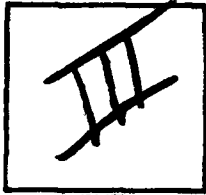


MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS - 1963 - A

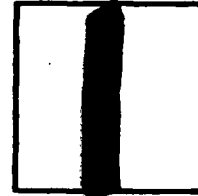
PHOTOGRAPH THIS SHEET

AD A 121986

DTIC ACCESSION NUMBER



LEVEL



INVENTORY

RIME: Vol. II, Section II

DOCUMENT IDENTIFICATION

Rept. No. TR-80-02-B

Contract F33600-78-C-0524

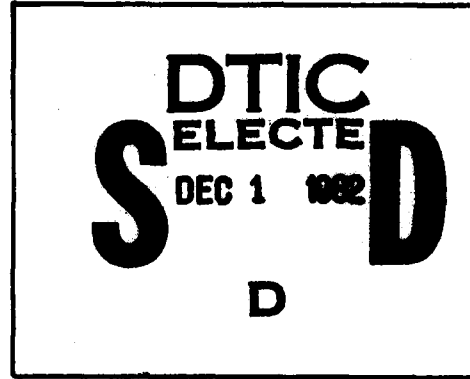
May 80

DISTRIBUTION STATEMENT A

Approved for public release
Distribution Unlimited

DISTRIBUTION STATEMENT

ACCESSION FOR	
NTIS	GRA&I
DTIC	TAB
UNANNOUNCED	
JUSTIFICATION	
BY	
DISTRIBUTION /	
AVAILABILITY CODES	
DIST	AVAIL AND/OR SPECIAL
A	



DATE ACCESSIONED



DISTRIBUTION STAMP

Empty rectangular box for date received in DTIC

DATE RECEIVED IN DTIC

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-DDA-2

RIME:

The Recoverable Item Management Evaluator:

Volume II, Section II

Stock Level Computation Programs

RIME:
The Recoverable Item Management Evaluator:
Volume II, Section II
Stock Level Computation Programs

by
W. Steven Demmy

May 1980

TR-80-02-B

DECISION SYSTEMS
2125 Crystal Marie Drive
Dayton, Ohio 45431

(513) 426-8515

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

82 11 30 026

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) RIME: The Recoverable Item Management Evaluator: Volume II: Program Listings and Narratives		5. TYPE OF REPORT & PERIOD COVERED INTERIM August 79-Jun. 80
7. AUTHOR(s) W. Steven Denny		6. PERFORMING ORG. REPORT NUMBER TR-80-02-D
9. PERFORMING ORGANIZATION NAME AND ADDRESS Decision Systems 3575 Charlene Drive Dayton, Ohio 45432		8. CONTRACT OR GRANT NUMBER(s) C-0524 F33600-78-D-0214
11. CONTROLLING OFFICE NAME AND ADDRESS 2750th ABW/PMA BLDG 1, Area C Wright-Patterson AFB, Ohio		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 23041A5
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE May 1980
		13. NUMBER OF PAGES 298
		15. SECURITY CLASS. (of this report) Unclassified
		16a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) A		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Recoverable item, multi-echelon, inventory/repair, simulation, METRIC, MOD-METRIC, AFLC		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report describes the Recoverable Item Management Evaluator (RIME), a FORTRAN simulation model for evaluating the relative cost-effectiveness of analytic optimization procedures proposed for use in Air Force Logistics Command recoverable item management systems. Major features of the model include (a) the use of actual Air Force demand histories to drive the model demand processes, (b) modeling of current Air Force statistical estimation procedures, and (c) modeling of the dynamic interactions among initial provisioning, replenish- ment and distribution policies. Volume II documents the programs for RIME.		

FORM 1473

SECTION OF 1 NOV 68 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Section II

Levels Computation System Programs

List of Programs

Program

DMSGNI

EVALUATE

GETBSO

GETDAT

INTPRO

LEVLP1

LEVLP1

METINP

ONEIND

OVHSTL

SAVDAT

SPNDMS

TWOIND/INPUT

TWOIND/OUTPUT

Subroutine: DMSGN1

Function:

This routine generates input data sets for the Levels Computation System.

Description:

This routine reads D041 historical data for LRU/SRU data sets, and produces as output appropriate data cards required to drive MOD-METRIC computation routines. For a detailed discussion of this program, see the Levels Computation System Chapter in Volume I.


```

1  *#RUM=IRINE/0BS/DMSGN1.0(BCD,ULIBYRINE/0BJ/READZ,0)
2  *# RINE/0BJ/ISY-27,0;RSQS/RANDU,0;RARIIE/0BJ/BASEDA,0,R;
3  *# #RINE/D)4IFLE"07",R;RINE/HOB.IDY05".R
4  *DMSGN1.S
5  C
6  C
7  *-----*-----*-----*-----*-----*-----*-----*-----*-----*
8  COMMON/ENTRY/ENTRY
9  COMMON/EDBUS/EDBUS
10 COMMON/EPGRP/EPGRP
11 COMMON/ELGRP/ELGRP
12 COMMON/EGROUP/EGROUP
13 COMMON/ITLT/ITLT(40)
14 COMMON/ITIME/ITIME
15 COMMON/IBASES/IBASES
16 COMMON/IJOB/IJOB
17 COMMON/ISRU/ISRU
18 COMMON/IISRN/IISRN
19 COMMON/IGUT/IGUT
20 COMMON/IWT/IWT(207)
21 C
22 C
23 C
24 C          PROGRAM HISTORY ARRAYS
25 C
26 COMMON/IBREN/IBREN(16,40)
27 COMMON/IRTS/IRTS(16,40)
28 COMMON/IBCOND/IBCOND(16,40)
29 COMMON/IRNITS/IRNITS(16,40)
30 COMMON/IDCOND/IDCOND(16,40)
31 COMMON/IOVCND/IOVCND(16,40)
32 COMMON/IPROG/IPROG(16,40)
33 COMMON/IBREN/IBREN(16,40)
34 C
35 C          ITR DATA COMMONS
36 C
37 COMMON/PSS/PSS(40)
38 COMMON/PCOST/PCOST(40)
39 COMMON/ETADN/ETADN(40)
40 COMMON/LTR0D/LTR0D(40)
41 COMMON/IBDT/IBDT(40)
42 COMMON/IBDTT/IBDTT(40)
43 COMMON/IDBT/IDBT(40)
44 COMMON/I0ST/I0ST(40)
45 COMMON/SIBDR/SIBDR(40)
46 COMMON/RISF/RISF(40)
47 COMMON/PDHP/PDHP(40)
48 COMMON/ROTF/ROTF(80)
49 COMMON/INSTRF/INSTRF(40)
50 COMMON/SPPDR/SPPDR(40)
51 COMMON/PJBCND/PJBCND(40)
52 COMMON/PNARC/PNARC(40)

```

DMSGN1


```

05      C
06      C***** I.B
07      C
08      C          READ SIMULATION PARAMETERS
09      C
10      WRITE(6,17) " NFGRP, NLCGRP,INOTE,NDNIS"
11      17 FORMAT(IV)
12      READ(5,13) NFGRP, NLCGRP,INOTE,NDNIS
13      WRITE(6,28) NFGRP, NLCGRP,INOTE,NDNIS
14      23 FORMAT(I9,4I7)
15      C
16      ITOT=NDNIS+INOTE
17      IF(ITOT.LE.16) GO TO 19
18      PRINT,"NDNIS+INOTE CANNOT EXCEED 16. STOP RUN"
19      STOP
20      C
21      C***** I.C
22      C          READ WRITE FLAG CARD.
23      C
24      19 CONTINUE
25      WRITE(6,17) "WRITE FLAGS "
26      READ(5,13) IWT
27      WRITE(6,31) IWT
28      3 FORMAT(//WRITE FLAGS=",5(I2,4X))
29      C
30      INBUG=INT(1)
31      C
32      C***** I.D
33      C          READ TITLE AND COMMENT CARD.
34      C
35      C
36      WRITE(6,13) "IDENT      COMMENT"
37      ENCODE(TITLE,37)XNSTR,XNQBAS,XCOST,KNSTR,KNQBAS,KCOST,
38      &      ININSTR,DOMINI,KNINSTR,KNONENR
39      37 FORMAT( 3I2," ",3I2," ",3I2," ",3I2," ")
40      &      I2,4I3)
41      WRITE(6,13) IDENT,TITLE
42      WRITE(1,69) IDENT,TITLE
43      WRITE(8,68) IDENT,TITLE
44      69 FORMAT("629,IX,I3,IX,"INITIAL PROV".,A40)
45      68 FORMAT("629,IX,I3,IX,"REFLN".,A40)
46      C
47      C***** I.E
48      C          READ EVALUATOR PARAMETER DATA.
49      C
50      WRITE(6,17) "NOCR, KBO, CPAC9"
51      READ(5,19) NOCR,KBO,CPAC9
52      WRITE(6,13) NOCR,KBO,CPAC9
53      WRITE(8,88) NOCR,KBO,CPAC9
54      WRITE(1,88) NOCR,KBO,CPAC9
55      68 FORMAT("869,IX,2I2I6.2)
56      C

