 1 1 1 0 1 1 1 1 1
8 T3D AG D B2S	0 1 1 0 1 1 1 1 1 1 0 0 1 1 1 1
9 T3D AGLT	0 1 1 0 1 0 1 1 1 1 0 0 1 1 1 1
10 T3D AGLT B2S	0 1 1 0 1 1 1 1 1 1 0 0 1 1 1 1
11 T3D CENT	1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1
12 T3D CENT B2S	0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1
13 SV3A AG D	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
14 SV3A AG D B2S	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
15 SV3C CENT	0 0 0 0 1 0 0 1 1 1 0 1 1 1 0 0
16 SV3C CENT B2S	0 0 0 0 1 0 0 1 1 1 0 0 1 1 1 0
17 SV3X CENT	0 0 0 0 1 0 0 1 1 1 0 0 1 1 1 1
18 SV3X CENT B2S	0 0 0 0 1 0 0 1 1 1 0 0 1 1 1 1
19 T3M AGLT	0 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1
20 T3M AGLT B2S	0 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1

********** SOLUTION **********

38	229.38	249.81	479.19

MISSION TITLE	CHARACTERISTIC VELOCITY(FT/SEC)	PAYLOAD (LBS)	LAUNCH YEAR	NUMBER OF LAUNCHES	OPTIMUM LAUNCH VEHICLE
MARS71	41500.	1100.	1971	2.00	T3D AG D
PIONER	49000.	460.	1974	1.00	T3D AG D B2S
			1975	1.00	T3D AG D B2S
MARS73	40000.	6000.	1975	2.00	T3M AGLT
MERCRY	38900.	1000.	1978	0.50	T3B AG D B2S
GRDTRT	49500.	800.	1978	0.50	T3D AG D B2S
MARS75	39400.	3000.	1978	1.00	T3D AG D
COMET	37200.	2000.	1978	1.00	T3D AG D
ASTRA	26300.	7000.	1977	0.50	T3B AG D
RELTIV	14200.	2000.	1977	0.50	T3B AG D
PIONDE	51400.	1000.	1978	0.50	T3M AGLT B2S
ATS	33600.	4000.	1972	1.00	T3D AG D
			1973	1.00	T3D AG D
DRELAY	14200.	2000.	1977	0.50	T3B AG D
USAMSC	33600.	4100.	1978	0.50	T3D AG D
USAMSM	30000.	6800.	1978	1.00	T3D AG D
********** S O L U T I O N **********					
38		234.74		249.81	484.54

MISSION TITLE	CHARACTERISTIC VELOCITY(FT/SEC)	PAYLOAD (LBS)	LAUNCH YEAR	NUMBER OF LAUNCHES	OPTIMUM LAUNCH VEHICLE
MARS71	41500.	1100.	1971	2.00	T3D AG D
PIONER	49000.	460.	1974	1.00	T3D AG D B2S
			1975	1.00	T3D AG D B2S
MARS73	40000.	6000.	1975	2.00	T3M AGLT
MERCRY	38900.	1000.	1978	0.50	T3B AG D B2S
GRDTRT	49500.	800.	1978	0.50	T3D AG D B2S
MARS75	39400.	3000.	1978	1.00	T3D AG D
COMET	37200.	2000.	1978	1.00	T3D AG D
ASTRA	26300.	7000.	1977	0.50	T3B AG D
RELTIV	14200.	2000.	1977	0.50	T3B AG D
PIONDE	51400.	1000.	1978	0.50	T3M AGLT B2S
ATS	33600.	4000.	1972	1.00	T3D AG D
			1973	1.00	T3D AG D
DRELAY	14200.	2000.	1977	0.50	T3B AG D
USAMSC	33600.	4100.	1978	0.50	T3D AG D
USAMSM	30000.	6800.	1978	1.00	T3D AG D

THE OPTIMUM SOLUTION HAS BEEN DETERMINED

RECURRING COST DATA

KEY	NAME	UNIT COST
1	T3B AG D	9.33
2	T3B AG DB2S	9.14
7	T3D AG D	13.80
8	T3D AG DB2S	15.53
19	T3H AGLT	29.69
20	T3H AGLTB2S	29.00

REFERENCE YEAR 1969. TEST CASE

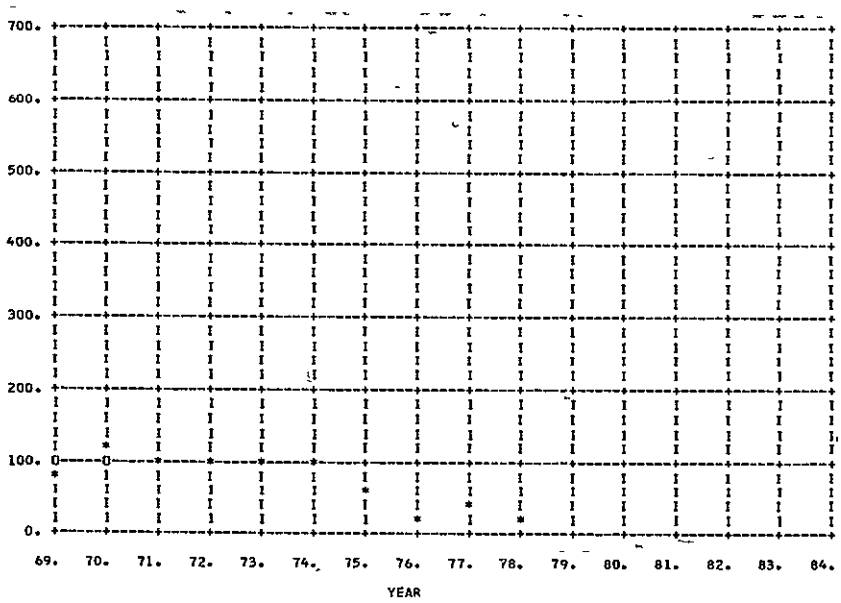
PN	NAME	START	DEVL	YRS	SUST	SS	SD	RS	RD	RECURRING OR FIXED ITEMS				
1	MARS71	1969.	0.	0.	4.	1	3	1	3	7.	14.	7.		
								3	2	1.	2.			
2	PIONER	1902.	0.	0.	0.	1	0	70	5	1.	4.	11.	12.	4.
3	MARS73	1902.	0.	0.	0.	1	0	71	4	0.	15.	30.	15.	
4	MERCRY	1903.	0.	0.	0.	1	0	73	4	0.	1.	3.	2.	
5	GRDTRT	1904.	0.	0.	0.	1	0	72	4	0.	2.	4.	2.	
6	MARS75	1903.	0.	0.	0.	1	0	73	4	1.	3.	7.	3.	
7	COMET	1902.	0.	0.	0.	1	0	74	4	1.	3.	7.	3.	
8	ASTRA	1901.	0.	0.	0.	1	0	74	4	0.	1.	1.	2.	
9	RELTIV	1903.	0.	0.	0.	1	0	72	4	0.	1.	1.	2.	
10	PIONOE	1903.	0.	0.	0.	1	0	73	4	1.	3.	7.	4.	
11	ATS	1900.	0.	0.	0.	1	0	70	5	1.	3.	10.	10.	3.
12	DRELAY	1901.	0.	0.	0.	1	0	74	4	0.	1.	1.	2.	
13	USAMSC	1903.	0.	0.	0.	1	0	73	4	0.	1.	3.	2.	
14	USAMSH	1902.	0.	0.	0.	1	0	74	4	1.	3.	7.	3.	
15	IMAGE	1969.	76.	5.	31.	4	4	0	0	0.				
16	PRETEN	1969.	3.	4.	14.	3	3	0	0	0.				
								6	1	15.				
17	DEV 2	1969.	0.	0.	3.	1	7	0	0					
								1	2	10.	20.			
18	DEV 4	1969.	36.	3.	2.	2	9	0	0					
19	DEV 6	1969.	27.	6.	0.	4	0	0	0					
20	DEV 7	1971.	1.	3.	0.	2	7	0	0					
21	DEV 9	1969.	0.	0.	6.	1	7	0	0					
22	DEV 10	1969.	0.	0.	3.	1	7	0	0					
23	DEV 11	1969.	49.	2.	0.	2	0	0	0					
24	DEV 12	1969.	3.	1.	0.	1	0	0	0					
TOTAL			195.		271.									

TOTAL PROGRAM COSTS AND LAUNCH VEHICLE SCHEDULE

YEAR	1969.	1970.	1971.	1972.	1973.	1974.	1975.	1976.	1977.	1978.	1979.	1980.	1981.	1982.	1983.	1984.	1985.	1986.	1987.	1988.
PROGRAM																				
1 MARS71	10.	17.	11. 7*	2.																
2 PIONER	0.	0.	1.	4.	11.	12. 8*	4. 8*													
3 MARS73	0.	0.	0.	0.	15.	30.	15. 19*													
4 MERCRY	0.	0.	0.	0.	0.	0.	0.	1.	3.	2. 2*										
5 GRDTRT	0.	0.	0.	0.	0.	0.	0.	2.	4.	2. 8*										
6 MARS75	0.	0.	0.	0.	0.	0.	1.	3.	7.	3. 7*										
7 COMET	0.	0.	0.	0.	0.	0.	1.	3.	7.	3. 7*										
8 ASTRA	0.	0.	0.	0.	0.	0.	1.	1.	2. 1*											
9 RELTIV	0.	0.	0.	0.	0.	0.	1.	1.	2. 1*											
10 PIONOE	0.	0.	0.	0.	0.	0.	1.	3.	7.	4. 20*										
11 ATS	1.	3.	10.	10. 7*	3. 7*															
12 ORELAY	0.	0.	0.	0.	0.	0.	1.	1.	2. 1*											
13 USAMSC	0.	0.	0.	0.	0.	0.	0.	1.	3.	2. 7*										
14 USAMSN	0.	0.	0.	0.	0.	0.	1.	3.	7.	3. 7*										
15 IMAGE	7.	19.	24.	49.	38.	31.	31.													
16 PRETEN	0.	1.	15.	15.	14.	15.														
17 DEV 2	13.	23.	3.	3.	3.	3.	3.													
18 DEV 4	9.	19.	11.	2.	2.	2.	2.	2.	2.	2.										
19 DEV 6	2.	5.	7.	7.	5.	2.														
20 DEV 7	0.	0.	0.	1.	0.	0.	0.	0.	0.	0.										
21 DEV 9	6.	6.	6.	6.	6.	6.	6.													
22 DEV 10	3.	3.	3.	3.	3	3.	3.													
23 DEV 11	24.	24.																		
24 DEV 12	2.																			
SUM	78.	119.	91.	101.	100.	102.	70.	21.	48.	21.										
FIXED	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.										
TOTAL	78.	119.	91.	101.	100.	102.	70.	21.	48.	21.										
	*	*	*	*	*	*														
LEVEL	100.	100.	100.	100.	100.	100.	100.	100.	100.	100.										

RMS = 13. SMOOTHING INTERVAL 1969. THRU 1974.

ITERATION 1



REFERENCE YEAR 1969. TEST CASE

*PN	NAME	START	DEVL	YRS		SUST	SS	SD	RS	RD	RECURRING OR FIXED ITEMS				
1	MARS71	1969.	0.	0.		4.	1	3	1	3	7.	14.	7.		
									3	2	1.	2.			
2	PIONER	1902.	0.	0.		0.	1	0	70	5	1.	4.	11.	12.	4.
3	MARS73	1902.	0.	0.		0.	1	0	71	4	0.	15.	30.	15.	
4	MERCRY	1903.	0.	0.		0.	1	0	73	4	0.	1.	3.	2.	
5	GRDTRT	1904.	0.	0.		0.	1	0	72	4	0.	2.	4.	2.	
6	MARS75	1903.	0.	0.		0.	1	0	73	4	1.	3.	7.	3.	
7	COMET	1902.	0.	0.		0.	1	0	74	4	1.	3.	7.	3.	
8	ASTRA	1901.	0	0.		0.	1	0	74	4	0.	1.	1.	2.	
9	RELTIV	1903.	0.	0.		0.	1	0	72	4	0.	1.	1.	2.	
10	PIONDE	1903.	0.	0.		0.	1	0	73	4	1.	3.	7.	4.	
11	ATS	1900.	0.	0.		0.	1	0	70	5	1.	3.	10.	10.	3.
12	DRELAY	1901.	0.	0.		0.	1	0	74	4	0.	1.	1.	2.	
13	USAMSC	1903.	0.	0.		0.	1	0	73	4	0.	1.	3.	2.	
14	USAMSN	1902.	0.	0.		0.	1	0	74	4	1.	3.	7	3.	
15	IMAGE	1969.	76.	5.		31.	4	4	0	0	0.				
16	PRETEN	1969.	3.	4.		14.	3	3	0	0	0.				
									6	1	15.				
17	DEV 2	1969.	0.	0.		3.	1	7	0	0					
									1	2	10.	20.			
18	DEV 4	1969.	38.	4.	*	2.	3	9	1	0					
19	DEV 6	1969.	27.	6.		0.	4	0	0	0					
20	DEV 7	1971.	1.	4.	*	0.	3	7	1	0					
21	DEV 9	1969.	0.	0.		6.	1	7	0	0					
22	DEV 10	1969.	0.	0.		3.	1	7	0	0					
23	DEV 11	1969.	49.	2.		0.	2	0	0	0					
24	DEV 12	1969.	3.	1.		0.	1	0	0	0					
TOTAL			197.			272.									

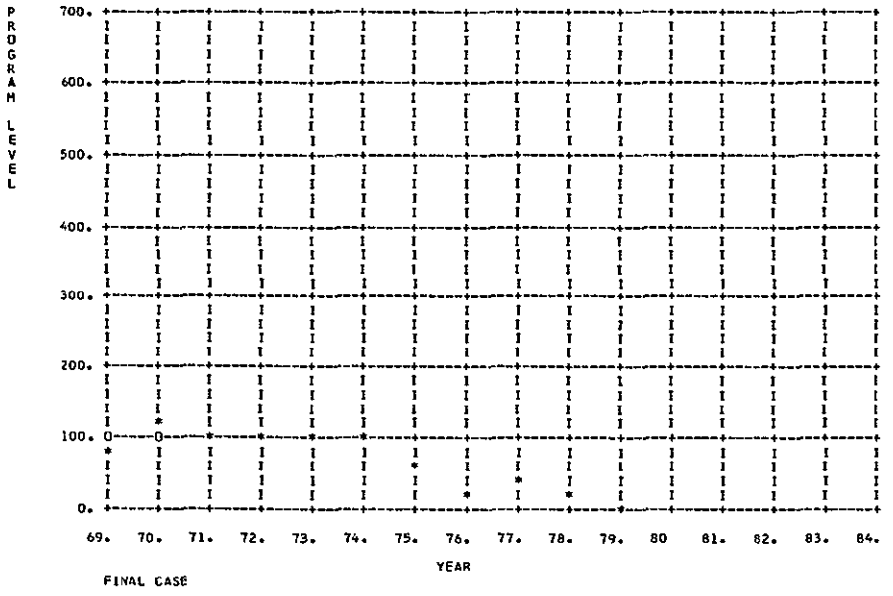
* INDICATES CHANGE FROM INPUT DATA

TOTAL PROGRAM COSTS AND LAUNCH VEHICLE SCHEDULE

YEAR	1969.	1970.	1971.	1972.	1973.	1974.	1975.	1976.	1977.	1978.	1979.	1980.	1981.	1982.	1983.	1984.	1985.	1986.	1987.	1988.
PROGRAM																				
1 MARS71	10.	17.	11. 7*	2.																
2 PIONER	0.	0.	1.	4.	11.	12. 8*	4. 8*													
3 MARS73	0.	0.	0.	0.	15.	30.	15. 19*													
4 MERCRY	0.	0.	0.	0.	0.	0.	0.	1.	3.	2. 2*										
5 GRDTRT	0.	0.	0.	0.	0.	0.	0.	2.	4.	2. 8*										
6 MARS75	0.	0.	0.	0.	0.	0.	1.	3.	7.	3. 7*										
7 COMET	0.	0.	0.	0.	0.	0.	1.	3.	7.	3. 7*										
8 ASTRA	0.	0.	0.	0.	0.	0.	1.	1.	2. 1*											
9 RELTIV	0.	0.	0.	0.	0.	0.	1.	1.	2. 1*											
10 PIONOE	0.	0.	0.	0.	0.	0.	1.	3.	7.	4. 20*										
11 ATS	1.	3.	10.	10. 7*	3. 7*															
12 ORELAY	0.	0.	0.	0.	0.	0.	1.	1.	2. 1*											
13 USAMSC	0.	0.	0.	0.	0.	0.	0.	1.	3.	2. 7*										
14 USAMSM	0.	0.	0.	0.	0.	0.	1.	3.	7.	3. 7*										
15 IMAGE	7.	19.	24.	49.	38.	31.	31.													
16 PRETEN	0.	1.	15.	15.	14.	15.														
17 DEV 2	13.	23.	3.	3.	3.	3.	3.													
18 DEV 4	6.	13.	15.	8.	2.	2.	2.	2.	2.	2.	2.	2.								
19 DEV 6	2.	5.	7.	7.	5.	2.														
20 DEV 7	0.	0.	0.	0.	1.	0.	0.	0.	0.	0.	0.	0.								
21 DEV 9	6.	6	6.	6.	6.	6.	6.													
22 DEV 10	3.	3.	3.	3.	3.	3.	3.													
23 DEV 11	24.	24.																		
24 DEV 12	2.																			
SUM	74.	114.	94.	107.	100.	102.	70.	21.	48.	21.	2.									
FIXED	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.									
TOTAL	74.	114.	94.	107.	100.	102.	70.	21.	48.	21.	2.									
LEVEL	* 100.	* 100.	* 100.	* 100.	* 100.	* 100.	* 100.	100.	100.	100.	100.									

RMS = 13. SMOOTHING INTERVAL 1969. THRU 1974.

ITERATION 3



FINAL CASE

LAUNCH VEHICLE REQUIREMENTS BY YEAR

YEAR		1969.	1970.	1971.	1972.	1973.	1974.	1975.	1976.	1977.	1978.	1979.
LV	TOTAL											
1	1.50	0.0	0.0	0 0	0.0	0.0	0.0	0.0	0.0	1.5	0.0	0.0
2	0.50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0 5	0.0
7	7.50	0.0	0.0	2.0	1.0	1.0	0.0	0.0	0.0	0.0	3.5	0.0
8	2.50	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.5	0.0
19	2.00	0.0	0.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0
20	0.50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0

INPUT ASSIGNMENT IS OPTIMUM SMOOTHED SOLUTION

OPTIMUM ASSIGNMENT WITHIN BUDGET CONSTRAINTS HAS BEEN DETERMINED

END OF DATA - JOB COMPLETE

Appendix G
FLOW CHARTS

G.1 DESCRIPTION

Flow charts are provided in this section for each of the major subroutines and the main program MASTER. They appear in alphabetical order by subroutine name. A short description of the purpose of each subroutine is provided in the program listing in Appendix H. Subroutines AFRMT, INPUT, PLOT, and PACK were written in 360 Assembler Language so a description of each subroutine appears in this section rather than a flow chart.

G.2 MAJOR SUBROUTINE CHARTS

The subroutine flow charts follow.

SUBROUTINE AFRMT

IDENTIFICATION

Subroutine AFRMT
Deck Name MOX02AT
Fortran IV subroutine coded in 360 Assembler Language
Written by R. E. Slye

PURPOSE

This subroutine converts a variable from integer to A format

USAGE

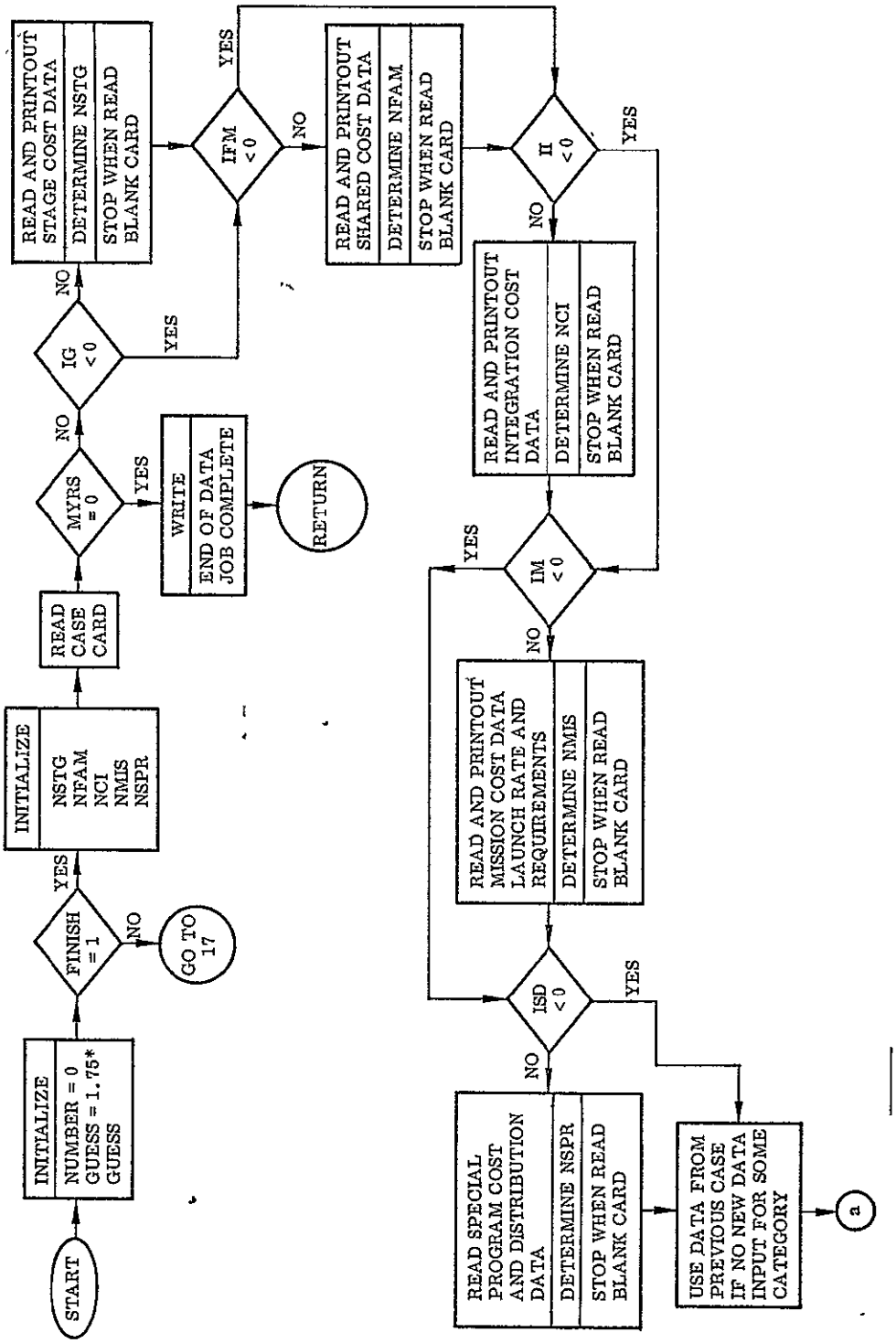
CALL AFRMT (I, X)

where

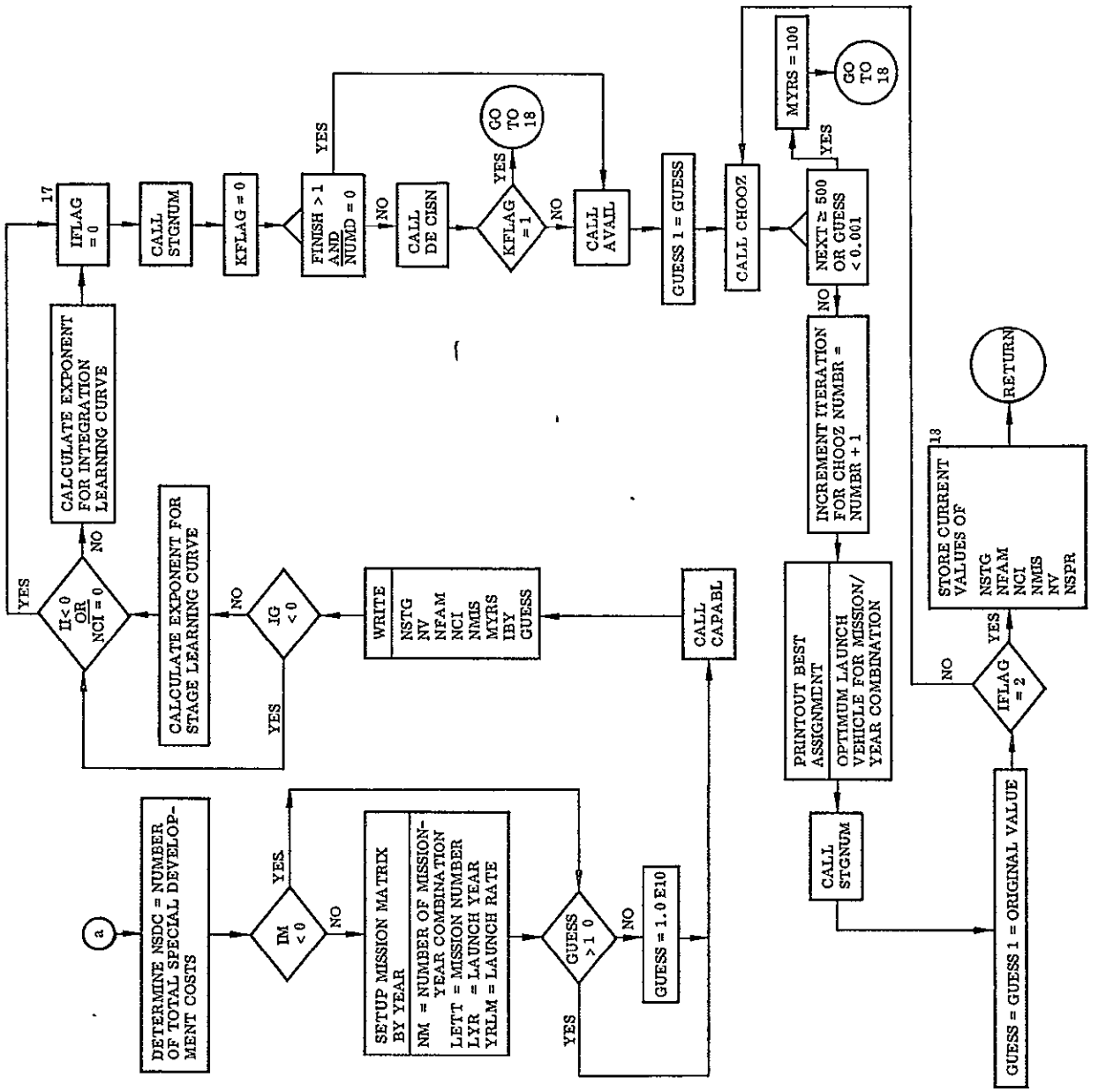
I is the name of the variable (may be one element of an array) in integer format

X is the name of the result returned in A4 format

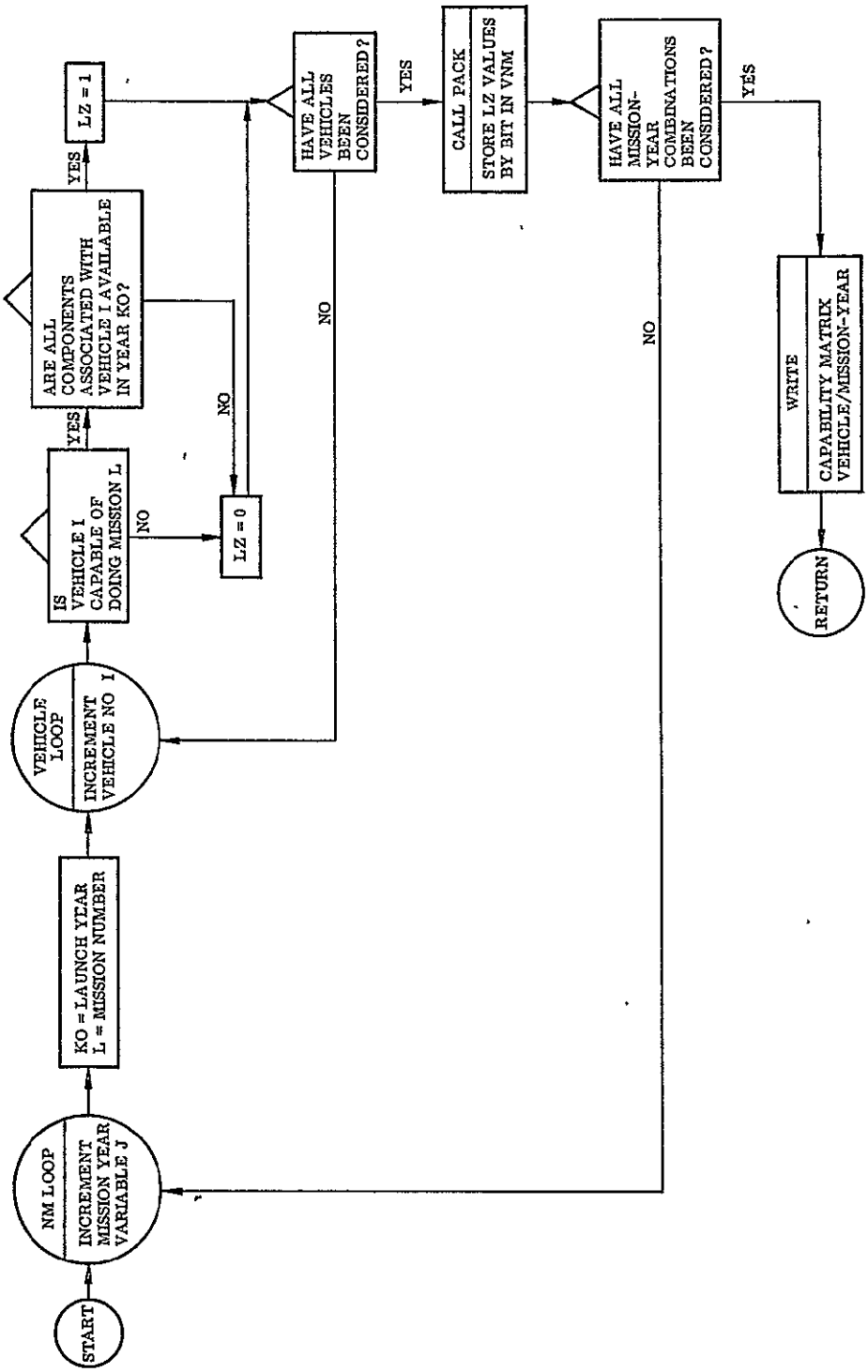
SUBROUTINE ASSIGN



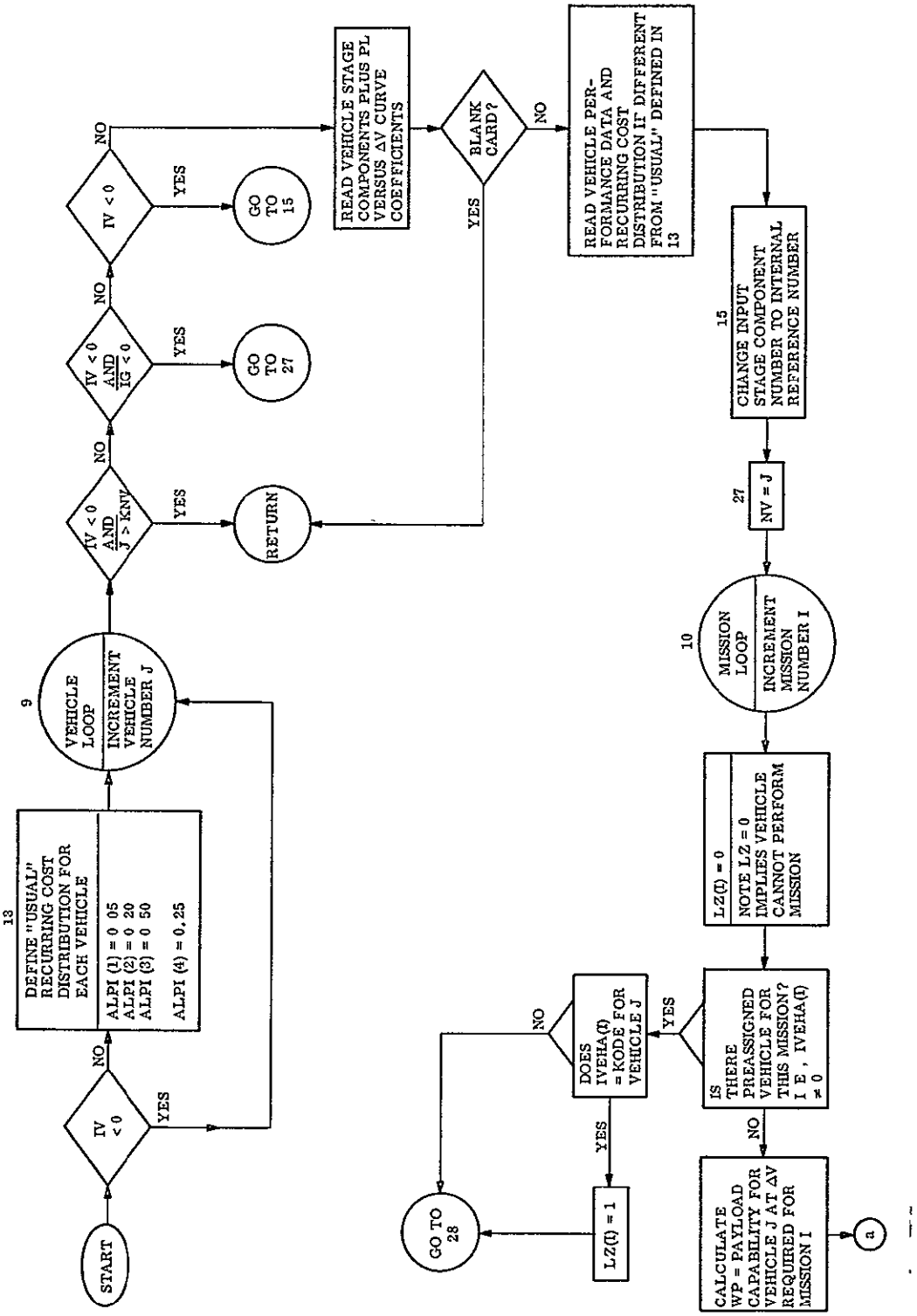
SUBROUTINE ASSIGN (Cont.)