```

2 01 10-20-79 10.356 28

```

57 C
58 C***** I.F
59 C      READ OUTPUT CONTROL DATA
60 C
61 WRITE(6,13)  NOUT,IPENT,IPNCH,IBSO
62 READ(5,13)  NOUT,IPENT,IPNCH,IBSO
63 WRITE(6,13)  NOUT,IPENT,IPNCH,IBSO
64 WRITE(8,03)  NOUT,IPENT,IPNCH,IBSO
65 WRITE(1,03)  NOUT,IPENT,IPNCH,IBSO
66 03 FORMAT('09',1X,12)
67 C
68 C***** I.G
69 C      READ OPTIMIZATION PARAMETERS.
70 C
71 WRITE(6,17)  NDIS,BETA,BSTART,BSTOP,CFAC,FBINC
72 READ(5,13)  NDIS,BETA,BSTART,BSTOP,CFAC,FBINC
73 WRITE(6,18)  NDIS,BETA,BSTART,BSTOP,CFAC,FBINC
74 WRITE(8,103)  NDIS,BETA,BSTART,BSTOP,CFAC,FBINC,KNINCK,SONZK,KEBBS
75 WRITE(1,103)  NDIS,BETA,BSTART,BSTOP,CFAC,FBINC,KNINCK,SONZK,KEBBS
76 103 FORMAT('09',1X,I3,F6.2,I6,F6.4,I6)
77 C
78 C***** I.H
79 C      READ FLYING HOUR PROGRAM AND GET DATA.
80 C
81 CALL BASDA
82 C
83 C***** II.A
84 C      COMPUTE TOTAL FLYING HOURS PER BASE
85 C
86 PAKPF=0.
87 OTFH=.
88 IS=NDNIS+1
89 IE=NDNIS+8
90 IF(IE.GT.23)IE=23
91 DO 120 IQ=IS,IE
92   TFH=0.
93   DO 110 KOT=KESNS
94     TFH=TFH+DPH(IQ,K)
95 110 CONTINUE
96   OTFH=OTFH+TFH
97   IF(PAKPF.LV.TFH)PAKPF=TFH
98 120 CONTINUE
99   AVTFH=OTFH/FLOAT(IE-IS+1)
00   PAKPF=PAKPF/AVTFH
01 C
02 C      CONVERT QFN DATA TO HOURS
03 AVHPR=AVTFH/3.
04 WRITE(6,18)  OTFH,OTFH, AVHPR,AVHPR,
05 " PAKPF,PAKPF
06 C
07 C***** II.B  INITIALIZED RANDOM NO. GENERATOR AND
08 C      RECORD SEQUENCE NO. (ISNS)

```

T 01 10-20-79 10.366 JS

```
09      R=RANDU(=.1)
10      ISEQ=0
11      C
12      C
13      C
14      C***** IS,C
15      C      SAYS THE FIRST (NPERD=1) LRU GROUPS
16      C
17      CALL READP1
18      NSKIP=NPERD-1
19      IF(NSKIP.LE.0) GO TO 60
20      DO 50 N=1,NSKIP
21      NGROUP=NG
22      CALL READPL(IENDP)
23      IF(IENDP.NE.1) GO TO 1200
24      50 CONTINUE
25      60 CONTINUE
26      C
27      C      BEGIN GROUP LOOP
28      C
29      DO 100 N=NPERD,NLGRP
30      NGROUP=NG
31      WRITE(6,11)'BEGIN LRU GROUP=',NG
32      13 FORMAT(V)
33      C
34      C
35      C***** III,A
36      C      SET LRU AND SRU HEADER DATA
37      C
38      CALL READPL(IENDPT)
39      IF(IENDP.NE.1) GO TO 1200
40      C
41      C      START PROCESSING RECORDS.
42      C
43      C-----*-----* FILE SECTION *-----*
44      C
45      C      ZNOTE= THE NUMBER OF PERIODS OF DATA TO BE
46      C      COMPUTED.
47      C
48      C      ZDNIS = INITIAL NUMBER OF PERIODS USED IN
49      C      ESTIMATE.
50      C
51      C-----*-----*-----*-----*
52      C
53      C***** III,B
54      C      COMPUTE OST BY BASE USING LRU OST TIME AS THE GRAND
55      C      AVERAGE.
56      C
57      OSTLRU=10ST(1)
58      AVOST=0.
59      WRITE(6,13)' OSTLRU=',10ST(1)
60      DO 15 N=1,NBASES
```

F 01 10-20-79 10.366 JS

```

61      OST(K)=OST(K)+OSTDLT(K)
62      IF(OST(K).LT.5.)OST(K)=5.
63      AVOST=EVOST+OST(K)
64      150 CONTINUE
65      AVOST=EVOST/FLOAT(NBASES)
66      C
67      C
68      C***** III.C
69      C      BEGIN QUARTER LOOP
70      C
71      DO 500 IQTR=1,INQTR,2
72      C
73      IF(INBUS.GE.1)WRITE(6,153)IQTR,NG
74      153 FORMAT(/,15('=-= '), "START QTR=",I3,
75              " FOR GROUP=",I3,5("=-8-8//")
76      C
77      C
78      C***** III.C,1
79      C      SET AVERAGE FUTURE FLYING HOURS PER QUARTER
80      C      USING AVERAGE OF THE NEXT 8-QUARTERS.
81      C
82      TOPH=0.
83      DO 177 N=1,NBASES
84      BPHR(K)=0.0
85      IS=NDNIS+QTR
86      IN=MIN(NDNIS+QTR+7,23)
87      DO 176 I=IS,IE
88      176 BPHR(K)=BPHR(K)+BPH(I,K)
89      BPHR(K)=BPHR(K)/FLOAT(IE-IS+1)
90      177 TOPH=TOPH+BPHR(K)
91      AVHACH=TOPH/J.
92      C
93      IF(INBUS.GE.1)WRITE(6,173)TOPH,IS,IE,(BPHR(K),K=1,NBASES)
94      173 FORMAT(/,9 "TOPH=",F10.0," = TOTAL PROJECTED AIRCRAFT",
95              " PROGRAM FOR QTR ",I2," THRU QTR ",I2/
96              " PROGRAM BY BASE = BPHR(K) =",8F10.0)
97      C
98      C
99      C***** III.C,2
100     C      COMPUTE AVERAGE LRU INSTALLED PROGRAM
101     C      IN THE NEXT 8 QUARTERS.
102     C
103     IPEAKP=0
104     IPROG=0
105     IS=MIN(NDNIS+QTR,16)
106     IE=MIN(NDNIS+QTR+7,16)
107     DO 178 I=IS,IE
108     EX=IPROG(I,4)
109     IF(IPEAKP.LT.EX)IPEAKP=IX
110     IPROG=IPROG+EX
111     178 CONTINUE
112     C

```

BT 01 10-20-79 10.366 JS

```

113 C***** IIX.C.3
114 C      CONVERT PROGRAM FROM 1000S OF HOURS TO HOURS,
115 C      THEN ESTIMATE HOURS/ MONTH ON THE HOST AIRCRAFT
116 C
117      IPROG= 100 * IPROG
118      IPEAKP=100 * IPEAKP
119 C
120      AVQPHR=FLOAT(SPROG)/FLOAT(IE&I0+1)
121 C      ESTIMATE FLYING HOURS RELATED TO THIS PROGRAM
122 C
123      PEAKPF=FLOAT(IPEAKP)/AVQPHR
124      AVQPHR=AVQPHR/FLOAT(IOPA(1))
125      AVMPHR=AVQPHR/3.
126      IF(INBUS,GE,1)WRITE(6,8173) IE&IE,IPROG,AVQPHR,AVMPHR,PEAKPF
127 8173 FORMAT(" AVERAGE LRU PROGRAM CALCULATIONS"/
128 & " I2," IE," I2," IE," I2," SPROG=" ,I10,
129 & " AVQPHR=" ,F10.0," AVMPHR=" ,F10.0," PEAKPF=" ,F6.3)
130 C
131 C***** IIX.C.4
132 C      COMPUTE FUTURE MONTHLY INSTALLED PROGRAM
133 C      BY BASE PROPORTIONEL TO AIRCRAFT FLYING PROGRAMS,
134 C      IF INSTALLED BASE PROGRAM IS LESS THAN 1.0,
135 C      RESET IT TO 1.0.
136 C      THIS IS TO PREVENT DIVIDS AND LOOPING ERRORS
137 C      IN THE MOD-METRIC ROUTINES.
138 C
139 C
140      AVBPHR=0.
141      DO 179 R=1,NBASES
142      PH(K)=AVBPHR+(BPHR(K)/TOPH)
143      IF(PH(R).LT.1.)PH(K)=1.0
144      AVBPHR=AVBPHR+PH(K)
145 179 CONTINUE
146      AVBPHR=AVBPHR/FLOAT(NBASES)
147 C
148      WRITE(6,183)IQTR,AVBPHR,(PH(K)K=1,NBASES)
149 183 FORMAT(/ /9 "*****DNEGN1=-IQTR*E12;" AVE BASE FLYING",
150 & " PROGRAM FOR AIRCRAFT OVER NEXT 6-0TES =",
151 & F10.2:/T20,PROGRAM BY BASE * PH(K)=" ,BPH.2)
152 C***** IIX.C.5
153 C
154      IF(IQTR.GE.1) GO TO 180
155 C
156 C      SET PARAMETERS FOR INITIAL PROVISIONING CALCULATIONS
157 C
158      IFILE=1
159      JEQAS=IEQAS
160      JMETH=IMETH
161      JCOST=ICOST
162      GO TO 184
163 C
164 C      SET PARAMETERS FOR REPLENISHMENT CALCULATIONS

```

01 10-20-79 10.366 25

```

165 C
166 180 CONTINUE
167 IFILE=8
168 JEQDAS=REQDAS
169 JHETH=KHETH
170 JCOST=KCOST
171 C
172 C***** IX.C.6
173 C CONSTRUCT PROGRAM FLYING DATA
174 C
175 184 CONTINUE
176 C
177 C IF JEQDAS=1 THEN OUTPUT A "97" RECORD WITH
178 C FH(K),K=1,...,NBASES SET EQUAL TO AVEFH, AND
179 C OST(K),K=1,...,NBASES SET EQUAL TO AVEOST.
180 C OTHERWISE, USE ORIGINAL FH AND OST ARRAYS
181 C
182 C
183 C***** IX.C.6.1
184 C WRITE OUT FLYING HOUR PROGRAM AND OST DATA
185 C
186 C
187 IF(NBASES.LE.8) GO TO 79
188 73 FORMAT("979,1X,I3,6(F6.0,F3.0)/2"97",T7,
189 & 6(F6.0,F3.0))
190 C
191 WRITE(IWRITE,73)
192 & NBASES,((FH(K),OST(K)),K=1,NBASES)
193 GO TO 81
194 C
195 74 FORMAT("979,1X,I3,6(F6.0,F3.0))
196 C
197 79 CONTINUE
198 WRITE(IWRITE,74)
199 & NBASES,((FH(K),OST(K)),K=1,NBASES)
200 C
201 C***** IX.C.6.2
202 C COMPUTE QUARTERS TO BE USED FOR PLANNING FACTOR CALCULATIONS
203 C ALWAYS USE AT LEAST 4 QUARTERS OF DATA IN
204 C COMPUTING PLANNING FACTORS
205 C
206 81 CONTINUE
207 IS=MAX(4,NBNS+IQTR-8)
208 IS=MIN(8,IS)
209 IS=MIN(8,IS)
210 IS=MIN(8,IS)
211 IS=MAX(8,IS)
212 C LOOP THRU N NUMBER OF ITEMS:
213 75 DO 100 J=1,NITEMS
214 C***** IX.C.6.3.A
215 C WRITE OUT TO FILE#7 THE LRU/SRU FACILITY RECORDS.
216 C IDENTIFY LRUs AND SRUs FOR IC1

```


ST 01 10-20-79 10.366 JS

```

117      IC2=1
118      IC1=3
119      IF(J.EQ.1) IC1=2
120      IF(JMETH.EQ.3) IC1=1
121      IF(JMETH.EQ.4) IC1=1
122      C
123      C          JMETH=1 INDICATES OWNED RUN.
124      C
125      IF(JMETH.EQ.1) IC1=1
126      IF(JMETH.EQ.1) IC2=1
127      C
128      C***** III.C.6.3.B
129      C          COMPUTE PLANNING FACTORS--XHTBD,XWRTS,CONS
130      C
131      IPT=J
132      CALL NETEND(XS,IE,IPT)
133      C
134      C***** III.C.6.3.C
135      C          COMPUTE PROCUREMENT LEADTIME IN MONTHS
136      IFLT(J)=(ETADN(J)+LTPROD(J))/30
137      C
138      C
139      C          SCALE PARAMETERS FOR OUTPUT IN MODMETRIC FORMAT
140      C          ROUNDING ALL CALCULATIONS.
141      C***** III.C.6.3.D SCALE PARAMETERS FOR OUTPUT IN
142      C          MODMETRIC FORMAT
143      C
144      ICON1=IFIX(100.0*CONS(J)+0.5)
145      IWRTS1=IFIX(100.0*WRTS(J)+0.5)
146      IHTBD1=IFIX(XHTBD(J)+0.5)
147      ICOST=IFIX(UCOST(J)+0.5)
148      IDRCD=IDDT(J)+IDRT(J)
149      C
150      C***** III.C.6.3.E
151      C          COMPUTE OVERHAUL STOCK LEVEL
152      C
153      IPOTR=IOTR
154      JJ=J
155      CALL OVHSTL(IPOTR,JJ,IYSTL)
156      C
157      C          IF JMETH IS NOT EQUAL TO 4,
158      C          SKIP THE KPCR $7-27 CALCULATION.
159      IF(JMETH.EQ.4) GO TO 300
160      C
161      C***** III.C.6.3.F
162      C          SET PARAMETERS FOR INITIAL PROVISIONING CALCULATIONS
163      C
164      IF(IOTR.EQ.1) GO TO 300
165      DRCD=IDRCD
166      DRCD=IDRCD
167      FLT=IFLT(J)
168      QPA=IQPA(J)

```

01 10-20-79 13.366 JS

```

189 C
190 C STATE INITIAL PROV. PROGRAM DATA IN 100'S OF HOURS
191 C
192 C AVHPPG=SYNCR/100.
193 C BRKPS=BRKPS/AVHPPG
194 C
195 C CALL INTPRO(FSN(J),AVHPPG,UCOST(J),XHTD(J),XHTS(J),
196 C CONS(J),DECD,DECDPRT,QDA,FSN(0),AVHPPG,
197 C BRKPS ,ITTR)
198 C
199 C
200 C WRITE ITTR DATA
201 C
202 C 300 CONTINUE
203 C
204 C ***** III.C.6.3.G
205 C IF ACOST .EQ. 1. APPLT UNIT COST DISCOUNT
206 C
207 C IF(JCOST,EN,1) GO TO 310
208 C DISCNT=1-RHTS(J)
209 C IF(DISCNT.LT.0.10)DISCNT=0.10
210 C UCOST=UC(J)*(DISCNT*UCOST(J))
211 C
212 C 310 CONTINUE
213 C ***** III.C.6.3.H OUTPUT HOURLY ITTR DATA RECORD
214 C
215 C LAMBDX=0
216 C JOBID=100000000*IDENT+1000000*LANBDA+10000*NO+100*ITTR+J
217 C WRITE(17,43) IC1,IC2,JOBYD,ISVTL,UCOST,
218 C INTD1,INTS1,ICN1,IOQA(J)
219 C
220 C INT(J),IDCD,IVL(J)
221 C 43 FORMAT(2I1,1X,2I10,1X,3I1,27I6,1X,3I2,3I3)
222 C
223 C ***** III.C.6.3.I
224 C WRITE DELIVERY SCHEDULE CARD
225 C
226 C IC2=5
227 C IF((JOBID%100000000).AND.(ITTR%1000000))
228 C WRITE(17,49)IC1,IC2,JOBYD,
229 C ISVTL,ITTR
230 C 49 FORMAT(2I1,1X,2I10,1X,3I1,27I6,1X,3I2,3I3)
231 C
232 C
233 C END OF ITTR LOOP
234 C
235 C 100 CONTINUE
236 C
237 C END OF ONE LOOP
238 C
239 C 500 CONTINUE
240 C
241 C END OF GROUP LOOP
242 C READ NEXT LNU/BRN FAMILY.
243 C
244 C 1000 CONTINUE
245 C WRITE(6,13)"DNSCH2 SUCCESSFULY COMPLETION"

```

ST 01 10-20-79 10.966 28

521 STOP
522 1200 CONTINUE
523 WRITE(6,13)"END OF FILE 07 REACHED. STOP RUN"
524 WRITE(6,13)"LAST LRU READ WAS BSN=".PSN(17
525 STOP
526 END
*W 7 MEMORY EXPANDED. USE SLIMITS OR CONN= OPTION FOR NEXT RUN

Subroutine: EVALUATE**Function:**

This program determines an optimal distribution of a given number of LRU/SRU assets.

Description:

The program EVALUATE determines an optimal distribution of stock among the depot and using bases. A discussion of the functions of this routine and input requirements is presented in AFLCR 57-13, and will not be discussed further here. The routine utilized by RIME is identical to the routine discussed in AFLCR 57-13 with one exception. Provisions have been added to the RIME version to permit output of computed levels to the file IBSO using the same record formats as output by the program GETBSO. Specific program lines that were changed for the RIME version are 421-425; 1295-1315; and 3581-3675. The portions of changed code are marked by comment statements in the text of the program.

02 10-20-79 10.561 EVALUATE

```

1      C      EVALUATE
2      C
3      C      COMPILED FILE IS MODERNTRIC/CONVAL
4      COMMON DATA,PRD,NDASHS,NGUO,NAPO,PN(FEYOT,OST,TRST
5      COMMON SYSPIL
6      REAL NAPP
7      REAL ONTRU(45), ONSTRUC(45), CONRU(45)
8      REAL ONSRU(45), ONTRU(45), ONTRU(45), ONTRU(45), ONTRU(45)
9      REAL OST(23), PN(23), ONTR(23), ADLT(23), TR(23)
0      DIMENSION NDA(10), ID(6), IDLU(6), XDSRU(45,6),
1      *      ON(4), ON(45,6)
2      DIMENSION CON(45), CONSRU(45), OFLT(45)
3      *      ,DATA(45)
4      *      ,SPLTS(45)
5      INTEGER ONLNU, ONSRU(45), ONDRU(45)
6      INTEGER OFDLNU
7      INTEGER SPP(1,6), SPP(45,6), OSN(10), OSN(45,6)
8      INTEGER STOCK(23,46), LSTOCK(23), SSTOCK(23,45)
9      REAL BAK(23,46), BBACK(23), BBACK(23,45)
0      EQUIVALENCE (STOCK, LSTOCK), (STOCK(1:27), SSTOCK)
1      DATA NBR/4000 /
2      DATA SBR/100*40 /
3      EQUIVALENCE (BAK, BBACK), (BAK(1:27), BBACK)
4      C
5      CHARACTER CARD*72, NAME
6      C
7      DIMENSION IDORD(9)
8      C
9      CALL IOST(IUNUM, IOCND)
0      CALL DOTS(IDATE)
1      300 READ(4,10,END=9)NAME
2      IF(NAME(20,"LAST ")#0)GO TO 300
3      BACKSPECS 0
4      WRITE(6,20)IDATE, IUNUM, IDORD
5      310 FORMAT(1E, A6)
6      320 FORMAT( " EVAL ", 1X, I6, 1X, X5E1X19A6)
7      330 FORMAT(1E, "LAST ")
8      WRITE(6,30)
9      9 CALL DETECH(6, ISTAT, )
0      C
1      C-----+-----+-----+-----+
2      C      SET LENS=1000 FOR GENERATING OUTPUT FILE
3      LINE=1000
4      C-----+-----+-----+-----+
5      C
6      C
7      196=0
8      197=0
9      198=0
0      ENDO=
1      INTC = 0
2      SYSPIL = 0:

```

EVALUATE

T 02 10-20-79 10.841 EVALUATE

```

53      ZSW11=0
54      ZSWLEU=0
55      ZSWSEU=0
56      MAXSEU=49
57      ZED=30
58      ZETA43=0
59      ZYBIX=0
60      ZYCOST=0
61      C
62      6  READ(5,7/END=990)IC1,ZC2,CARD
63      7  FORMAT(2E1,1X,A6)
64      8  IF(INOZ,SE,0)GO TO 2100
65      GO TO 110,20,30,30,6,6,6,6,901,IC1
66      C
67      90 IF(IZWSEU,SE,0)GO TO 1000
68      IF(IZW11,SE,0)GO TO 1000
69      GO TO 10,92,6,6,6,96,97,98,812)IC2
70      C
71      92 DECODE(CARD,192)NDC
72      192 FORMAT(18A4)
73      INDC=0
74      GO TO 6
75      C
76      96 Z96=1
77      INDC=0
78      DECODE(CARD,196)NOTE,K97,CPXC
79      196 FORMAT(2E2,7E,2)
80      IF(NOTE,SE,1)INOUR=0
81      IF(K97,IS,1)K97=0
82      GO TO 6
83      C
84      97 Z97=1
85      INDC=0
86      DECODE(CARD,197)NBASIS
87      197 FORMAT(2E)
88      IS=1
89      IS=INZ(6,NBASIS)
90      DECODE(CARD,297)(PH(I),OST(ZYET=IS,IS))
91      297 FORMAT(3E,6(9E,0,7E,0))
92      IF ( IS,SE,NBASIS ) GO TO 567
93      IS=IS+6
94      IS=INZ(28+6(NBASIS))
95      READ(5,7/IC1,IC2,CARD
96      IF(IC1,SE,9,AND,IC2,SE,7)GO TO 297
97      696 WRITE(6,97)
98      697 FORMAT(13X)FLYING HOUR CARD MESSNG/INCOMPLETE)
99      STOP
00      897 ADDR=0.0
01      PLOT=0.0
02      CHIN=1.235
03      CHAX=1.235
04      DO 697 IS=1,NBASIS