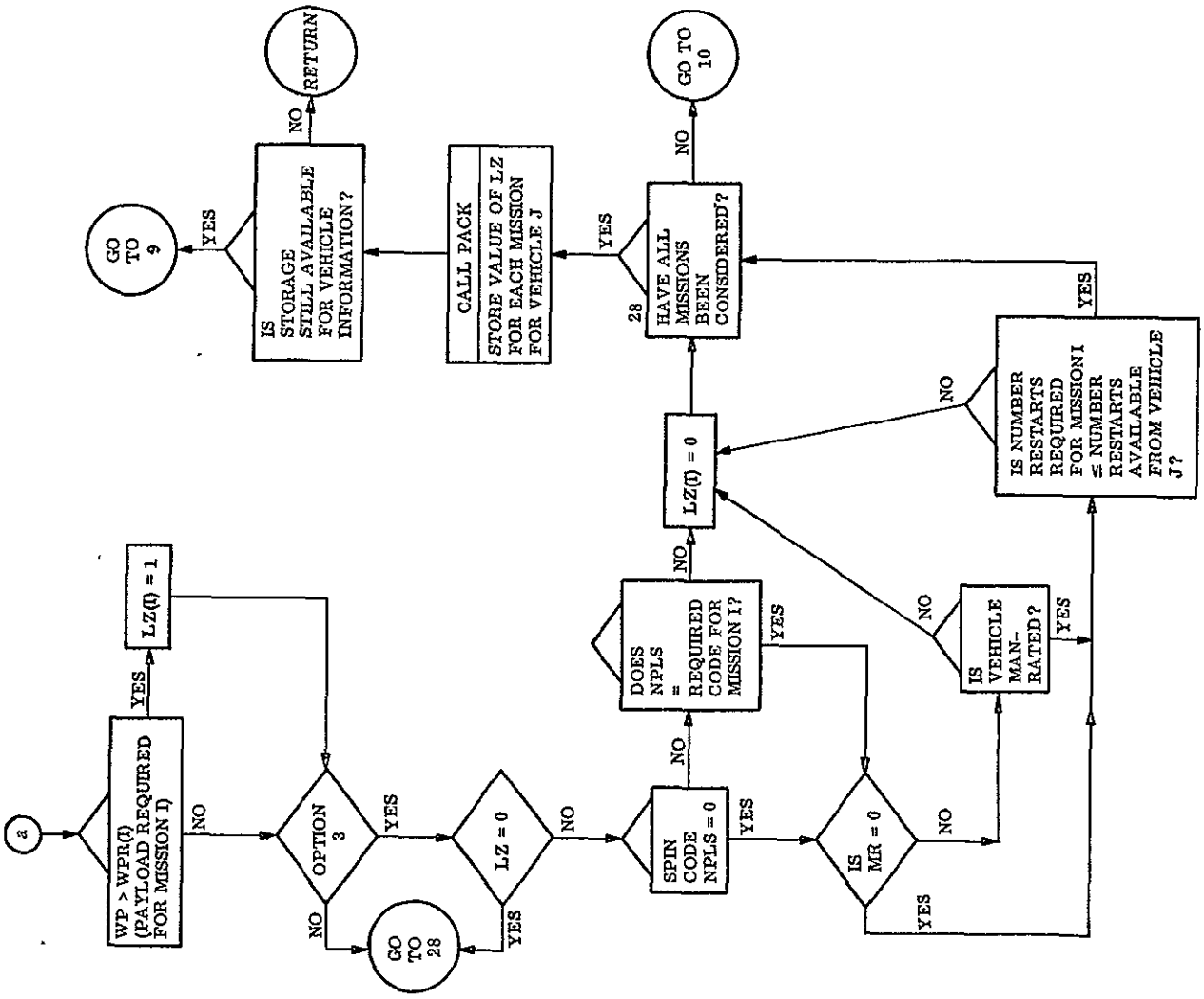
SUBROUTINE AVAIL



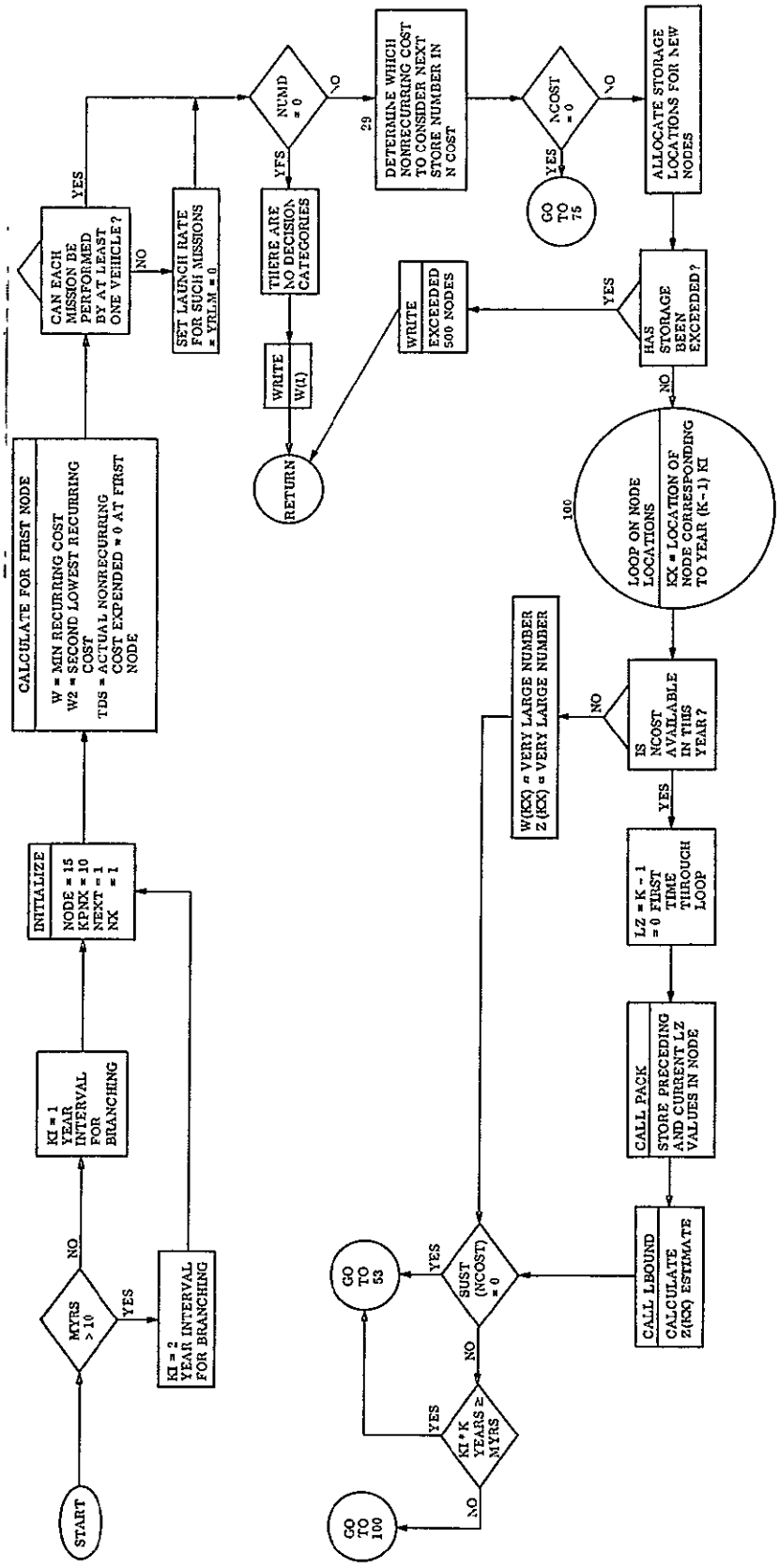
SUBROUTINE CAPABL



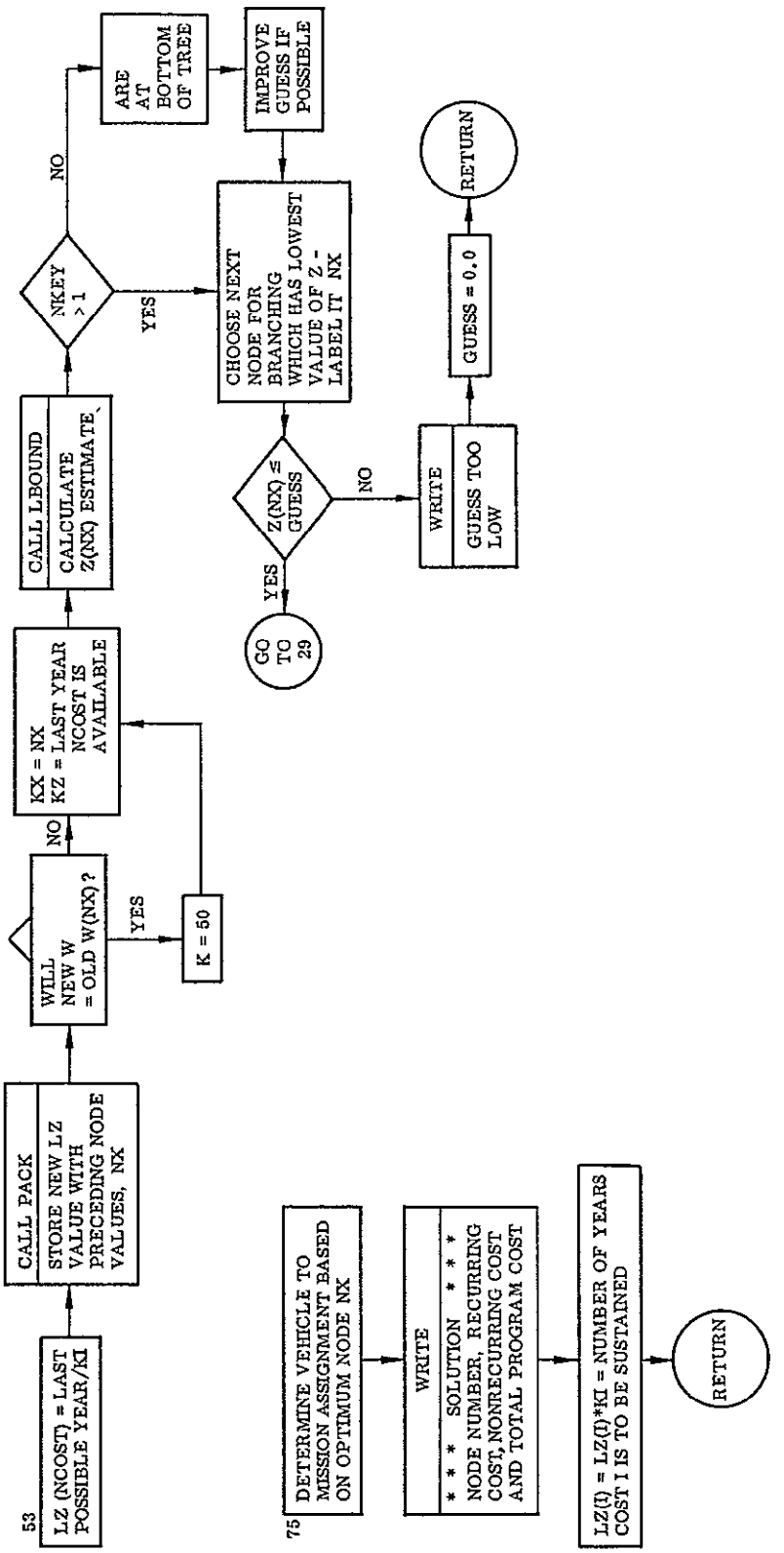
SUBROUTINE CAPABL (Cont.)



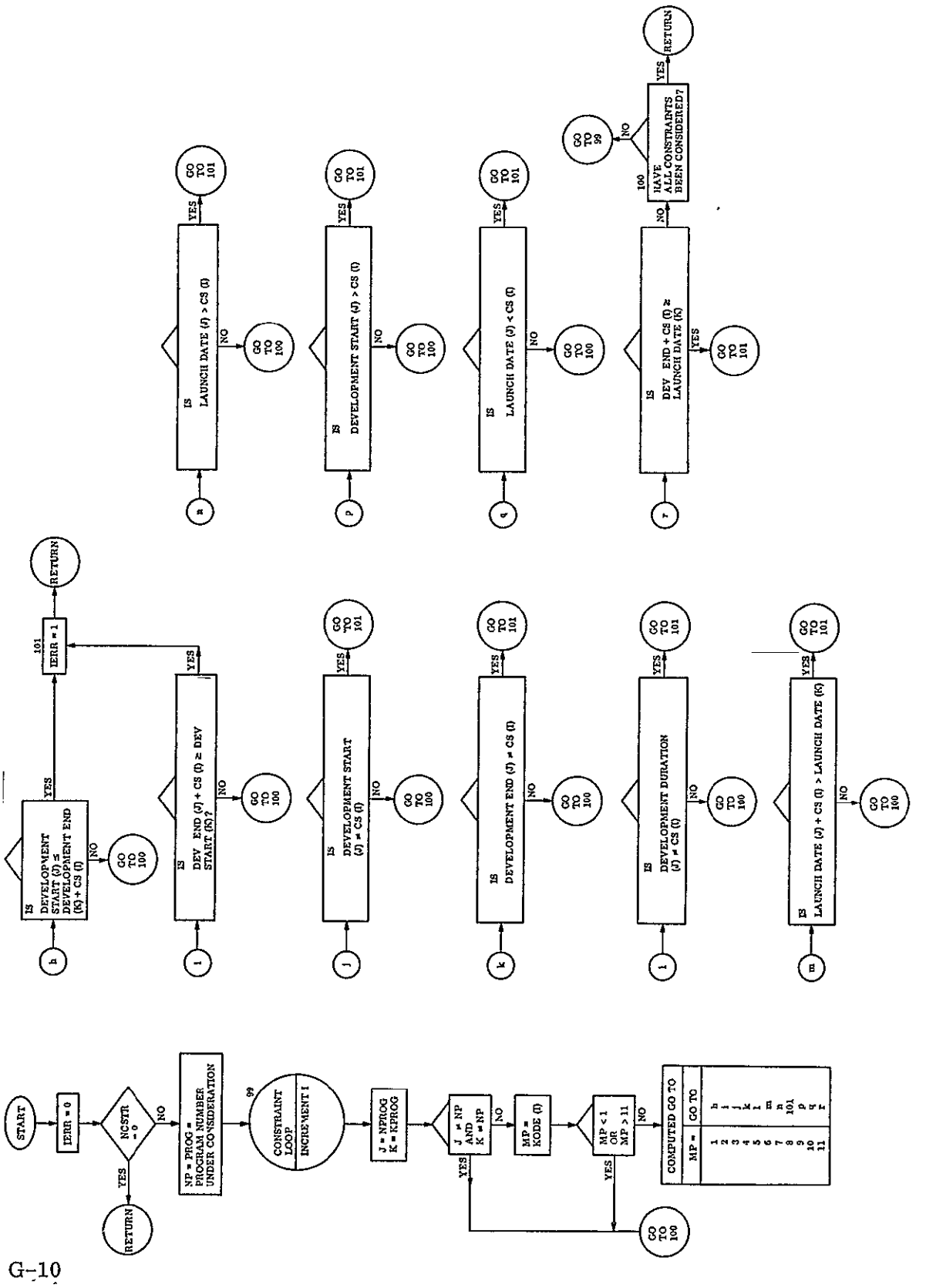
SUBROUTINE CHOOZ



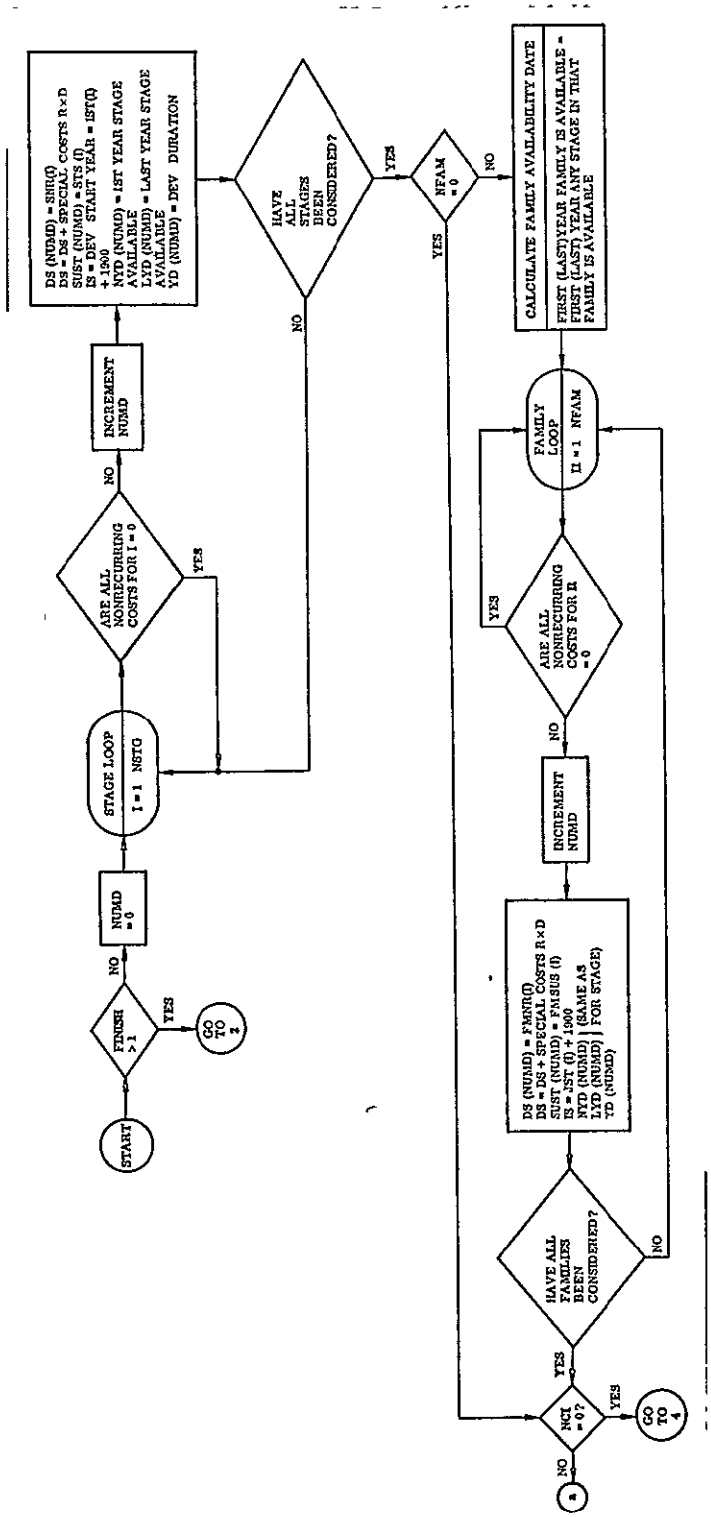
SUBROUTINE CHOOZ (Cont.)



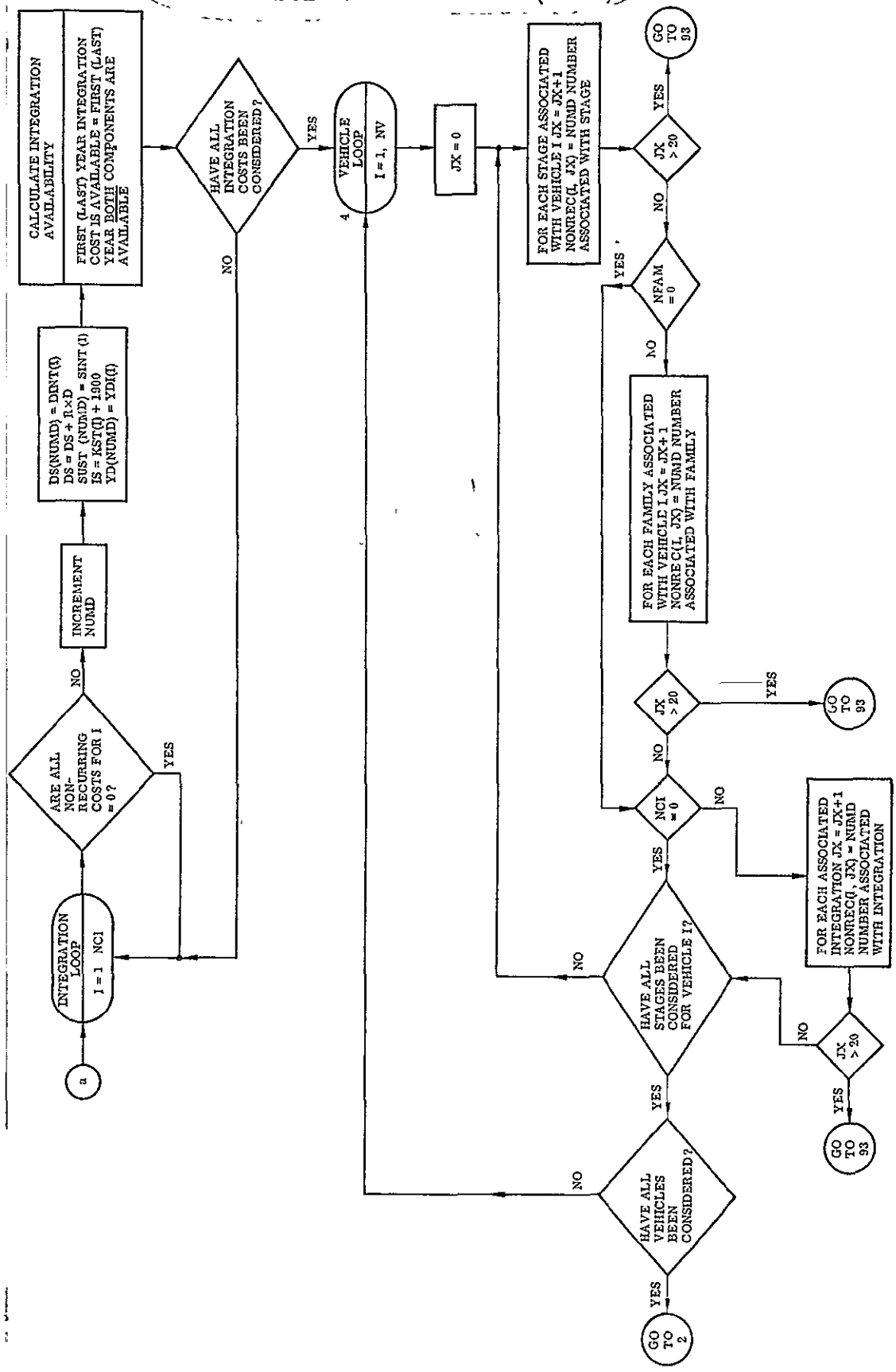
SUBROUTINE CONSTR



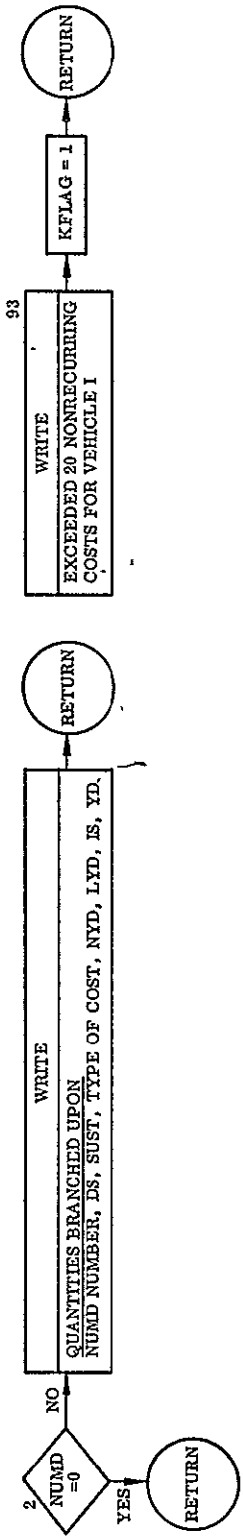
SUBROUTINE DECISN



SUBROUTINE DECISN (Cont.)



SUBROUTINE DECISN (Cont.)



SUBROUTINE INPUT

IDENTIFICATION

AL INPT Generalized Data Input Subroutine
360/Assembler Language
Written by R. E. Slye

PURPOSE

This subroutine provides for input of single-precision fixed and floating point numbers and Hollerith information. Usage is particularly convenient inasmuch as no format statements are required, and data may be loaded in any order irrespective of the order in the calling statement.

USAGE

The calling statement is

```
CALL INPUT (5HALPHA, ALPHA, 4BETA, BETA, ...)
```

In the above, the Hollerith literals represent the external names of variables or arrays as they should appear on data cards. The other arguments are the internal names of the variables and arrays as referenced in the source program. It will become apparent that by using the external names in addition to the symbolic location names, it is possible to enter data for a variable on an input card without regard to its relative location in the calling sequence of the program.

ACCEPTABLE INPUT DATA FORMS -

A. Floating Point General Form

Examples

Up to 9 decimal digits, with a decimal point permitted at the beginning, at the end or between two digits. A preceding plus or minus sign is optional. A decimal exponent preceded by E+ or + or - if negative may follow. If no decimal point appears, the exponent is mandatory. The magnitude of the number must be between the approximate limits of 10^{-75} and 10^{75}.

17.
5.0
.0003
5.0E3 (5.0×10^{3})
5.0E+3 (5.0×10^{3})
5.0E-7 (5.0×10^{-7})

B. Decimal Integers General Form

Examples

The magnitude of the number must be less than 2^{31}. A preceding plus or minus sign is optional.

3
+1
-28987

C. Hollerith Information General Form

Examples

Any number of characters, including blanks. The number of characters is specified by writing nH preceding the Hollerith information. n is the number of characters in the block following nH.

14HTHIS IS A TEST
6HALPHA

RULES FOR PREPARATION OF DATA CARDS

Blanks are ignored except within Hollerith data fields.

Data must be contained within card columns 1 through 72.

It is not necessary that variable names on the data cards appear in the same order as those in the calling sequence. The routine will search the list for the name and its core location.

Individual data items are separated by commas.

An equal sign separates the name of a variable and its first data item.

A comma separates the end of a data set and the next variable name.

A data input record is terminated by an asterisk (*).

It is not necessary to input a data set for each name in the calling sequence.

Elements of an array may be skipped by writing consecutive commas — i.e., no data between the commas; or by singly subscripting the array name. Double subscripting is illegal. Thus, if it is desired to input data into a three-element vector V, one could write:

$$V = 2.79,,1.32$$

No data would be entered into V(2). What was originally there remains there. Alternatively, the above could be written:

$$V(1) = 2.79,\ V(3) = 1.32$$

Special Feature. The card image is normally written on the system output unit, tape 6, prior to being processed by the routine. If an N is punched in column 73, the card will not be listed. If column 73 contains a C, the card is treated as a comment only; i.e., it is not scanned for data. If the card contains CE in columns 73 — 74, the card will be treated as a comment card, and a page will be ejected.

EXAMPLE

If the following call statement appeared in a FORTRAN program,

CALL INPUT (1HA, A, 1HB, B, 1HC, C, 1HD, D, 1HP, P, 1HR, R, 1HS, S)

the input cards could be punched as follows:

```
A    = 3.14159265, B = 707, C = 1870,      1st card
D    = 1., 2., 3., 4., 5., 6., 7., 8., 9., 2nd card
R(2) = 3, R(5) = 74., 42,                  3rd card
F    = 22HTHIS IS A CHECKOUT RUN*          4th card
```

Note that D must be dimensioned at least 9, R dimensioned at least 7 and P at least 6.

Also R(1), R(3), R(4), and R(6) are unchanged.

Even though S appears in the CALL statement, it is not necessary that it appear on one of the input cards. The * on card 4 signifies the end of the data record. This means that the routine will return control to the calling program.

RESTRICTIONS

The following errors will be detected by the subroutine. A diagnostic message and the card in error will be permited on the system output unit, tape 6.

1. Name on data card exceeds six characters.
2. Name on data card does not appear in the calling sequence.
3. Punctuation errors.
4. Name on data card begins with a non-alphabetic character.
5. Decimal or integer data out of range.

This subroutine may be used for reading double precision numbers; however, only the high order part of the number will be loaded. To clear the low order part of the number, write

```
DWORD = 1., 0,
```

ADDITIONAL INFORMATION

1. A slash (/) on a data card (not in an H field) indicates that information to the right of the slash is not to be scanned for data. Therefore, these columns may be used for comments.
2. In addition to the above means for entering Hollerith information, Hollerith may also be entered by enclosing it in apostrophes, i.e., P = 'THIS IS A CHECKOUT RUN'
3. Floating point and integer data may be repeated into consecutive locations by use of the letter X followed by the data; i.e.,

   D = 1., 4X2., 3.,

   is equivalent to

   D = 1., 2., 2., 2., 2., 3.,

4. Alphanumeric data may also be repeated. The use of the letter X is optional. For example, to set an array dimensioned 18 to blanks, write

   TITLE = 18' ',

   If the alphanumeric field exceeds 4 characters, only the last word will be repeated. For example,

   DATA = 3'ABCDEF', will result in
   ABCDEF EF EF

5. If a name on a data card is not followed by an equal sign, it will be retrieved from the calling program. For example, if in the calling program, X and ALPHA are dimensioned at least 2, then the following data card

   X = 3.1, ALPHA(2),

   will result in the current value of ALPHA(2) being stored in X(2).

As an additional example, suppose that the calling FORTRAN program has the following sequence:

```
      LOGICAL
       . . .
       . . .
      TRUE = .TRUE.
      FALSE = .FALSE.
       . . .
       . . .
      CALL INPUT (. . ., 'OK', OK, 'TRUE', TRUE,
                         'FALSE', FALSE, . . .)
```

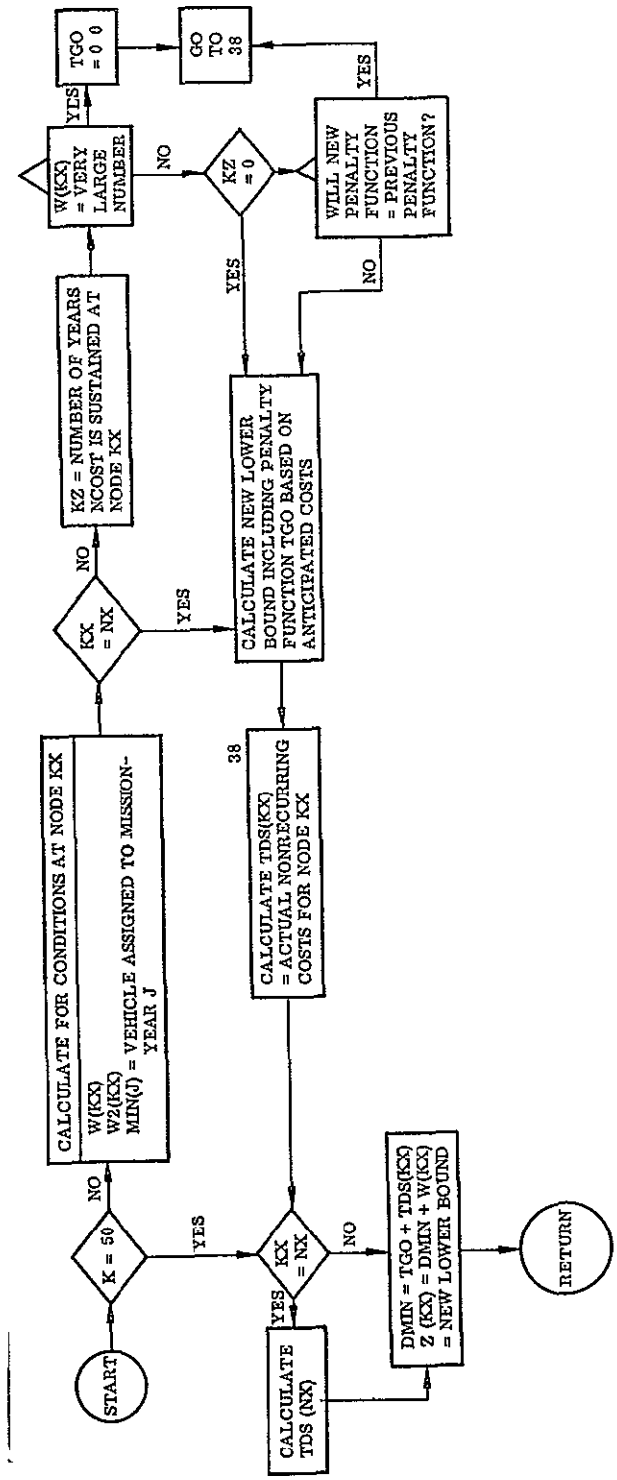
Then a data card written as follows,

OK = TRUE,

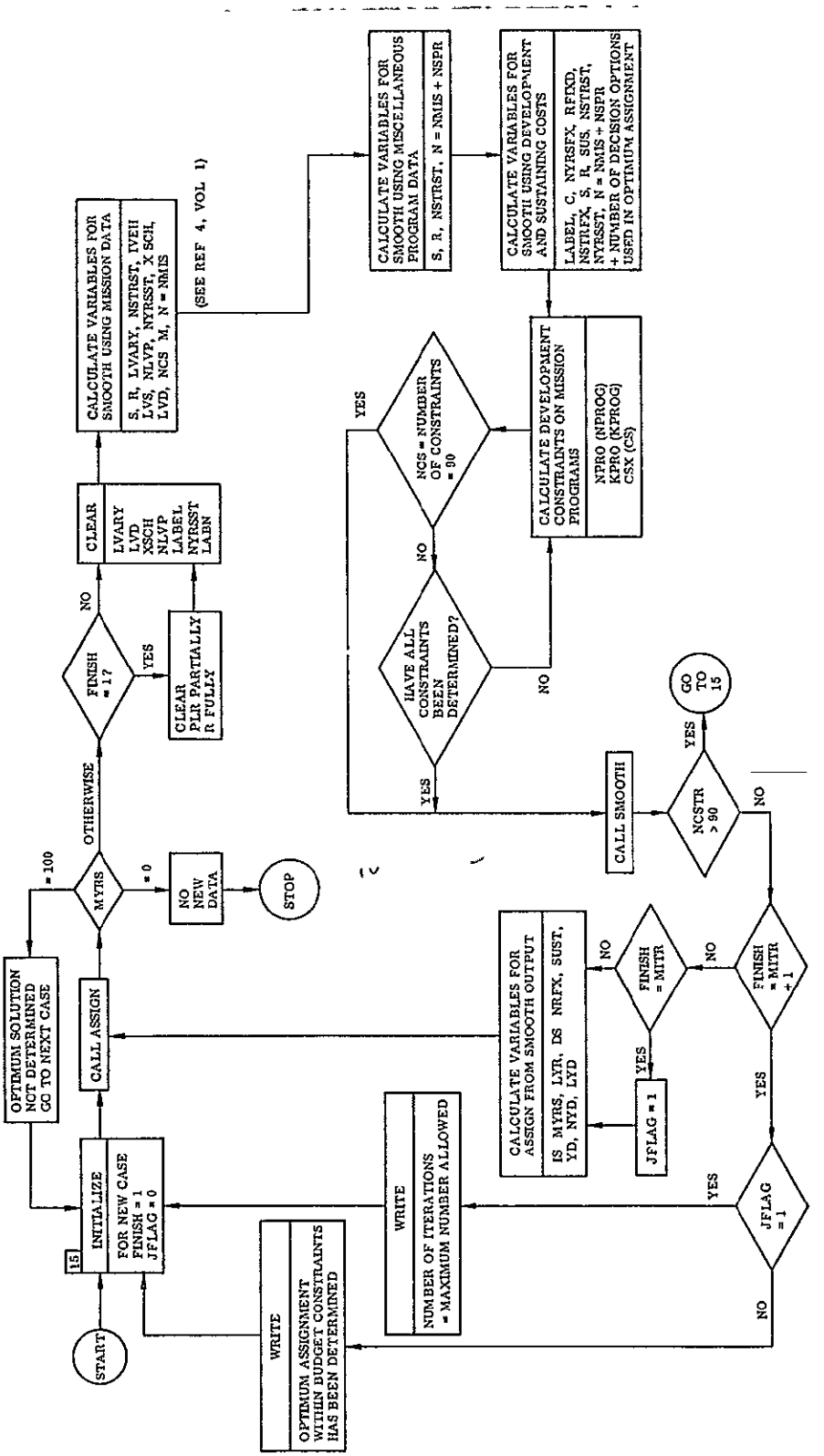
will result in the input of logical data to the program.

6. If a comma is omitted from a data card, a warning will be written on the system output unit and execution will continue. However, for any other type of error, execution will be suppressed, and the remaining data cards will be scanned for errors.
7. This subroutine will accept data cards punched on either a 026 for 029 keypunch.

SUBROUTINE LBOUND



SUBROUTINE MASTER



SUBROUTINE PACK

IDENTIFICATION

Subroutine PACK
Deck Name MOX01PK
Fortran IV subroutine coded in 360 Assembler Language (also MAP coded for the 7094)
Written by R. E. Slye

PURPOSE

This subroutine is used to pack an array of integer or logical data into a smaller array in a packed binary format.

METHOD

The unpacked (source) data is treated as an array of unsigned integers. The integer words are truncated on the left and only the N low order bits are retained. The N low order bits are then placed sequentially, left adjusted, in a packed array word until that word is filled. Packing then continues into the next word, etc., until the source data is exhausted.

Since a storage word contains 32 bits, a packed word may contain 32/N data items. Note that since only the N low order bits are retained, the largest integer item that will be represented correctly is $2^N - N$. For example, if N = 4, the packed items will represent digits from 0 to 15. For a larger integer, the packed item will in effect be the modulus of the source item.

USAGE

This subroutine has three entry points. The three entries are PACK, UNPACK, and ITEM. To pack data, the Fortran call statement is

CALL PACK (L, M, I, N)

where

L is the name of the array containing the source data.
M is the name of the array containing the packed data.
I is the number of data items in L.
N is the number of low order bits to be retained.

The array L should be dimentioned I.
The array M should be dimentioned [(I-1)/[32/N]] + 1
To unpack data, the Fortran call statement is

CALL UNPACK (L, M, I, N)

where the arguments are as listed above.

I may be less than the actual number of items in the packed array.

Packed data in the array M is unpacked and placed right adjusted in the array L. (The unused high order part of the word is cleared.)

The third entry point to the routine may be used to recover a single item from the packed array M. It is called by the Fortran statement

J = ITEM (M, I, N)

The Ith item in the packed array M is returned to the calling program.

ADDITIONAL INFORMATION

If [32/N] is not an even integer, some low order bits in a packed word are unused. For example, if N = 6 the word may contain 5 items and the last 2 bits are unused. The 6th item will then start at the beginning of the 2nd word.

This subroutine is also available for use on the 7094. Since the 7094 has 36 bits/word rather than 32, the data will be packed differently. This should not concern the user except that the size of the M array may be slightly smaller.

PRECAUTION

Integers in the source data may be negative. However, if negative integers are used, the results will be different on the 360 from that on the 7094 since negative 360 integers are carried in complement form.

SUBROUTINE PLOT

IDENTIFICATION

UMPLOT, Drawing of Graphs by Use of the Printer
360/Assembler Language
Ames Modification of SHARE Library Routine UM PLOT

PURPOSE

This subroutine is used for the purpose of drawing plots, along with the printing of the usual type of numerical output, by use of the printer.

PRELIMINARY REMARKS

Several changes have been incorporated in the FORTRAN IV version of UM PLOT. The maximum width of the plot has been increased from 101 columns to 119 columns. The original program included entries for use in SAP and MAD coded routines, whereas the present version may be entered only from FORTRAN IV or MAP coded programs.

METHOD

A region of core is treated here much as a piece of graph paper. This region of core is called the "image region." The image region is cleared, and then a grid, consisting of I's and -'s, with +'s at grid intersection points, is formed. The program will place any given BCD character at the appropriate place in the image region, corresponding to an ordinate – abscissa pair. Each point is written in the image region independently of those previously written, and so data to be plotted need not be sorted. Any number of points (consistent with the specified size of the image) may be plotted, with any Hollerith plotting character whatever. Points which fall on previously plotted points replace the latter, and points which fall on a grid line replace the grid line character.

Points which lie outside of the specified grid limits are not plotted. When all desired points have been placed in the image region, the latter is written out onto a standard BCD tape (i.e., tape 6, 7, 9, or 11) for subsequent printing.