```


2 02 10-20-79 10.341 SYALUATB

```

55      DO 315 K01, K02
56      315 ONLRO=ONLRO+0.5L(1,K)
57      IF(K01,0,0)GO TO 6
58      DO 415 K01, K02
59      415 ONLNU=ONLNU+LPP(1,K)
60      GO TO 6

```

```

61      C
62      16 DECODE(CARD, 116)NSH
63      116 FORNKT(25X, 644)
64      GO TO 6

```

```

65      C
66      C
67      20 GO TO 121,6,6,6,25,26,6,6,67,2C2
68      C

```

```

69      21 IF(ISHR0,0)GO TO 1000
70      IF(ISH(1,SHR,0)0)GO TO 1000
71      ISHLNU=1
72      DECODE(CARD, 111)IDLRU,C,INTSDTYN25,CQNLRU,NAPE.
73      = ONLNU,DATE,PLXL
74      INRC = INRC + 1
75      IF(INRC,0,1)NAPE=1.
76      DO 221 K01, NAPE
77      221 IF(KT=ONIK)NAPEZ(XNTSD=30.Y
78      J=0
79      NSRU=0
80      PL=PL+L
81      ONLNU = DATE + KDDRT
82      CONLCANRO
83      GO TO 611

```

```

84      C
85      25 DECODE(CARD, 125)SD
86      125 FORNKT(644)
87      DO 1125 Z = 1,6
88      1125 IF(Z0) .NE. IDERU(Z))GO TO 6
89      DECODE(CARD, 225)05L,EP9
90      225 FORNKT(25X, (6X2, 1X, 4X2), 1X, 6X2)
91      DO 325 K01, K02
92      325 ONLRO=ONLRO+0.5L(1,K)
93      IF(K01,0,0)GO TO 6
94      DO 425 K01, K02
95      425 ONLNU=ONLNU+LPP(1,K)
96      GO TO 6

```

```

97      C
98      26 DECODE(CARD, 126)NSH
99      126 FORNKT(25X, 644)
100     GO TO 6

```

```

101     C
102     30 GO TO 131,6,6,6,35,36,6,6,67,2C2
103     C

```

```

104     31 IF(ISHR0,0)GO TO 6
105     J=1
106     NSRU=NSRU+1

```


02 10-20-79 10.541 EVALUATE

```

17 ZSWSHU=1
18 IF (NHRN, ST, MAXSHR) GO TO 2200
19 DECOMB(CARR, I11, I2DRN(J, X) * I1, 0), CSRU(J), SHTSU(J), SHTSU(0)
20 * CONSRU(J), SAPP(J),
21 * DRTSHU(J), DRSS(J), SPLTS(J)
12 C
13 IF (SAPP(J) * LG, I1, SAPP(J)) = 1,
14 IF (SHTSU(J), DO, 0) SHTSU(J) = 000000
15 IF (DRTSHU(J), DO, 0) DRTSHU(J) = 0
16 IF (DRTSHU(J), DO, 0) DRTSHU(J) = 0
17 DRTSHU(J) = SHTSU(J) * ADHRF
18 CON(J) = CONSRU(J)
19 SPLT(J) = SPLTS(J)
20 IF (SHTSU(J), DO, 0) GO TO 231
21 CON(J) = CON(J) * SHTSU(J)
22 SPLT(J) = CON(J) * SPLT(J) * 301 * 0.01 / SHTSU(J)
23 DRTSHU(J) = SPLT(J) * (SHTSU(J) * CON(J)) * DRTSHU(J) / SHTSU(J)
24 GO TO 6
25 231 SHTSU(J) = CON(J)
26 SPLT(J) = 0
27 DRTSHU(J) = 0
28 GO TO 6
29 C
30 35 DECOMB(CARR, I35) 20
31 135 FORMAT(6A4)
32 DO 135 K = 1, 6
33 135 IF (I35(K), DO, I35SHU(J, K)) GO TO 6
34 DECOMB(CARR, I35) (OSR(J, X) * I35(I35)) (S55(J, X), X = 1, 6)
35 235 FORMAT(25X, (4I2, 1X, 4I2), 1X, 02V)
36 DO 235 K = 1, NHRN
37 335 OSRSHU(J) = OSRSHU(J) + OSR(J, K)
38 IF (K = 7, DO, 0) GO TO 6
39 DO 435 K = 1, NHRN
40 435 OSRSHU(J) = OSRSHU(J) + OSR(J, K)
41 GO TO 6
42 C
43 36 DECOMB(CARR, I36) (SWSH(J, X), X = 1, 6)
44 136 FORMAT(25X, 4A4)
45 GO TO 6
46 C
47 990 IBOJ=1
48 IF (ISWHRN, NHR, 0) GO TO 1000
49 IF (ISWHRN, NHR, 0) GO TO 1000
50 GO TO 8100
51 C
52 1000 IF (I99: NHR, 1) AND I97, DO, 1) AND (I96, DO, 1) GO TO 1020
53 WRITE(6, 1010) I99, I97, I96
54 1010 FORMAT(15I12, 90W, I2, 5X, 4I12, 1X, 5X, 4I12, 1X, 40B, 1)
55 STOP
56 C
57 1020 ISWHRN=0
58 ISWLRN=0

```

10-30-75 10.54 SYLVAN

```

59 WRITE(OUT,1026)GO TO 1047
60 IF(2ND,10,1026)GO TO 1026
61 WRITE(OUT,1027)
62 1030 FORMAT(101,1024)
63 STOP
64 WRITE(OUT,1035)NOTE,K09,STPC
65 1038 FORMAT(1,1040,1041,1042,1043)
66   102,17,1041,1042,1043,1044
67   SYMPTOM PROVISIONS QUANTITIES,EX
68   SRT(10,1042)
69 GO TO 1040
70 1040 WRITE(OUT,1045)K,PH(B),OSTIKY
71 1045 FORMAT(10,1045,1046)
72 1047 WRITE(OUT,1048)
73 1048 FORMAT(1048,1049)
74 WRITE(OUT,1050)
75 1050 FORMAT(1050 WORK 9999 NO.112)
76   *SH SST QUANTITY THAN TEN(SST 9999 COND SST DST 999 )
77   *SH 1048,1049
78   *1050 1051,1052,1053 COST OR HAND BETWEEN DEMANDS SST,
79   *SST,PH=,10,1051,1052,1053)
80
81 C-----
82 WRITE(OUT,1055)EDLEU,CONLUNTED,STRTS,CONLU,STLRV,BRLNAPP
83   *STLRV,STLRV
84 1055 FORMAT(1055,1056,1057,1058,1059,1060,1061,1062,1063,1064,1065,1066,1067,1068,1069,1070)
85   *1056,1057,1058,1059,1060,1061,1062,1063,1064,1065,1066,1067,1068,1069,1070
86   *1055,1056,1057,1058,1059,1060,1061,1062,1063,1064,1065,1066,1067,1068,1069,1070)
87 1070 FORMAT(1070) *STLRV - END STOCK LESS THAN ZERO*/
88   IF(1070,1070)WRITE(OUT,1071)
89 1071 FORMAT(1071) *STLRV - END LESS THAN OR EQUAL TO ZERO*/
90   IF(1071,1071)WRITE(OUT,1072)
91 1072 FORMAT(1072) *STLRV - END LESS THAN OR EQUAL TO ZERO*/
92   IF(1072,1072)GO TO 1080
93   WRITE(OUT,1073)
94 1073 FORMAT(1073) *FATAL ERROR F**** - NTS LESS THAN ONE*/
95   STOP
96 1080 IF(1073,1073)GO TO 1082
97   WRITE(OUT,1074)
98 1074 FORMAT(1074) *FATAL ERROR F**** - NTS IS INCORRECT*/
99   STOP
100 1082 IF(1074,1074)AND,CONLU,1,1)GO TO 1084
101   WRITE(OUT,1075)
102 1075 FORMAT(1075) *FATAL ERROR F**** - CONDENATION IS INCORRECT*/
103   STOP
104 1086 IF(1075,1075)AND,PLT,1)GO TO 1088
105   GO TO 1088
106 1088 WRITE(OUT,1076)
107 1076 FORMAT(1076) *FATAL ERROR F**** - CONDENATION > 0 AND PLT < 1 */
108   *
109   STOP
110
111 C

```

T 02 10-20-79 10.561 EVALUATE

```

11 1055 IF(NERS,NO.0)GO TO 2060
12 C
13 WRITE(ROUT,1057)
14 1057 FORMAT(4H SRU)
15 DO 1060 N=1,NBRU
16 C-----WRITE SRU ITEM DATE
17 WRITE(ROUT,1055)(IDSRU(K),STRTM(1,0),CONSR(K),ONSRU(K),ONRSD(K),
18 *SRTS(N),CONSRU(K),BRTSRU(K),BRTS(K),SKPR(K),SPZS(K)
19 * (SWSNR(K),ST,0)
20 IF(ONSRU(K),ST,0)WRITE(ROUT,3007
21 3000 FORMAT(1X,*** WARNING - SRU STOCK LESS THAN ZERO*/)
22 IF(BRTSRU(K),LE,0)WRITE(ROUT,3002)
23 3002 FORMAT(1X,*** WARNING - SRU LESS THAN OR EQUAL TO ZERO*/)
24 IF(DTS(N),LE,0)WRITE(ROUT,3004)
25 3004 FORMAT(1X,*** WARNING - DRT LESS THAN OR EQUAL TO ZERO*/)
26 IF(SRTSD(K),GE,1760 TO 3160
27 WRITE(ROUT,3006)
28 3006 FORMAT(1X,***** FATAL ERROR ***** - NTRD LESS THAN ONE*/)
29 STOP
30 3100 IF(SWSNR(K),GE,0 .AND. SRTS(N),LE,1)GO TO 3110
31 WRITE(ROUT,3008)
32 3008 FORMAT(1X,***** FATAL ERROR ***** - NRTS IS INCORRECT*/)
33 STOP
34 3110 IF(CONSRU(K),GE,0 .AND. CONSR(K),LE,1)GO TO 3120
35 WRITE(ROUT,3010)
36 3010 FORMAT(1X,***** FATAL ERROR ***** - CONDENMATION IS INCORRECT*/)
37 STOP
38 3120 IF(CONSRU(K),GT,0 .AND. SWS(N),ST,1760 TO 3160
39 GO TO 1060
40 3140 WRITE(ROUT,3150)
41 3150 FORMAT(1X,***** FATAL ERROR ***** -
42 * 1X,"CONDENMATION > 0 AND DLT < 1 %/)
43 STOP
44 1060 SRTSD(N)=SRTSD(K)/SWS(K)
45 1065 FORMAT(1X,E3.2X,E6.4X,F9.0,XE17.0,F9.1,XE17.0,2.2X,E6.2,1X,E12.0
46 * 1X,E3.0,2X,E2.0,2X,E6.0,2X,E6.0)
47 C
48 WRITE(ROUT,2000)(K,K=1,NBRU)
49 2000 FORMAT(20H0 SRU FILL RXTS BY NBRU/7X,E3E6)
50 DO 2030 J=1,NBRU
51 SDDSD(J)=
52 DO 2010 N=1,NBRU
53 2010 SDDSD(J)=SRTSD(J)*FN(K)*SWS(K)/SRTSD(J)*30.7
54 DO 2020 N=1,NBRU
55 2020 SRT(K)=BRTSRU(J)
56 C
57 2030 CALL NERS(SDDSD(J),BRTSRU(J),ONSRU(J),ONRSD(J),
58 * SRTSD(J),SRTS(J),SKPR ,STOCK,J+1)
59 DO 2050 N=1,NBRU
60 ADLT(N)=0
61 DO 2040 N=1,NBRU
62 2040 ADLT(N)=ADLT(K)+BBACK(K)J)

```

02 10-20-79 10.561 EVALUATE

63 2050 ADULT(N)=ADULT(K)/(TD(K)+1)+ENRSTS)

64 DO TO 2066

65 2060 DO 2066 J,JBASNS

66 2065 ADULT(N2065)=0.

67 2066 CONTINUE

68 DBDRZ=0.

69 DO 2070 N=1,JBASNS

70 DBDRZ=DBDRZ+ENRSTS*TD(K)+ENR

71 2070 ENR(K)=ENRTRU+ADULT(K)

72 C

73 CALL MENU(DDRL,DTLNU,ONERU,OPDLRU,

74 * YRTRD,YRTRS,BAK ,STOCKS 1)

75 BOMU,

76 DO 2086 N=1,JBASNS

77 2080 BOMU+BAN(K,1)

78 KCONU=1/2*(OPDLRU)+.99*ENRTRU

79 KDBNS=OPDLRU-LCONU

80 ENRST(ENR,2090)(K,K=1,JBASNS)

81 2090 FORMAT/7/6X,37NOPTIMAL DETERMINATION OF ON-HAND STOCK/1

82 * 5X,ENRST,23X,5NTOTAL,17X,ENRST/

83 * 5X,ENRST,23X,2ONH-HAND ENRST CON:36X3)

84 C-----O-----O-----OUTPUT LRU LEVELS

85 ENRST(ENR,2100)ENRTRU,ONERU,KDBNS,LCONU,(LSTOCK(K),K=1,JBASNS)

86 2100 FORMAT16X0 LRU ,6A4,I5,8X12/8X,23,23I3)

87 C-----O-----O-----ONERU

88 IF(IENR,0)WRITE(IENR,2103)LINE,IENRUCSTLRU,

89 * ONERU,KDBNS,LCONU,(LSTOCK(K),K=1,JBASNS)

90 LNR=(IENR+1)

91 2103 FORMAT(1X,I5,"N1",1X,6A4,1X,23I3)

92 TCOST=0 ONERU

93 C-----O-----O-----OUTPUT SHU LEVELS

94 IF(NENR,0)GOTO 2140

95 WRITE(ENR,2110)

96 2110 FORMAT(8X,SHU)

97 DO 2120 J=1,NSRU

98 CSTRUCRU(J)*ENRST(J)

99 YCOST=YCOST+CSTRUCRU(J)+ONERU/J

100 KCONU=1/2*(OPDRU(J)+.99*ENRST(J)/DTRU(J)

101 KDBNS=OPDRU(J)-KCONU

102 2120 WRITE(ENR,2130)J,(ENRST(J),ENRST(J)+1.6),ONERU(J),KDBNS,

103 * KCONU (LSTOCK(K),K=1,JBASNS)

104 2130 FORMAT23,I2,2X,6A4,I5,8X12/8X,23,23I3)

105 C-----O-----O-----

106 IF(IENR,0)WRITE(IENR,2103)LINE,(ENRST(J,X),I=1,6),

107 * CSTRUCRU(ENR(J),KDBNS),ENRST(J),(LSTOCK(K),K=1,JBASNS)

108 LNR=(IENR+1)

109 C-----O-----O-----

110 2140 WRITE(ENR,2150)BO,TCOST

111 2150 FORMAT11X,23X,21EXPERIENR ENRST ENRST W,P13,0/

112 * 26X,12EXPERIENR ENRST ENRST W,P13,0)

113 C

114 SYBKK=ENRST+BO

```

15 SYCOST=SYCOST+TCOST
16 ONLNU=0
17 OPDLNU=0
18 IF(NSRV,0.0)GO TO 8

```

c

```

19 WRITE(ROUT,2160)(K,K=1,NSRST
20 2160 FORMAT(1X,NO. OF WAIT TIME IN THIS RUN DUE TO
21 * 31% NON-AVAILABILITY OF NEEDED SRU ?/SN BASE,2225)
22 WRITE(ROUT,2170)(ADULT(K),K=1,NSRST
23 2170 FORMAT(6X, WAIT ,23F5.1)

```

c

```

24 DO 2175 N=1,NSRV
25 OPDSRU(K)=0
26 2175 ONSRU(N)=0
27 DO 2176 N=1,NSRV
28 DO 2177 N=1,NSRST
29 2177 SSTACK(J,K)=0
30 GO TO 8

```

c

```

31 2180 WRITE(ROUT,2185)SYSBAK,SYCOSTE SYSPIL/IRSC
32 2185 FORMAT(//14(6X-+---)/25X719SYSTEM BACKORDERS #.F13.84
33 *24X,20TOTAL SYSTEMS COST =F52.0/
34 *24X,20AVERAGE SYSTEM FILE RATE,F52.3/14(6X-+---)/24
35 *5X,11END OF DATA)
36 STOP

```

c

```

37 2200 WRITE(ROUT,2210)NSRV,IDLNU
38 2210 FORMAT(19HOTOO MANY SRU,210F50.624)
39 J=0
40 NSRV=0
41 ISVLRU=0
42 ISVSRU=0
43 GO TO 8
44 END

```

*W 1470 EQUALITY OR NON-EQUALITY COMPARISON MAY NOT BE MEANINGFUL IN LOGICAL IF STATE
 *W 7 MEMORY EXPANDED. USE SLIMITS OR GORR= OPTION FOR NEXT RUN

Subroutine: GETBSO

Function:

This program records up to 20 sets of LRU/SRU stock levels, with each set of levels corresponding to a different Buy Support Objective (BSO). It then scans this set of points to identify the particular set closest to a given BSO.

Description:

This program inputs up to 20 sets of LRU/SRU stock levels, with each set corresponding to a different Buy Support Objective (BSO). It then scans these points, and determines the particular point closest to a given desired Buy Support Objective, and outputs the data set of levels associated with that point. A discussion of the computations performed by this routine and input and output records is given in AFLC 57-13, and will not be discussed further here.

The RIME version of GETBSO is identical to the AFLCR 57-13 routine with one exception. In program lines 2120-2200 logic has been inserted which replaces all blanks in the part number identification field with zeros. This is to facilitate sorting steps which are performed at a later stage of Levels Computation System job stream.

T 01 10-20-79 13.897 JS

```

1  *#RUN=IRINS/682/GETBSO.O(BCD.W0607
2  *GETBSO.S
3  C      GETBSO
4  C      COMPILED FILE IS MODNETRIC/08850
5  C
6  C
7  C      READ AND SCORE(TEMPORARY BSR FILE) ALL DATA
8  C      EQUAL TO THE SAME WORK UNIT CODE(WUC).
9  C      THERE MAY BE UP TO 20 SETS PER WUC. EACH SET
10 C      IS IDENTIFIED BY AN "88" CARD. THE DATA CARDS
11 C      FOLLOWING EACH "88" CARD BELONG TO THAT SET.
12 C      EACH SET HAS A UNIQUE BACKORDER,CONT & BSO.
13 C      EACH SET EQUATES TO A POINT ON THE GRAPH.
14 C
15 C
16 C
17 C      I
18 C      I
19 C      I
20 C      I
21 C      IX (A)
22 C      I
23 C      X (B)
24 C      I
25 C      X (C)
26 C      I
27 C      X (D)
28 C      B I
29 C      X (E)
30 C      I
31 C      X (F)
32 C      I
33 C      X (G)
34 C      I
35 C      X (H)
36 C      I
37 C      X (I)
38 C      I X Y I I I I X I I I X I X I X I X I X I
39 C      C
40 C      B = BACKORDERS      C = COST
41 C
42 C
43 C      ANALYZE ALL SETS(POINTS). DISCARD THE POINTS
44 C      THAT ARE NOT VALID
45 C      EX: POINTS "B", "D", & "G" ARE NOT ACCEPTABLE.
46 C      THEREFORE THEY ARE "NOGOOD".
47 C
48 C      SEARCH ALL VALID POINTS FOR THE POINT CLOSEST TO
49 C      THE DESIRED BUY SUPPORT OBJECTIVE(BSBOY AND
50 C      OUTPUT THE DATA FOR THAT POINT
51 C
52 C