USAGE

This subroutine has four main entries and two auxiliary entries. The four main entires are PLOT 1, PLOT 2, PLOT 3, and PLOT 4. Each performs a specific function, and normally they are taken in the order listed above. Exceptions to the normal sequence are discussed below. The two auxiliary entries are OMIT and PLTAPE. The first of these is used for the purpose of causing portions of the grid to be deleted, and the second is used if it is desired to output on a tape other than logical tape 6.

Each of the entries is discussed below in detail, following which the calling sequence arguments are defined. It may be noted that the four main entries can be taken by use of either a standard CALL statement [e.g., CALL PLOT 1( )] or an arithmetic statement [e.g., R = PLOT 1( )]. The advantage of the latter is that if certain error conditions arise, they can be detected by interrogation of R, whereas the programmer has no way to detect an error condition if the CALL type entry is used. The details concerning error conditions and the interrogation of R will be found in Section D to follow.

A. The Four Main Entries

CALL PLOT 1 (NSCALE, NHL, NSBH, NBL, NSBV)

or

R = PLOT 1 (NSCALE, NHL, NSBH, NBL, NSBV)

This entry is used to set up grid spacing and the total length and width of the graph. The location of decimal points, and the scale factors (powers of 10) for values of the ordinate and abscissa to be printed along the axes of the plot are also specified. If both standard grid and standard scale factors are desired (to be described subsequently), then this entry need not be taken. If several plots are to be printed, all having the same scale factors and grid specifications, then this entry need only be taken one time.

CALL PLOT 2 (IMAGE, XMAX, XMIN, YMAX, YMIN, IDIM)

or

R = PLOT 2 (IMAGE, XMAX, XMIN, YMAX, YMIN, IDIM)

This entry clears the image region and prepares the grid lines of I's and -'s, with +'s at grid line intersection points. It establishes internally formula for computing the location in the image region that corresponds to a given abscissa – ordinate (X_I, Y_I) pair, based on maximum and minimum values as entered through the calling sequence.

CALL PLOT 3 (BCD, X, Y, NDATA)

or

R = PLOT 3(BCD, X, Y, NDATA)

This entry causes a specified Hollerith plotting character to be placed in the appropriate place in the image region for each of the abscissa – ordinate pairs, which are stored in arrays X and Y. This entry may not be taken unless entry PLOT 2 has been taken previously. This entry may be taken repeatedly, if desired, in order to write several sets of data in the image region before it is read out on tape.

CALL PLOT 4 (NCHAR, LABEL)

R = PLOT 4(NCHAR, -LABEL)

This entry causes the contents of the image region to be written out on logical tape 6 (unless a different tape has been specified by use of the entry PLTAPE, discussed later). The topmost line of the graph will appear one space below the last line previously printed. The ordinate label is specified, and it will appear to the left of the graph. Abscissa labels may be printed above or below the graph by use of standard printout statments. The entry PLOT 4 can be taken repeatedly to obtain several copies of the same graph, if desired. The entry PLOT 2 must have been taken at least once prior to the entry PLOT 4. It is permissible to alter a graph (in the image region) by use of the entry PLOT 3 and then print the result using PLOT 4, without returning to the entry PLOT 2.

B. The Arguments For The Four Main Entries Are Described Here

Note that certain of them may be either integers or floating point quantities, as for example NHL (integer) or HL (floating equivalent of NHL).

NSCALE is an array of dimension 5 that supplies the subroutine with grid and scale factor information

NSCALE(1) = 0, standard grid and scale factors (see note (a), to follow)
≠ 0, grid and scale factors are as defined in NSCALE (2) - NSCALE (5)

NSCALE(2) = I, scale factor such that printed values of the ordinate are 10^I times the actual values

NSCALE(3) = J, J digits will appear to the right of the decimal point in printed ordinate values (J < 8)

NSCALE(4) = K, scale factor such that printed values of the abscissa are 10^K times the actual values

NSCALE(5) = M, M digits will appear to the right of the decimal point in printed abscissa values ($M < 8$)

NHL (or HL) is the number of horizontal grid lines (NHL > 0)

NSBH (or BH) is the number of spaces between horizontal grid lines (NSBH > 0)

NVL (or VL) is the number of vertical grid lines (NVL > 0)

NSBV (or SBV) is the number of spaces between vertical grid lines (NSBV > 0, and NSVB*NVL ≤ 119)

Note (a). Standard scale factors correspond to values of I, J, K, and M of 0, 3, 0, 3, respectively. A standard grid is available which is 101 columns wide starting at column 13, and 51 lines long. It has 10 vertical grid lines and 5 horizontal grid lines, with 10 spaces between both horizontal and vertical grid lines. If both the standard scale factors and standard grid are desired, then the PLOT 1 entry need not be taken. It should be noted, however, that if PLOT 1 has been entered for the purpose of setting up nonstandard conditions, then the latter prevail until PLOT 1 is reentered with different arguments.

Any combination of vertical and horizontal grid lines may be specified, but the vertical grid always starts at column 13. It may extend as far to the right as column 132. The length of the grid is limited only by the dimensions of the image region in core.

Note (b). Integers are printed for the ordinate and/or abscissa scales if $J \leq -1$ and/or $M \leq -1$.

Note (c). If a scale factor is such that overflow or underflow would occur, then the scale factor is treated as zero. The subroutine may shift abscissa scale printout in order to accommodate all of the desired numbers. If the value of an ordinate or abscissa is too large to be printed in the allowed space to the left of the graph it will be truncated from the left.

IMAGE (or AIMAGE) is an array, dimensioned IDIM, which is used as the image region by the subroutine

XMAX is the value of the abscissa at the rightmost grid line

XMIN is the value of the abscissa at the leftmost grid line (XMIN < XMAX)

YMAX is the value of the ordinate at the uppermost grid line

YMIN is the value of the ordinate at the lowermost grid line (YMIN < YMAX)

IDIM is the dimension of the array IMAGE, where IDIM = N*(NSBH*NHL+1) and

$$N = \frac{K}{6} \text{ rounded up for the IBM 7094, or}$$

$$N = \frac{K}{4} \text{ rounded up for the IBM 360}$$

and where

$$K = NSBV*NVL+1$$

(The square brackets in the formula for N signify "integral value.")

Note (d). Set IDIM equal to at least 867 for the standard grid. (1326 for 360).

BCD is the Hollerith plotting character, any character whatever (see note (e), to follow)

X is the array (or single location) that contains the abscissa of the points to be plotted

Y is the array (or single location) that contains the ordinates of the points to be plotted

NDATA (or DATA) is the number of points to be plotted (NDATA > 0)

Note (e). The plotting character may be loaded into cell BCD by use of a DATA statement, that is,

```
DATA BCD/1H*/.
```

or, alternatively, it may be entered as a Hollerith literal in the PLOT 3 entry statement, for example,

```
CALL PLOT 3 (1H*, X, Y, NDATA)
```

(The arithmetic statement entry R = PLOT 3 ( ) may not be used in the latter case.)

Note (f). If it is desired to write a single point at a time into the image array, set NDATA equal to 1.

N CHAR (or CHAR) is the number of Hollerith characters, including blanks, in the ordinate label (N CHAR ≤ NHL*NSBH+1)

LABEL is an array which contains the Hollerith characters that constitute the ordinate lable to be printed along the leftmost grid line. (See note (g), below)

Note (g). The ordinate label can be entered in array LABEL by use of the DATA statement, that is,

```
DATA (LABEL (J), J = 1, 3)/17HbbbORDINATEbLABEL/
```

Alternatively, it can be loaded as a Hollerith literal in the PLOT 4 entry statement, for example,

```
CALL PLOT 4 (17, 17HbbbORDINATEbLABEL)
```

(The arithmetic statement entry, R = PLOT 4 ( ), may not be used in the latter case.)

R (See Section D, to follow)

C. The Two Auxiliary Entries and Their Arguments

CALL PLTAPE (NTAPE)

This entry is used, prior to PLOT 4, if it is desired that the output be on a tape other than tape 6. Here, NTAPE is the tape number upon which the output is to take place (7, 9, or 11). The output tape number remains as set by this entry until PLTAPE is called again with a different value for NTAPE.

CALL OMIT (NARG)

This entry causes certain portions of the graph to be deleted. It is taken prior to the entry PLOT 4. The settings for NARG are tabulated below

NARG	Effect
1	Numerical values of the abscissa are not printed
2	Numerical values of the ordinate are not printed
3	Combines the effect of NARG = 1 and NARG = 2
4	The complete bottom horizontal grid line is deleted
5	Combines the effect of NARG = 1 and NARG = 4
6	Combines the effect of NARG = 2 and NARG = 4
7	Combines the effect of NARG = 1 NARG = 2, and NARG = 4

D. Error Conditions

If arguments are incompatible with certain restrictions, then the message

IMPROPER ARGUMENT { PLOT 1, or / PLOT 2, / etc.

is printed, thus indicating the entry where the improper entry appears. If such errors occur in PLOT 1 or PLOT 2, subsequent entries into PLOT 3 and PLOT 4 are deleted with no further comment. The argument restrictions are

$$NHL > 0$$
$$NSBH > 0$$
$$NVL > 0$$
$$NSBV > 0$$
$$NSBV * NVL \leq 119$$
$$XMAX > XMIN$$
$$YMAX > YMIN$$

BCD must be a single left-adjusted Hollerith character

If the user attempts to execute PLOT 3 or PLOT 4 without having previously executed PLOT 2, (or without execution of PLOT 2 subsequent to the execution of PLOT 1), the comment

NO PREVIOUS PLOT 2

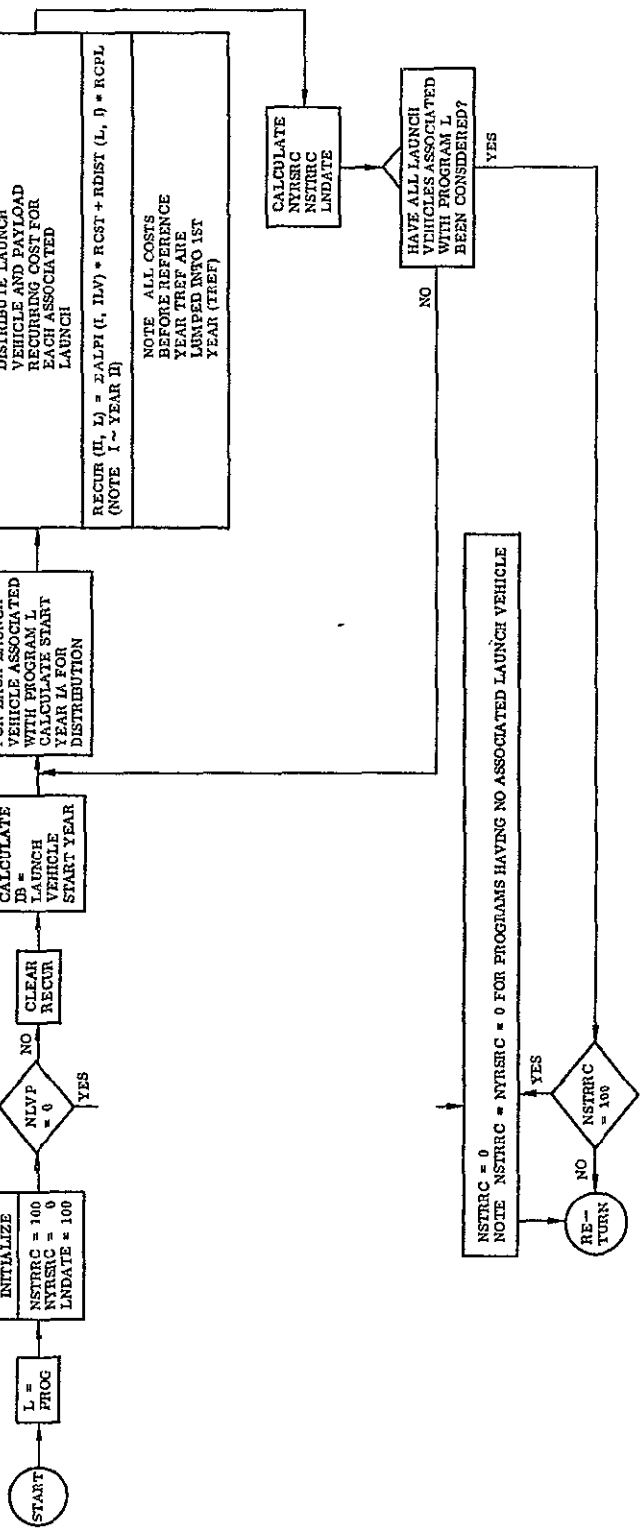
will be printed.

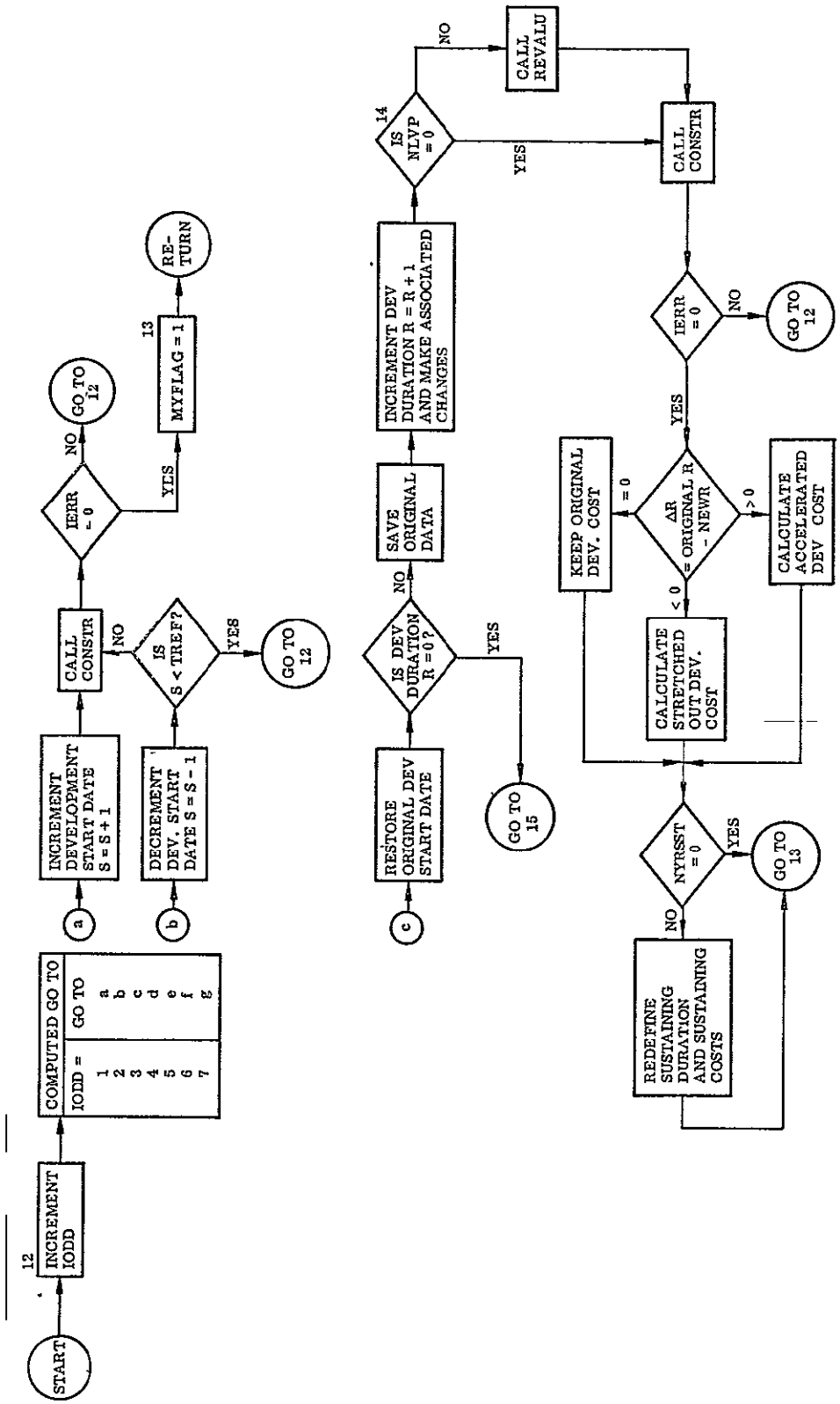
If the arithmetic statement (rather than the CALL statement) is used for the four main entries, then the user may take appropriate action in the case of such errors as would lead to the printouts described above. An error in the arguments, or one due to the

unsuccessful completion of an earlier entry, will cause a + 1.0, + 2.0, + 3.0, or + 4.0 to be loaded in cell R for entries PLOT 1, PLOT 2, PLOT 3, or PLOT 4, respectively. Cell R contains + 0.0 if no error condition arises. The user simply tests R following each attempt-to-enter the subroutine via PLOT 1, PLOT 2, PLOT 3, or PLOT 4.

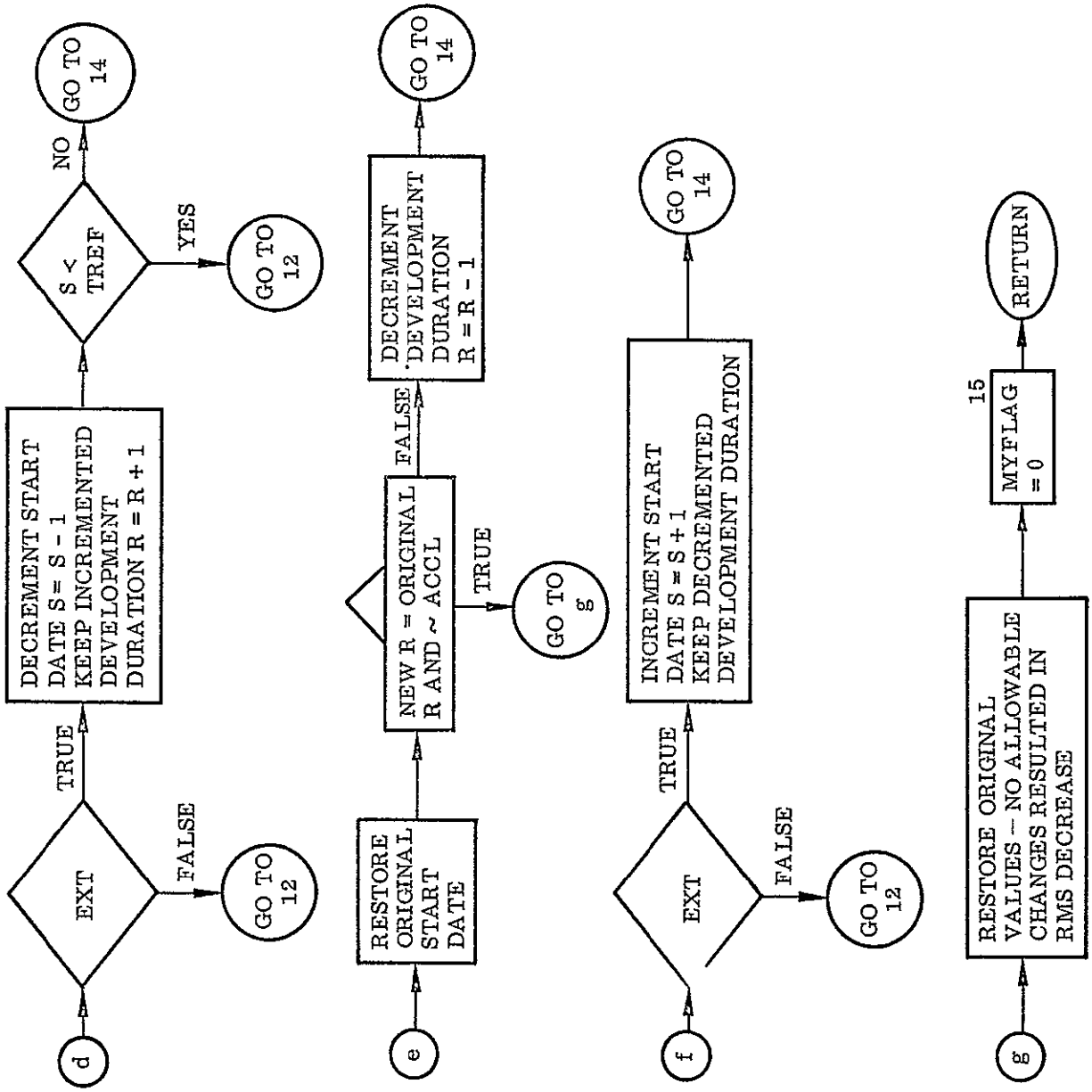
If any points are not plotted by PLOT 3, then the number - 3.0 will be found in R . This might arise if points lie outside the stated minimum and maximum limits of the ordinate and abscissa, and need not be considered an error.

SUBROUTINE REVALU

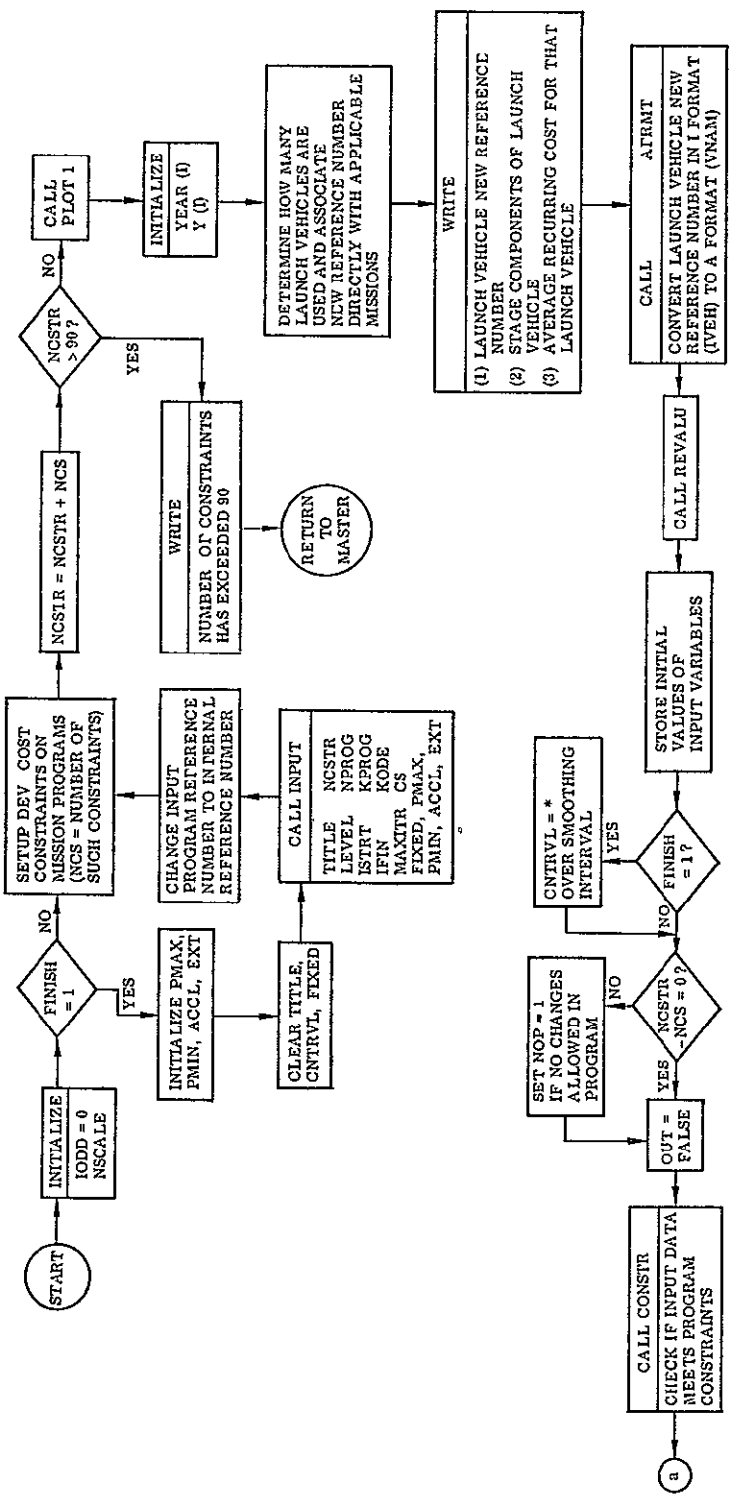



START
12
INCREMENT IODD
COMPUTED GO TO
IODD = 1 2 3 4 5 6 7
GO TO a b c d e f g
a
b
c
INCREMENT DEVELOPMENT START DATE S = S + 1
DECREMENT DEV. START DATE S = S - 1
CALL CONSTR
IS S < TREF?
YES
NO
GO TO 12
IERR = 0
YES
NO
GO TO 12
13
MYFLAG = 1
RE-TURN
IS DEV DURATION R = 0 ?
YES
NO
RESTORE ORIGINAL DEV START DATE
GO TO 15
SAVE ORIGINAL DATA
INCREMENT DEV DURATION R = R + 1 AND MAKE ASSOCIATED CHANGES
14
IS NLVP = 0
NO
YES
CALL REVALU
CALL CONSTR
IERR = 0
NO
GO TO 12
YES
ΔR = ORIGINAL R - NEWR
= 0
< 0
> 0
KEEP ORIGINAL DEV. COST
CALCULATE STRETCHED OUT DEV. COST
CALCULATE ACCELERATED DEV COST
NYRSST = 0
YES
NO
GO TO 13
REDEFINE SUSTAINING DURATION AND SUSTAINING COSTS

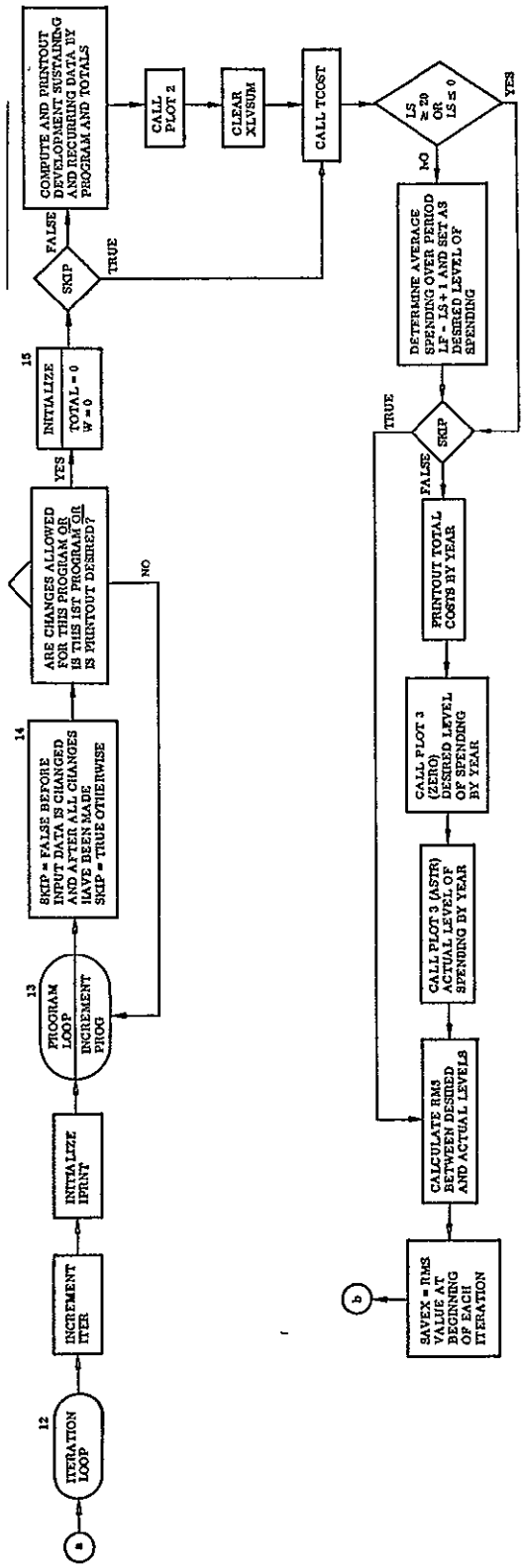
SUBROUTINE SHIFT (Cont.)



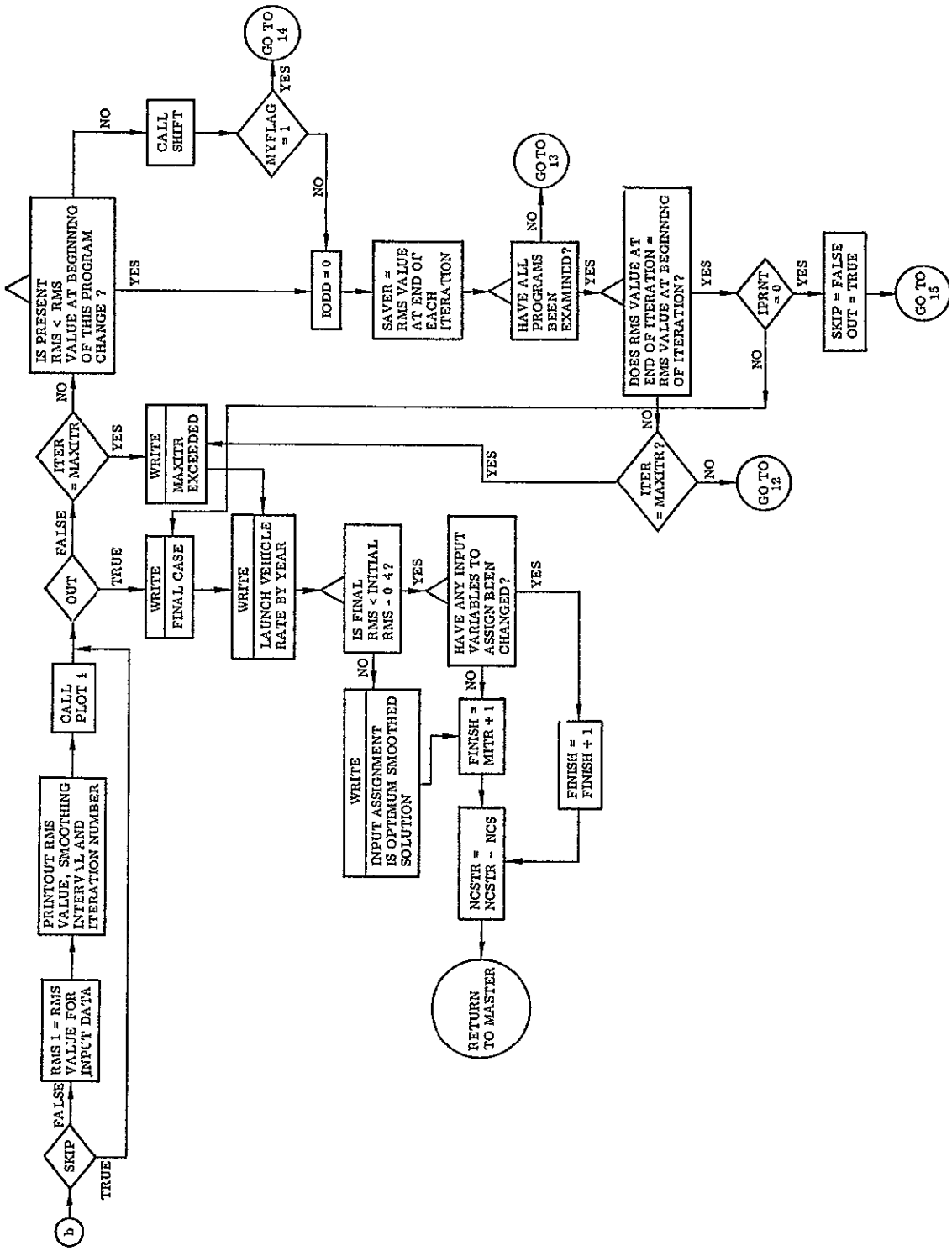
SUBROUTINE SMOOTH



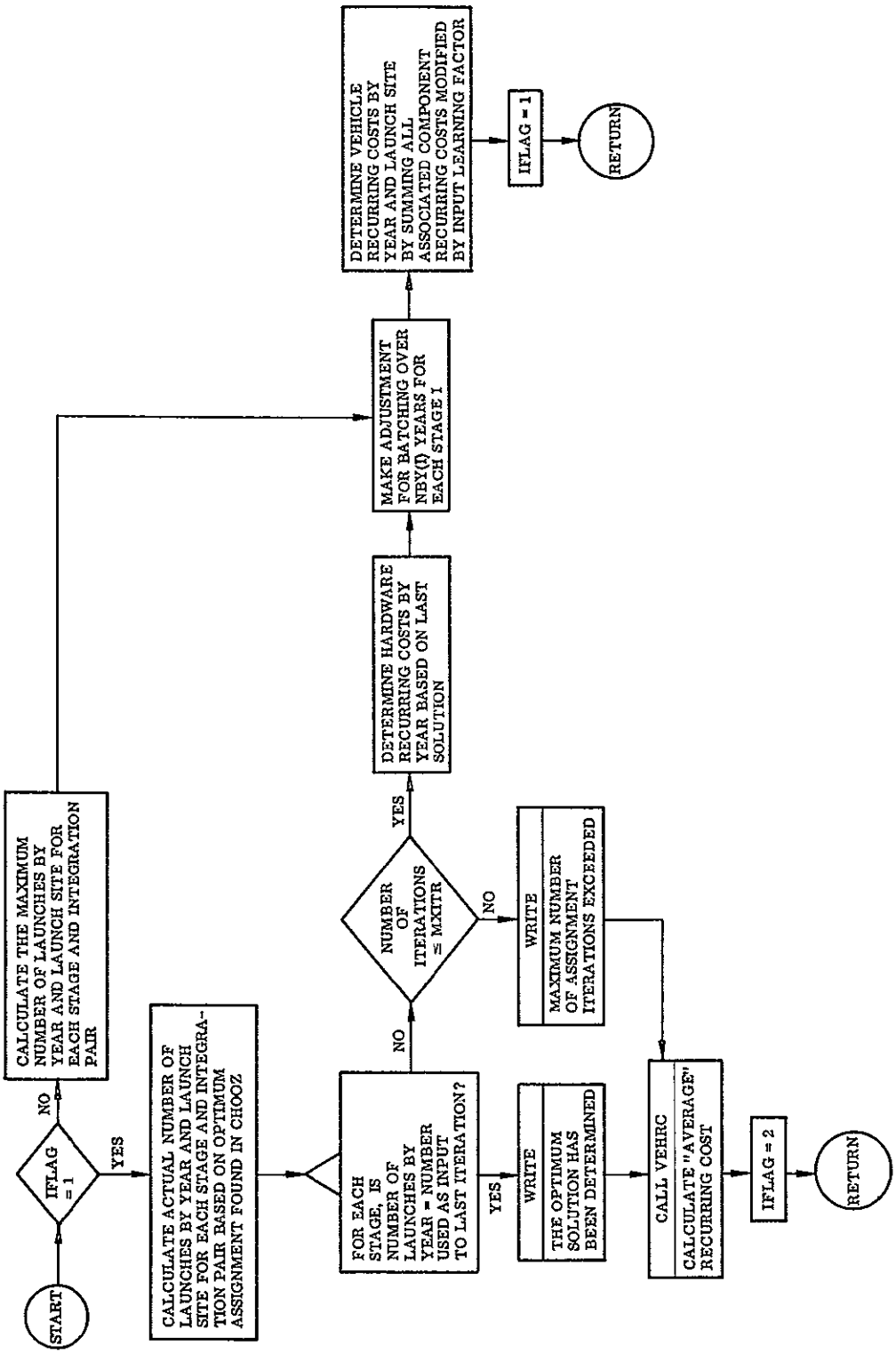
SUBROUTINE SMOOTH (Cont.)



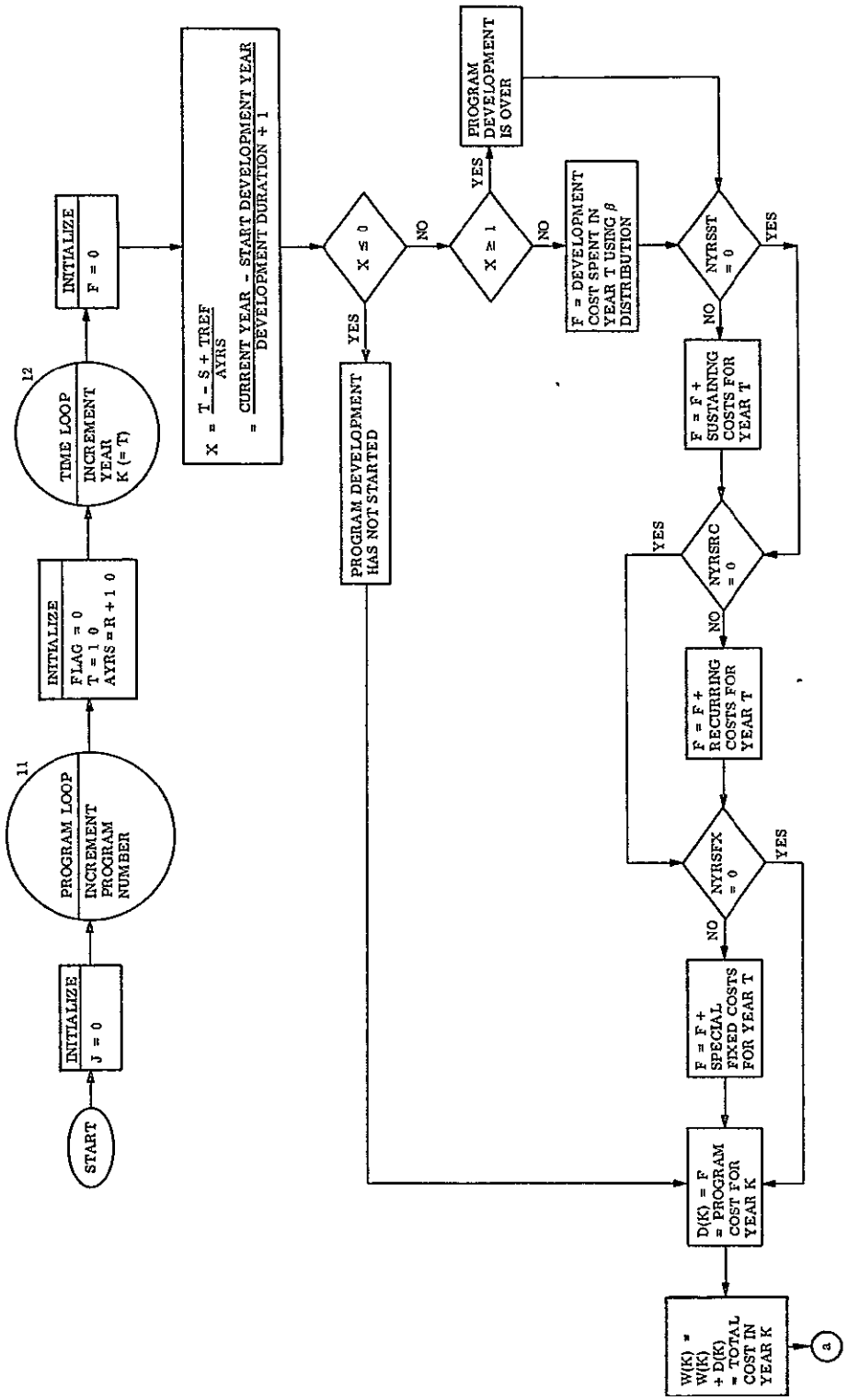
SUBROUTINE SMOOTH (Cont.)