```

GETBSO

```

53 CHARACTER CARD*150
54 CHARACTER CARD88*150
55 CHARACTER OUT1*24,OUT2*89,PRINT
56 CHARACTER REST*24
57 CHARACTER WUPP8,SWUC*8
58 CHARACTER NAME
59 DIMENSION ICARD(9)
60 DIMENSION B(81),C(21),BSO(24)
61 INTEGER B0000/8880000/,PLK(220)
62 C
63 CALL ISST(ISNUMB,ICARD)
64 CALL DATE(DATE)
65 500 READ(4,510,ERR=9,END=9)NAME
66 IF(NAME.EQ."LAST ")GO TO 560
67 BACKSPACE 6
68 WRITE(6,820)DATE,ISNUMB,ICARD
69 510 FORMAT(1X,A6)
70 820 FORMAT(" OUT80 ",1X,I6,1X,I5E1X,9A6)
71 530 FORMAT(1X,"LIST ")
72 WRITE(6,830)
73 9 CALL DETCH(6,ISTAT, )
74 C
75 C SET LAMBDA COUNTER
76 C
77 WRITE(6)
78 C
79 LINE#1000
80 IEND#0
81 DO 535 I=1,28
82 535 FLAG(I)=0
83 C
84 C GET A TEMPORARY DISK FILE
85 C
86 CALL CREATE(8,3000,0,ISTAT)
87 C
88 C
89 C NEED "95" CARD AND DETERMINE OUTPUT(ROUT),INPUT(INN)
90 C AND DESIRED BUY SUPPORT OBJECTIVE(BBSO)
91 C
92 1 CONTINUE
93 READ(9,95,END=850)IC,ROUT,EWR,BBSO
94 5 FORMAT(12,1X,I12I2I2I2I2,5)
95 IF(IC.EQ.95) GO TO 15
96 7 WRITE(6,10)
97 10 FORMAT(1JONMISSING "95" CARDS-8880 / ROUT)
98 STOP
99 15 IF(ROUT.EQ.0)GO TO 7
100 IF(INN.EQ.0)GO TO 7
101 IF(DNS.EQ.888888)GO TO 16
102 TCT=0.
103 XBACK#0.
104 16 SDBSO=BBSO

```



```

105      C
106      REWIND INH
107      C
108      C          INCREMENT LAMBDA COUNTS
109      C
110      KNTD50=KNTD50+1
111      C
112      WRITE(ROUT,17)LINE,KNTD50,DESD
113      17 FORMAT(1X,14X,A1501,"KNTD50 =",I2,".18X,14X,1X,INDICATED DESD,18,
114      14      &      S12,S1)
115      LINE=LINE+1
116      WRITE(ROUT,18)LINE
117      18 FORMAT(1X,14X,A1501,".37X,"Y D C B1888"Y
118      LINE=LINE+1
119      IPT=0
120      C
121      C          PROCESS FIRST "88" CARD AND RELATED DATA
122      C
123      20 READ(INH,25)IC,CARD
124      25 FORMAT(1X,12,1X,A1501
125      IF(IC.NE.88)GO TO 20
126      IPT=IPT+1
127      DECODE(CARD,87)DSO(IPT),WUC,WUNDB(IPT),C(IPT)
128      C          B88 - BUY SUPPORT OBJECTIVE
129      C          WUC - WORK UNIT COSTS
130      C          WUNDB - NUMBER OF DATA LINES FOLLOWING 88 CARD
131      C          B - BACKORDERS
132      C          C - COST
133      27 FORMAT(1X,12,5X,A8,13,1X,12,8,1X,12,8)
134      30 SWUC=WUC
135      WRITE(8)IC,CARD
136      C
137      DO 40 I=1,WUNDB
138      READ(INH,25)IC,CARD
139      WRITE(8)IC,CARD
140      40 CONTINUE
141      IPT=IPT+1
142      C
143      C          STORE ALL INPUT DATA PERTAINING TO THE SAME WUC
144      C          WHEN NEW WUC IS FOUND PROCESS STORED DATA
145      C
146      200 READ(INH,25,WUNDB+60)IC,CARD
147      IF(IC.NE.88)GO TO 200
148      DECODE(CARD,87)DSO(IPT),WUC,WUNDB(IPT),C(IPT)
149      IF(WUC.NE.SWUC)GO TO 90
150      WRITE(8)IC,CARD
151      C
152      205 DO 50 I=1,WUNDB
153      READ(INH,25)IC,CARD
154      50 WRITE(8)IC,CARD
155      IPT=IPT+1
156      GO TO 200

```

```

57      C
58      60 IEND01
59      IPT=IPT-1
60      GO TO 100
61      C
62      C      HOLD NEW VUC DATA
63      C
64      90 HISTOZPT
65      IPT=IPT-1
66      NO=NO+1
67      SWUC=VUC
68      CARDS=CARD
69      ZC=ZPT
70      C
71      C      FIND THE GOOD SLOPES AND FLAG THE BAD POINTS
72      C
73      IF(IPT.EQ.1) GO TO 136
74      100 DO 135 I=1,IPT
75      NM=1
76      108 IF(I+M.LT.2) GO TO 135
77      IF(FLAG(I+M).NE.NOGOOD) GO TO 109
78      S1=(B(I+M)-B(I))/ (C(I+M)-C(I))
79      S2=(B(I)-B(I-M))/ (C(I)-C(I-M))
80      IF(S2.EQ.S1) GO TO 134
81      FLAG(I+M)=NOGOOD
82      109 NM=NM+1
83      GO TO 108
84      134 BSO(I)=S2
85      135 CONTINUE
86      C
87      136 CONTINUE
88      IF(IPT.LT.1) GO TO 306
89      C
90      C      SEARCH FOR THE BSO THAT IS DESIRED
91      C
92      DO 300 I=1,IPT
93      IF(FLAG(I).NE.NOGOOD) GO TO 101
94      IPT=IPT-1
95      IF(BSO(I).EQ.DBSO) GO TO 303
96      300 CONTINUE
97      C
98      303 IF(FLAG(IPT).NE.NOGOOD) GO TO 304
99      IPT=IPT-1
100     GO TO 303
101     304 REWIND J
102     IF(IPT.EQ.1) IPT=1
103     DO 310 X=1,KPT
104     305 READ(3) IC,CARD
105     IF(IC.NE.05) GO TO 305
106     310 CONTINUE
107     DECODE(CARD,27)E1,D2,NUMB,XT,HW,CT
108     XBACK=XBACK+XT*HW

```

01 10-20-79 10.001 5

```

009      TCT=TCT+CT
010      C
011      DO 350 I=1,NUMB
012      READ(3)IN,CARD
013      LINE=LINE+1
014      DECODE(CARD,320)OUT1,FSW,OUT2
015      320 FORMAT(A5,A10,A5)
016      C-----SND LINE MODIFICATIONS
017      DECODE(OUT1,323)XDRUM,LANDBASELWUP,XC,XVAL,ENST
018      323 FORMAT(2Z,A5)
019      XVAL=100000000*XDRUM+100000000*LANDBASELWUP+10000*XC+100*XVAL+ENST
020      WRITE(OUT,326)XLINE,XC,XVAL,ENST.
021      &
022      326 FORMAT(1X,I4,1X,I4,1X,I10,A4,A5)
023      IFLAG=1
024      IF(IFLAG,NO:1) GO TO 350
025      C-----SND LINE MODIFICATIONS
026      C
027      WRITE(OUT,325)LINE,XC,OUT1,OUT2
028      325 FORMAT(1E,20,1X,I4,1X,A20,A5)
029      LINE=LINE+1
030      IC=IC+3
031      WRITE(OUT,330)LINE,XC,OUT1,FSW
032      330 FORMAT(1X,20,1X,I4,1X,A20,4X,A16)
033      350 CONTINUE
034      C
035      IPT=1
036      REWIND 3
037      IF(XLINE,NO:1)GO TO 300
038      BSO(IPT)@BSO(NEXT)
039      B(IPT)@B(NIPT)
040      C(IPT)@C(NIPT)
041      NUMB@N
042      DO 355 I=1,20
043      355 FLAG(I)=0
044      WRITE(8)XC@C,CARD@C
045      GO TO 300
046      C
047      300 LINE=LINE+1
048      WRITE(OUT,440)LINE,TCT,XBACK
049      440 FORMAT(1X,I4,4X,I11,24X,"TOTAL COST",F13.0,2X,"REASON@BS@B"
050      &
051      F12.8)
052      LINE=LINE+1
053      WRITE(OUT,18)LINE
054      ENDDO
055      DO 450 I=1,20
056      450 FLAG(I)=0
057      GO TO 1
058      C
059      550 STOP
060      END

```

01 10-20-79 16.899 28

1970 EQUALITY OR NON-EQUALITY COMPASSION MAY NOT BE NEARLY AS SOCIAL IF...

Subroutine: GETDAT**Functions:**

This program reads output from GETBSO and retrieves corresponding LRU/SRU item description data from random file 08. It then outputs MOD-METRIC item description records with "IC2 = 1", and delivery schedule cards for later processing by the program EVALUATE.

Descriptions:

This routine is used in conjunction with the program SAVDAT. Program SAVDAT saves item description data on a random file for subsequent retrieval by this program.

Program GETDAT begins by reading MOD-METRIC input card types 92, 96, 98, and 99 from random file 08, and immediately writes these cards to the output file 09. It then reads the first "97" card to determine the number of input records (NCARD) that is needed to hold the flying hour data. Next, the flying hour data cards are read.

The program then reads an LRU/SRU levels card from file 05, and decodes the sequence number that was associated with this LRU/SRU by the subroutine SAVDAT. This sequence number, ISEQ, is then used to retrieve the appropriate

item description data from random file 08. This information is then used to output a MOD-METRIC item description record with "IC2 = 1" to file 09. Next, the program writes a "95" delivery schedule card to file 05. These steps are then repeated for each LRU/SRU record on file 05.

The output from this routine serves as input to the program EVALUATE. The program EVALUATE reads the total number of assets recorded on the delivery schedule card, and determines an optimal allocation of these assets among the bases.

7 08 40023-70 11/20/70 70

```

1  PROGRAMMING/CONF/NUMBER/0100010000)
2  GETDAT.S
3  C      THIS PROGRAM READS OUTPUT FROM CONSOLE,
4  C      AND RETRIEVE THE DATA FROM FILE 'QNT'.
5  C      IT THEN PRODUCES 'X0019' OUTPUT CARDS, AND DELIVERS
6  C      SCHEDULE CARDS FOR LATER PROGRAMMING BY
7  C      SUBTASKS.
8  C
9  C      CHARACTER ZCARD=00, ZCARD1=10, SCARDS=50
10 C      CHARACTER INCARD=7
11 C
12 C
13 C      SET RECORD SIZE FOR RANDOM FILE '00'
14 C
15 C      COBL NUMBER(00013)
16 C
17 C      BEGINN 00
18 C      INCR=0
19 C      IONT ' 0
20 C
21 C      READ 02,001,001000 AND 07 CARDS, IN THAT ORDER?
22 C
23 C      DO 4 SUBS
24 C      READ(001)INCARD
25 C
26 C      WRITE THESE CARDS TO FILE 9
27 C      WRITE(9,1)INCR=0
28 C      IONT=000001
29 C      3  FORMAT(1Y2)
30 C      4  CONTINUE
31 C
32 C      READ NEXT 'QNT' CARD, AND DETERMINE THE NUMBER OF
33 C      CARDS (NCARD) NEEDED TO HOLD THE NEXT 1000 DATA?
34 C
35 C      READ(001)INCR=0
36 C      READ(001)INCR=7,INCR=0
37 C      7  FORMAT(1Y2)
38 C      PRINT, " NUMBER= ", INCR
39 C      NCARD=INCR*100 + 1
40 C
41 C      10 CONTINUE
42 C      READ(001)INCR=001,INCR=0,INCR=0
43 C      23 FORMAT(1Y,1Y,1Y,1Y,1Y)
44 C      INCR=INCR+1
45 C
46 C      IF THIS IS NOT AN "X0019" CARD, GO TO 90
47 C
48 C      IF (ICR.EQ.1) GO TO 000
49 C
50 C      READ SEQUENCE NUMBER FOR THIS DATA SET?
51 C      THEN OBTAIN DESIRED ITEM DATA FROM FILE '00'
52 C

```

GETDAT.S

12 09 40023-79 17,800 28

83 DECODE(BCARD,03)X880
84 23 FORMAT(10X1R2)

85
86 XZXKXZXKX(1) GO TO 870
87 DECODE(BCARD,04)X880X7800X207L
88 23 FORMAT(10X1R2X1R2)
89 DECODE(BCARD,04)X880
90 24 FORMAT(10X1R2)

91
92 READ(10,100)XHCARD
93 DECODE(XHCARD,03)X8800
94 23 FORMAT(10X1R2)

95
96 XE THIS IS AN END GO TO 80
97 OTHERWISE, SEND THE '99' DATA FOR THIS LRU GROUP.

98
99 XZVXLEU78(1) GO TO 80
100

101
102 XH800X800XHCARD
103 GO GO XH1XHCARD
104 XH2XZH800X81
105 READ(10,100)XHCARD
106 XH2X800XHCARD
107 XH1XHCARD

108 GO CONTINUE

109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

999 CONTINUE
PRINT, " END OF RUN: ITEM RECORDS INPUT =", IREC
PRINT, " RECORDS OUTPUT =", IOUT
END

2 08 30-23-79 17,886 58

05 STOP
06 348

Subroutine: INTPRO

Function:

This program computes initial provisioning requirements according to logic specified in AFLCR 57-27.

Description:

This subroutine is called by the program DMSGN1 to compute initial provisioning requirements according to logic specified in AFLCR 57-27. The program was originally written by Mr. Terry Mitchell/AFALD/XRS as a CREATE Time-Sharing program. The program was subsequently modified for use in the Levels Computation System for use as a callable subroutine.

If the write flag IWT(3) equals 1, the routine prints out the details of its computations. The printed details are in the same form as an AFLCR 57-27 Recoverable Item Requirements Computation Worksheet, AFLC Form 614. An example of this form is presented in Figure II-2 of reference 1.

```

1  *NRUN=RIME/OBJ/R57=27.0(RGD,NOGO)
2  *R57=27,S
3  SUBROUTINE INTPRO(WUC,OST,COST,RMTBD,RNRTS,COND,BRCD,DRCD,
4  & PLT,OPA,NSN,AVGMP,PEAKP,ITER)
5  CHARACTER WUC*8,NSN*15,CARD*70,DELSCH*42,WUCPN*24
6  INTEGER C1,C2,C2H
7  COMMON/INT/IWT(5)
8  C
9  IDBUG=IWT(3)
10 12 FORMAT(V)
11 IF(IDBUG,GE,1)WRITE(6,12) '====INTPRO==',
12 & "INITIAL PROVISIONING CALCULATION"
13 IF(IDBUG,EO,1) WRITE(6,25) WUC,OST,COST,RMTBD,RNRTS,
14 & COND,BRCD,DRCD,PLT,OPA,NSN,
15 & AVGMP,PEAKP
16 25 FORMAT("0","WUC=",A9," OST=",F5.1," COST=",F11.2,
17 & " RMTBD=",F8.1," RNRTS=",F5.2," COND=",F5.2," BRCD=",
18 & F5.1," DRCD=",F5.1," PLT=",F5.1," OPA=",F5.1,/5X,
19 & "NSN=",A15,/10X,"AVGMP=",F8.0," PEAKP=",F5.2//)
20 IF(IDBUG,EO,1) WRITE(06,30) WUC,NSN
21 30 FORMAT("0",12X,"R57=27 COMP FOR WUC ",A9," NSN ",A15)
22 IQPA=QPA
23 IF(IDBUG,EO,1) WRITE(06,40) IQPA,COST
24 40 FORMAT("0",17X,I5,3X,F10.2)
25 DRCM=AINT( (DRCD/30.) * 10. + 0.5 ) / 10.
26 IF(IDBUG,EO,1) WRITE(06,50) PLT,BRCD,DRCD,DRCM,OST,AVGMP,PEAKP
27 50 FORMAT("0",10X,F5.0,F5.0,F5.0,F5.1,F5.0,F5.0,6X,F6.1//)
28 BPP=1.0-RNRTS
29 TOIM=AINT( (100./RMTBD) * 10000. + 0.5 ) / 10000.
30 TOIMD=AINT( RNRTS * TOIM * 10000. + 0.5 ) / 10000.
31 TOIMB= TOIM-TOIMD
32 WOP=AINT( RNRTS * COND * 10000. + 0.5 ) / 10000.
33 WOR=AINT( TOIM * WOP * 10000. + 0.5 ) / 10000.
34 IF(IDBUG,EO,1)
35 & WRITE(06,60) COND,BPP,RNRTS,TOIM,TOIMD,TOIMB,WOP,WOR
36 60 FORMAT("0",5X,F5.2,F5.2,F5.2,5F7.4//)
37 PCSL=1.0 * AVGMP * QPA
38 PCSLR=AINT( PCSL * WOR * 10. + 0.5 ) / 10.
39 PCSLD= PCSLR * COST
40 IF(IDBUG,EO,1) WRITE(06,70) PCSL,WOR,PCSLR,PCSLD
41 70 FORMAT("0",25X,F15.4,F10.4,F10.4,F10.2)
42 RLTP= PLT * AVGMP * QPA
43 RLTPR = AINT( RLTP * WOR * 10. + 0.5 ) / 10.
44 RLTPD= RLTPR * COST
45 IF(IDBUG,EO,1) WRITE(06,70) RLTP,WOR,RLTPR,RLTPD
46 DRCP= DRCM * AVGMP * QPA
47 DRCPR = AINT( DRCP * TOIMD * 10. + 0.5 ) / 10.
48 DRCPD= DRCPR * COST
49 IF(IDBUG,EO,1) WRITE(06,70) DRCP,TOIMD,DRCPR,DRCPD
50 IF(IDBUG,EO,1) WRITE(06,80)
51 80 FORMAT("0")
52 BRCP=AINT( (BRCD * PEAKP * QPA / 30. ) * 10000. + 0.5 ) / 10000.

```

```

53      BRCPR=AINI( BRCP * TOIMB * 10. + 0.5 ) / 10.
54      BRCPD= BRCPR * COST
55      IF(IDBUG.EQ.1) WRITE(06,70) BRCP,TOIMB,BRCPR,BRCPD
56      BOSTP=AINI( ( OST * PEAKP * QPA / 30. ) * 10000. + 0.5 ) / 10000.
57      BOSTPR=AINI( BOSTP * TOIMD * 10. + 0.5 ) / 10.
58      BOSTPD= BOSTPR * COST
59      IF(IDBUG.EQ.1) WRITE(06,70) BOSTP,TOIMD,BOSTPR,BOSTPD
60      IF(IDBUG.EQ.1) WRITE(06,80)
61      BSLR= BRCPR + BOSTPR
62      BSLD= BSLR * COST
63      IF(IDBUG.EQ.1) WRITE(06,90) BSLR,BSLD
64      90  FORMAT("0",50X,F10.4,F10.2)
65      TRR=AINI( PCSLR + RLTPR + DRCPR + BSLR + 0.5 )
66      TRD=TRR * COST
67      IF(IDBUG.EQ.1) WRITE(06,90) TRR,TRD
68      ITRR=TRR
69      RETURN
70      END

```

Subroutine: LEVLDP

Function:

This routine produces duplicates of levels records read in on file 07, and outputs the duplicates to file 08. The program is used whenever AFLCR 57-27 logic is used to compute initial provisioning levels.

Description:

Programs within the MOD-METRIC system produce a different set of stock levels for each given Buy Support Objective (BSO). In our study, five different BSOs were used, resulting in five sets of stock level records for each LRU and SRU being considered. However, a desired Buy Support Objective is not considered when AFLCR 57-27 logic is used. This program is employed to produce the same number of LRU/SRU stock level records for the AFLCR 57-27 computation logic as is produced whenever a MOD-METRIC computation routine is used.

The program begins by reading the "BSO-FILE" which contains the set of desired Buy Support Objectives to be evaluated in the current computational run. It reads this file and counts the number of BSO records on the file, NBSO. The routine then produces NBSO duplicates of each levels record read in from file 07. These duplicates are output to file 08. A counter in the part number identification field, representing the number of BSO being considered, is updated for each output record.