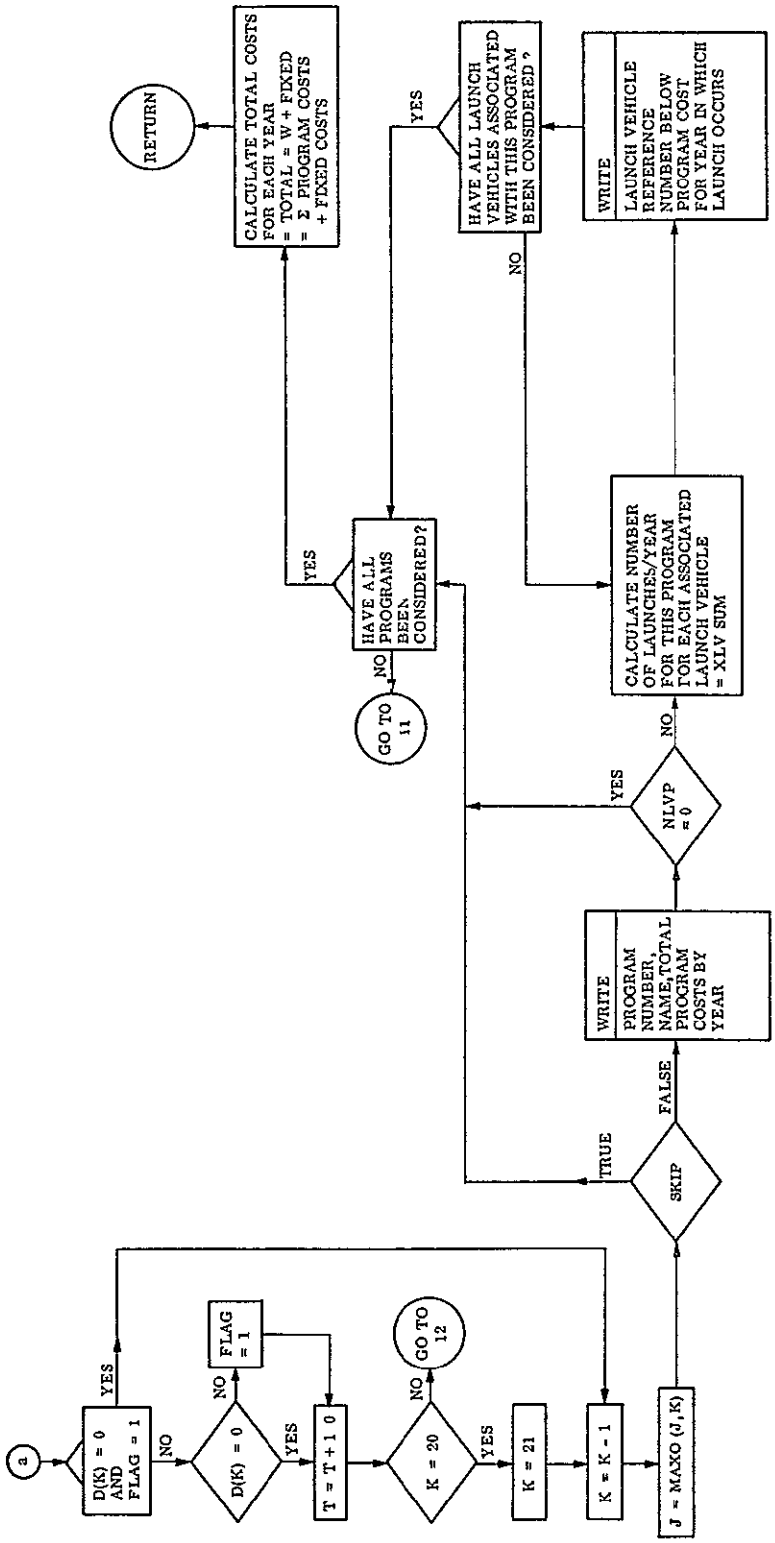


–SUBROUTINE STGNUM

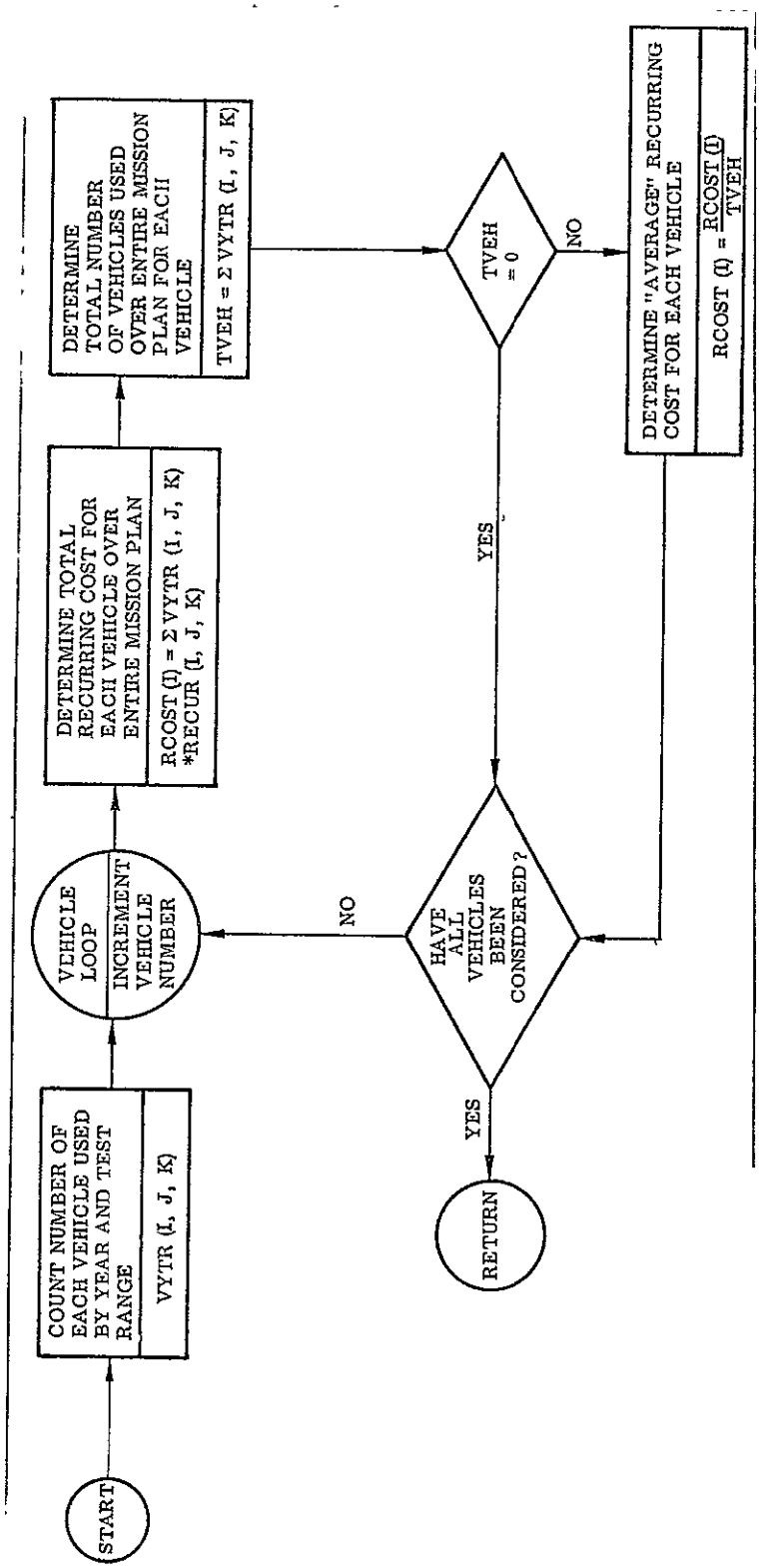


SUBROUTINE TCOST




a
D(K) = 0 AND FLAG = 1
YES
NO
D(K) = 0
NO
FLAG = 1
YES
T = T + 1 0
K = 20
NO
GO TO 12
YES
K = 21
K = K - 1
J = MAXO (J, K)
SKIP
TRUE
FALSE
WRITE
PROGRAM NUMBER, NAME, TOTAL PROGRAM COSTS BY YEAR
NLVP = 0
YES
NO
CALCULATE NUMBER OF LAUNCHES/YEAR FOR THIS PROGRAM FOR EACH ASSOCIATED LAUNCH VEHICLE = XLV SUM
WRITE
LAUNCH VEHICLE REFERENCE NUMBER BELOW PROGRAM COST FOR YEAR IN WHICH LAUNCH OCCURS
HAVE ALL LAUNCH VEHICLES ASSOCIATED WITH THIS PROGRAM BEEN CONSIDERED ?
YES
NO
HAVE ALL PROGRAMS BEEN CONSIDERED?
YES
NO
GO TO 11
CALCULATE TOTAL COSTS FOR EACH YEAR = TOTAL = W + FIXED = Σ PROGRAM COSTS + FIXED COSTS
RETURN

SUBROUTINE VEHRC



Appendix H
PROGRAM LISTING

H.1 DESCRIPTION

A compile-and-save Fortran listing of each major subroutine in the Budget Smoothing and Vehicle Assignment Integrated program is included in this section. Storage requirements for each subroutine are listed on the output along with the code name under which the subroutine was saved. Total storage requirements are listed at the beginning of the sample case presented in Appendix F. Comment cards describing the logical function of each subsection and defining any variables whose names are not mnemonic are liberally distributed throughout the deck so that new users may readily become familiar with the programs.

Subroutines CLEAR, INPUT and PLOT are stored for general NASA use and are described in Ref. 4. Therefore, no listing is included here; however, a description of each is provided in Appendix G for completeness. Subroutines AFRMT and PACK are written in 360 assembler language, so their listings are provided in that language.

Labeled common blocks were used for storage whenever possible to avoid long argument lists for each subroutine. These blocks are found at the beginning of each listing with a brief explanation concerning the nature of the variables found in each block. The block labeled SCRACH stores variables only required in that subroutine so that the same storage locations may be used for storing new variables in the next subroutine. All other labeled common blocks contain variables used in several subroutines.

The listings are presented in alphabetical order according to subroutine name for easy reference.

H.2 COMPILE-AND-SAVE LISTINGS

The compile-and-save listings follow.

```
SYMBOL   TYPE ID  ADDR  LENGTH LD ID                                        16.07  4/14/70

AFRMT     SD  01 000000 000040
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                          F01FE869   4/14/70
000000                                1 AFRMT   CSECT
000000                                2         USING *,15          REG 15 FOR BASE
000000 5020 001C            0001C     3         ST    2,28(0,13)    SAVE REG 2
000004 9812 1000            00000     4         LM    1,2,0(1)      LOAD ADDRESSES OF ARGS TO REGS 1-2
000008 5810 1000            00000     5         L     1,0(0,1)      DATA TO REG 1
00000C 4E10 F038            00038     6         CVD   1,WORK        CONVERT TO DECIMAL
000010 F332 2000 F03D 00000 0003D     7         UNPK  0(4,2),WORK+5(3)   UNPACK 4 DIGITS
000016 96F0 2003      00003           8         OI    3(2),X'F0'    INSERT ZONES
00001A 4110 0004            00004     9         LA    1,4
00001E 95F0 2000      00000          10 LOOP    CLI   0(2),C'0'     SCAN OUT LEADING ZEROS
000022 4770 F032            00032    11         BNE   RETURN
000026 9240 2000      00000          12         MVI   0(2),C' '     INSERT BLANK
00002A 4120 2001            00001    13         LA    2,1(0,2)      BUMP POINTER
00002E 4610 F01E            0001E    14         BCT   1,LOOP        LIMIT TO 4 CHARACTERS
000032 5820 001C            0001C    15 RETURN  L     2,28(0,13)    RESTORE REG 2
000036 07FE                          16         BR    14            RETURN
000038                               17 WORK    DS    D
                                     18         END
```

```
                                CROSS-REFERENCE

SYMBOL    LEN  VALUE  DEFN     REFERENCES                                  4/14/70

AFRNT     00001 000000 0001
LOOP      00004 00001E 0010     0014
RETURN    00004 000032 0015     0011
WORK      00008 000038 0017     0006  0007


NO STATEMENTS FLAGGED IN THIS ASSEMBLY
   32 PRINTED LINES
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,24576)          DEFAULT OPTION(S) USED
IEW0000      NAME MOXO2AT(R)
****MOXO2AT   NOW REPLACED IN DATA SET
```

```
FORTRAN IV G LEVEL 1, MOD 4            ASSIGN          DATE = 70105        09/14/08

0001              SUBROUTINE ASSIGN
                  C
                  C     THIS PROGRAM GENERATES THE LEAST COST ASSIGNMENT OF LAUNCH
                  C     VEHICLES TO SPACE MISSIONS.  A BRANCH AND BOUND TECHNIQUE IS USED
                  C     TO REDUCE THE COMBINATORIAL COMPLEXITY OF THE PROBLEM. SEVERAL BRANCHES
                  C     ARE CREATED AT EACH NODE.  ONE OF THE BRANCHES EXCLUDES THE NEXT
                  C     COST AND THE OTHERS    ASSUME EXPENDITURE OF A NON-RECURRING COST
                  C     WITH I-2 YEARS OF SUSTAINING COST ADDED AT EACH NODE.
                  C     PENALTY FUNCTIONS ARE USED TO SHARPEN THE LOWER BOUND.
                  C     MODIFIED PENALTY FUNCTION IN LOWER BOUND BASED ON VEHICLE INFORMATION
                  C ****THIS VERSION USES RATE EFFECTS IN RECURRING COSTS****
                  C
0002                  DOUBLE PRECISION NAME
0003                  INTEGER YDPL,FINISH
0004                  INTEGER*2 LYR,LETT,MIN,IS,NONREC,NYD,LABS,LABF,LABI,MAT,VEH,LYD,
                     1  NYRSST
                  C
                  C  THE FOLLOWING STORAGE IS USED IN ASSIGN AND MASTER
0005                  COMMON/SAVEA/NN,YRLM(250),LYR(252),LETT(250),MIN(250),
                     1  DS(50),SUST(50),YD(50),IS(106),     NUMO,NONREC( 60,20),NYD(50),
                     2  LABS(40),LABF(30),LABI(40),                    RXD(12,50),MAT(50)
0006                  COMMON/SAVEA1/YDPL(56),IBY,NSFX(50),NRFX(50),NSYR(50)
                  C  THE FOLLOWING STORAGE IS USED IN MASTER AND SMOOTH AND ASSIGN
0007                  COMMON/SAVEB/NAME(56), MITR,  ALPI(4, 60),PLD(72),PLS(72),
                     1  PLR(50),RDIST(56,4),RFIXD(12,72),NSTRFX(72),NYRSFX(72),TREF
0008                  COMMON/SAVEB1/NMIS,VEH(4,60),FINISH,RCOST(61),STG(40),NSPR,
                     1   NYRSST(72)
                  C   THE FOLLOWING STORAGE IS USED IN ASSIGN AND CHOOZ
0009                  COMMON/SAVEC2/RECUR( 60,20,2),VNM(4,250),LYD(50),NEXT,GUESS,
                     1  LTR(50)
                  C   STORAGE USED IN DECISN AND ASSIGN
0010                  COMMON/ASGN/IST(40),JST(30),KST(40),KODEM(50),YDS(40),YDF(30),
                     1   YDI(40)
0011                  COMMON/SAVEDC/KFLAG,NFAM,LSA(40),SNR(40),STS(40),NYS(40),FMNR(30),
                     1  KODEF(30),FMSUS(30),FAM(30),DINT(40),SINT(40)
                  C  THE FOLLOWING STORAGE IS USED IN STGNUM
0012                  COMMON/SAVESM/IFLAG,STGYTR(40,20,2),RINTYR(40,20),NBY(40),NCI,
                     1       PLCINT(40),                NFML(40),NFS(40,4),NFMU(40),
                     2   MODE(40,3),SR(40,3),PLC(40,3),POJ(3),SRJ(3,3),RINT(40)
                  C  THE FOLLOWING STORAGE IS USED IN CAPABL AND ASSIGN
0013                  COMMON/SAVECL/NOPT,IV,KNV,IG,     PLMD(50),KODEV(60),KODS(40),
                     2  IVEHA(50),VLR(50),WPR(50),VM(2,60),NPLS(50),MR(50),NRR(50)
                  C  STORAGE USED IN CAPABL ONLY BUT SAVED FOR BATCHING
```

```
FORTRAN IV G LEVEL 1, MOD 4            ASSIGN          DATE = 70105        09/14/08

0014                  COMMON/SAVED/B1(60),B2(60),B3(60),B4(60),NVS(60),MRV(60),NRP(60)
0015                  COMMON/SAVELZ/LZ(60)
0016                  COMMON/SAVENV/NV,MYRS
0017                  COMMON/BATCH/KNSTG,KNFAM,KNCI,KNMIS,KNSP,KODESP(6)
0018                  COMMON/SAVESG/NSTG
0019                  COMMON/SAVENR/NUMBR,MXITR
                  C  THE FOLLOWING STORAGE NEED NOT BE SAVED
0020                  COMMON/SCRACH/MISN(50,20),KVEH(50),PB(50),IIS(56),DUMA(3429)
                  C
0021                  NUMBR = 0
0022                  GUESS = 1.75*GUESS
0023                  IF(FINISH.GT.1)  GO TO 17
0024               11 NSTG = 0
0025                  NFAM = 0
0026                  NCI = 0
0027                  NMIS = 0
0028                  NSPR = 0
0029                  READ(5,100)NOPT,MYRS,IBY,GUESS,MITR,TREF,MXITR,IG,IFM,II,IM,ISD,IV
                  C     ***IG IFM II IM ISD AND IV ARE VARIABLES FOR BATCHING ONLY ***
0030                  IF (MYRS.EQ.0) GO TO 806
0031                  WRITE (6,104)
0032                  IF(IG.LT.0)  GO TO 12
0033                  WRITE(6,213)
0034                  LX = 0
                  C  NSDC = NUMBER OF SPECIAL DEVELOPMENT COSTS
0035                  NSDC = 1
0036                  DO 8000 I = 1,50
0037                  READ(5,101)  KODS(I),STG(I),(SR(I,J),J=1,3),(PLC(I,J),J=1,3),
                     1          SNR(I),STS(I),NYS(I),LSA(I),NBY(I),(NFS(I,J),J=1,4),
                     2  (MODE(I,J), J = 1,3)
0038                  IF(KODS(I).EQ.0)  GO TO 12
0039                  READ(5,111)  YDS(I), IST(I), NSFX(NSDC)
0040                  NSTG = NSTG + 1
0041                  LABS(I) = 0
0042                  WRITE (6,8001) STG(I),(SR(I,J),PLC(I,J),J=1,3),SNR(I),STS(I),
                     1  NYS(I),LSA(I),(NFS(I,J),J=1,4),NBY(I)
0043                  DO 8002 J = 1,3
0044                  IF (MODE(I,J).EQ.0) GO TO 8002
0045                  LX = LX + 1
0046                  MODE(I,J) = LX
0047                  READ(5,8003) (SRJ(LX,K), K = 1,3), POJ(LX)
0048                  WRITE(6,8004) J,POJ(LX),SRJ(LX,1),POJ(LX),(SRJ(LX,K),K=2,3)
0049               8002 CONTINUE
```

```
FORTRAN IV G LEVEL 1, MOD 4          ASSIGN          DATE = 70105          09/14/08

0050              IF(NSFX(NSDC).EQ.0)  GO TO 8000
0051              READ(5,110) NRFX(NSDC), (RXD(J,NSDC), J = 1,12)
0052              LAB$(I) = NSDC
0053              NSDC = NSDC + 1
0054         8000 CONTINUE
0055           12 IF(IFM.LT.0)  GO TO 14
0056              DO 13 J = 1,30
0057              READ(5,102) I,FAM(I),FMNR(I),FMSUS(I),YDF(I),JST(I),NSFX(NSDC)
0058              IF(I.EQ.0)  GO TO 14
0059              IF(J.EQ.1)  WRITE(6,214)
0060              NFAM = NFAM + 1
0061              LABF(I) = 0
0062              KODEF(J) = I
0063              WRITE(6,2141)KODEF(J), FAM(I),FMNR(I),FMSUS(I)
0064              IF(NSFX(NSDC).EQ.0)  GO TO 13
0065              READ(5,110) NRFX(NSDC), (RXD(J1,NSDC),  J1=1,12)
0066              LABF(I) = NSDC
0067              NSDC = NSDC + 1
0068           13 CONTINUE
0069           14 IF(II.LT.0)  GO TO 1716
0070              DO 1715 I = 1,40
0071              READ(5,103)  NFML(I),NFMU(I),RINT(I),PLCINT(I),DINT(I),SINT(I),
                 1  YDI(I),KST(I),NSFX(NSDC)
0072              IF(NFML(I).EQ.0)  GO TO 1716
0073              IF(I.EQ.1)   WRITE(6,215)
0074              NCI = NCI + 1
0075              LABI(I) = 0
0076              J=NFML(I)
0077              K=NFMU(I)
0078              WRITE(6,216) FAM(J),FAM(K),RINT(I),PLCINT(I),DINT(I),SINT(I)
0079              IF(NSFX(NSDC).EQ.0)  GO TO 1715
0080              READ(5,110) NRFX(NSDC), (RXD(J,NSDC), J = 1,12)
0081              LABI(I) = NSDC
0082              NSDC = NSDC + 1
0083         1715 CONTINUE
0084         1716 IF(IM.LT.0)  GO TO 19
0085              DO 1719 I=1,MYRS
0086         1719 LZ(I)=IBY+I-1
0087              WRITE(6,217) (LZ(I),I=1,MYRS)
0088         1717 DO 1718 I=1,50
0089              READ(5,105) KODEM(I),NAME(I),PB(I),NSYR(I),NYRSFX(I),VLR(I),WPR(I)
                 1 ,(MISN(I,J),J=1,MYRS)
0090              IF(KODEM(I).EQ.0)  GO TO 19
```

```
FORTRAN IV G LEVEL 1, MOD 4          ASSIGN          DATE = 70105          09/14/08

0091              NMIS = NMIS + 1
0092              READ (5,107) PLR(I),PLS(I),PLD(I),YDPL(I),(RDIST(I,L),L=1,4),
                 1  PLMD(I),NPLS(I),MR(I),LTR(I),NRR(I),IIS(I),IVEHA(I)
0093              IS(I) = 1900 +IIS(I)
0094              IF(NYRSFX(I).EQ.0)  GO TO 1718
0095              READ(5,110)  NSTRFX(I), (RFIXD(J,I),J=1,12)
0096         1718 WRITE(6,219) I,NAME(I),  VLR(I),WPR(I),PB(I),LTR(I),
                 1  (MISN(I,J),J=1,MYRS)
0097           19 IF(ISD.LT.0)  GO TO 20
               C   INPUT SPECIAL PROGRAMS HAVING NO ASSOCIATED LAUNCHES
               C   KODESP GT 100
0098              DO 1725 I = 1,6
0099              READ(5,106) KODESP(I),NAME(NMIS+1),PLD(I+NMIS),YDPL(I+NMIS),
                 1 IIS(I+NMIS),PLS(I+NMIS),NST,NYRSFX(I+NMIS)
0100              NYRSST(I+NMIS) = NST
0101              IF(KODESP(I).EQ.0)  GO TO 20
0102              NSPR = NSPR + 1
0103              IS(I + NMIS) = 1900 + IIS(I+NMIS)
0104              IF(NYRSFX(I+NMIS).GT.0)
                 1READ(5,110) NSTRFX(I+NMIS),(RFIXD(J,I+NMIS),J=1,12)
0105         1725 CONTINUE
0106           20 IF(IG.LT.0)  NSTG = KNSTG
0107              IF(IFM.LT.0)  NFAM = KNFAM
0108         c    IF(II.LT 0)  NCI = KNCI
0109              IF(IM.LT.0)  NMIS = KNMIS
0110              IF(ISD.LT.0)  NSPR=KNSP
0111              NSDC = NSDC - 1
0112              IF(IM.LT.0)  GO TO 3000
               C
               C     ***SET UP MISSION MATRIX BY YEAR***
0113              NM = 0
0114           3  DO 4 I = 1,NMIS
0115              DO 4 J=1,MYRS
0116              IF(MISN(I,J).EQ.0) GO TO 4
0117              NM = NM + 1
0118              YRLM(NM)= FLOAT(MISN(I,J)) * PB(I)
0119              LETT(NM)= I
0120              LYR(NM) = J
0121            4 CONTINUE
0122         3000 CONTINUE

0123              IF(GUESS.GT.1.0)  GO TO 3005
0124              GUESS = 1.0E10
```

```
FORTRAN IV G LEVEL 1, MOD 4          ASSIGN          DATE = 70105          09/14/08

 0125        3005 CONTINUE
             C
 0126             CALL CAPABL
             C
 0127          16 WRITE(6,2001) NSTG,NV,NFAM,NCI,NMIS,MYRS,IBY,GUESS
             C    CALCULATE EXPONENT FOR LEARNING CURVE
 0128         8020 ALOG2 = ALOG(2.)
 0129              IF (IG.LT.0) GO TO 8030
 0130              DO 660 I=1,NSTG
 0131              DO 660 J=1,3
 0132              IF (MODE(I,J).NE.0) GO TO 660
 0133              PLC(I,J) = ALOG(PLC(I,J))/ALOG2
 0134          660 CONTINUE
 0135         8030 IF(II.LT.0.OR.NCI.EQ.0)  GO TO 17
 0136              DO 680 I=1,NCI
 0137          680 PLCINT(I) = ALOG(PLCINT(I))/ALOG2
             C
 0138           17 IFLAG = 0
 0139              CALL STGNUM
             C
 0140           61 KFLAG = 0
 0141              IF(FINISH.GT 1.AND.NUMD.EQ 0)  GO TO 305
             C
 0142              CALL DECISN
             C
 0143              IF(KFLAG.EQ.1)  GO TO 1
             C
 0144          305 CALL AVAIL
             C
 0145              GUESS1 = GUESS
 0146          620 CALL CHOOZ
 0147              IF(NEXT.GE.500.OR.GUESS.LT..001)  MYRS = 100
 0148              IF(NEXT.GE.500.OR.GUESS.LT..001)  GO TO 1
 0149              NUMBR = NUMBR + 1
             C
             C    ***  PRINT OUT BEST ASSIGNMENT  ***
 0150              WRITE (6,4010)
 0151              DO 805 J=1,NM
 0152              L=LETT(J)
 0153              K=LYR(J)
 0154              M=1899+IBY+K
 0155              IF(YRLM(J).NE.0.0) GO TO 804
 0156              MIN(J) = 0

FORTRAN IV G LEVEL 1, MOD 4          ASSIGN          DATE = 70105          09/14/08

 0157              IF(LETT(J-1).NE.L) WRITE(6,206)NAME(L),  VLR(L),WPR(L),M,YRLM(J)
 0158              IF(LETT(J-1).EQ.L) WRITE(6,2061) M,YRLM(J)
 0159              GO TO 805
 0160          804 I=MIN(J)
 0161              IA=VEH(1,I)
 0162              IB=VEH(2,I)
 0163              IC=VEH(3,I)
 0164              ID=VEH(4,I)
 0165              IF(LETT(J-1).NE.L) WRITE(6,202)NAME(L),  VLR(L),WPR(L),M,YRLM(J)
                  1  ,STG(IA),STG(IB),STG(IC),STG(ID)
 0166               IF(LETT(J-1).EQ.L) WRITE(6,2021) M,YRLM(J),  STG(IA),STG(IB),
                   1  STG(IC),STG(ID)
 0167          805 CONTINUE
             C
 0168              CALL STGNUM
             C
 0169              GUESS = GUESS1
 0170              IF(IFLAG.EQ.2)  GO TO 1
 0171              GO TO 620
 0172            1 KNSTG = NSTG
 0173              KNFAM = NFAM
 0174              KNCI = NCI
 0175              KNMIS = NMIS
 0176              KNV = NV
 0177              KNSP = NSPR
 0178              RETURN
 0179          806 WRITE(6,4102)
 0180           99 RETURN
 0181          100 FORMAT (3I5,F12.2,I5,F5.1,I5,25X,6I2)
 0182          101 FORMAT (I2,1X,A4,3F6.3,3F5 3,3X,  2F6.3,7I3,1X,3I1)
 0183          102 FORMAT(I2,1X,A4,2F10.0,F4.1,2I3)
 0184          103 FORMAT (2X,2I3,4F10.0,F4.1,2I3)
 0185          104 FORMAT (1H1)
 0186          105 FORMAT (I2,A6,F4.2,2X,2I2,F7.0,6X,F7.0,2X,20I2)
 0187          106 FORMAT(I3,   A6,F10.2,I5,I2,F10.2,2I2)
 0188          107 FORMAT (2X,3F10.2,I5,  4F5.3,F10.2,5I2,I3)
 0189          110 FORMAT (I3,12F6.2)
 0190          111 FORMAT(F3.1,2I3)
 0191          202 FORMAT (1X,A6,6X,F10 0,4X,F10 0,5X,I4,4X,F5.2,9X,4(A4,1X))
 0192          206 FORMAT (1X,A6,6X,F10.0,4X,F10.0,5X,I4,4X,F5.2,9X,32HNO LAUNCH VEHI
                  1CLE CAN ACCOMPLISH)
 0193          213 FORMAT (16H STAGE COST DATA/6HOTITLE,3(16H  RECURRING  LC ),68H  D
                  1EVELOPMENT  SUSTAINING  AVAILABLE  SHARED COST GROUPS  BATCH FACT/
```

```
FORTRAN IV G LEVEL 1, MOD 4          ASSIGN          DATE = 70105          09/14/08

                2  10X,10H(HARDWARE),
                3       6X,10H(ETR ONLY),6X,10H(WTR ONLY),30X,8HFROM  TO//)
-0194          214 FORMAT (1HO///17HOSHARED COST DATA/37HONO. TITLE   DEVELOPMENT   S
                   1USTAINING//)
 0195         2141 FORMAT (1X,I2,2X,A4,2X,2F13.2)
 0196          215 FORMAT (1HO///22HOINTEGRATION COST DATA/59HOLOWER   UPPER    RECUR
                   1RING   LC   DEVELOPMENT   SUSTAINING/14H GROUP   GROUP//)
 0197          216 FORMAT (2X,A4,4X,A4,F11.2,F7.3,2F13.2)
 0198          217 FORMAT (14H1MISSION MODEL/48HO   MISSION   VELOCITY   PAYLOAD    P
                   1RIORITY  TR, 17X, 15HLAUNCH SCHEDULE//50X,20I4/1H /)
 0199          219 FORMAT (1X,I2,1X,A6,2X,2F10.0,F10.2,4X,I2,2X,20I4)
 0200         2001 FORMAT (17H1NUMBER OF STAGES,8X,I5/19HONUMBER OF VEHICLES,6X,I5/
                   1  19HONUMBER OF FAMILIES,6X,I5/28HONUMBER OF INTEGRATION COSTS,I2/
                   2  19HONUMBER OF MISSIONS,6X,I5/16HONUMBER OF YEARS,9X,I5/
                   3  17HOLAUNCH BASE YEAR,8X,I5/20HOTOTAL COST ESTIMATE,F12.2)
 0201         2021 FORMAT (42X,I4,4X,F5.2,9X,4(A4,1X))
 0202         2061 FORMAT (42X,I4,4X,F5.2,9X,32HNO LAUNCH VEHICLE CAN ACCOMPLISH)
 0203         4010 FORMAT(8H1MISSION,4X,14HCHARACTERISTIC,4X,7HPAYLOAD,4X,6HLAUNCH,
                   1  4X,6HNUMBER,10X,7HOPTIMUM/7H  TITLE,4X,16HVELOCITY(FT/SEC),4X,
                   2  5H(LBS),6X,4HYEAR,3X,11HOF LAUNCHES,4X,14HLAUNCH VEHICLE//)
 0204         4102 FORMAT (1HO///5X,26HEND OF DATA - JOB COMPLETE)
 0205         8001 FORMAT
                   1  (1X,A4,1X,3(F9.2,F7.3),F13.2,F12.2,2X,I4,1X,I4,2X,4I4,I9)
 0206         8003 FORMAT (4X,4F10.3)
 0207         8004 FORMAT (3X,19HRECURRING COST TYPE,I2,22H  FOR X LESS THAN OR =,
                   1 F6.2,14H, TOTAL COST =,F6.2,19H.FOR X GREATER THAN,F6.2,
                   2 14H, TOTAL COST =,F6.2,4H X +,F6.2)
 0208               END

FORTRAN IV G LEVEL 1, MOD 4          ASSIGN          DATE = 70105          09/14/08

  TOTAL MEMORY REQUIREMENTS 002130 BYTES
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,NCAL,MAP
        VARIABLE OPTIONS USED - SIZE=(126976,24576)                    DEFAULT OPTION(S) USED
IEW0000      NAME MOX02AN(R)
IEW0461  IBCOM#
IEW0461  CAPABL
IEW0461  STGNUM
IEW0461  DECISN
IEW0461  AVAIL
IEW0461  CHOOZ
IEW0461  ALOG

                                        MODULE MAP

  CONTROL SECTION                        ENTRY
    NAME     ORIGIN  LENGTH              NAME   LOCATION     NAME   LOCATION     NAME   LOCATION     NAME   LOCATI

  ASSIGN        00    2130
  SAVEA       2130    2160
  SAVEA1      4290     33C
  SAVEB       45D0    1BD0
  SAVEB1      61A0     410
  SAVECZ      65B0    3654
  ASGN        9C08     438
  SAVEDC      A040     5A8
  SAVESM      A5E8    30F8
  SAVECL      D6E0    08F8
  SAVED       DFD8     690
  SAVELZ      E668      F0
  SAVENV      E758       8
  BATCH       E760      2C
  SAVESG      E790       4
  SAVENR      E798       8
  SCRACH      E7A0    47A4

 ENTRY ADDRESS       00
 TOTAL LENGTH     12F48

****MOX02AN   NOW REPLACED IN DATA SET

                                    DIAGNOSTIC MESSAGE DIRECTORY

(17)          OS/360  FORTRAN H                                DATE  70.104/16.08.05

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
   ISN 0002            SUBROUTINE AVAIL
                     C     *** ADD AVAILABILITY TO VEHICLE CAPABILITY MATRIX***
   ISN 0003                INTEGER*2 LYR,LETT,MIN,IS,NONREC,NYD,LABS,LABF,LABI,MAT,VEH,LYD,
                          1  NYRSST
                     C  THE FOLLOWING STORAGE IS USED IN ASSIGN AND MASTER
   ISN 0004                COMMON/SAVEA/NM,YRLN(250),LYR(252),LETT(250),MIN(250),
                          1  DS(50),SUST(50),YD(50),IS(106),     NUMD,NONREC( 60,20),NYD(50),
                          2  LABS(40),LABF(30),LABI(40),                        RXD(LZ,50),MAT(50)
                     C  THE FOLLOWING STORAGE IS USED IN MASTER AND SMOOTH AND ASSIGN
   ISN 0005                COMMON/SAVEB1/NMIS,VEH(4,60),FINISH,RCOST(61),STG(40),NSPR,
                          1   NYRSST(72)
                     C   THE FOLLOWING STORAGE IS USED IN ASSIGN AND CHOOZ
   ISN 0006                COMMON/SAVECZ/RECUR( 60,20,2),VNH(4,250),LYD(50),NEXT,GUESS,
                          1  LTR(50)
                     C  THE FOLLOWING STORAGE IS USED IN CAPABL AND ASSIGN
   ISN 0007                COMMON/SAVECL/NOPT,IV,KNV,IG,     PLMD(50),KODEV(60),KODS(40),
                          2  (VEHA(50),VLR(50),HPR(50),VH(2,60),NPLS(50),HR(50),NRR(50)
   ISN 0008                COMMON/SAVELZ/LZ(60)
   ISN 0009                COMMON/SAVENV/NV,HYRS
                     C
   ISN 0010                DO 39 J = 1,NM
   ISN 0011                KO = LYR(J)
   ISN 0012                L = LETT(J)
   ISN 0013                DO 35 I = 1,NV
   ISN 0014                IF(ITEN(VH(I,I),L,1).EQ.0)  GO TO 38
   ISN 0016                DO 36 M = 1,20
   ISN 0017                IF(NONREC(I,M).EQ.0)  GO TO 37
   ISN 0019                NO = NONREC(I,M)
   ISN 0020                IF(KO.LT.NYD(NO))  GO TO 38
   ISN 0022                IF(KO.GT.LYD(NO))  GO TO 38
   ISN 0024              36 CONTINUE
   ISN 0025              37 LZ(I) = 1
   ISN 0026                 GO TO 35
   ISN 0027              38 LZ(I) = 0
   ISN 0028              35 CONTINUE
   ISN 0029                 CALL PACK(LZ,VNH(1,J),NV,1)
   ISN 0030              39 CONTINUE
                     C     NM LESS THAN 136 FOR PRESENT FORMATS
   ISN 0031             284 WRITE (6,4000)
   ISN 0032                 KNM = MINO(45,NM)
   ISN 0033                 K = 1
   ISN 0034             285 WRITE(6,4002)  (LETT(J), J = K,KNM)
```

```
ISN 0035            DO 420 I=1,NV
ISN 0036            IA=VEH(1,I)
ISN 0037            IB=VEH(2,I)
ISN 0038            IC=VEH(3,I)
ISN 0039            ID=VEH(4,I)
ISN 0040            DO 286 J = K,KNM
ISN 0041            LZ(J&1-K) = ITEM(VNM(1,J),1,1)
ISN 0042          286 CONTINUE
ISN 0043            WRITE(6,4100)I,STG(IA),STG(IB),STG(IC),STG(ID),(LZ(J&1-K),J=K,KNM)
ISN 0044          420 CONTINUE
ISN 0045            IF(NM.LE.KNM)  RETURN
ISN 0047            IF(NM.GT.90.AND.K.EQ.46)  GO TO 339
ISN 0049            KNM = MINO(90,NM)
ISN 0050            K = 46
ISN 0051            WRITE(6,4001)
ISN 0052            GO TO 285
ISN 0053          339 KNM = MINO(135,NM)
ISN 0054            K = 91
ISN 0055            WRITE(6,4003)
ISN 0056            GO TO 285
ISN 0057         4000 FORMAT (1H1,34X,51HV E H I C L E / M I S S I O N   C A P A B I L I
                     1 T Y/46X,30H(1 = POSSIBLE, 0 = IMPOSSIBLE)/1H0,43X,10(2H1 ),
                     2  10(2H2 ),10(2H3 ),6(2H4 )/18H VEHICLE / MISSION,9X,4(20H1 2 3 4
                     35 6 7 8 9 0 ),9H1 2 3 4 5//)
ISN 0058         4001 FORMAT(1H0/ 1H0,25X,4(2H4 ),10(2H5 ),10(2H6 ),10(2H7 ),10(2H8 ),-
                     1  2H9 /18H VEHICLE / MISSION,9X,9H6 7 8 9 0,4(20H1 2 3 4 5 6 7 8 9
                     2 0 1//)
ISN 0059         4002 FORMAT (1H0,7X,14HMISSION NUMBER,   4X,45I2)
ISN 0060         4003 FORMAT(1H0/ 1H0,25X,9(2H9 ),10(2H10),10(2H11),10(2H12),6(2H19)/
                     1   18H VEHICLE / MISSION, 9X,4(20H1 2 3 4 5 6 7 8 9 0 1,
                     2       9H1 2 3 4 5//)
ISN 0061         4100 FORMAT (1H ,I2,1X,4(A4,1X),2X,45I2)
ISN 0062              END

****** END OF COMPILATION ******
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,24576)          DEFAULT OPTION(S) USED
IEW0000       NAME MOX02AL(R)
IEW0461  ITEM
IEW0461  PACK
IEW0461  IBCOM=
```

CROSS REFERENCE TABLE

CONTROL SECTION NAME	ORIGIN	LENGTH
AVAIL	00	710
SAVEA	710	2160
SAVEB1	2870	410
SAVECZ	2C80	3654
SAVECL	6208	8F8
SAVELZ	6B00	F0
SAVENV	6CC0	8

LOCATION	REFERS TO SYMBOL	IN CONTROL SECTION	LOCATION	REFERS TO SYMBOL	IN CONTROL SECTION
2F0	SAVEA	SAVEA	2F4	SAVEA	SAVEA
2F8	SAVEB1	SAVEB1	2FC	SAVECZ	SAVECZ
300	SAVECZ	SAVECZ	304	SAVECZ	SAVECZ
308	SAVECL	SAVECL	30C	SAVELZ	SAVELZ
310	SAVENV	SAVENV	314	ITEM	$UNRESOLVED
318	PACK	$UNRESOLVED	31C	IBCOM=	$UNRESOLVED
250	SAVELZ	SAVELZ	258	SAVENV	SAVENV

```
ENTRY ADDRESS       00
TOTAL LENGTH      6CC8

****MOX02AL   NOW REPLACED IN DATA SET
```

DIAGNOSTIC MESSAGE DIRECTORY

IEW0461 WARNING - SYMBOL PRINTED IS AN UNRESOLVED EXTERNAL REFERENCE, NCAL WAS SPECIFIED.

```
(17)          OS/360  FORTRAN H                                    DATE  70.104/16.09.01

         COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
   ISN 0002          SUBROUTINE CAPABL
                  C   VEHICLE DATA IS INPUT
                  C  THE ORIGINAL CAPABILITY MATRIX BETWEEN VEHICLE AND MISSION IS SET UP
                  C
   ISN 0003          DOUBLE PRECISION NAME
   ISN 0004          INTEGER*2 VEH,NYRSST
                  C  THE FOLLOWING STORAGE IS USED IN MASTER AND SMOOTH AND ASSIGN
   ISN 0005          COMMON/SAVEB/NAME(56), MITR,  ALPI(4, 60),PLD(72),PLS(72),
                    1  PLR(50),RDIST(56,4),RFIXD(12,72),NSTRFX(72),NYRSFX(72),TREF
   ISN 0006          COMMON/SAVEB1/NMIS,VEH(4,60),FINISH,RCOST(61),STG(40),NSPR,
                    1  NYRSST(72)
                  C  THE FOLLOWING STORAGE IS USED IN CAPABL AND ASSIGN
   ISN 0007          COMMON/SAVECL/NOPT,IV,KNV,IG,     PLHD(50),KODEV(60),KODS(40),
                    2  IVEHA(50),VLR(50),WPR(50),VMI(2,60),NPLS(50),MR(50),NRR(50)
   ISN 0008          COMMON/SAVESG/NSTG
   ISN 0009          COMMON/SAVELZ/LZ(60)
   ISN 0010          COMMON/SAVENV/NV,MYRS
                  C   STORAGE USED IN SUBROUTINE ONLY & SAVED FOR BATCHING RUNS
   ISN 0011          COMMON/SAVED/B1(60),B2(60),B3(60),B4(60),NVS(60),NRV(60),NRP(60)
   ISN 0012          COMMON/SCRACH/DUMC(1156),NEH(4,60),DUMA(3189)
   ISN 0013          IF(IV.LT.0)  GO TO  14
   ISN 0015          DO 2 I = 1,60
   ISN 0016          ALPI(1,I) = .05
   ISN 0017          ALPI(2,I) = .20
   ISN 0018          ALPI(3,I) = .50
   ISN 0019        2 ALPI(4,I) = .25
   ISN 0020       14 DO 281 J = 1,60
   ISN 0021          IF(IV.LT.0.AND.J.GT.KNV)  RETURN
   ISN 0023          IF(IV.LT.0.AND.IG.LT.0)  GO TO 27
   ISN 0025          IF(IV.LT.0)  GO TO 15
   ISN 0027          READ(5,106)  (NEH(I,J),I=1,4),B1(J),B2(J),B3(J),B4(J),KODEV(J)
   ISN 0028          IF(KODEV(J).EQ.0)  RETURN
   ISN 0030          DO 16 K = 1,4
   ISN 0031       16 VEH(K,J) = NEH(K,J)
   ISN 0032          READ (5,108) NVS(J),NRV(J),NRP(J),JKEY
   ISN 0033          IF (JKEY.EQ.0) GO TO 15
   ISN 0035          READ (5,109) (ALPI(I,J),I=1,4)
   ISN 0036       15 DO 26 I = 1,4
   ISN 0037          IF(VEH(I,J).EQ.0)  GO TO 27
   ISN 0039          DO 25 K = 1,NSTG
   ISN 0040          IF(VEH(I,J).NE.KODS(K))  GO TO 25
   ISN 0042          VEH(I,J)  = K

   ISN 0043          GO TO 26
   ISN 0044       25 CONTINUE
   ISN 0045       26 CONTINUE
   ISN 0046       27 NV = J
   ISN 0047          C1 = B1(J)
   ISN 0048          C2 = B2(J)
   ISN 0049          C3 = B3(J)
   ISN 0050          C4 = B4(J)
   ISN 0051          DO 28 I=1,NMIS
   ISN 0052          LZ(I)=0
   ISN 0053          IF (IVEHA(I).EQ.0) GO TO 21
   ISN 0055          IF (IVEHA(I).EQ.KODEV(J)) LZ(I) = 1
   ISN 0057          GO TO 28
   ISN 0058       21 VLX=VLR(I)-25573.
   ISN 0059          IF(VLX.GE.C4) GO TO 28
   ISN 0061          WP=EXP(C1-C2*VLX-C3/(C4-VLX))
   ISN 0062          IF(WP.GT.WPR(I)) LZ(I)=1
   ISN 0064          IF(NOPT.NE.3)  GO TO 28
   ISN 0066          IF(LZ(I).EQ.0)  GO TO 28
   ISN 0068          IF(NPLS(I).EQ.0)  GO TO 8023
   ISN 0070          IF(NPLS(I).NE.NVS(J))   GO TO 8024
   ISN 0072     8023 IF(MR(I).EQ.0)  GO TO 8025
   ISN 0074          IF(MRV(J).NE.1)   GO TO 8024
   ISN 0076     8025 IF(NRR(I).LE.NRP(J))    GO TO 28
   ISN 0078     8024 LZ(I) = 0
   ISN 0079       28 CONTINUE
   ISN 0080          CALL PACK(LZ,VMI(1,J),NMIS,1)
   ISN 0081      281 CONTINUE
   ISN 0082       99 RETURN
   ISN 0083      106 FORMAT (4I2,4E13.6,18X,I2)
   ISN 0084      108 FORMAT (3X,3I2,70X,I1)
   ISN 0085      109 FORMAT (3X,4F5.2)
   ISN 0086          END

****** END OF COMPILATION ******
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
        VARIABLE OPTIONS USED - SIZE=(126976,24576)              DEFAULT OPTION(S) USED
IEW0000      NAME HOX02CL(R)
IEW0461  PACK
IEW0461  EXP
IEW0461  IBCOM=

                                        CROSS REFERENCE TABLE

  CONTROL SECTION                       ENTRY
    NAME     ORIGIN   LENGTH             NAME   LOCATION     NAME   LOCATION     NAME   LOCATION     NAME   LOCATION
  CAPABL         00      652
  SAVEB         658     1B00
  SAVEB1       2228      410
  SAVECL       2638      8F8
  SAVESG       2F30        4
  SAVELZ       2F38       F0
  SAVENV       3028        8
  SAVED        3030      690
  SCRACH       36C0     47A4


  LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION          LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION
      120           SAVEB          SAVEB                     124           SAVEB          SAVEB
      128           SAVEB1         SAVEB1                    12C           SAVECL         SAVECL
      130           SAVESG         SAVESG                    134           SAVELZ         SAVELZ
      138           SAVENV         SAVENV                    13C           SAVED          SAVED
      140           SCRACH         SCRACH                    144           SCRACH         SCRACH
      148           PACK           $UNRESOLVED               14C           EXP            $UNRESOLVED
      150           IBCOM=         $UNRESOLVED                A0           SAVELZ         SAVELZ
       A8           SAVEB1         SAVEB1
 ENTRY ADDRESS      00
 TOTAL LENGTH     7E68

****HOX02CL   NOW REPLACED IN DATA SET

                                        DIAGNOSTIC MESSAGE DIRECTORY
```

```
FORTRAN IV G LEVEL 1, MOD 4          CHOOZ            DATE = 70104        16/34/16

0001            SUBROUTINE CHOOZ
              C  DETERMINE OPTIMUM VEHICLE TO MISSION ASSIGNMENT
              C
              C  THE FOLLOWING STORAGE IS USED IN ASSIGN AND MASTER
0002              INTEGER*2    NSAVE,LYR,LETT,MIN,IS,NONREC,NYD,LABS,LABF,LABI,MAT,
                 1  LYD
              C
0003              COMMON/SAVEA/NM,YRLM(250),LYR(252),LETT(250),MIN(250),
                 1  DS(50),SUST(50),YD(50),IS(106),     NUMD,NONREC( 60,20),NYD(50),
                 2  LABS(40),LABF(30),LABI(40),                  RXD(12,50),MAT(50)
              C  THE FOLLOWING STORAGE IS USED IN ASSIGN AND CHOOZ
0004              COMMON/SAVECZ/RECUR( 60,20,2),VNM(4,250),LYD(50),NEXT,GUESS,
                 1  LTR(50)
              C
0005              COMMON/SCRACH/NODE(4,500),       WR(50),COST(2,250),
                 1  W2(500),Z(500),W(500),NSAVE(10),TOS(500),DUM(30)
0006              COMMON/SAVEL8/KX,KI,NX,K,KZ,NCOST,MZ(60)
0007              COMMON/SAVELZ/LZ(60)
0008              COMMON/SAVENV/NV,MYRS
              C
0009              DATA IONES /-1/
              C
0010              IF(MYRS.GT.10) GO TO 2
0011              KI = 1
0012              KNEX = MYRS
0013              GO TO 3
0014            2 KI = 2
0015              KNEX = (MYRS + 1)/2
0016            3 CONTINUE
              C
              C     ***   INITIALIZE FUNCTIONS   ***
0017            7 NEXT=1
0018              NX=1
0019              KPNX = 10
0020              DO 16 I=1,4
0021           16 NODE(I,1)=IONES
                  :
                  :     ***   FIND W(1) = SUM OF COLUMN MINIMUMS OF FIRST CASE   ***
0022              W(1)=0.0
0023              W2(1) = 0.0
0024              TOS(1) = 0.0
0025              DO 19 J=1,NM
0026              IY = LYR(J)
```

```
FORTRAN IV G LEVEL 1, MOD 4          CHOOZ          DATE = 70104        16/34/16

0027              JX = LETT(J)
0028              ITR = LTR(JX)
0029              CALL UNPACK(MZ,VNM(1,J),NV,1)
0030              COST(1,J) = 1.0E30
0031              COST(2,J) = 1.0E30
0032              DO 18 I=1,NV
0033              IF(MZ(I).EQ.0)  GO TO 18
0034              CX = YRLM(J)*RECUR(I,IY,ITR)
0035              IF(CX.GE.COST(2,J))  GO TO 18
0036              IF(CX.LT.COST(1,J))  GO TO 176
0037              COST(2,J) = CX
0038              GO TO 18
0039            176 COST(2,J) = COST(1,J)
0040              COST(1,J)  = CX
0041              MIN(J) = I
0042             18 CONTINUE
0043              IF(COST(1,J).LT.1.0E25)  GO TO 20
0044              YRLM(J)=0.0
0045              COST(1,J) = 0.0
0046              COST(2,J) = 0.0
0047             20 W2(1) = W2(1) + COST(2,J)
0048             19 W(1) = W(1) + COST(1,J)
0049              IF(NUMD.NE.0)  GO TO 29
0050              WRITE(6,211)  W(1)
0051            211 FORMAT(1HO//25H PROGRAM RECURRING COST =, F12.2)
0052              RETURN
                C
                C     ***   PICK COST TO CONSIDER NEXT   ***
                C
0053             29 NCOST = 0
0054              NKEY = 0
0055              FMAX = -1.0E35
0056              IF(KPNX.EQ.NX)  GO TO 30
0057              CALL UNPACK (LZ,NODE(1,NX),NUMD,4)
0058             30 DO 35 NIC = 1,NUMD
0059              IF(LZ(NIC).LT.15) GO TO 35
0060              NKEY = NKEY + 1
0061              IF(KPNX.EQ.NX)  GO TO 300
0062              WR(NIC) = 0.0
0063              DO 33 J= 1,NM
0064              IF(YRLM(J).EQ.0.0) GO TO 33
0065              CALL UNPACK(MZ,VNM(1,J),NV,1)
0066              CMIN = 1.0E30
```

```
FORTRAN IV G LEVEL 1, MOD 4          CHOOZ          DATE = 70104        16/34/16

0067              KO = LYR(J)
0068              JX = LETT(J)
0069              ITR = LTR(JX)
0070              DO 32 I = 1,NV
0071              IF(MZ(I).EQ.0)  GO TO 32
0072              DO 31 M = 1, 20
0073              IF(NONREC(I,M).EQ.0)  GO TO 315
0074              NO = NONREC(I,M)
0075              IF(NO.EQ.NIC)  GO TO 32
0076              IF(KI*LZ(NO).LT.KO)  GO TO 32
0077             31 CONTINUE
0078            315 CX = YRLM(J)*RECUR(I,KO,ITR)
0079              IF(CX.LT.CMIN)  CMIN = CX
0080             32 CONTINUE
0081              WR(NIC) = WR(NIC) + CMIN
0082             33 CONTINUE
0083            300 PF = WR(NIC) - W(NX)
0084              IF(SUST(NIC).GT.4.0)  DF = DS(NIC)*0.5 + SUST(NIC) + PF
0085              IF(SUST(NIC).GE..001.AND.SUST(NIC).LE.2.0)  DF = 0.5*DS(NIC) +
                 1 SUST(NIC) + PF - 1.0E32
0086              IF(SUST(NIC).GT.2.0.AND.SUST(NIC).LE.4.0)  DF=0.5*DS(NIC) +
                 1  SUST(NIC) + PF - 1.0E5
0087              IF(SUST(NIC).LT..001)  DF = 0.5*DS(NIC) + 4.0 + PF
0088              IF(DF.LE.FMAX)  GO TO 35
0089              FMAX = DF
0090              NCOST = NIC
0091             35 CONTINUE
0092             36 IF(NCOST.EQ.0) GO TO 75
                C
0093            295 CONTINUE
                C
                C     ALLOCATE SPACE FOR NEW NODES
                C
0094              IF(SUST(NCOST).GE..001)  J=0 + (NYD(NCOST) - 1)/KI
0095              IF(SUST(NCOST).LT..001)   J = KNEX - 1
0096              IF(NEXT.EQ.1)  GO TO 41
0097              DO 40  I = 2,NEXT
0098              K = NEXT + 2 - I
0099              IF(Z(K).LE.GUESS)  GO TO 40
0100              J=J+1
0101              IF(SUST(NCOST).GE..001)  NSAVE(J) = K
0102              IF(SUST(NCOST).LT..001)  NSAVE(1) = K
0103              IF(J.EQ.KNEX) GO TO 44
```

```
FORTRAN IV G LEVEL 1, MOD 4          CHOOZ          DATE = 70104          16/34/16

0104           40 CONTINUE
0105           41 IF(J.EQ.KNEX)  GO TO 44
0106              J=J+1
0107              NEXT=NEXT+1
0108              IF(NEXT.EQ.500) GO TO 74
0109              IF(SUST(NCOST).GE..001)  NSAVE(J) = NEXT
0110              IF(SUST(NCOST).LT..001)  NSAVE(1) = NEXT
0111              GO TO 41
          C
          C       ***   BRANCH WITH VARYING YEARS OF SUSTAINING COST   ***
0112           44 DO 52 K=1,10
0113              IF(SUST(NCOST).GE..001.AND.K.LT.1+(NYD(NCOST)-1)/KI)  GO TO 52
0114              KX=NSAVE(K)
0115              IF((K-1)*KI.LT.LYD(NCOST)) GO TO 45
0116              W(KX) = 1.0E30
0117              Z(KX) =20.0E30
0118              GO TO 509
0119           45 DO 46 I=1,4
0120           46 NODE(I,KX)=NODE(I,NX)
0121              LZ(NCOST)=K-1
0122              IF(K.EQ.1 + (NYD(NCOST)-1)/KI)   LZ(NCOST) = 0
0123              CALL PACK(LZ,NODE(1,KX),NUMD,4)
          C
0124              CALL LBOUND
          C
0125          509 IF(SUST(NCOST).LT..001) GO TO 53
0126              IF(KI*K.GE.MYRS) GO TO 53
0127           52 CONTINUE
          C
          C       ***   BRANCH INCLUDING NCOST AND ALL SUSTAINING - PUT IN NODE NX   ***
0128           53 LZ(NCOST) = (LYD(NCOST) + KI - 1)/KI
0129              CALL PACK (LZ,NODE(1,NX),NUMD,4)
0130              IF(W(NX).GT.W(KX)-.0001.AND.W2(NX).GT.W2(KX)-0.0001)  K = 50
0131              IF(W(NX).GT.W(KX)-.0001.AND.W2(NX).GT.1.0E25.AND.W2(KX)-W2(NX).LT.
                 1  1.0E25)  K = 50
0132              KX = NX
0133              KZ = LYD(NCOST)
          C
0134              CALL LBOUND
          C
          C
          C       ***   IMPROVE GUESS IF AT BOTTOM OF TREE   ***
0135              IF(NKEY.GT.1) GO TO 55
```

```
FORTRAN IV G LEVEL 1, MOD 4          CHOOZ          DATE = 70104          16/34/16

0136              IF(Z(NX).LT.GUESS)  GUESS = Z(NX)
0137              DO 51 I=1,KNEX
0138              IF(SUST(NCOST).GE..001.AND.I.LT.1+(NYD(NCOST)-1)/KI)  GO TO 51
0139              K=NSAVE(I)
0140              IF(Z(K).LT.GUESS)  GUESS = Z(K)
0141              IF(SUST(NCOST).LT..001) GO TO 55
0142           51 CONTINUE
          C
          C       PICK NEXT NODE FOR BRANCHING AS THE ONE WITH LEAST LOWER BOUND Z
0143           55 KPNX = NX
0144              NX = 1
0145              DO 59  I=2,NEXT
0146              IF(Z(NX).GT.Z(I))  NX = I
0147           59 CONTINUE
0148              IF(Z(NX).LE.GUESS)  GO TO 29
0149              WRITE(6,202)
0150              GUESS = 0.0
0151              GO TO 99
0152           74 WRITE(6,203)
0153              GO TO 99
0154           75 DO 80 J=1,NM
0155              IF(YRLM(J).EQ.0.0) GO TO 80
0156              CALL UNPACK(MZ,VNM(1,J),NV,1)
0157              CMIN=1.0E30
0158              KO = LYR(J)
0159              JX = LETT(J)
0160              ITR = LTR(JX)
0161              DO 78 I=1,NV
0162              IF(MZ(I).EQ.0)  GO TO 78
0163              DO 77 K=1,20
0164              IF(NONREC(I,K).EQ.0) GO TO 775
0165              NO =NONREC(I,K)
0166              IF(KI*LZ(NO).LT.KO)     GO TO 78
0167           77 CONTINUE
0168          775 CX=YRLM(J)*RECUR(I,KO,ITR)
0169              IF(CX.GE.CMIN) GO TO 78
0170              CMIN=CX
0171              MIN(J)=I
0172           78 CONTINUE
0173           80 CONTINUE
0174              DMIN = Z(NX) - W(NX)
0175              WRITE(6,201) NX,W(NX),DMIN,Z(NX)
0176              DO 76 NO = 1,NUMD
```

```
FORTRAN IV G LEVEL 1, MOD 4            CHOOZ            DATE = 70104          16/34/16

0177           76 LZ(NO) = LZ(NO)*K1
0178          201 FORMAT (1HO,13(1H*),17H S O L U T I O N ,12(1H*)/1H ,I3,29X,
                1  3(F9.2,5X))
0179          202 FORMAT(14H1GUESS TOO LOW)
0180          203 FORMAT (19H1EXCEEDED 500 NODES)
0181           99 RETURN
0182              END

FORTRAN IV G LEVEL 1, MOD 4            CHOOZ            DATE = 70104          16/34/16

  TOTAL MEMORY REQUIREMENTS 001544 BYTES
```

```
F68-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,NCAL,MAP
        VARIABLE OPTIONS USED - SIZE=(126976,24576)                    DEFAULT OPTION(S) USED
IEW0000      NAME MOXO2CZ(R)
IEW0461  UNPACK
IEW0461  IBCOM*
IEW0461  PACK
IEW0461  LBOUND

                                                MODULE MAP

  CONTROL SECTION                          ENTRY
    NAME     ORIGIN  LENGTH                 NAME   LOCATION     NAME   LOCATION     NAME   LOCATION     NAME   LOCATION

  CHOOZ          00    1544
  SAVEA        1548    2160
  SAVECZ       36A8    3654
  SCRACH       6000    47A4
  SAVELB       B4A8     108
  SAVELZ       B5B0      F0
  SAVENV       B6A0       8

  ENTRY ADDRESS        00
  TOTAL LENGTH       B6A8

****MOXO2CZ   NOW REPLACED IN DATA SET

                                        DIAGNOSTIC MESSAGE DIRECTORY

   IEW0461 WARNING - SYMBOL PRINTED IS AN UNRESOLVED EXTERNAL REFERENCE, NCAL WAS SPECIFIED.
```

```
(17)          OS/360  FORTRAN H                                DATE  70.104/16.34.49

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
   ISN 0002          SUBROUTINE CONSTR
                  C    DETERMINE IF ANY PROGRAM CONSTRAINTS HAVE BEEN VIOLATED
   ISN 0003           INTEGER PROG
   ISN 0004           INTEGER*2 IVEH,LVS,NSTRST,LVARY,LVD,       NLVP,LABEL,LNDATE
   ISN 0005           COMMON/PLSAVE/TITLE(10),FIXED(20),LEVEL(20),CNTRVL(20),
                     1  PMAX,PMIN,ACCL,EXT,ISTRT,IFIN,MAXITR,NCSTR
   ISN 0006           COMMON/SCRACH/TOTAL(20),W(20),NSL(10),NDP(72),LVSF(66),D(20),
                     1  XLVSUM(20,50),XOUT(20),VOUT(20),RF(72),CF(72),SF(72),FLAGR(72),
                     2  FLAGS(72),PROG,IODD, KVEHI(50),RRR(20),IMAGE(830),YEAR(20),
                     3    Y(20),NSSF(72),NSRF(72),NSXF(72),NDSF(72),SUSTF(72),VNAM(66),
                     4NSCALE(5),RECUR(20,50),NSTRRC(72),NYRSRC(72),LNDF(72),KVEH( 60),
                     5  IERR,SKIP,MYFLAG,NPRO(90),KPRO(90),DUMS(145)
                  C  THE FOLLOWING STORAGE IS USED IN MASTER AND SMOOTH
   ISN 0007           COMMON/SAVES/N,M,IVEH(66),LVS(66),S(72),NSTRST(72),R(72),LVARY(66)
                     1 ,LVD(66),          NLVP(72),XSCH(10,66),LABEL(50),NCS,CS(90),
                     2  NPROG(90),KPROG(90),KODE(90),LNDATE(72)
   ISN 0008           IERR = 0
   ISN 0009           IF (NCSTR.EQ.0) RETURN
   ISN 0011           NP = PROG
   ISN 0012           DO 100 I=1,NCSTR
   ISN 0013           J = NPROG(I)
   ISN 0014           K = KPROG(I)
   ISN 0015           IF (J.NE.NP.AND.K.NE.NP)  GO TO 100
   ISN 0017           NP= KODE(I)
   ISN 0018           IF (NP.LT.1.OR.NP.GT.11)  GO TO 100
   ISN 0020           GO TO (10,20,30,40,50,60,70,110,90,91,92),  NP
   ISN 0021        10 DT = CS(I)
   ISN 0022           IF (S(J).LT.(S(K)&R(K) & DT)) GO TO 110
   ISN 0024           GO TO 100
   ISN 0025        20 DT = CS(I)
   ISN 0026           IF ((S(J)&R(J)&DT).GT.S(K))  GO TO 110
   ISN 0028           GO TO 100
   ISN 0029        30    IF(S(J).NE.CS(I)) GO TO 110
   ISN 0031           GO TO 100
   ISN 0032        40 IF((S(J) & R(J) - 1.0).NE.CS(I))  GO TO 110
   ISN 0034           GO TO 100
   ISN 0035        50      IF (R(J).NE.CS(I)) GO TO 110
   ISN 0037           GO TO 100
   ISN 0038        60 DT = LNDATE(J)
   ISN 0039           ET = LNDATE(K)
   ISN 0040           IF((S(J)&DT&CS(I)).GT.(S(K)&ET))
                     1  GO TO 110
```

```
ISN 0042            GO TO 100
ISN 0043         70 DT = LNDATE(J) - 1
ISN 0044            IF ((S(J) & DT).GT.CS(I))   GO TO 110
ISN 0046            GO TO 100
ISN 0047         90  IF(S(J).LT.CS(I))  GO TO 110
ISN 0049            GO TO 100
ISN 0050         91 DT = LNDATE(J) - 1
ISN 0051            IF ((S(J) & DT).LT.CS(I))  GO TO 110
ISN 0053            GO TO 100
ISN 0054         92 DT = LNDATE(K) -1
ISN 0055            IF ((S(J)&R(J)&CS(I)).GT.(S(K)&DT))  GO TO 110
ISN 0057        100 CONTINUE
ISN 0058            RETURN
ISN 0059        110 IERR = 1
ISN 0060        120 RETURN
ISN 0061            END

****** END OF COMPILATION ******
```

```
 F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
1        VARIABLE OPTIONS USED - SIZE=(126976,24576)              DEFAULT OPTION(S) USED
IEW0000      NAME HOX02CR(R)

                                        CROSS REFERENCE TABLE

  CONTROL SECTION                    ENTRY
    NAME     ORIGIN  LENGTH            NAME   LOCATION     NAME   LOCATION     NAME   LOCATION     NAME   LOCATION
  CONSTR         00     3EC
  PLSAVE        3F0     138
  SCRACH        528    47A4
  SAVES        4CD0    1660


  LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION          LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION
       F0           PLSAVE          PLSAVE                    F4           SCRACH          SCRACH
       F8           SCRACH          SCRACH                    FC           SCRACH          SCRACH
      100           SCRACH          SCRACH                   104           SAVES           SAVES
      108           SAVES           SAVES
 ENTRY ADDRESS      00
 TOTAL LENGTH     6330

****HOX02CR   NOW REPLACED IN DATA SET
```

```
(171)      OS/360  FORTRAN H                              DATE  70.104/16.35.17

      COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
ISN 0002          SUBROUTINE DECISN
                  C
                  C  THIS SUBROUTINE SETS UP DS COSTS, CALCULATES AVAILABILITY OF EACH DECISION
                  C   COST, AND MATCHES THESE COSTS WITH EACH VEHICLE THEN PRINTS THEM OUT
                  C
ISN 0003          INTEGER FINISH
ISN 0004          INTEGER*2 LYR,LETT,MIN,IS,NONREC,NYO,LABS,LABF,LABI,MAT,VEH,LYD,
                 1 NYRSST
                  C  STORAGE USED IN DECISN AND ASSIGN
ISN 0005          COMMON/ASGN/1ST(40),JST(30),KST(40),KODEM(50),YDS(40),YOF(30),
                 1   YDI(40)
ISN 0006          COMMON/SAVEDC/KFLAG,NFAM,LSA(40),SNR(40),STS(40),NYS(40),FMNR(30),
                 1  KODEF(30),FMSUS(30),FAM(30),DINT(40),SINT(40)
                  C  THE FOLLOWING STORAGE IS USED IN ASSIGN AND MASTER
ISN 0007          COMMON/SAVEA/NM,YRLN(250),LYR(252),LETT(250),MIN(250),
                 1  DS(50),SUST(50),YD(50),IS(106),    NUMD,NONREC( 60,20),NYD(50),
                 2  LABS(40),LABF(30),LABI(40),                    RXD(12,50),MAT(50)
                  C  THE FOLLOWING STORAGE IS USED IN MASTER AND SMOOTH AND ASSIGN
ISN 0008          COMMON/SAVEB1/NMIS,VEHI(4,60),FINISH,RCOST(61),STG(40),NSPR,
                 1   NYRSST(22)
                  C   THE FOLLOWING STORAGE IS USED IN ASSIGN AND CHOOZ
ISN 0009          COMMON/SAVEC2/RECURI 60,20,2),VNM(4,250),LYD(50),NEXT,GUESS,
                 1  LTR(50)
                  C  THE FOLLOWING STORAGE IS USED IN STGNUM
ISN 0010          COMMON/SAVESH/IFLAG,STGYTR(40,20,2),RINTYR(40,20),NBY(40),NCI,
                 1     PLCINT(40),                  NFHL(40),NFS(40,4),NFHU(40),
                 2   MODE(40,3),SR(40,3),PLC(40,3),POJ(3),SRJ(3,3),RINT(40)
ISN 0011          COMMON/SAVENV/NV,NYRS
ISN 0012          COMMON/SAVESG/NSTG
ISN 0013          COMMON/SCRACH/HAS(40), LYF(30),NYF(30),MAF(30),HAIC(40),DUMD(4415)
                  C
ISN 0014          IF(FINISH.GT.1)  GO TO 2
                  C
                  C     ***SET UP DS  COSTS FOR BRANCH AND BOUND PROCEDURE***
                  C     CALCULATE AVAILABILITY OF EACH DECISION COST
                  C
ISN 0016          NUMD = 0
ISN 0017          DO 3 I = 1,NSTG
ISN 0018          LSA(I) = MIND(LSA(I),NYRS)
ISN 0019          HAS(I)=0
ISN 0020          X = LABS(I)
ISN 0021          IF(SNR(I)&STS(I)&X.LT..01)  GO TO 3

ISN 0023          NUMD = NUMD & 1
ISN 0024          DS (NUMD)=SNR(I)
ISN 0025          IF(LABS(I).EQ.0)  GO TO 302
ISN 0027          L = LABS(I)
ISN 0028          DO 301 K = 1,12
ISN 0029       301 DS(NUMD) = DS(NUMD) & RXD(K,L)
ISN 0030       302 SUST (NUMD)=STS(I)
ISN 0031          MAT(NUMD) = I
ISN 0032          HAS(I) = NUMD
ISN 0033          NYD(NUMD) = NYS(I)
ISN 0034          LYD(NUMD) = LSA(I)
ISN 0035          YD(NUMD) = YDS(I)
ISN 0036          IS(NUMD & NMIS&NSPR)  = IST(I) & 1900
ISN 0037        3 CONTINUE
ISN 0038          IF(NFAM.EQ. 0) GO TO 601
                  C     CALCULATE FAMILY AVAILABILITY DATE
                  C FIRST YR. FAMILY IS AVAIL. = 1ST YR. ANY STAGE IN THAT FAMILY IS AVAIL.
ISN 0040          DO 422 I1 = 1,NFAM
ISN 0041          I = KODEF(I1)
ISN 0042          LYF(I) = 0
ISN 0043       422 NYF(I) = NYRS
ISN 0044          DO 423 J = 1,NSTG
ISN 0045          DO 424 MS = 1,4
ISN 0046          I = NFS(J,MS)
ISN 0047          IF(I.EQ.0)  GO TO 423
ISN 0049          NYF(I) = MIND (NYF(I),NYS(J))
ISN 0050          LYF(I) = MAXO(LYF(I),LSA(J))
ISN 0051       424 CONTINUE
ISN 0052       423 CONTINUE
ISN 0053          DO 6 I1 = 1,NFAM
ISN 0054          I = KODEF(I1)
ISN 0055          MAF(I)=0
ISN 0056          X = LABF(I)
ISN 0057          IF(FMNR(I)&FMSUS(I)&X.LT..01)  GO TO 6
ISN 0059          NUMD = NUMD & 1
ISN 0060          DS (NUMD) = FMNR(I)
ISN 0061          IF(LABF(I).EQ.0)  GO TO 304
ISN 0063          L = LABF(I)
ISN 0064          DO 303 K = 1,12
ISN 0065       303 DS(NUMD) = DS(NUMD) & RXD(K,L)
ISN 0066       304 SUST (NUMD)=FMSUS(I)
ISN 0067          MAT(NUMD) = -I
```

```
ISN 0068           MAP(I) = NUMD
ISN 0069           NYD(NUMD) = NYF(I)
ISN 0070           LYD(NUMD) = LYF(I)
ISN 0071           YD(NUMD) = YDF(I)
ISN 0072           IS(NUMD & NMIS & NSPR) = JST(I) & 1900
ISN 0073         6 CONTINUE
ISN 0074        601 IF(NCI.EQ.0) GO TO 4
ISN 0076           DO 60 I = 1,NCI
ISN 0077           MAIC(I)=0
ISN 0078           X = LABI(I)
ISN 0079           IF(DINT(I)&SINT(I)&X.LT..01)  GO TO 60
ISN 0081           NUMD = NUMD & 1
ISN 0082           DS (NUMD) = DINT(I)
ISN 0083           IF(LABI(I).EQ.0)  GO TO 306
ISN 0085           L = LABI(I)
ISN 0086           DO 305 K = 1,12
ISN 0087        305 DS(NUMD) = DS(NUMD) & RXD(K,L)
ISN 0088        306 SUST (NUMD)=SINT(I)
ISN 0089           MAT(NUMD) = -100 -I
ISN 0090           MAIC(I) = NUMD
                C FIRST YR. INT. COST IS AVAIL. = 1ST YR. BOTH FAMS. ARE AVAIL.
ISN 0091           JF = NFHL(I)
ISN 0092           KF = NFHU(I)
ISN 0093           NYD(NUMD) = MAX0(NYF(JF),NYF(KF))
ISN 0094           LYD(NUMD) = MIN0(LYF(JF),LYF(KF))
ISN 0095           YD(NUMD) = YDI(I)
ISN 0096           IS(NUMD & NMIS & NSPR) = KST(I) & 1900
ISN 0097         60 CONTINUE
                C
                C     ***MATCH DECISION COSTS WITH EACH VEHICLE***
                C
ISN 0098          4 DO 66 I = 1,NV
ISN 0099           JX = 1
ISN 0100           DO 64 J = 1,20
ISN 0101        64 NONREC(I,J) = 0
ISN 0102        25 DO 65 MS= 1,4
ISN 0103           K= VEH(MS,I)
ISN 0104           IF(K.EQ.0) GO TO 66
ISN 0106           IF(MAS(K).EQ. 0)  GO TO 63
ISN 0108           NONREC  (I,JX)   = MAS(K)
ISN 0109           JX = JX & 1
ISN 0110           IF(JX.GT.20) GO TO 93
```

```
ISN 0112           IF(NFAM.EQ.0)  GO TO 21
                C     ***   PICK UP SHARED COSTS   ***
ISN 0114        63 DO 885 KY=1,4
ISN 0115           KX=NFS(K,KY)
ISN 0116           IF(KX.EQ.0) GO TO 885
ISN 0118           IF(MAP(KX).EQ.0) GO TO 885
ISN 0120           NONREC(I,JX)=MAP(KX)
ISN 0121           JX=JX&1
ISN 0122           IF(JX.GT.20) GO TO 93
ISN 0124        885 CONTINUE
                C     ***   PICK UP INTEGRATION COSTS   ***
ISN 0125        21 IF(NCI.EQ.0) GO TO 65
ISN 0127           IF(MS.EQ.4) GO TO 65
ISN 0129           IF(VEH(MS&1,I).EQ.0) GO TO 65
ISN 0131           K1=VEH(MS&1,I)
ISN 0132           DO 89 J=1,NCI
ISN 0133           DO 887 KY=1,4
ISN 0134           IF(NFHL(J).NE.NFS(K,KY)) GO TO 887
ISN 0136           DO 886 KZ=1,4
ISN 0137           IF(NFHU(J).EQ.NFS(K1,KZ)) GO TO 888
ISN 0139        886 CONTINUE
ISN 0140        887 CONTINUE
ISN 0141           GO TO 89
ISN 0142        888 IF(MAIC(J).EQ.0) GO TO 89
ISN 0144           NONREC(I,JX) = MAIC(J)
ISN 0145           JX = JX & 1
ISN 0146           IF(JX.GT.20) GO TO 93
ISN 0148        89 CONTINUE
ISN 0149        65 CONTINUE
ISN 0150        66 CONTINUE
                C
                C     ***PRINT OUT DECISION COST CATEGORIES***
ISN 0151         2 IF(NUMD.EQ.0)  RETURN
ISN 0153           WRITE(6,211)
ISN 0154           DO 925 I = 1,NUMD
ISN 0155           J=MAT(I)
ISN 0156           IF(J.LT.-100) GO TO 345
ISN 0158           IF(J.LT.0) GO TO 340
ISN 0160           WRITE (6,208) I,DS(I),SUST(I),STG(J),NYD(I),LYD(I),IS(I&NMIS&
                  1  NSPR), YD(I)
ISN 0161           GO TO 925
ISN 0162        340 JX=-J
```

```
ISN 0163            WRITE(6,209) I, DS(I),   SUST(I), FAN(JX),NYD(I),LYD(I),IS(IENMISG
                   1  NSPR), YD(I)
ISN 0164            GO TO 925
ISN 0165          345 JX=-J-100
ISN 0166            JY=NFHL(JX)
ISN 0167            JZ=NFHU(JX)
ISN 0168            WRITE(6,210) I, DS(I),  SUST(I), FAN(JY),FAN(JZ),NYD(I),LYD(I),
                   1  IS(IENMISGNSPR),  YD(I)
ISN 0169          925 CONTINUE
ISN 0170            RETURN
ISN 0171           93 WRITE(6,220) I
ISN 0172            KFLAG = 1
ISN 0173           99 RETURN
ISN 0174          208 FORMAT (I4,6X,2F12.2,5X,A4,1X,5HSTAGE,26X,I3,9X,I3,8X,I5,7X,F5.0)
ISN 0175          209 FORMAT (I4,6X,2F12.2,5X,A4,1X,6HSHARED,25X,I3,9X,I3,8X,I5,7X,F5.0)
ISN 0176          210 FORMAT(I4,6X,2F12.2,5X,15HINTEGRATION OF ,A4,5H AND ,A4,8X,I3,9X,
                   1  I3,8X,I5,7X,F5.0)
ISN 0177          211 FORMAT(25H1QUANTITIES BRANCHED UPON/1HO,11X,11HDEVELOPMENT,2X,
                   1  10HSUSTAINING,38X,10HYEAR AVAIL,2X,9HLAST YEAR,2X,9HDEV START,
                   2  2X,12HDEV DURATION//)
ISN 0178          220 FORMAT(45HOEXCEEDED 20  NON-RECURRING COSTS FOR VEHICLE,I4)
ISN 0179            END

****** END OF COMPILATION ******
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,24576)              DEFAULT OPTION(S) USED
IEW0000      NAME MOX02DC(R)
IEW0461  IBCOM#
```

CROSS REFERENCE TABLE

CONTROL SECTION NAME	ORIGIN	LENGTH	ENTRY NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
DECISN	00	ED8								
ASGN	ED8	438								
SAVEDC	1310	5A8								
SAVEA	1888	2160								
SAVEB1	3A18	410								
SAVECZ	3E28	3654								
SAVESM	7480	30F8								
SAVENV	A578	8								
SAVESG	A580	4								
SCRACH	A588	47A4								

LOCATION	REFERS TO SYMBOL	IN CONTROL SECTION	LOCATION	REFERS TO SYMBOL	IN CONTROL SECTION
248	ASGN	ASGN	24C	SAVEDC	SAVEDC
250	SAVEA	SAVEA	254	SAVEA	SAVEA
258	SAVEB1	SAVEB1	25C	SAVECZ	SAVECZ
260	SAVECZ	SAVECZ	264	SAVECZ	SAVECZ
268	SAVESM	SAVESM	26C	SAVESM	SAVESM
270	SAVESM	SAVESM	274	SAVENV	SAVENV
278	SAVESG	SAVESG	27C	SCRACH	SCRACH
280	IBCOM#	$UNRESOLVED			

ENTRY ADDRESS 00
TOTAL LENGTH ED30

```
****MOX02DC   NOW REPLACED IN DATA SET
```

DIAGNOSTIC MESSAGE DIRECTORY

```
H(17)    OS/360 FORTRAN H                                DATE  70.104/16.48.10

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
ISN 0002            SUBROUTINE LBOUND
                C   THIS SUBROUTINE CALCULATES THE RECURRING AND NON-RECURRING LOWER
                C     BOUND WITH A PENALTY FUNCTION INCLUDED IF W NE 1.E30
                C
ISN 0003            INTEGER*2    NSAVE,LYR,LETT,MIN,IS,NONREC,NYD,LABS,LABF,LABI,MAT,
                   1  LYD,KEEP
ISN 0004            COMMON/SAVEA/NM,YRLM(250),LYR(252),LETT(250),MIN(250),
                    1  DS(50),SUST(50),YD(50),IS(106),    NUMD,NONREC( 60,20),NYD(50),
                    2  LABS(40),LABF(30),LABI(40),                    RXD(12,50),MAT(50)
ISN 0005            COMMON/SAVECZ/RECUR( 60,20,2),VNM(4,250),LYD(50),NEXT,GUESS,
                    1  LTR(50)
ISN 0006            COMMON/SAVEL6/KX,KI,NX,K,KZ,NCOST,MZ(60)
ISN 0007            COMMON/SAVELZ/LZ(60)
ISN 0008            COMMON/SAVENV/NV,MYRS
ISN 0009            COMMON/SCRACH/NODE(4,500),       WR(50),  COST(2,250),
                    1  W2(500),Z(500),W(500),NSAVE(10),TDS(500),KEEP(50),DUM(5)
                C
ISN 0010            IF(K.EQ.50)  GO TO 54
                C     ***FIND NEW RECURRING LOWER BOUND***
ISN 0012         49 W(KX)=0.
ISN 0013            W2(KX) = 0.0
ISN 0014            DO 50 J=1,NM
ISN 0015            IF(YRLM(J).EQ.0.0) GO TO 50
ISN 0017            CALL UNPACK(MZ,VNM(1,J),NV,1)
ISN 0018            COST(1,J) = 1.0E30
ISN 0019            COST(2,J) = 1.0E30
ISN 0020            KO = LYR(J)
ISN 0021            JX = LETT(J)
ISN 0022            ITR = LTR(JX)
ISN 0023            DO 48 I=1,NV
ISN 0024            IF(MZ(I).EQ.0)  GO TO 48
ISN 0026            DO 47 M=1,20
ISN 0027            IF(NONREC(I,M).EQ.0) GO TO 475
ISN 0029            NO = NONREC(I,M)
ISN 0030            IF(KI*LZ(NO).LT.  KO  ) GO TO 48
ISN 0032         47 CONTINUE
ISN 0033        475 CX=YRLM(J)*RECUR(I,KO,ITR)
ISN 0034            IF(CX.GE.COST(2,J))  GO TO 48
ISN 0036            IF(CX LT.COST(1,J))  GO TO 43
ISN 0038            COST(2,J) = CX
ISN 0039            GO TO 48
ISN 0040         43 COST(2,J) = COST(1,J)

ISN 0041            COST(1,J) = CX
ISN 0042            MIN(J) = I
ISN 0043         48 CONTINUE
ISN 0044            W(KX)=W(KX)&COST(1,J)
ISN 0045            W2(KX) = W2(KX) & COST(2,J)
ISN 0046         50 CONTINUE
ISN 0047            IF(KX.EQ.NX)  GO TO 510
ISN 0049            KZ = KI*LZ(NCOST)
ISN 0050            IF(W(KX).LT.1.0E20)  GO TO 508
ISN 0052            TGO = 0.0
ISN 0053            GO TO 38
ISN 0054        508 IF(KZ.EQ.0)  GO TO 510
ISN 0056            KY = NSAVE(K-1)
ISN 0057        512 IF(W(KX).GT.W(KY)-.0001.AND.W2(KX).GT.W2(KY)-.0001)  GO TO 38
ISN 0059            IF(W(KX).GT.W(KY)-.0001.AND.W2(KX).GT.1.0E25.AND.W2(KY)-W2(KX).LT.
                   1  1.0E25)  GO TO 38
                C
                C     CALCULATE LOWER BOUND USING PENALTY FUNCTION BASED ON VEHICLES
ISN 0061        510 DO 350 NIC = 1,NUMD
ISN 0062            KEEP(NIC) = 1
ISN 0063            IF(LZ(NIC).LT.15)  KEEP(NIC) = 0
ISN 0065        350 CONTINUE
ISN 0066        355 TGO = 0.0
ISN 0067            IV = 0
ISN 0068            TG = 0.0
ISN 0069        354 DO 351 IX = 1,NV
ISN 0070            IF(IX.EQ.IV)  GO TO 351
ISN 0072            VGO = 0.0
ISN 0073        330 DO 90 J = 1,NM
ISN 0074            IF(YRLM(J).EQ.0.0) GO TO 90
ISN 0076            IF(MIN(J).EQ.IX)  GO TO 91
ISN 0078         90 CONTINUE
ISN 0079            GO TO 351
ISN 0080         91 PF = 0.0
ISN 0081            KTV = 0
ISN 0082            DO 341 M = 1,20
ISN 0083            IF(NONREC(IX,M).EQ.0)  GO TO 3415
ISN 0085            NO = NONREC(IX,M)
ISN 0086            IF(KEEP(NO).EQ.0)  GO TO 341
ISN 0088            VGO = VGO &DS(NO) & FLOAT(KI)*SUST(NO)
ISN 0089            KTV = 1
ISN 0090        341 CONTINUE
```

```
ISN 0091      3415 IF(KIV.EQ.0)  GO TO 351
ISN 0093           DO 331J= 1,NM
ISN 0094           IF(YRLH(J).EQ.0.0) GO TO 331
ISN 0096           IF(MIN(J).NE.IX)  GO TO 331
ISN 0098           PF = PF & COST(2,J) - COST(1,J)
ISN 0099       331 CONTINUE
ISN 0100           VGO = AMIN1(VGO,PF)
ISN 0101           IF(VGO.LT.TG)  GO TO 351
ISN 0103           IV = IX
ISN 0104           TG = AMAX1(TG,VGO)
ISN 0105       351 CONTINUE
ISN 0106           IF(TG.LT.GUESS*.01)  GO TO 37
ISN 0108           TGO = TG & TGO
ISN 0109           TG = 0.0
ISN 0110           DO 352 M = 1,20
ISN 0111           IF(NONREC(IV,M).EQ 0)  GO TO 354
ISN 0113           NO = NONREC(IV,M)
ISN 0114           KEEP(NO) = 0
ISN 0115       352 CONTINUE
ISN 0116        37 TGO = TGO & TG
ISN 0117        38 IF(KZ.EQ.0)  TDS(KX) = TDS(NX)
ISN 0119           IF(KZ.GT.0.AND.KX NE.NX)  TDS (KX) = TDS(NX)
                  1          & DS(NCOST) & FLOAT(IK-1)*KI-NYD(NCOST)&1)*SUST(NCOST)
ISN 0121        54 IF(KX.EQ.NX)
                  1TDS(NX) = TDS(NX) &
                  2          DS(NCOST) & FLOAT(LYD(NCOST)-NYD(NCOST)&1)*SUST(NCOST)
ISN 0123           DMIN = TGO & TDS(KX)
ISN 0124       507 Z(KX) = DMIN & W(KX)
ISN 0125           RETURN
ISN 0126           END

****** END OF COMPILATION ******
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,24576)              DEFAULT OPTION(S) USED
IEW0000      NAME HDX02LD(R)
IEW0461  UNPACK
```

CROSS REFERENCE TABLE

CONTROL SECTION NAME	ORIGIN	LENGTH	ENTRY NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
LBOUND	00	952								
SAVEA	958	2160								
SAVECZ	2AB8	3654								
SAVELB	6110	108								
SAVELZ	6218	F0								
SAVENV	6308	8								
SCRACH	6310	47A4								

LOCATION	REFERS TO SYMBOL	IN CONTROL SECTION	LOCATION	REFERS TO SYMBOL	IN CONTROL SECTION
130	SAVEA	SAVEA	134	SAVEA	SAVEA
138	SAVECZ	SAVECZ	13C	SAVECZ	SAVECZ
140	SAVECZ	SAVECZ	144	SAVELB	SAVELB
148	SAVELZ	SAVELZ	14C	SAVENV	SAVENV
150	SCRACH	SCRACH	154	SCRACH	SCRACH
158	SCRACH	SCRACH	15C	SCRACH	SCRACH
160	UNPACK	$UNRESOLVED	74	SAVELB	SAVELB
7C	SAVENV	SAVENV			

ENTRY ADDRESS 00
TOTAL LENGTH AAB8

```
****HDX02LD   NOW REPLACED IN DATA SET
```

DIAGNOSTIC MESSAGE DIRECTORY

```
IEW0461 WARNING - SYMBOL PRINTED IS AN UNRESOLVED EXTERNAL REFERENCE, NCAL WAS SPECIFIED.
```

```
FORTRAN IV G LEVEL 1, MOD 4            MAIN              DATE = 70104        16/48/39

            C MASTER PROGRAM FOR BUDGET SMOOTHING AND VEHICLE ASSIGNMENT INTEGRATION
            C
            C  A PROGRAM EITHER EQUALS A MISSION WITH LAUNCH SCHEDULE OR A
            C  DEVELOPMENT OR SUSTAINING PROGRAM OR A MISC. PROGRAM
            C  A MISSION MUST HAVE AT LEAST ONE AND NO MORE THAN 10 LAUNCH YEARS
            C
 0001               DOUBLE PRECISION NAME
 0002               INTEGER YDPL,FINISH
 0003               INTEGER*2 IVEH,LVS,NSTRST,LVARY,LVD,NYRSST,NLVP,LABEL,LNDATE,
                   1  LYR,LETT,MIN,IS,NONREC,NYD,LABS,LABF,LABI,MAT,VEH,LYD
            C  THE FOLLOWING STORAGE IS USED IN MASTER AND SMOOTH
 0004               COMMON/SAVES/N,M,IVEH(66),LVS(66),S(72),NSTRST(72),R(72),LVARY(66)
                   1 ,LVD(66),          NLVP(72),XSCH(10,66),LABEL(50),NCS,CS(90),
                   2  NPROG(90),KPROG(90),KODE(90),LNDATE(72)
            C  THE FOLLOWING STORAGE IS USED IN ASSIGN AND MASTER
 0005               COMMON/SAVEA/NM,YRLM(250),LYR(252),LETT(750),MIN(250),
                   1  DS(50),SUST(50),YD(50),IS(106),    NUMD,NONREC( 60,20),NYD(50),
                   2  LABS(40),LABF(30),LABI(40),                    RXD(12,50),MAT(50)
 0006               COMMON/SAVEA1/YDPL(56),IBY,NSFX(50),NRFX(50),NSYR(50)
            C  THE FOLLOWING STORAGE IS USED IN MASTER AND SMOOTH AND ASSIGN
 0007               COMMON/SAVEB/NAME(56), HITR,  ALPI(4, 60),C(72),SUS(72),
                   1  PLR(50),RDIST(56,4),RFIXD(12,72),NSTRFX(72),NYRSFX(72),TREF
 0008               COMMON/SAVEB1/NMIS,VEH(4,60),FINISH,RCOST(61),STG(40),NSPR,
                   1   NYRSST(72)
            C  STORAGE FOR SMOOTH ONLY THAT MUST BE SAVED
 0009               COMMON/PLSAVE/TITLE(10),FIXED(20),LEVEL(20),CNTRVL(20),
                   1  PHAX,PMIN,ACCL,EXT,ISTRT,IFIN,MAXITR,NCSTR
            C   THE FOLLOWING STORAGE IS USED IN ASSIGN AND CHOOZ
 0010               COMMON/SAVECZ/RECUR( 60,20,2),VNH(4,250),LYD(50),NEXT,GUESS,
                   1  LTR(50)
            C   STORAGE USED IN DECISN AND ASSIGN
 0011               COMMON/ASGN/IST(40),JST(30),KST(40),KODEH(50),YDS(40),YDF(30),
                   1   YDI(40)
 0012               COMMON/SAVEDC/KFLAG,NFAM,LSA(40),SNR(40),STS(40),NYS(40),FMNR(30),
                   1  KODEF(30),FMSUS(30),FAM(30),DINT(40),SINT(40)
            C  THE FOLLOWING STORAGE IS USED IN STGNUM
 0013               COMMON/SAVESM/IFLAG,STGYTR(40,20,2),RINTYR(40,20),NBY(40),NCI,
                   1      PLCINT(40),                NFML(40),NFS(40,4),NFMU(40),
                   2    MODE(40,3),SR(40,3),PLC(40,3),POJ(3),SRJ(3,3),RINT(40)
            C  THE FOLLOWING STORAGE IS USED IN CAPABL AND ASSIGN
 0014               COMMON/SAVECL/NOPT,IV,KNV,IG,      PLMD(50),KODEV(60),KODS(40),
                   2  IVEHA(50),VLR(50),WPR(50),VM(2,60),NPLS(50),MR(50),NRR(50)
            C  STORAGE USED IN CAPABL ONLY BUT SAVED FOR BATCHING
```

```
FORTRAN IV G LEVEL 1, MOD 4            MAIN              DATE = 70104        16/48/39

 0015               COMMON/SAVED/B1(60),B2(60),B3(60),B4(60),NVS(60),MRV(60),NRP(60)
 0016               COMMON/SAVELZ/LZ(60)
 0017               COMMON/SAVENV/NV,MYRS
 0018               COMMON/BATCH/KNSTG,KNFAM,KNCI,KNMIS,KNSP,KODESP(6)
 0019               COMMON/SAVESG/NSTG
 0020               COMMON/SCRACH/DUM1(4260),NPRO(90),KPRO(90),CSX(90),LABN(50),DUM(5)
            C  THE FOLLOWING STORAGE IS FOR MASTER USE ONLY
 0021               DIMENSION          PLD(72),PLS(72),PRGLV(4)
 0022               EQUIVALENCE  (SUS(1),PLS(1)),    (C(1),PLD(1))
 0023               DATA BLANK /1H /
 0024               DATA PRGLV /16HPROGRAM LEVEL   /
 0025               DATA ASTR /1H*/
 0026               DATA ZERO /1H0/
 0027               RCOST( 61) = BLANK
 0028               LYR(252) = 0
 0029             9 FINISH = 1
 0030               JFLAG = 0
 0031            10 CALL ASSIGN
 0032               IF(MYRS.EQ.0)  GO TO 99
 0033               IF(MYRS.EQ.100)  GO TO 9
 0034               IF(FINISH.GT.1) GO TO 12
 0035               NNH = NMIS + 1
 0036               DO 11 I = NNH,72
 0037            11 PLR(I) = 0.0
 0038               NNH = NNH + NSPR
 0039               TREF = 1900.0 + TREF
 0040               CALL CLEAR (R(1),R(72))
 0041            12 DO 13 I = 1,66
 0042               LVARY(I) = 0
 0043               LVD(I) = 0
 0044               DO 131 J = 1,10
 0045           131 XSCH(J,I) = 0.0
 0046            13 CONTINUE
 0047               DO 132 I = 1,72
 0048           132 NLVP(I) = 0
 0049               DO 133 I = 1,50
 0050           133 LABEL(I) = 0
 0051               DO 134 I = 1,NMIS
 0052           134 NYRSST(I) = 0
 0053               DO 14 I = 1, NUMD
 0054            14 LABN(I) = 0
            C
            C  CALCULATE VARIABLES FOR SMOOTH FROM MISSION DATA
```

```
(FORTRAN IV G LEVEL 1, MOD 4            MAIN              DATE = 70104         16/48/39

            C
0055              M = 1
0056              DO 120 K = 1,NM
0057              IF(MIN(K).EQ.0)  GO TO 120
0058              I = LYR(K)
0059              J = LETT(K)
0060              IF(J.EQ.LETT(K-1))  GO TO 105
0061              IF(FINISH.GT.1)  GO TO 104
0062              S(J) = IS(J)
0063              R(J) = YDPL(J)
0064            104 LVARY(J) = M
0065              NSTRST(J) =  INT(2.0*R(J)/3.0 + .999)
0066              IF(R(J).EQ.0)  NSTRST(J)=1
0067              GO TO 108
0068            105 L1 = LVARY(J)
0069              M0 = M-1
0070              DO 106 L = L1,M0
0071              IF(MIN(K).NE.IVEH(L))  GO TO 106
0072              M1 = L
0073              GO TO 110
0074            106 CONTINUE
0075            108 IVEH(M) = MIN(K)
0076              LVS(M) = I - IS(J) + 1900 + IBY
0077              NLVP(J) = NLVP(J) + 1
0078              M1 = M
0079              M = M + 1
0080            110 M3 = IS(J)
0081              IF(SUS(J).LE..001)  GO TO 111
0082              NX = NYRSST(J)
0083              M4 = NSTRST(J)
0084              M5 = NSYR(J)
0085              NYRSST(J) = MAXO(NX,I - M4 - M3 + 1900 + IBY + M5)
0086            111 M2 = LVS(M1)
0087              XSCH(I - M2 - M3 + 1900 + IBY + 1,M1) = YRLM(K)
0088              NX = LVD(M1)
0089              LVD(M1) = MAXO(NX,I-M2-M3+1900+IBY+1)
0090            120 CONTINUE
0091              M = M - 1
0092              NCS = 0
0093              N = NMIS
0094              IF(NSPR.EQ.0)  GO TO 170
0095              DO 150 I = 1,NSPR
0096              N = N + 1


FORTRAN IV G LEVEL 1, MOD 4             MAIN              DATE = 70104         16/48/39              PAGE 0004

0097              IF(FINISH.GT.1)  GO TO  140
0098              S(N) = IS(N)
0099              R(N) = YDPL(N)
0100            140 NSTRST(N) = INT(2.0*R(N)/3.0 + .999)
0101              IF(R(N).EQ.0)  NSTRST(N) = 1
0102            150 CONTINUE
            C
            C  CONTINUE TO CALCULATE VARIABLES FOR SMOOTH USING DEV. AND SUST. COSTS
            C
0103            170 IF(NUMD.EQ.0)  GO TO 260
0104              DO 210 I = 1,NUMD
0105              IF(LZ(I).EQ.0)  GO TO 210
0106              N = N + 1
0107              LABEL(N-NMIS-NSPR) = I
0108              LABN(I) = N
0109              C(N) = DS(I)
0110              L = MAT(I)
0111              NYRSFX(N) = 0
0112              IF(L.LT.-100)  J = LABT(-L-100)
0113              IF(L.LT.0.AND.L.GE.-100)  J = LABF(-L)
0114              IF(L.GT.0)  J = LABS(L)
0115              IF(J.EQ.0)  GO TO 206
0116              DO 205  K = 1,12
0117              RFIXD(K,N) = RXD(K,J)
0118            205 C(N) = C(N) - RXD(K,J)
0119              NYRSFX(N) = NSFX(J)
0120              NSTRFX(N) = NRFX(J)
0121            206 S(N) = IS(I) + NMIS + NSPR)
0122              R(N) = YD(I)
0123              SUS(N) = SUST(I)
0124              NSTRST(N) = INT(2.0*R(N)/3.0 + .999)
0125              IF(R(N).EQ.0)  NSTRST(N)=1
0126              NYRSST(N) = LZ(I) - NYD(I) + INT(YD(I)) - NSTRST(N) + 1
0127              IF(SUS(N).LT..0001)  NYRSST(N) = 0
0128            210 CONTINUE
            C
            C  CALCULATE DEVELOPMENT CONSTRAINTS ON MISSION PROGRAMS
0129              DO 250  K = 1,NM
0130              IF(MIN(K).EQ.0)   GO TO 250
0131              J = LETT(K)
0132              IF(NLVP(J).EQ.1.AND.J.EQ.LETT(K-1))  GO TO 250
0133              IV = MIN(K)
0134              DO 211 I = 1,10
```

```
FORTRAN IV G LEVEL 1, MOD 4            MAIN              DATE = 70104        16/48/39            PAGE 0005

0135              IF(J.NE.LETT(K-I)) GO TO 215
0136              IF (IV.EQ.MIN(K-I)) GO TO 250
0137          211 CONTINUE
0138          215 X = LYR(K) - LYR(K-I+1)
0139              DO 220 K1 = 1,20
0140              IF(NONREC(IV,K1).EQ.0)  GO TO 250
0141              NO = NONREC(IV,K1)
0142              J1 = LARN(NO)
0143              IF(C(J1).LT..0001)  GO TO 220
0144              NCS = NCS + 1
0145              NPRO(NCS) = J1
0146              KPRO (NCS) = J
0147              CSX(NCS) = -1.0 -X
0148              IF(NCS.GE.90)  GO TO 255
0149          220 CONTINUE
0150          250 CONTINUE
0151              GO TO 260
0152          255 WRITE(6,1002)
0153         1002 FORMAT(52HONUMBER OF DEVELOPMENT CONSTRAINTS HAS BEEN EXCEEDED)
             C
0154          260 CALL SMOOTH(PRGLV,ASTR,BLANK,ZERO)
0155              IF(NCSTR.GT.90)  GO TO 9
0156              IF(FINISH.EQ.MITR + 1.AND.JFLAG.EQ.1)  GO TO 401
0157              IF(FINISH.EQ.MITR + 1)  GO TO 402
0158              IF(FINISH.EQ.MITR)   JFLAG = 1
             C
             C  CALCULATE VARIABLES FOR ASSIGN FROM SMOOTH VARIABLES
             C
0159              MXRS = MYRS
0160              DO 300  K = 1,NM
0161              I = LYR(K)
0162              J = LETT(K)
0163              IF(J.EQ LETT(K-I))  GO TO 305
0164              IS(J) = S(J)
0165              IX = IS(J) + LNDATE(J) - 1900 - IBY
0166              IDIFF = IX - I
0167          305 IF (IDIFF.EQ.0)  GO TO 300
0168              MYRS = MAXO(MYRS,IDIFF + I)
0169              LYR(K) = I + IDIFF
0170          300 CONTINUE
0171              IF(N.EQ.NMIS+NSPR)  GO TO 10
0172              DO 350 I = NNH , N
0173              J = LABEL(I-NMIS-NSPR)


FORTRAN IV G LEVEL 1, MOD 4            MAIN              DATE = 70104        16/48/39            PAGE 0006

0174              DS(J) = C(I)
0175              L = MAT(J)
0176              IF(L.LT.-100)  J1 = LARI(-L-100)
0177              IF(L.LT.0.AND.L.GE.-100)  J1 = LARF(-L)
0178              IF(L.GT.0)  J1 = LARS(L)
0179              IF(J1.EQ.0)  GO TO 320
0180              DO 310 K = 1,12
0181          310 DS(J) = DS(J) + RFIXD(K,I)
0182              NRFX(J1) = NSTRFX(I)
0183          320 SUST(J) = SUS(I)
0184              YD(J) = R(I)
0185              NYD(J) = INT(S(I) + R(I)) - 1900 - IBY
0186              IF(NYD(J).LE.0)   NYD(J) = 1
0187              IS(J + NMIS + NSPR) = S(I)
0188          350 CONTINUE
0189              DO 349 I = 1,NUMD
0190              IF(LYD(I).EQ.MXRS)   LYD(I) = MYRS
0191          349 CONTINUE
0192              GO TO 10
0193          401 WRITE(6,500)
0194              GO TO 9
0195          402 WRITE(6,501)
0196              GO TO 9
0197          500 FORMAT (28HONUMBER OF ITERATIONS = MITR)
0198          501 FORMAT ( 65HOOPTIMUM ASSIGNMENT WITHIN BUDGET CONSTRAINTS HAS BEEN
                   1 DETERMINED)
0199           99 STOP
0200              END
```

```
FORTRAN IV G LEVEL 1, MOD 4          MAIN          DATE = 70104          16/48/39          PAGE 0007

   TOTAL MEMORY REQUIREMENTS 001528 BYTES
```

```
 F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,NCAL,MAP
         VARIABLE OPTIONS USED - SIZE=(126976,24576)                    DEFAULT OPTION(S) USED
IEW0000      NAME MOX02MS(R)
IEW0461  ASSIGN
IEW0461  CLEAR
IEW0461  IBCOM#
IEW0461  SMOOTH
IEW0461  MAXO
```

MODULE MAP

CONTROL SECTION			ENTRY							
NAME	ORIGIN	LENGTH	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
MAIN	00	1528								
SAVES	1528	1660								
SAVEA	2B88	2160								
SAVEA1	4CE8	33C								
SAVEB	5028	1BD0								
SAVEB1	6BF8	410								
PLSAVE	7008	138								
SAVECZ	7140	3654								
ASGN	A798	438								
SAVEDC	A8D0	5A8								
SAVESM	B178	30F8								
SAVECL	E270	8F8								
SAVED	EB68	690								
SAVELZ	F1F8	F0								
SAVENV	F2E8	8								
BATCH	F2F0	2C								
SAVESG	F320	4								
SCRACH	F328	47A4								

```
 ENTRY ADDRESS       00
 TOTAL LENGTH     13AD0

****MOX02MS   NOW REPLACED IN DATA SET
```

DIAGNOSTIC MESSAGE DIRECTORY

EXTERNAL SYMBOL DICTIONARY

```
SYMBOL   TYPE ID  ADDR   LENGTH LD ID

PACK      SD  01 000000 0000E8
UNPACK    LD     000052         01
ITEM      LD     000096         01
```

```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                  F01FEB69   4/14/70

                                    1 *       SUBROUTINE  PACK ( L, M, I, N )
                                    2 *
                                    3 *            THIS ROUTINE PACKS I WORDS IN THE L-ARRAY  TO THE
                                    4 *            ARRAY M.  DATA ITEMS L ARE TRUNCATED ON THE LEFT
                                    5 *            AND ONLY THE N LOW ORDER BITS ARE RETAINED.
                                    6 *            PACKED DATA IN M IS LEFT JUSTIFIED WITH 32/N ITEMS
                                    7 *            PER WORD.
                                    8 *
000000                              9 PACK    CSECT
000000                             10         USING  *,15          USE REG 15 FOR BASE
000000 9027 D01C          0001C    11         STM    2,7,28(13)    SAVE REGS
000004 9825 1000          00000    12         LM     2,5,0(1)      LOAD ADDRESSES OF ARGUMENTS
000008 5844 0000          00000    13         L      4,0(4)        I TO REG 4 - NO. OF ITEMS TO BE PACKED
00000C 5875 0000          00000    14         L      7,0(5)        N TO REG 7 - NO. OF BITS/ITEM
000010 4270 F029          00029    15         STC    7,SHIFT+3     MODIFY SHIFT INST WITH NO. OF BITS
000014 1367                        16         LCR    6,7           NO. OF BITS SHIFT FOR DECREMENT
000016 0670                        17         BCTR   7,0           N-1 FOR COMPARAND
000018 1B11                        18         SR     1,1           ZERO REG 1
00001A 4150 0020          00020    19 WORD    LA     5,32          LOAD A 32 TO REG 5 FOR COUNT
00001E 5013 0000          00000    20         ST     1,0(3)        ZERO STORAGE AREA
000022 5802 0000          00000    21 LOOP    L      0,0(2)        LOAD DATA TO REG 0
000026 8C00 0000          00000    22 SHIFT   SRDL   0,0           SHIFT DATA TO REG 1
00002A 1B00                        23         SR     0,0           TRUNCATE ON LEFT FOR MOD 2**N
00002C 8D00 5000          00000    24         SLDL   0,0(5)        SHIFT BACK TO PROPER POSITION
000030 5603 0000          00000    25         O      0,0(3)        OR PACKED WORD TO REG 0
000034 5003 0000          00000    26         ST     0,0(3)        STORE BACK TO PACKED AREA
000038 4122 0004          00004    27         LA     2,4(2)        INCREMENT DATA ADDRESS
00003C 4640 F046          00046    28         BCT    4,NEXT        COUNT DOWN ON NO. OF ITEMS
000040 9827 D01C          0001C    29         LM     2,7,28(13)    RESTORE REGS
000044 07FE                        30         BR     14            RETURN
000046 8656 F022          00022    31 NEXT    BXH    5,6,LOOP      BRANCH BACK IF SPACE LEFT
00004A 4133 0004          00004    32         LA     3,4(3)        OTHERWISE INCREMENT STORAGE ADDRESS
00004E 47F0 F01A          0001A    33         B      WORD          AND CONTINUE
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                  FO1FEB69  4/14/70

                                  35 *       SUBROUTINE UNPACK ( L, M, I, N )
                                  36 *
                                  37 *            THIS ROUTINE UNPACKS I WORDS OF DATA FROM THE M
                                  38 *            ARRAY TO THE L ARRAY.  WORDS IN L ARE ZEROED AND N
                                  39 *            BITS ARE PLACED RIGHT JUSTIFIED FROM THE PACKED
                                  40 *            ARRAY M.
                                  41 *
                                  42         ENTRY  UNPACK
000052                            43         USING  *,15          USE REG 15 FOR BASE
000052 9027 D01C          0001C   44 UNPACK  STM    2,7,28(13)    SAVE REGS
000056 9825 1000          00000   45         LM     2,5,0(1)      LOAD ADDRESSES OF ARGUMENTS
00005A 5844 0000          00000   46         L      4,0(4)        I TO REG 4 - NO. OF ITEMS TO BE PACKED
00005E 5875 0000          00000   47         L      7,0(5)        N TO REG 7 - NO. OF BITS/ITEM
000062 4270 F025          00077   48         STC    7,LEFT+3      MODIFY SHIFT INST WITH NO. OF BITS
000066 1367                       49         LCR    6,7           NO. OF BITS SHIFT FOR DECREMENT
000068 0670                       50         BCTR   7,0           N-1 FOR COMPARAND
00006A 4150 0020          00020   51 DATA    LA     5,32          LOAD A 32 TO REG 5 FOR COUNT
00006E 5813 0000          00000   52         L      1,0(3)        LOAD PACKED DATA TO REG 1
000072 1800                       53 BACK    SR     0,0           ZERO REG 0
000074 8000 0000          00000   54 LEFT    SLDL   0,0           SHIFT N BITS TO REG 0
000078 5002 0000          00000   55         ST     0,0(2)        STORE IN L
00007C 4122 0004          00004   56         LA     2,4(2)        INCREMENT STORAGE ADDRESS
000080 4640 F038          0008A   57         BCT    4,MORE        COUNT DOWN ON NO. OF ITEMS
000084 9827 D01C          0001C   58         LM     2,7,28(13)    RESTORE REGS
000088 07FE                       59         BR     14            RETURN
00008A 8656 F020          00072   60 MORE    BXH    5,6,BACK      BRANCH BACK IF MORE DATA
00008E 4133 0004          00004   61         LA     3,4(3)        OTHERWISE INCREMENT DATA ADDRESS
000092 47F0 F018          0006A   62         B      DATA          AND CONTINUE
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                  FO1FEB69  4/14/70

                                  64 *       FUNCTION  ITEM ( M, I, N )
                                  65 *
                                  66 *            THIS ROUTINE RETRIEVES THE I TH ITEM FROM THE PACKED
                                  67 *            ARRAY M.
                                  68 *
                                  69         ENTRY  ITEM
000096                            70         USING  *,15
000096 9025 D01C          0001C   71 ITEM    STM    2,5,28(13)    SAVE REGS
00009A 9824 1000          00000   72         LM     2,4,0(1)      LOAD ADDRESSES OF ARGS TO REGS 2,3,4.
00009E 5833 0000          00000   73         L      3,0(3)        LOAD I TO REG 3
0000A2 0630                       74         BCTR   3,0           SUBTRACT 1 FOR I-1
0000A4 4100 0020          00020   75         LA     0,32          LOAD A 32 TO REG 0
0000A8 8D00 0020          00020   76         SRDA   0,32          SHIFT TO REG 1
0000AC 5D04 0000          00000   77         D      0,0(4)        DIVIDE BY N
0000B0 5010 F04E          000E4   78         ST     1,TEMP        NO. OF ITEMS/WORD
0000B4 1803                       79         LR     0,3           I-1 TO REG 0
0000B6 8D00 0020          00020   80         SRDA   0,32          SHIFT TO REG 1
0000BA 5D00 F04E          000E4   81         D      0,TEMP        DIVIDE I-1 BY NO. ITEMS/WORD
0000BE 1851                       82         LR     5,1           SAVE IN REG 5 TO INDEX ARRAY M
0000C0 8B50 0002          00002   83         SLA    5,2           MULTIPLY BY 4
0000C4 8D00 0020          00020   84         SRDA   0,32          REMAINDER TO REG 1
0000C8 5C04 0000          00000   85         M      0,0(4)        MULTIPLY BY N
0000CC 1831                       86         LR     3,1           LOAD TO REG 3 TO INDEX SHIFT
0000CE 5815 2000          00000   87         L      1,0(5,2)      LOAD DATA FROM M ARRAY
0000D2 8910 3000          00000   88         SLL    1,0(3)        LEFT ADJUST PROPER ITEM
0000D6 5846 0000          00000   89         L      4,0(4)        LOAD N TO REG 4
0000DA 8D00 4000          00000   90         SLDL   0,0(4)        SHIFT N BITS TO REG 0
0000DE 9825 D01C          0001C   91         LM     2,5,28(13)    RESTORE REGS
0000E2 07FE                       92         BR     14            RETURN
0000E4                            93 TEMP    DS     F
                                  94         END
```

```
                                CROSS-REFERENCE

SYMBOL    LEN  VALUE  DEFN     REFERENCES                                    4/14/70

BACK      00002 000072 0053     0060
DATA      00004 00006A 0051     0062
ITEM      00004 000096 0071     0069
LEFT      00004 000074 0054     0048
LOOP      00004 000022 0021     0031
MORE      00004 00008A 0060     0057
NEXT      00004 000046 0031     0028
PACK      00001 000000 0009
SHIFT     00004 000026 0022     0015
TEMP      00004 0000E4 0093     0078  0081
UNPACK    00004 000052 0044     0042
WORD      00004 00001A 0019     0033


NO STATEMENTS FLAGGED IN THIS ASSEMBLY
  120 PRINTED LINES
```

```
F68-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,24576)          DEFAULT OPTION(S) USED
IEW0000      NAME MOX01PK(R)
****MOX01PK   NOW REPLACED IN DATA SET
```

```
(17)       F  OS/360  FORTRAN H                          DATE  70.104/16.49.36

         COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
   ISN 0002            SUBROUTINE REVALU
                     CTHIS SUBROUTINE RECALCULATES THE APPROPRIATE VALUES FOR RECURRING COSTS
   ISN 0003            DOUBLE PRECISION NAME
   ISN 0004            INTEGER H,PROG
   ISN 0005            INTEGER*2 IVEH,LVS,NSTRST,LVARY,LVD,NYRSST,NLVP,LABEL,LNDATE,VEH
   ISN 0006            COMMON/SCRACH/TOTAL(20),W(20),NSL(10),NOP(72),LVSF(66),D(20),
                      1  XLVSUM(20,50),XOUT(20),VOUT(20),RF(72),CF(72),SF(72),FLAGR(72),
                      2  FLAGS(72),PROG,IODD, KVEHI(50),RRR(20),IMAGE(830),YEAR(20),
                      3     Y(20),NSSF(72),NSRF(72),NSXF(72),NDSF(72),SUSTF(72),VNAM(66),
                      4NSCALE(5),RECUR(20,50),NSTRRC(72),NYRSRC(72),LNDF(72),KVEH( 60),
                      5  IERR,SKIP,NYFLAG,NPRO(90),KPRO(90),DUMS(145)
   ISN 0007            COMMON/SAVES/N,H,IVEH(66),LVS(66),S(72),NSTRST(72),R(72),LVARY(66)
                      1 ,LVD(66),          NLVP(72),XSCH(10,66),LABEL(50),NCS,CS(90),
                      2  NPROG(90),KPROG(90),KODE(90),LNDATE(72)
   ISN 0008            COMMON/SAVEB/NAME(56), MITR,  ALPI(4, 60),C(72),SUS(72),
                      1  PLR(50),RDIST(56,4),RFIXD(12,72),NSTRFX(72),NYRSFX(72),TREF
   ISN 0009            COMMON/SAVE81/NMIS,VEHI(4,60),FINISH,RCOST(61),STG(40),NSPR,
                      1   NYRSST(72)
   ISN 0010            L = PROG
   ISN 0011            NSTRRC(L) = 100
   ISN 0012            NYRSRC(L) = 0
   ISN 0013            LNDATE(L) = 100
   ISN 0014            IF (NLVP(L).EQ.0) GO TO 21
   ISN 0016            DO 34 LC = 1,20
   ISN 0017         34 RECUR(LC,L) = 0.0
   ISN 0018            IJ = NLVP(L)
   ISN 0019            H = LVARY(L)
   ISN 0020            IB = LVS(H)
   ISN 0021            IF (IB.LT.4) IB = 4
   ISN 0023            DO 38 K=1,IJ
   ISN 0024            IF (LVD(H).EQ.0) GO TO 38
   ISN 0026            IA = LVS(H)-3
   ISN 0027            IF (IA.LT.1) IA=1
   ISN 0029            IK = LVD(H)
   ISN 0030            ILV = IVEH(H)
   ISN 0031            DO 37 J=1,IK
                     C  RCST = VEH. RECURRING COST/YR. BY MISSION
   ISN 0032               RCST = XSCH(J,H)*RCOST(ILV)
                     C RCPL = PAYLOAD RECURRING COST/YR.
   ISN 0033            RCPL = XSCH(J,H)*PLR(L)
   ISN 0034            DO 36 I=1,4
   ISN 0035            II = LVS(H)-IB&I&J-1

   ISN 0036               IF (II.LT.1) II=1
                     C  DISTRIBUTE RECURRING COST BY YEAR
   ISN 0038            36 RECUR(II,L) = RECUR(II,L)&ALPI(I,ILV)*RCST & RDIST(L,I)*RCPL
   ISN 0039            37 CONTINUE
   ISN 0040               NYRSRC(L) = MAXO (NYRSRC(L),II)
   ISN 0041               NSTRRC(L) = MINO (NSTRRC(L),IA)
   ISN 0042               LSUB = LNDATE(L)
   ISN 0043               LVSUB = LVS(H)
   ISN 0044               LNDATE(L) = MINO(LSUB,LVSUB)
   ISN 0045            38 H = H & 1
                     C  NYRSRC & NSTRRC = 0 FOR DEVELOPMENT PROGRAMS
   ISN 0046            21 IF (NSTRRC(L).EQ.100) NSTRRC(L) = 0
   ISN 0048            99 RETURN
   ISN 0049               END

****** END OF COMPILATION ******
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
        VARIABLE OPTIONS USED - SIZE=(126976,24576)              DEFAULT OPTION(S) USED
IEW0000          NAME HDX02RU(R)

                                        CROSS REFERENCE TABLE

  CONTROL SECTION                    ENTRY
    NAME    ORIGIN  LENGTH             NAME   LOCATION     NAME   LOCATION     NAME   LOCATION     NAME   LOCATION
  REVALU        00     474
  SCRACH       478    47A4
  SAVES       4C20    1660
  SAVEB       6280    1BD0
  SAVEB1      7E50     410


  LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION          LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION
       F0           SCRACH          SCRACH                   F4           SCRACH          SCRACH
       F8           SCRACH          SCRACH                   FC           SCRACH          SCRACH
      100           SAVES           SAVES                   104           SAVES           SAVES
      108           SAVEB           SAVEB                   10C           SAVEB           SAVEB
      110           SAVEB1          SAVEB1
ENTRY ADDRESS       00
TOTAL LENGTH      8260

****HDX02RU   NOW REPLACED IN DATA SET
```

```
(17)       * OS/360  FORTRAN H                              DATE  70.105/09.09.44

         COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
   ISN 0002          SUBROUTINE SHIFT
                     C  THIS SUBROUTINE SHIFTS THE DEVELOPMENT START DATES AND DURATION IN ORDER
                     C    TO ACHIEVE A SMOOTHER LEVEL OF SPENDING
                     C
   ISN 0003           DOUBLE PRECISION NAME
   ISN 0004           LOGICAL SKIP,ACCL,EXT
   ISN 0005           INTEGER H,PROG
   ISN 0006           INTEGER*2 IVEH,LVS,NSTRST,LVARY,LVD,NYRSST,NLVP,LABEL,LNDATE,VEH
   ISN 0007           COMMON/PLSAVE/TITLE(10),FIXED(20),LEVEL(20),CNTRVL(20),
                     1  PMAX,PMIN,ACCL,EXT,ISTRT,IFIN,MAXITR,NCSTR
   ISN 0008           COMMON/SCRACH/TOTAL(20),W(20),NSL(10),NOP(72),LVSF(66),D(20),
                     1  XLVSUM(20,50),XOUT(20),VOUT(20),RF(72),CF(72),SF(72),FLAGR(72),
                     2  FLAGS(72),PROG,IODD, KVEHI(50),RRR(20),IMAGE(830),YEAR(20),
                     3    Y(20),NSSF(72),NSRF(72),NSXF(72),NDSF(72),SUSTF(72),VNAM(66),
                     4NSCALE(5),RECUR(20,50),NSTRRC(72),NYRSRC(72),LNDF(72),KVEH( 60),
                     5  IERR,SKIP,MYFLAG,NPRO(90),XPRO(90),DUMS(145)
   ISN 0009           COMMON/SAVES/N,M,IVEH(66),LVS(66),S(72),NSTRST(72),R(72),LVARY(66)
                     1 ,LVD(66),          NLVP(72),XSCH(10,66),LABEL(50),NCS,CS(90),
                     2  NPROG(90),KPROG(90),KODE(90),LNDATE(72)
   ISN 0010           COMMON/SAVEB/NAME(56), HITR,  ALPI(4, 60),C(72),SUS(72),
                     1  PLR(50),RDIST(56,4),RFIXD(12,72),NSTRFX(72),NYRSFX(72),TREF
   ISN 0011           COMMON/SAVEB1/NMIS,VEH(4,60),FINISH,RCOST(61),STG(40),NSPR,
                     1   NYRSST(72)
                     C
   ISN 0012        110 IODD = IODD & 1
   ISN 0013            GO TO (140,150,160,168,170,178,180),  IODD
   ISN 0014        140 STR = S(PROG)
   ISN 0015            S(PROG) = STR & 1.0
   ISN 0016        145 CALL CONSTR
   ISN 0017            IF (IFRR.NE.0) GO TO 110
   ISN 0019            GO TO 14
   ISN 0020        150 S(PROG) = STR - 1.0
   ISN 0021            IF (S(PROG).LT.TREF) GO TO 110
   ISN 0023            GO TO 145
   ISN 0024        160 S(PROG) =  STR
   ISN 0025            IF (R(PROG).EQ.0.) GO TO 190
   ISN 0027            CKR = R(PROG)
   ISN 0028            CKC = C(PROG)
   ISN 0029            CKS = SUS (PROG)
   ISN 0030            NDS = NYRSST(PROG)
   ISN 0031            NSS = NSTRST(PROG)
   ISN 0032            NSR = NSTRRC(PROG)
```

```
ISN 0033          NYRC = NYRSRC(PROG)
ISN 0034          NSX = NSTRFX(PROG)
ISN 0035          NSO = LNDATE(PROG)
ISN 0036          R(PROG) = CKR & 1.0
ISN 0037          NSTRST(PROG) = INT(2.0*R(PROG)/3.0 & .999)
ISN 0038          NSTRRC(PROG) = NSR & 1
ISN 0039          NSTRFX(PROG) = NSX & 1
ISN 0040          LNDATE(PROG) = NSO & 1
ISN 0041          IF (NLVP(PROG).EQ.0) GO TO 165
ISN 0043          IJ = NLVP(PROG)
ISN 0044          H = LVARY(PROG)
ISN 0045          DO 162 I=1,IJ
ISN 0046          NSL(I) = LVS(H)
ISN 0047          LVS(H) = NSL(I) & 1
ISN 0048        162 H = H & 1
ISN 0049          DO 34 LC = 1,20
ISN 0050         34 RRR(LC) = RECUR(LC,PROG)
ISN 0051        164 CALL REVALU
ISN 0052        165 CALL CONSTR
ISN 0053          IF (IERR.NE.0) GO TO 110
ISN 0055          IF(RF(PROG) - R(PROG)) 9010,9020,9030
                C  DEVELOPMENT DURATION IS STRETCHED OUT
ISN 0056         9010 C(PROG) = (.8 & .2*R(PROG)/RF(PROG)) * CF(PROG)
ISN 0057          GO TO 9050
ISN 0058         9020 C(PROG) = CF(PROG)
ISN 0059          GO TO 9050
                C  DEVELOPMENT DURATION IS ACCELERATED - CRASH PROGRAM
ISN 0060         9030 X = AINT (.5*RF(PROG) & .99)
ISN 0061          IF(R(PROG).LT.X)  R(PROG) = X
ISN 0063          C(PROG) = CF(PROG) * EXP ((1. -R(PROG)/RF(PROG))/ (R(PROG)/
                 1' RF(PROG) - .4))
ISN 0064         9050 IF (NYRSST(PROG).EQ.0) GO TO 14
ISN 0066          NYRSST(PROG) = NDSF(PROG) & 1 & NSS - NSTRST(PROG)
                C  THE FOLLOWING DEFN. OF NYRSST IS THE ORIGINAL
                C     NYRSST(PROG) = R(PROG)/RF(PROG)*FLOAT(NDSF(PROG))&.001
ISN 0067          X = NYRSST(PROG)
ISN 0068          SUS (PROG) = C(PROG)/CF(PROG)*SUSTF(PROG)/X*FLOAT(NDSF(PROG))
ISN 0069          GO TO 14
ISN 0070        168 IF(.NOT.EXT)  GO TO 110
ISN 0072          S(PROG) = STR - 1.0
ISN 0073          IF(S(PROG).LT.TREF)  GO TO 110
ISN 0075          IF(NLVP(PROG).EQ.0)  GO TO 165

ISN 0077          GO TO 164
ISN 0078        170 S(PROG) = STR
ISN 0079          IF (CKR.EQ.RF(PROG).AND..NOT.ACCL)  GO TO 180
ISN 0081          R(PROG) = CKR - 1.0
ISN 0082          NSTRST(PROG) = INT(2.0*R(PROG)/3.0 & .999)
ISN 0083          NSTRRC(PROG) = NSR - 1
ISN 0084          NSTRFX(PROG) = NSX - 1
ISN 0085          LNDATE(PROG) = NSO - 1
ISN 0086          IF (NLVP(PROG).EQ.0) GO TO 165
ISN 0088          IJ = NLVP(PROG)
ISN 0089          H = LVARY(PROG)
ISN 0090          DO 172 I=1,IJ
ISN 0091          LVS(H) = NSL(I) - 1
ISN 0092        172 H = H & 1
ISN 0093        175 GO TO 164
ISN 0094        178 IF(.NOT.EXT)  GO TO 110
ISN 0096          S(PROG) = STR & 1.0
ISN 0097          IF(NLVP(PROG).EQ.0)  GO TO 165
ISN 0099          GO TO 164
ISN 0100        180 S(PROG) = STR
ISN 0101          R(PROG) = CKR
ISN 0102          C(PROG) = CKC
ISN 0103          SUS (PROG) = CKS
ISN 0104          NYRSST(PROG) = NDS
ISN 0105          NSTRST(PROG) = NSS
ISN 0106          NSTRRC(PROG) = NSR
ISN 0107          NYRSRC(PROG) = NYRC
ISN 0108          NSTRFX(PROG) = NSX
ISN 0109          LNDATE(PROG) = NSO
ISN 0110          IF (NLVP(PROG).EQ.0) GO TO 190
ISN 0112          IJ = NLVP(PROG)
ISN 0113          H = LVARY(PROG)
ISN 0114          DO 182 I=1,IJ
ISN 0115          LVS(H) = NSL(I)
ISN 0116        182 H = H & 1
ISN 0117          DO 36 LC = 1,20
ISN 0118         36 RECUR(LC,PROG) = RRR(LC)
ISN 0119        190 MYFLAG = 0
ISN 0120          RETURN
ISN 0121         14 MYFLAG = 1
ISN 0122          RETURN
ISN 0123          END
```

```
 F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,24576)              DEFAULT OPTION(S) USED
IEW0000      NAME MOX02ST(R)
IEW0461  EXP
IEW0461  CONSTR
IEW0461  REVALU

                                        CROSS REFERENCE TABLE

  CONTROL SECTION                       ENTRY
    NAME     ORIGIN  LENGTH              NAME   LOCATION     NAME   LOCATION     NAME   LOCATION     NAME   LOCATION
  SHIFT          00     708
  PLSAVE        708     138
  SCRACH        910    47A4
  SAVES        5088    1660
  SAVEB        6718    1AD0
  SAVEB1       82E8     410


  LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION          LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION
      140           PLSAVE          PLSAVE                   144           SCRACH          SCRACH
      148           SCRACH          SCRACH                   14C           SCRACH          SCRACH
      150           SCRACH          SCRACH                   154           SAVES           SAVES
      158           SAVES           SAVES                    15C           SAVEB           SAVEB
      160           SAVEB           SAVEB                    164           SAVEB1          SAVEB1
      168           EXP             $UNRESOLVED              16C           CONSTR          $UNRESOLVED
      170           REVALU          $UNRESOLVED
 ENTRY ADDRESS      00
 TOTAL LENGTH     86F8

****MOX02ST   NOW REPLACED IN DATA SET

                                        DIAGNOSTIC MESSAGE DIRECTORY

   IEW0461 WARNING - SYMBOL PRINTED IS AN UNRESOLVED EXTERNAL REFERENCE, NCAL HAS SPECIFIED.
```

```
FORTRAN IV G LEVEL 1, MOD 4            SMOOTH            DATE = 70105          09/10/22

0001              SUBROUTINE SMOOTH(PRGLV,ASTR,BLANK,ZERO)
                  C    BUDGET SMOOTHING PROGRAM - R E. SLYE
                  C     MODIFIED BY C.J. GOLDEN
0002                  REAL LEVEL
0003                  DOUBLE PRECISION NAME
0004                  LOGICAL SKIP,OUT,ACCL,EXT
0005                  INTEGER PROG,H,FINISH
0006                  INTEGER*2 IVEH,LVS,NSTRST,LVARY,LVD,NYRSST,NLVP,LABEL,LNDATE,VEH
                  C   THE FOLLOWING STORAGE IS FOR SUBROUTINE USE ONLY BUT MUST BE SAVED
0007                  COMMON/PLSAVE/TITLE(10),FIXED(20),LEVEL(20),CNTRVL(20),
                     1  PMAX,PMIN,ACCL,EXT,ISTRT,IFIN,MAXITR,NCSTR
                  C  THE FOLLOWING STORAGE IS FOR SUBROUTINE USE ONLY & NEED NOT BE SAVED
0008                  COMMON/SCRACH/TOTAL(20),W(20),NSL(10),NOP(72),LVSF(66),D(20),
                     1  XLVSUM(20,50),XOUT(20),VOUT(20),RF(72),CF(72),SF(72),FLAGR(72),
                     2  FLAGS(72),PROG,IODD, KVEH(50),RRR(20),IMAGE(830),YEAR(20),
                     3   Y(20),NSSF(72),NSRF(72),NSXF(72),NDSF(72),SUSTF(72),VNAM(66),
                     4NSCALE(5),RECUR(20,50),NSTRRC(72),NYRSRC(72),LNDF(72),KVEH( 60),
                     5  IERR,SKIP,MYFLAG,NPRO(90),KPRO(90),CSX(90),DUMSX(55)
                  C  THE FOLLOWING STORAGE IS USED IN MASTER AND SMOOTH
0009                  COMMON/SAVES/N,M,IVEH(66),LVS(66),S(72),NSTRST(72),R(72),LVARY(66)
                     1 ,LVD(66),           NLVP(72),XSCH(10,66),LABEL(50),NCS,CS(90),
                     2  NPROG(90),KPROG(90),KODE(90),LNDATE(72)
                  C  THE FOLLOWING STORAGE IS USED IN MASTER AND SMOOTH AND ASSIGN
0010                  COMMON/SAVEB/NAME(56), HITR,  ALPI(4, 60),C(72),SUS(72),
                     1  PLR(50),RDIST(56,4),RFIXD(12,72),NSTRFX(72),NYRSFX(72),TREF
0011                  COMMON/SAVEB1/NMIS,VEH(4,60),FINISH,RCOST(61),STG(40),NSPR,
                     1   NYRSST(72)
0012                  COMMON/SAVENV/NV,NYRS
                  C  STORAGE USED IN DECISN AND ASSIGN / KODEH IS USED IN SMOOTH
0013                  COMMON/ASGN/IST(40),JST(30),KST(40),KODEH(50),YDS(40),YDF(30),
                     1   YDI(40)
0014                  COMMON/BATCH/KNSTG,KNFAH,KNC1,KNMIS,KNSP,KODESP(6)
0015                  DIMENSION PRGLV(4)
0016                  EQUIVALENCE (LS,LEVEL(1)),(LF,LEVEL(2))
0017                  IODD = 0
0018                  NSCALE(1) = 1
0019                  NSCALE(2) = 0
0020                  NSCALE(3) = 0
0021                  NSCALE(4) = 0
0022                  NSCALE(5) = 0
0023                  IF(FINISH.GT.1)  GO TO 18
0024                  PMAX = 5000.
0025                  PMIN = 1500.


FORTRAN IV G LEVEL 1, MOD 4            SMOOTH            DATE = 70105          09/10/22

                  C   ACCL = TRUE  IMPLIES USE ACCELERATION OPTION
0026                  ACCL = .TRUE.
                  C    EXT = TRUE   IMPLIES USE EXTENSION OPTION
0027                  EXT = .TRUE.
0028                  DO 5 I=1,10
0029                5 TITLE (I) = BLANK
0030                  DO 6  I = 1,20
0031                  CNTRVL(I) = BLANK
0032                6 FIXED(I) = 0.0
0033                  WRITE(6,399)
0034               16 CALL INPUT (6HTITLE , TITLE, 6HLEVEL ,LEVEL, 6HISTRT ,ISTRT,
                     X  6HIFIN  ,IFIN, 6HMAXITR,MAXITR,6HNCSTR ,NCSTR,6HNPROG ,NPROG,
                     X   6HKPROG ,KPROG,6HKODE  ,KODE,6HCS    ,CS,6HFIXED ,FIXED,
                     X   6HPMAX  ,PMAX, 6HPMIN  ,PMIN, 6HACCL  , ACCL, 6HEXT   ,EXT)
0035                  DO 8 I = 1,NCSTR
0036                  DO 2 I1 = 1,NMIS
0037                  IF(NPROG(I).EQ.KODEH(I1))  GO TO 3
0038                2 CONTINUE
0039                  I1 = 0
0040                3  NPROG(I) = I1
0041                  DO 1 I1 = 1,NMIS
0042                  IF(KPROG(I).EQ.KODEH(I1))  GO TO 4
0043                1 CONTINUE
0044                  I1 = 0
0045                4 KPROG(I) = I1
0046                8 CONTINUE
0047                  IF(NSPR.EQ.0)  GO TO 18
0048                  DO 510 I = 1,NCSTR
0049                  DO 502 I1 = 1,NSPR
0050                  IF(NPROG(I).EQ.KODESP(I1))  GO TO 503
0051              502 CONTINUE
0052                  I1 = -NMIS
0053              503 NPROG(I) = I1 + NMIS
0054                  DO 501 I1 = 1,NSPR
0055                  IF(KPROG(I).EQ.KODESP(I1))  GO TO 504
0056              501 CONTINUE
0057                  I1 = -NMIS
0058              504 KPROG(I) = I1 + NMIS
0059              510 CONTINUE
0060               18 IF(NCS.EQ.0)  GO TO 20
0061                  DO 19 I = 1,NCS
0062                  KODE(NCSTR + I) =  I1
0063                  CS(NCSTR + I)  = CSX(I)
```

```
FORTRAN IV G LEVEL 1, MOD 4          SMOOTH          DATE = 70105        09/10/22

0064              NPROG(NCSTR + I) = NPRO(I)
0065              KPROG(NCSTR + I) = KPRO(I)
0066           19 CONTINUE
0067              NCSTR = NCSTR + NCS
0068              IF(NCSTR.GT.90)  GO TO 1000
0069           20 CALL PLOT1 (NSCALE,7,5,15,6)
0070              T = 1.0
0071              DO 17 I=1,20
0072              YEAR(I) = TREF + T - 1.
0073              Y(I) = AMOD(YEAR(I),100.)
0074           17 T = T + 1.0
0075              WRITE (6,903)
0076              NLV = 0
0077              DO 33 I = 1,NV
0078              DO 31 J = 1,N
0079              IF(IVEH(J).NE.I)  GO TO 31
0080              NLV = NLV + 1
0081              KVEH(I) = NLV
0082              KVEHI(NLV) = I
0083              GO TO 32
0084           31 CONTINUE
0085              GO TO 33
0086           32 IA = VEH(1,I)
0087              IB = VEH(2,I)
0088              IC = VEH(3,I)
0089              ID = VEH(4,I)
0090              WRITE(6,905) I,STG(IA),STG(IB),STG(IC),STG(ID),RCOST(I)
0091           33 CONTINUE
0092              DO 335 I = 1,N
0093              NX = IVEH(I)
0094           335 CALL AFRMT (NX,VNAM(I))
0095              DO 39 PROG = 1,N
0096           39 CALL REVALU
0097           22 DO 23 I=1,N
0098              NOP(I) = 0
0099              RF(I) = R(I)
0100              SF(I) = S(I)
0101              CF(I) = C(I)
0102              SUSTF(I) = SUS (I)
0103              NDSF(I) = NYRSST(I)
0104              NSSF(I) = NSTRST(I)
0105              NSXF(I) = NSTRFX(I)
0106              LNDF(I) = LNDATE(I)

FORTRAN IV G LEVEL 1, MOD 4          SMOOTH          DATE = 70105        09/10/22

0107           23 NSRF(I) = NSTRRC(I)
0108              DO 24 I=1,N
0109           24 LVSF(I) = LVS(I)
0110              IF(FINISH.GT.1)  GO TO 21
0111              DO 25 I=ISTRT,IFIN
0112           25 CNTRVL(I) = ASTR
0113           21 IF(NCSTR - NCS.EQ.0)  GO TO 27
0114              IL = NCSTR - NCS
               C  NOP = 1 IF NO CHANGES ARE ALLOWED IN PROGRAM VARIABLES
0115              DO 26 I = 1,IL
0116              J = NPROG(I)
0117              IF (KODE(I).EQ.8) NOP(J) = 1
0118           26 CONTINUE
0119           27 OUT = .FALSE.
0120              DO 61 PROG = 1,N
0121              IF(NOP(PROG).EQ.1)  GO TO 61
0122              CALL CONSTR
0123              IF (IERR.NE.0) WRITE (6,91) PROG
0124           61 CONTINUE
0125           91 FORMAT('0WARNING - CONSTRAINT VIOLATED IN PROGRAM NUMBER',I3)
0126              DO 300 ITER = 1,MAXITR
0127              IPRNT = 0
0128              IF (ITER.EQ.MAXITR)  IPRNT = 1
0129              DO 200 PROG = 1,N
               C  IODD INDICATES WHAT TYPE OF CHANGE IS BEING MADE- IT = 0 INITIALLY
0130           14 SKIP = (IPRNT.EQ.0.AND.ITER.GT.1).OR.PROG.NE.1.OR.IODD.NE.0
0131              IF (SKIP.AND.NOP(PROG).EQ.1.AND.PROG.NE.1) GO TO 200
0132           15 DO 30 J=1,20
0133              TOTAL(J) = 0.
0134           30 W(J) = 0.
0135              IF (SKIP) GO TO 55
0136           40 XT = 0.
0137              ST = 0.
0138              DO 50 I=1,N
               C  FLAGR = * INDICATES A CHANGE IN DEVELOPMENT DURATION
0139              FLAGR(I) = BLANK
               C  FLAGS = * INDICATES A CHANGE IN START DATE OF DEVELOPMENT
0140              FLAGS(I) = BLANK
0141              IF (R(I).NE.RF(I)) FLAGR(I) = ASTR
0142              IF (S(I).NE.SF(I)) FLAGS(I) = ASTR
0143              X = NYRSST(I)
0144              ST = ST + SUS(I)*X
0145           50 XT = XT + C(I)
```

```
FORTRAN-IV G LEVEL 1, MOD 4            SMOOTH            DATE = 70105        09/10/22

0146              WRITE (6,90) TREF,TITLE
0147              WRITE (6,92)
0148              DO 53 I=1,N
0149              IF(I.GT.NMIS+NSPR)   GO TO 52
0150              K = NYRSRC(I)
0151              IF(I.GT.NMIS)  RECUR(1,I)  = 0.0
0152              WRITE (6,94) I,NAME(I),S(I),FLAGS(I),C(I),R(I),FLAGR(I),SUS (I),
                 X  NSTRST(I),NYRSST(I),NSTRRC(I),NYRSRC(I),(RECUR(J,I),J=1,K)
0153              GO TO 51
0154           52 WRITE(6,93) I,LABEL(I-NMIS-NSPR),S(I),FLAGS(I),C(I),R(I),FLAGR(I),
                 1 SUS(I),
                 X  NSTRST(I),NYRSST(I),NSTRRC(I),NYRSRC(I)
0155           51 K = NYRSFX(I)
0156              IF (K.EQ.0) GO TO 53
0157              WRITE (6,98) NSTRFX(I),NYRSFX(I),(RFIXD(J,I),J=1,K)
0158           53 CONTINUE
0159              WRITE (6,95) XT,ST
0160              IF (ITER.NE.1) WRITE (6,902)
0161              WRITE (6,96) (YEAR(I),I=1,20)
0162              WRITE (6,97)
0163              CALL PLOT2 (IMAGE,Y(16),Y(1),PMAX,PMIN)
0164              DO 54 I=1,1000
0165           54 XLVSUM(I,1) = 0.0
                C
0166              55 CALL TCOST(J,BLANK,ASTR)
                C
0167                 IF(LS.GT.20.OR.LS.LE.0)  GO TO 78
0168                 XL = 0.
                C  IF LEVEL(1) AND (2) ARE INPUT AS INTEGER YEARS, THEN THE PROGRAM
                C  TAKES THE AVERAGE SPENDING OVER THE PERIOD ENCOMPASSED BY THESE
                C  YEARS AS THE DESIRED BUDGET LEVEL
0169                 DO 76 I=LS,LF
0170              76 XL = XL+TOTAL(I)
0171                 XL = XL/FLOAT(LF-LS+1)
0172                 DO 77 I=1,20
0173              77 LEVEL(I) = XL
0174              78 IF (SKIP) GO TO 80
0175                 WRITE (6,99) (H(I),I=1,J)
0176                 WRITE (6,990) (FIXED(I),I=1,J)
0177                 WRITE (6,991) (TOTAL(I),I=1,J)
0178                 WRITE (6,993) CNTRVL
0179                 WRITE (6,992) (LEVEL(I),I=1,J)
0180                 CALL PLOT3 (ZERO,Y,LEVEL,IFIN)


FORTRAN IV G LEVEL 1, MOD 4            SMOOTH            DATE = 70105        09/10/22

0181                 CALL PLOT3 (ASTR,Y,TOTAL,J)
0182              80 SQD = 0
0183                 DO 100 I=ISTRT,IFIN
0184                 SQD = (TOTAL(I)-LEVEL(I))**2 + SQD
0185             100 CONTINUE
0186                 RMS = SQRT (SQD/FLOAT(IFIN-ISTRT+1))
                C   SAVEX = RMS VALUE AT BEGINNING OF ITERATION
0187                 IF (PROG.EQ.1.AND.IODD.EQ.0) SAVEX = RMS
0188                 IF (SKIP) GO TO 110
                C  RMS1 = VALUE OF RMS USING INPUT DATA
0189                 IF(ITER.EQ.1)  RMS1 = RMS
0190                 WRITE (6,199) RMS,YEAR(ISTRT),YEAR(IFIN)
0191                 WRITE (6,298) ITER
0192                 WRITE (6,399)
0193                 CALL PLOT4 (13,PRGLV)
0194                 WRITE (6,499)
0195             110 IF (OUT) GO TO 400
0196                 IF (ITER.EQ.MAXITR) GO TO 300
                C  SAVER = RMS VALUE AT BEGINNING OF PROGRAM CHANGE CONSIDERATIONS
0197                 IF(IODD.EQ.0)  SAVER = RMS
0198                 IF(RMS.LT.SAVER)  GO TO 190
                C
0199                 CALL SHIFT
                C
0200                 IF(MYFLAG.EQ.1)  GO TO 14
0201             190 IODD = 0
                C   SAVER  = VALUE OF RMS AT END OF ITERATION
0202                 IF(RMS.LT.SAVER.AND.PROG.EQ.N)  SAVER = RMS
0203             200 CONTINUE
0204                 IF (SAVEX.NE.SAVER) GO TO 300
0205                 IF (IPRNT.NE.0) GO TO 400
0206                 SKIP = .FALSE.
0207                 OUT = .TRUE.
0208                 GO TO 15
0209             300 CONTINUE
0210                 WRITE (6,390)
0211                 GO TO 403
0212             400 WRITE (6,299)
0213             403 WRITE (6,906) (YEAR(I),I=1,J)
0214                 WRITE (6,907)
0215                 DO 402 I=1,NLV
0216                 XLVTOT = 0.0
0217                 DO 401 II=1,J
```

```
FORTRAN IV G LEVEL 1, MOD 4            SMOOTH            DATE = 70105        09/10/22

0218          401 XLVTOT = XLVTOT + XLVSUM(II,I)
             C  XLVSUM(II,I) = NUMBER OF LAUNCHES IN YEAR II FOR VEH. KVEHI(I)
0219          402 WRITE (6,908) KVEHI(I),XLVTOT,(XLVSUM(II,I),II=1,J)
0220              IF(SAVER.LT.RMS1 - .4)  GO TO 404
0221              WRITE(6,909)
0222          909 FORMAT (46HOINPUT ASSIGNMENT IS OPTIMUM SMOOTHED SOLUTION)
0223              GO TO 7
0224          404 NNMI = NMIS + NSPR
0225              DO 9 I = 1,NNMI
0226              IF(ABS(S(I) + R(I) - SF(I) - RF(I)).GE..01) GO TO 13
0227              IF(NYRSST(I).NE.NDSF(I))  GO TO 13
0228              IF(NLVP(I).EQ.0)  GO TO 9
0229              IF(LNDATE(I).NE.LNDF(I))  GO TO 13
0230              IJ = NLVP(I)
0231              H = LVARY(I)
0232              DO 11 II = 1,IJ
0233              X = LVS(H)
0234              IF(ABS(S(I)+X-SF(I)-FLOAT(LVSF(H))).GE..01)  GO TO 13
0235           11 H = H + 1
0236            9 CONTINUE
0237              IF(N.EQ.NNMI)   GO TO 7
0238              NNMI = NNMI + 1
0239              DO 10 I = NNMI,N
0240              IF(ABS(S(I) + R(I) - SF(I) - RF(I)).GE.01)  GO TO 13
0241              IF(NYRSST(I).NE.NDSF(I))  GO TO 13
0242              IF(ABS(CF(I)-C(I)).GE..001)  GO TO 13
0243              IF(ABS(SUS(I) - SUSTF(I)).GE..001)  GO TO 13
0244           10 CONTINUE
0245            7 FINISH = MITR + 1
0246              GO TO 12
0247           13 FINISH = FINISH + 1
0248           12 NCSTR = NCSTR - NCS
0249              RETURN
0250         1000 WRITE(6,1001)
0251         1001 FORMAT(38HONUMBER OF CONSTRAINTS HAS EXCEEDED 90)
0252              RETURN
0253           90 FORMAT (1H1,15X,14HREFERENCE YEAR,F7.0,5X,10A4)
0254           92 FORMAT (78HOPN  NAME  START    DEVL  YRS   SUST  SS  SD  RS  RD  R
                 XECURRING OR FIXED ITEMS   /1H )
0255           93 FORMAT (I3,1X,4HDEV ,I2,F6.0,1X,A1,F7.0,F4.0,1X,A1,F5.0,4I4)
0256           94 FORMAT (I3,1X,A6,F6 0,1X,A1,F7.0,F4.0,1X,A1,F5.0,4I4,12F6.0)
0257           95 FORMAT (20X,4H----,8X,4H----/2X,5HTOTAL,12X,F6.0,F11.0)
0258           96 FORMAT (1H1,30X,47HTOTAL PROGRAM COSTS AND LAUNCH VEHICLE SCHEDULE


FORTRAN IV G LEVEL 1, MOD 4            SMOOTH            DATE = 70105        09/10/22

                  *  /6HOYEAR ,4X,20F6.0)
0259           97 FORMAT (8HOPROGRAM)
0260           98 FORMAT(44X,2I4,12F6.0)
0261           99 FORMAT (6HOSUM  ,4X,20F6.0)
0262          107 FORMAT (44X,2I4,12F6.1)
0263          990 FORMAT (6H FIXED,4X,20F6.0)
0264          991 FORMAT (6H TOTAL,4X,20F6 0)
0265          992 FORMAT (6H LEVEL,4X,20F6.0)
0266          993 FORMAT (8X,20(5X,A1))
0267          199 FORMAT (6HORMS =,F8.0,5X,18HSHOOTHING INTERVAL,F6.0,5H THRU,F6.0)
0268          298 FORMAT (10HOITERATION, I3)
0269          299 FORMAT (11X,11H FINAL CASE)
0270          390 FORMAT (11X,16H MAXITR EXCEEDED )
0271          399 FORMAT (1H1)
0272          499 FORMAT (1HO,50X,4HYEAR)
0273          902 FORMAT (1HO,40X,34H* INDICATES CHANGE FROM INPUT DATA)
0274          903 FORMAT (1H1,30X,19HRECURRING COST DATA  /1HO,8X,3HKEY,10X,
                  *  4HNAME,24X,9HUNIT COST  /1H )
0275          905 FORMAT (10X,I2,10X,4A4,10X,F10.2)
0276          906 FORMAT (1H1,30X,35HLAUNCH VEHICLE REQUIREMENTS BY YEAR  /
                  *  6HOYEAR ,6X,20F6.0)
0277          907 FORMAT (11HOLV    TOTAL)
0278          908 FORMAT (1X,I2,F8.2,20F6.1)
0279              END
```

TOTAL MEMORY REQUIREMENTS 00213C BYTES

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,NCAL,MAP
        VARIABLE OPTIONS USED - SIZE=(126976,24576)          DEFAULT OPTION(S) USED
IEW0000      NAME HOX028S(R)
IEW0461  IBCOM#
IEW0461  INPUT
IEW0461  PLOT1
IEW0461  AFRMT
IEW0461  REVALU
IEW0461  LUNSTN
IEW0461  PLOT2
IEW0461  TCOST
IEW0461  PLOT3
IEW0461  PLOT4
IEW0461  SHIFT
IEW0461  SQRT
```

MODULE MAP

CONTROL SECTION NAME	ORIGIN	LENGTH	ENTRY NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
SMOOTH	00	213C								
PLSAVE	2140	138								
SCRACH	2278	47A4								
SAVES	6A20	1660								
SAVEB	8080	1BD0								
SAVEB1	9C50	410								
SAVENV	A060	8								
ASGN	A068	438								
BATCH	A4A0	2C								

ENTRY ADDRESS 00
TOTAL LENGTH A4D0

****HOX028S NOW REPLACED IN DATA SET

DIAGNOSTIC MESSAGE DIRECTORY

IEW0461 WARNING - SYMBOL PRINTED IS AN UNRESOLVED EXTERNAL REFERENCE, NCAL WAS SPECIFIED.

```
(17)          OS/360  FORTRAN H                              DATE  70.105/09.11.01

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
ISN 0002            SUBROUTINE STGNUM
                  C DETERMINE NUMBER OF COMPONENTS ACTUALLY USED AND ASSOCIATED
                  C   RECURRING COSTS
                  C
ISN 0003            INTEGER*2 LYR,LETT,MIN,IS,NONREC,NYD,LABS,LABF,LABI,MAT,VEH,LYD,
                   1  NYRSST
                  C  THE FOLLOWING STORAGE IS USED IN STGNUM
ISN 0004            COMMON/SAVESH/IFLAG,STGYTR(40,20,2),RINTYR(40,20),NBY(40),NCI,
                   1      PLCINT(40),                    NFML(40),NFS(40,4),NFMU(40),
                   2    MODE(40,3),SR(40,3),PLC(40,3),POJ(3),SRJ(3,3),RINT(40)
                  C  THE FOLLOWING STORAGE IS USED IN ASSIGN AND MASTER
ISN 0005            COMMON/SAVEA/NM,YRLM(250),LYR(252),LETT(250),MIN(250),
                   1  DS(50),SUST(50),YD(50),IS(106),     NUMD,NONREC( 60,20),NYD(50),
                   2  LABS(40),LABF(30),LABI(40),                    RXD(12,50),MAT(50)
                  C  THE FOLLOWING STORAGE IS USED IN MASTER AND SMOOTH AND ASSIGN
ISN 0006            COMMON/SAVEB1/NMIS,VEH(4,60),FINISH,RCOST(61),STG(40),NSPR,
                   1   NYRSST(72)
                  C   THE FOLLOWING STORAGE IS USED IN ASSIGN AND CHOOZ
ISN 0007            COMMON/SAVEC2/RECUR( 60,20,2),VNM(4,250),LYD(50),NEXT,GUESS,
                   1  LTR(50)
ISN 0008            COMMON/SAVENV/NV,NYRS
ISN 0009            COMMON/SAVESG/NSTG
ISN 0010            COMMON/SAVENR/NUMBR,MXITR
                  C  THE FOLLOWING STORAGE NEED NOT BE SAVED
ISN 0011            COMMON/SCRACH/STGYHW(40,20),RINTMX(40,20),STGMAX(40,20,2),
                   1  DUMM(1385)
                  C
ISN 0012            IF(IFLAG.EQ.1)  GO TO 621
                  C     FIND MAX NUM OF EACH STAGE AND INTEGRATION POSSIBLE
ISN 0014            DO 661 J=1,NYRS
ISN 0015            STGYTR(1,J,1) = 0.0
ISN 0016          661 STGYTR(1,J,2) = 0.0
ISN 0017            DO 662 I = 1,NM
ISN 0018            JX = LETT(I)
ISN 0019            J = LYR(I)
ISN 0020            K = LTR(JX)
ISN 0021          662 STGYTR(1,J,K) = YRLM(I) & STGYTR(1,J,K)
ISN 0022            DO 668 J = 1,NYRS
ISN 0023            RINTYR(1,J) = STGYTR(1,J,1) & STGYTR(1,J,2)
ISN 0024          668 STGYHW(1,J) = RINTYR(1,J)
ISN 0025            DO 665 I=2,NSTG
ISN 0026            DO 665 J=1,NYRS
```

```
ISN 0027          STGYTR(I,J,1) = STGYTR(1,J,1)
ISN 0028          STGYTR(I,J,2) = STGYTR(1,J,2)
ISN 0029        665 STGYHW(I,J) = STGYHW(1,J)
ISN 0030          IF(NCI.EQ.0)  GO TO 673
ISN 0032          DO 9100 I = 2,NCI
ISN 0033          DO 9100 J = 1,MYRS
ISN 0034       9100 RINTYR(I,J) = RINTYR(1,J)
ISN 0035          GO TO 673
                C
                C DETERMINE NUMBER OF EACH STAGE AND INTEGRATION USED IN EAST ITERATION BY YEAR
ISN 0036        621 DO 623 I=1,NSTG
ISN 0037          DO 623 J=1,MYRS
ISN 0038          DO 623 K=1,2
ISN 0039          STGMAX(I,J,K) = STGYTR(I,J,K)
ISN 0040        623 STGYTR(I,J,K) = 0.0
ISN 0041          IF(NCI.EQ.0)  GO TO 9000
ISN 0043          DO 624 I=1,NCI
ISN 0044          DO 624 J=1,MYRS
ISN 0045          RINTMX(I,J) = RINTYR(I,J)
ISN 0046        624 RINTYR(I,J) = 0 0
ISN 0047       9000 DO 622 J=1,NM
ISN 0048          IF(YRLM(J).EQ.0.0) GO TO 622
ISN 0050          I = MIN(J)
ISN 0051          K = LYR(J)
ISN 0052          JX = LETT(J)
ISN 0053          ITR = LTR(JX)
ISN 0054          DO 625 MS = 1,4
ISN 0055          L = VEH(MS,I)
ISN 0056          IF (L.EQ.0) GO TO 622
ISN 0058          STGYTR(L,K,ITR) = STGYTR(L,K,ITR) & YRLM(J)
ISN 0059          IF (NCI.EQ.0) GO TO 625
ISN 0061          IF (MS.EQ.4) GO TO 625
ISN 0063          IF (VEH(MS&1,1).EQ.0) GO TO 625
ISN 0065          L1 = VEH(MS&1,I)
ISN 0066          DO 626 MI=1,NCI
ISN 0067          DO 627 KY=1,4
ISN 0068          IF(NFML(MI).NE.NFS(L,KY)) GO TO 627
ISN 0070          DO 628 KZ = 1,4
ISN 0071          IF (NFMU(MI).EQ.NFS(L1,KZ)) GO TO 629
ISN 0073        628 CONTINUE
ISN 0074        627 CONTINUE
ISN 0075          GO TO 626
```

```
ISN 0076        629 RINTYR(MI,K) = RINTYR(MI,K) & YRLM(J)
ISN 0077        626 CONTINUE
ISN 0078        625 CONTINUE
ISN 0079        622 CONTINUE
ISN 0080          IF(NCI.EQ.0)  GO TO 9001
ISN 0082          DO 691 I=1,NCI
ISN 0083          DO 691 J=1,MYRS
ISN 0084        691 IF(RINTYR(I,J).EQ.0.0)RINTYR(I,J) = RINTMX(I,J)
ISN 0086       9001 DO 676 I = 1,NSTG
ISN 0087          DO 676 J = 1,MYRS
ISN 0088          DO 676 K = 1,2
ISN 0089          IF(ABS(STGYTR(I,J,K) - STGMAX(I,J,K)).LT.0.001)  GO TO 676
ISN 0091          IF (STGYTR(I,J,K).GT.0.001) GO TO 677
ISN 0093        676 CONTINUE
ISN 0094          WRITE(6,4101)
ISN 0095       4101 FORMAT (1H0,4X, 40HTHE OPTIMUM SOLUTION HAS BEEN DETERMINED)
                C
ISN 0096        678 CALL VEHRC
                C
ISN 0097          IFLAG = 2
ISN 0098          RETURN
ISN 0099        677 IF(NUMBR.LE.MX(ITR)  GO TO 679
ISN 0101          WRITE(6,8005)
ISN 0102       8005 FORMAT(49H0MAXIMUM NUMBER OF ASSIGNMENT ITERATIONS EXCEEDED)
ISN 0103          GO TO 678
                C
                C     DETERMINE HARDWARE COSTS BY YEAR BASED ON LAST ITERATION
ISN 0104        679 DO 8013 I = 1,NSTG
ISN 0105          DO 8013 J = 1,MYRS
ISN 0106          IF(STGYTR(I,J,1).GT.0.0.OR .STGYTR(I,J,2).GT.0.0) GO TO 200
ISN 0108       8014 STGYTR(I,J,1) = STGMAX(I,J,1)
ISN 0109          STGYTR(I,J,2) = STGMAX(I,J,2)
ISN 0110        200 STGYHW(I,J) = STGYTR(I,J,1) & STGYTR(I,J,2)
ISN 0111       8016 IF(STGYTR(I,J,1).EQ.0.0) STGYTR(I,J,1) = STGMAX(I,J,1)
ISN 0113          IF (STGYTR(I,J,2).EQ.0.0) STGYTR(I,J,2) = STGMAX(I,J,2)
ISN 0115       8013 CONTINUE
                C
                C     MAKE ADJUSTMENT FOR BATCHING OVER YEARS
ISN 0116        673 DO 663 I = 1,NSTG
ISN 0117          IF (NBY(I).EQ.1) GO TO 663
ISN 0119          IA = 2
ISN 0120          IB = NBY(I)
```

```
ISN 0121          IC = 1
ISN 0122        666 DO 664 J= IA,IB
ISN 0123            IF (J.GT.MYRS) GO TO 700
ISN 0125        664 STGYHW(I,IC) = STGYHW(I,IC) & STGYHW(I,J)
ISN 0126        700 DO 667 J = IA,IB
ISN 0127            IF (J.GT.MYRS) GO TO 663
ISN 0129        667 STGYHW(I,J) = STGYHW(I,IC)
ISN 0130            IA = IA & NBY(I)
ISN 0131            IB = IB & NBY(I)
ISN 0132            IC = IC & NBY(I)
ISN 0133            GO TO 666
ISN 0134        663 CONTINUE
                C
                C  DETERMINE VEHICLE RECURRING COSTS BY YEAR AND LAUNCH SITE
ISN 0135            DO 632 I=1,NV
ISN 0136            DO 635 J=1,MYRS
ISN 0137            RECUR(I,J,1) = 0.0
ISN 0138        635 RECUR(I,J,2) = 0.0
ISN 0139            DO 633 MS = 1,4
ISN 0140            K = VEH(MS,I)
ISN 0141            IF (K.EQ 0) GO TO 632
ISN 0143       9004 DO 634 J= 1,MYRS
ISN 0144            IF(STGYHW(K,J).LT.0 001)  GO TO 634
ISN 0146            IF(MODE(K,1).NE.0)  GO TO 8015
ISN 0148            HDWR = SR(K,1)*STGYHW(K,J)**PLC(K,1)
ISN 0149            GO TO 8010
ISN 0150       8015 LX = MODE(K,1)
ISN 0151            IF(STGYHW(K,J).LE.POJ(LX))   HDWR = SRJ(LX,1)/STGYHW(K,J)
ISN 0153            IF(STGYHW(K,J).GT.POJ(LX))  HDWR = SRJ(LX,2)&SRJ(LX,3)/STGYHW(K,J)
ISN 0155       8010 DO 692 L = 1,2
ISN 0156            M = L & 1
ISN 0157            IF(STGYTR(K,J,L).EQ.0.0)  GO TO 692
ISN 0159            IF(MODE(K,M).NE.0)  GO TO 8011
ISN 0161            RECUR(I,J,L)=RECUR(I,J,L)&HDWR&SR(K,M)*STGYTR(K,J,L)**PLC(K,M)
ISN 0162            GO TO 692
ISN 0163       8011 LX = MODE(K,M)
ISN 0164            IF(STGYTR(K,J ,L).LE.POJ(LX)) RECUR(I,J ,L) = RECUR(I,J ,L)
                   1  & SRJ(LX,1)/STGYTR(K,J ,L) & HDWR
ISN 0166            IF (STGYTR(K,J ,L).GT.POJ(LX)) RECUR(I,J ,L) = RECUR(I,J ,L) &
                   1         SRJ(LX,2) & SRJ(LX,3)/STGYTR(K,J ,L) & HDWR
ISN 0168        692 CONTINUE
ISN 0169        634 CONTINUE

ISN 0170            IF (MS.EQ.4) GO TO 633
ISN 0172            IF(VEH(MS&1,I).EQ.0) GO TO 633
ISN 0174            K1 = VEH(MS&1,I)
ISN 0175            IF(NCI.EQ.0)  GO TO 633
ISN 0177            DO 636 L=1,NCI
ISN 0178            DO 637 KY=1,4
ISN 0179            IF (NFHL(L).NE.NFS(K,KY)) GO TO 637
ISN 0181            DO 638 KZ = 1,4
ISN 0182            IF (NFHU(L).EQ.NFS(K1,KZ)) GO TO 639
ISN 0184        638 CONTINUE
ISN 0185        637 CONTINUE
ISN 0186            GO TO 636
ISN 0187        639 DO 640 J = 1,MYRS
ISN 0188            IF(RINTYR(L,J).EQ.0.0) GO TO 640
ISN 0190            HDWR = RINT(L)*RINTYR(L,J)**PLCINT(L)
ISN 0191            RECUR(I,J,1) = RECUR(I,J,1) & HDWR
ISN 0192            RECUR(I,J,2) = RECUR(I,J,2) & HDWR
ISN 0193        640 CONTINUE
ISN 0194        636 CONTINUE
ISN 0195        633 CONTINUE
ISN 0196        632 CONTINUE
ISN 0197            IFLAG = 1
ISN 0198         99 RETURN
ISN 0199            END

****** END OF COMPILATION ******
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,245760)              DEFAULT OPTION(S) USED
IEW0000      NAME MOX02SM(R)
IEW0461  VEHRC
IEW0461  FRXPR=
IEW0461  IBCOM=
```

CROSS REFERENCE TABLE

CONTROL SECTION NAME	ORIGIN	LENGTH	ENTRY NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
STGNUM	00	F98								
SAVESM	F98	30F8								
SAVEA	4090	2160								
SAVEB1	61F0	410								
SAVECZ	6600	3654								
SAVENV	9C58	8								
SAVESG	9C60	4								
SAVENR	9C68	8								
SCRACH	9C70	4744								

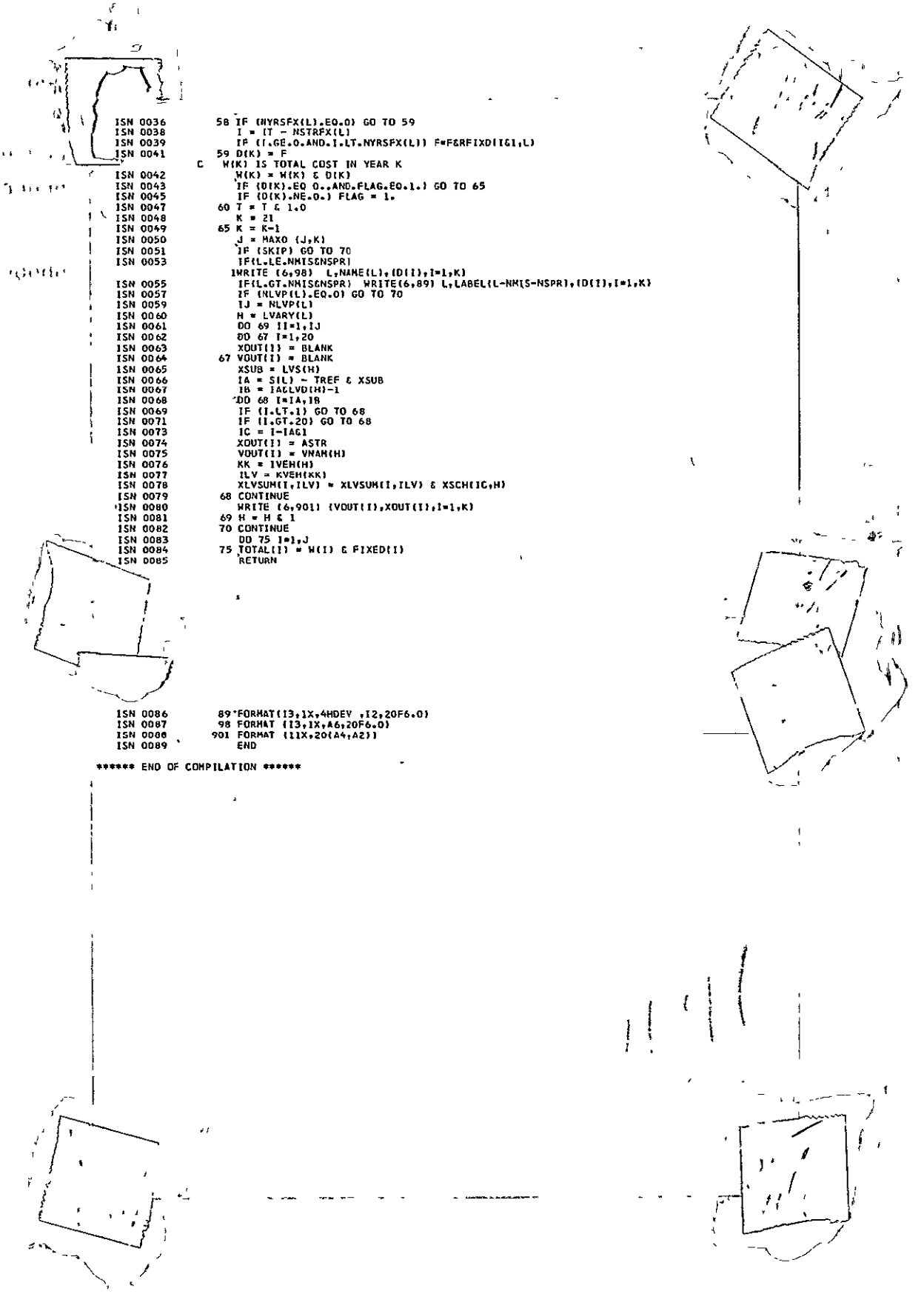
LOCATION	REFERS TO SYMBOL	IN CONTROL SECTION	LOCATION	REFERS TO SYMBOL	IN CONTROL SECTION
188	SAVESM	SAVESM	18C	SAVESM	SAVESM
190	SAVESM	SAVESM	194	SAVEA	SAVEA
198	SAVEA	SAVEA	19C	SAVEB1	SAVEB1
1A0	SAVECZ	SAVECZ	1A4	SAVECZ	SAVECZ
1A8	SAVECZ	SAVECZ	1AC	SAVENV	SAVENV
1B0	SAVESG	SAVESG	1B4	SAVENR	SAVENR
1B8	SCRACH	SCRACH	1BC	SCRACH	SCRACH
1C0	VEHRC	$UNRESOLVED	1C4	FRXPR=	$UNRESOLVED
1C8	IBCOM=	$UNRESOLVED			

ENTRY ADDRESS 00
TOTAL LENGTH E418

****MOX02SM NOW REPLACED IN DATA SET

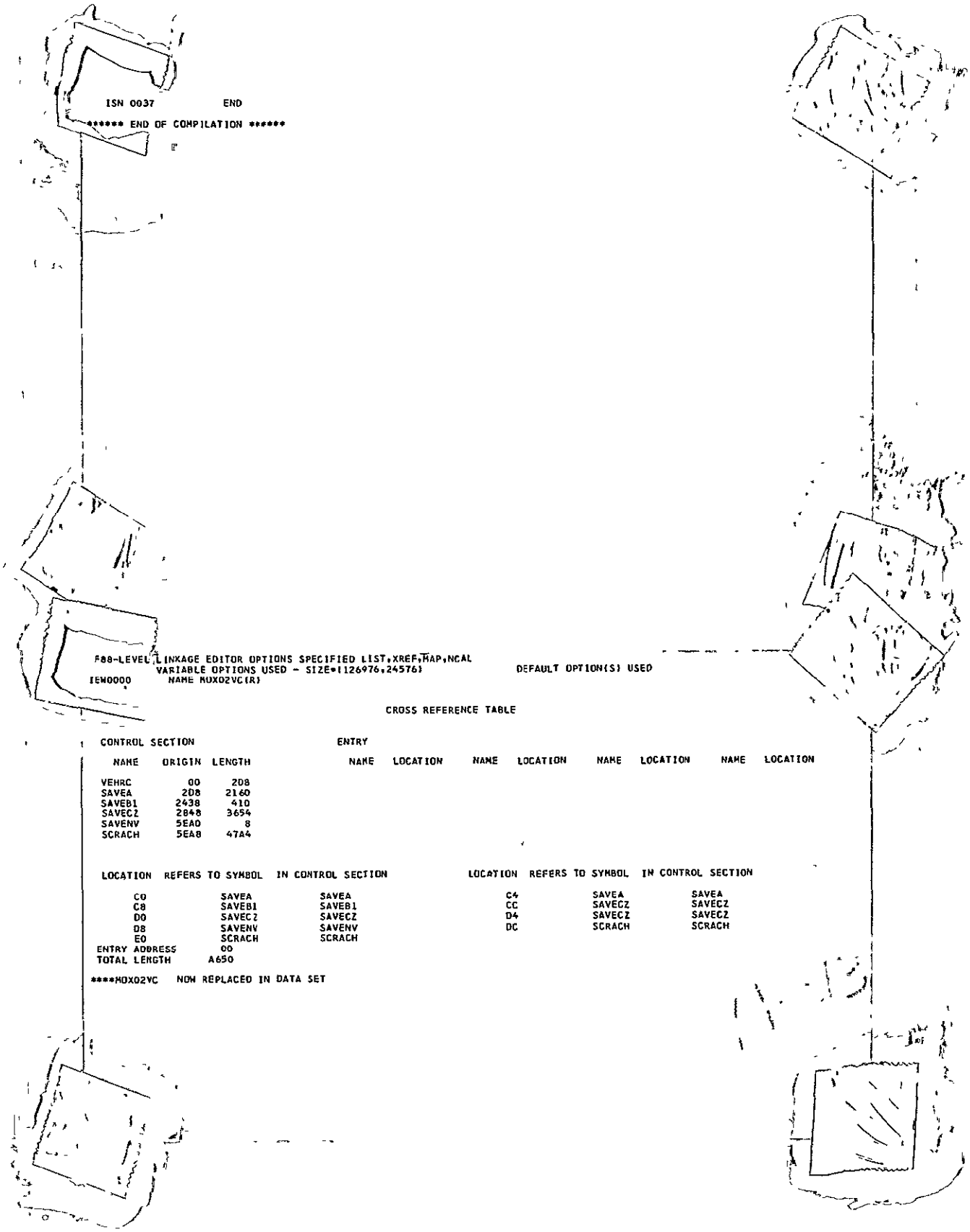
DIAGNOSTIC MESSAGE DIRECTORY

```
(17)          OS/360  FORTRAN H                    DATE  70.105/09.13 11

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
 ISN 0002            SUBROUTINE TCOST(J,BLANK,ASTR)
               C     CALCULATE TOTAL COSTS
 ISN 0003            DOUBLE PRECISION NAME
 ISN 0004            LOGICAL SKIP
 ISN 0005            INTEGER H
 ISN 0006            INTEGER*2 IVEH,LVS,NSTRST,LVARY,LVD,NYRSST,NLVP,LABEL,LNDATE,VEH
 ISN 0007            COMMON/PLSAVE/TITLE(10),FIXED(20),LEVEL(20),CNTRVL(20),
                    1  PMAX,PMIN,ACCL,EXT,ISTRT,IFIN,MAXITR,NCSTR
 ISN 0008            COMMON/SCRACH/TOTAL(20),W(20),NSL(10),NOP(72),LVSF(66),D(20),
                    1  XLVSUM(20,50),XOUT(20),VOUT(20),RF(72),CF(72),SF(72),FLAGR(72),
                    2  FLAGS(72),PROG,IODD, KVEHI(50),RRR(20),IMAGE(830),YEAR(20),
                    3     Y(20),NSSF(72),NSRF(72),NSXF(72),NDSF(72),SUSTF(72),VNAM(66),
                    4NSCALE(5),RECUR(20,50),NSTRRC(72),NYRSRC(72),LNDF(72),KVEH( 60),
                    5  IERR,SKIP,MYFLAG,NPRO(90),KPRO(90),DUMS(145)   )
 ISN 0009            COMMON/SAVES/N,M,IVEH(66),LVS(66),S(72),NSTRST(72),R(72),LVARY(66)
                    1 ,LVD(66),          NLVP(72),XSCH(10,66),LABEL(50),NCS,CS(90),
                    2  NPROC(90),KPROG(90),KODE(90),LNDATE(72)
 ISN 0010            COMMON/SAVEB/NAME(56), MITR,  ALP(4, 60),C(72),SUS(72),
                    1  PLR(50),RDIST(56,4),RFIXD(12,72),NSTRFX(72),NYRSFX(72),TREF
 ISN 0011            COMMON/SAVEB1/NMIS,VEH(4,60),FINISH,RCOST(61),STG(40),NSPR,
                    1   NYRSST(72)
               C
 ISN 0012         55 J = 0
 ISN 0013            DO 70 L=1,N
 ISN 0014            FLAG = 0.
 ISN 0015            T = 1.0
 ISN 0016            AYRS = R(L) & 1 0
 ISN 0017            DO 60 K=1,20
 ISN 0018            F = 0.
 ISN 0019            IT = T - S(L) & TREF
 ISN 0020            X = (T - S(L) & TREF) / AYRS
               C  X.LE.0 PROGRAM DEV. HASN'T STARTED YET - X.GE.1 PROGRAM DEV. IS OVER
 ISN 0021            IF (X.LE.0.) GO TO 59
 ISN 0023            IF (X.GE.1.) GO TO 56
               C  BETA DISTRIBUTION FOR C(L)
 ISN 0025            F = ((X*(1.-X))**2) * 30. * C(L) / AYRS
 ISN 0026         56 IF (NYRSST(L).EQ.0) GO TO 57
 ISN 0028            I = IT - NSTRST(L)
 ISN 0029            IF (I.GE.0.AND.I.LT.NYRSST(L)) F = F&SUS (L)
 ISN 0031         57 IF (NYRSRC(L).EQ.0) GO TO 58
 ISN 0033            I = IT - NSTRRC(L)
 ISN 0034            IF (I.GE.0 AND.I.LT.NYRSRC(L)) F=FGRECUR(I&I,L)
```



```
ISN 0036        58 IF (NYRSFX(L).EQ.0) GO TO 59
ISN 0038           I = IT - NSTRFX(L)
ISN 0039           IF (I.GE.0.AND.I.LT.NYRSFX(L)) F=F&RFIXD(I+1,L)
ISN 0041        59 D(K) = F
                C  W(K) IS TOTAL COST IN YEAR K
ISN 0042           W(K) = W(K) & D(K)
ISN 0043           IF (D(K).EQ 0..AND.FLAG.EQ.1.) GO TO 65
ISN 0045           IF (D(K).NE.0.) FLAG = 1.
ISN 0047        60 T = T & 1.0
ISN 0048           K = 21
ISN 0049        65 K = K-1
ISN 0050           J = MAXO (J,K)
ISN 0051           IF (SKIP) GO TO 70
ISN 0053           IF(L.LE.NMIS&NSPR)
                  1WRITE (6,98)  L,NAME(L),(D(I),I=1,K)
ISN 0055           IF(L.GT.NMIS&NSPR)  WRITE(6,89) L,LABEL(L-NM(S-NSPR),(D(I),I=1,K)
ISN 0057           IF (NLVP(L).EQ.0) GO TO 70
ISN 0059           IJ = NLVP(L)
ISN 0060           H = LVARY(L)
ISN 0061           DO 69 II=1,IJ
ISN 0062           DO 67 I=1,20
ISN 0063           XOUT(I) = BLANK
ISN 0064        67 VOUT(I) = BLANK
ISN 0065           XSUB = LVS(H)
ISN 0066           IA = SIL) - TREF & XSUB
ISN 0067           IB = IALLVD(H)-1
ISN 0068           DO 68 I=IA,IB
ISN 0069           IF (I.LT.1) GO TO 68
ISN 0071           IF (I.GT.20) GO TO 68
ISN 0073           IC = I-IAG1
ISN 0074           XOUT(I) = ASTR
ISN 0075           VOUT(I) = VNAM(H)
ISN 0076           KK = IVEH(H)
ISN 0077           ILV = KVEH(KK)
ISN 0078           XLVSUM(I,ILV) = XLVSUM(I,ILV) & XSCH(IC,H)
ISN 0079        68 CONTINUE
ISN 0080           WRITE (6,901) (VOUT(I),XOUT(I),I=1,K)
ISN 0081        69 H = H & 1
ISN 0082        70 CONTINUE
ISN 0083           DO 75 J=1,J
ISN 0084        75 TOTAL(I) = W(I) & FIXED(I)
ISN 0085           RETURN

ISN 0086        89 FORMAT(I3,1X,4HDEV ,I2,20F6.0)
ISN 0087        98 FORMAT (I3,1X,A6,20F6.0)
ISN 0088       901 FORMAT (11X,20(A4,A2))
ISN 0089           END

****** END OF COMPILATION ******
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
        VARIABLE OPTIONS USED - SIZE=(126976,24576)              DEFAULT OPTION(S) USED
IEW0000      NAME MOXOZTT(R)
IEW0461  IBCOM=

                                        CROSS REFERENCE TABLE

  CONTROL SECTION                    ENTRY

    NAME     ORIGIN  LENGTH           NAME   LOCATION     NAME   LOCATION     NAME   LOCATION     NAME   LOCATION

  TCOST         00     784
  PLSAVE       788     138
  SCRACH       8F0    47A4
  SAVES       5098    1660
  SAVEB       66F8    1B00
  SAVEB1      82C8     410


  LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION          LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION

      150           PLSAVE          PLSAVE                    154           SCRACH          SCRACH
      158           SCRACH          SCRACH                    15C           SCRACH          SCRACH
      160           SCRACH          SCRACH                    164           SAVES           SAVES
      168           SAVES           SAVES                     16C           SAVEB           SAVEB
      170           SAVEB           SAVEB                     174           SAVEB1          SAVEB1
      178           IBCOM=          $UNRESOLVED
ENTRY ADDRESS       00
TOTAL LENGTH      86D8

****MOXOZTT   NOW REPLACED IN DATA SET

                                        DIAGNOSTIC MESSAGE DIRECTORY

IEW0461 WARNING - SYMBOL PRINTFD IS AN UNRESOLVED EXTERNAL REFERENCE, NCAL WAS SPECIFIED.

(17)        OS/360  FORTRAN H                              DATE  70.105/09.13.41

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
   ISN 0002           SUBROUTINE VEHRC
                C     DETERMINE 'AVERAGE' RECURRING COST OF EACH VEHICLE
                C
   ISN 0003           INTEGER*2 LYR,LETT,MIN,IS,NONREC,NYD,LABS,LABF,LABI,MAT,VEH,LYD,
                     1  NYRSST
   ISN 0004           COMMON/SAVEA/NH,YRLH(250),LYR(252),LETT(250),MIN(250),
                      1  DS(50),SUST(50),YD(50),IS(106),     NUMD,NONREC( 60,20),NYD(50),
                      2  LABS(40),LABF(30),LABI(40),                    RXD(12,50),MAT(50)
   ISN 0005           COMMON/SAVEB1/NHIS,VEH(4,60),FIN(SH,RCUST(61),STG(40),NSPR,
                      1   NYRSST(72)
   ISN 0006           COMMON/SAVEC2/RECUR( 60,20,2),VNH(4,250),LYD(50),NEXT,GUESS,
                      1  LTR(50)
   ISN 0007           COMMON/SAVENV/NV,MYRS
                C    THE FOLLOWING STORAGE NEED NOT BE SAVED
   ISN 0008           COMMON/SCRACH/VYTR(60,20,2),DUMV(2185)
                C     COUNT NUMBER OF EACH VEHICLE USED BY YEAR AND TEST RANGE
   ISN 0009           DO 8032 I = 1,NV
   ISN 0010           DO 8032 J = 1,MYRS
   ISN 0011           DO 8032 K = 1,2
   ISN 0012      8032 VYTR(I,J,K) = 0.0
   ISN 0013           DO 8033 L = 1,NH
   ISN 0014           IF (YRLH(L).EQ.0.0) GO TO 8033
   ISN 0016           I = MIN(L)
   ISN 0017           J = LYR(L)
   ISN 0018           H = LETT(L)
   ISN 0019           K = LTR(H)
   ISN 0020           VYTR(I,J,K) = VYTR(I,J,K) & YRLH(L)
   ISN 0021      8033 CONTINUE
                C     DETERMINE 'AVERAGE' RECURRING COST OF EACH VEHICLE
   ISN 0022           DO 8034 I = 1,NV
   ISN 0023           RCOST(I) = 0.0
   ISN 0024           TVEH = 0.0
   ISN 0025           DO 8035 J = 1,MYRS
   ISN 0026           DO 8035 K = 1,2
   ISN 0027           IF (VYTR(I,J,K).EQ.0.0) GO TO 8035
   ISN 0029           RCOST(I) = RCOST(I) & VYTR(I,J,K) * RECUR(I,J,K)
   ISN 0030           TVEH = TVEH & VYTR(I,J,K)
   ISN 0031      8035 CONTINUE
   ISN 0032           IF (TVEH.EQ.0.0) GO TO 8034
   ISN 0034           RCOST(I) = RCOST(I)/TVEH
   ISN 0035      8034 CONTINUE
   ISN 0036        99 RETURN
```



```
ISN 0037          END
****** END OF COMPILATION ******
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,24576)          DEFAULT OPTION(S) USED
IEW0000     NAME HOX02VC(R)

                                CROSS REFERENCE TABLE

CONTROL SECTION                          ENTRY
  NAME    ORIGIN  LENGTH                 NAME  LOCATION    NAME  LOCATION    NAME  LOCATION    NAME  LOCATION

VEHRC         00     2D8
SAVEA        2D8    2160
SAVEB1      2438     410
SAVEC2      2848    3654
SAVENV      5EA0       8
SCRACH      5EA8    47A4


LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION          LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION

      C0          SAVEA         SAVEA                       C4          SAVEA         SAVEA
      C8          SAVEB1        SAVEB1                      CC          SAVEC2        SAVEC2
      D0          SAVEC2        SAVEC2                      D4          SAVEC2        SAVEC2
      D8          SAVENV        SAVENV                      DC          SCRACH        SCRACH
      E0          SCRACH        SCRACH
ENTRY ADDRESS     00
TOTAL LENGTH    A650

****HOX02VC   NOW REPLACED IN DATA SET
```