19T 01 10-20-79 15.518 JS

```

1 *SRUN=LEVIN/08/LEVLDP.0(SCD)*RINE/00/BSOFILE*05*.R
2 *LEVLDP.S
3 C
4 C THIS ROUTINE FIRST READS THE BSO-FILE, AND COUNTS
5 C THE NUMBER OF BSO-RECORDS; NBSO.
6 C
7 C THE ROUTINE THEN PRODUCES DUPLICATES OF LEVELS
8 C RECORDS READ IN ON FILE 07, AND OUTPUTS THE DUPLICATES
9 C TO FILE 08. A COUNTER, REPRESENTING THE PLAN, IS UPDATED
10 C ON EACH OUTPUT RECORD.
11 C
12 CHARACTER FIELD1*11, FIELD2*64
13 C
14 NBSO=0
15 5 READ(5,33,END=10)FIELD1
16 NBSO=NBSO+1
17 GO TO 5
18 C
19 C READ IN AND PRODUCE NBSO DUPLICATES OF THE LEVELS RECORDS.
20 C
21 10 CONTINUE
22 WRITE(6,18)NBSO
23 13 FORMAT(///20X,"NO. OF BSO'S 6"Y13,/)
24 C
25 REWIND 07
26 20 CONTINUE
27 READ(07,33,END=800)FIELD1,PLAN,FIELD2
28 FORMAT(11,2,A64)
29 IF(PLAN.LT.10)WRITE(6,35)FIELD1,PLAN,FIELD2
30 35 FORMAT(11,"07,I1,A64)
31 IF(PLAN.GE.10)WRITE(6,33)FIELD1,PLAN,FIELD2
32 C
33 C OUTPUT DUPLICATES
34 C
35 DO 60 PLAN=1,NBSO
36 IF(PLAN.GE.10)WRITE(8,33)FIELD1,PLAN,FIELD2
37 IF(PLAN.LT.10)WRITE(8,35)FIELD1,PLAN,FIELD2
38 60 CONTINUE
39 C
40 GO TO 20
41 C
42 C END OF RUN
43 C
44 800 CONTINUE
45 REWIND 08
46 WRITE(6,803)
47 803 FORMAT(120,"END OF RUN")
48 STOP
49 END

```

LEVLDP

Subroutine: LEVLPI

Function:

This program reads levels cards for quarter I, and outputs levels cards for quarters I and I+1.

Description:

In the Recoverable Item Management Evaluator, item stock levels are recomputed on a quarterly basis. Unfortunately, it was found that the MOD-METRIC computation system required very large amounts of computer time to determine stock levels for the item sets selected for consideration in this study. Consequently, to reduce compute time requirements to reasonable levels, it was decided to recompute stock levels on a six month basis, rather than quarterly. Program LEVLPI was developed to resolve interface problems associated with six months levels computations performed by the Levels Computation System, and the Quarterly Update calculations performed in the RIME simulation model.

The function of this program is to produce duplicates of input LRU/SRU stock level records. Specifically, the routine first reads a levels card for quarter I from file 07. It then writes out to file 08 a copy of this card, followed by an identical card with coding appropriate for quarter I+1. In RIME, levels are still recomputed quarterly; however, identical values are read for quarters 1 and 2, 3 and 4, and so on.

```
1 *#RUM=RIHE/GRJ/LEVL1,S(BCD)RIHE/GO/LEVELS.C*07",B;
2 *# RIHE/HOLD*08",W
3 *LEVL1,S
4 C
5 C READ LEVELS CARD FOR QUARTER I.
6 C OUTPUT LEVELS CARDS FOR QTR I AND I+1.
7 C
8 CHARACTER LEVLCD*77
9 C
10 IDBUG=1
11 CALL XPARAM(1,132)
12 IREC=0
13 C
14 REWIND 07
15 REWIND 08
16 C
17 10 CONTINUE
18 C
19 READ(7,23,END=800) LEVLCD
20 23 FORMAT(A77)
21 IREC=IREC+1
22 IF(IDBUG.GE.1)PRINT,IREC,LEVLCD
23 C
24 DECODE(LEVLCD,33)IQTR
25 33 FORMAT(T16,I2)
26 WRITE(8,23)LEVLCD
27 C
28 NQTR=IQTR+1
29 IF(NQTR.LT.10)ENCODE(LEVLCD,53)NQTR
30 IF(NQTR.GE.10)ENCODE(LEVLCD,33)NQTR
31 C
32 53 FORMAT(T16,"0",I1)
33 C
34 C WRITE OUT LEVELS CARD FOR QTR I+1=NQTR
35 C
36 WRITE(8,23)LEVLCD
37 GO TO 10
38 C
39 800 CONTINUE
40 PRINT," LEVL1--*--RECORDS PROCESSED=",IREC
41 STOP
42 END
```

LEVL1

Subroutine: METINP**Function:**

This routine estimates Mean Time Between Demands (XMTBD), NRTS rates (XNRTS), and condemnation rates (CONS) using historical D041 histories.

Calling Parameters:

- IS = The quarter number identifying the most recent historical period to be used in rates estimation. (Note: Quarter 1 represents the oldest quarter, while quarter 16 represents the most recent quarter for which data is available).
- IE = Quarter number identifying the oldest historical period to be used for rates estimation.
- J = The item index identifying the particular LRU or SRU for which rates are to be developed. J = 1 denotes the LRU, J = 2 denotes the first SRU, J = 3 denotes the second SRU, and so on.

Description:

Subroutine METINP begins by computing the total base reparable generations (ITBRGN), total LRU program (ITPROG), the total number of NRTS events

(ITNRTS), and the total number of base condemnations (ITBCND) occurring during the quarters IS through IE, inclusive. It then computes values for Mean Time Between Demands (XMTBD(J)), the NRTS fraction (XNRTS(J)), and the base condemnation fraction (CONS(J)) for item number J. To prevent "fatal" computation errors in subsequent MOD-METRIC routines, each of these values are compared to upper and lower bounds, and set to the appropriate bounds if they are out of range. If XMTBD(J) exceeds 99999, it is reset to this value. Similarly, if XMTBD(J) is less than 1, it is also set to 99999. This latter case implies that there has been no program for item number J, and consequently the estimated MTBD is set to a very large number.

Similarly, if there have been no base generations for item number J during the interval IS through IE, the estimated NRTS fraction is set to $XNRTS(J) = 0.01$. If there have been at least some base rep gens in past periods, but no NRTS actions have been recorded, the estimated NRTS fraction is also set to 0.01. Hence, the estimated NRTS fraction is never allowed to be less than 0.01. This is because the MOD-METRIC program TWOIND will not accept as input any item with a NRTS fraction less than this value.

95T 01 10-20-79 13.870

```

1      SUBROUTINE METINP(I3,I2,J)
2      COMMON/IBRGN/IBRGN(16,40)
3      COMMON/ITPROG/ITPROG(16,40)
4      COMMON/ITNRTS/ITNRTS(16,40)
5      COMMON/ITCND/ITCND(16,40)
6      COMMON/XMTBD/XMTBD(40)
7      COMMON/XNRTS/XNRTS(40)
8      COMMON/CONS/CONS(40)
9      COMMON/IWT/IWT(20)
10     C
11     IDBUG=IWT(4)
12     C      INITIALIZE INTERMEDIATE SUMS:
13     ITBRGN=0
14     ITPROG=0
15     ITNRTS=0
16     ITCND=0
17     C      COMPUTE INTERMEDIATE SUMS.
18     C      USE LRU PROGRAM IN XMTBD CALCULATIONS,
19     C      SINCE IT IS ASSUMED THAT SRU ARE PECULIAR
20     C      TO THE LRU. (THIS PREVENTS INCONSISTENCIES
21     C      THAT MIGHT CREEP IN IF THE PROGRAM DATA IS
22     C      INCONSISTENT.)
23     DO 10 J IWT3,I2
24     ITBRGN=ITBRGN+IBRGN(I,J)
25     ITPROG=ITPROG+ITPROG(I,J)
26     ITNRTS=ITNRTS+ITNRTS(I,J)
27     ITCND=ITCND+ITCND(I,J)
28     10 CONTINUE
29     C      CONVERT ITPROG TO HOUR.
30     ITPROG=ITPROG*100
31     C
32     IF(IDBUG.GE.1)WRITE(6,103)I3,I2,IWT3,I2,IPTROG,ITBRGN,
33     & ITNRTS,ITCND
34     103 FORMAT('-----METINP PROGRAM RESULTS FOR QTR ',I2,
35     & ' THRU QTR ',I2/' I20:',IPTROG',I10,' I2BRGN',
36     & ' I10,' ITNRTS=',I10,' ITCND=',I10)
37     C
38     C      COMPUTE MEAN TIME BETWEEN DEMANDS.
39     C
40     IF(ITBRGN.GT.0) GO TO 150
41     XMTBD(J)=99999.
42     GO TO 175
43     150 CONTINUE
44     XMTBD(J)=FLOAT(ITPROG)/FLOAT(ITBRGN)
45     IF(XMTBD(J).GT.99999.0) XMTBD(J)=99999.0
46     IF(XMTBD(J).LT. 1.) XMTBD(J)=99999.0
47     C      COMPUTE RATE RATE (NOT NEARLY THIS STATION.)
48     175 IF(ITBRGN.GT.0) GO TO 200
49     XNRTS(J)=0.01
50     GO TO 210
51     200 CONTINUE
52     XNRTS(J)=FLOAT(ITNRTS)/FLOAT(ITBRGN)

```

METINP

ST 01 10-20-79 12.870

```

53      IF(XWRITS(J),LT.0.01) XWRITS(J)=0.01
54      C
55      C   COMPUTE CONDENSATION RATE (FUNCTION OF WRTS RATE)
56      C   210 IF(ITWRTS.GT.0) GO TO 300
57      C   CONS(J)=0%
58      C   GO TO 310
59      C   300 CONTINUE
60      C   CONS(J)=FROAT(ITBCND)/FLOAF(ITWRTS)
61
62      C   310 CONTINUE
63      C   IF(IDBUS.GE.1)WRITE(6,313)J,SHRBD(J),XWRTS(J),CONS(J)
64      C   313 FORMAT(1X,I2," ESTIMATES=","T=","Y,"
65      C   " XRTSD=","F8.3," XWRTS=","F8.3," CONS=","
66      C   "F8.3)
67      C
68      C   RETURN
69      C   END

```

Subroutine: ONEIND

Function:

This MOD-METRIC program computes stock levels using assumptions embedded in Sherbrooke's original METRIC model.

Description:

This program computes optimal depot and base stock levels ignoring LRU/SRU relationships. The computational logic and input/output requirements for the program are defined in AFLCP 57-13, and will not be discussed further here.

The RIME version of this routine is identical with the original MOD-METRIC program with two exceptions. First, the input program has been modified to read values of the levels computation variables IMINSK, BOMIN, and JEQBAS from the "99" program parameter card. If JEQBAS = 1, the flying hour and order and ship time data by base is reset to the average values summed over all bases. These changes are implemented in program lines 1531-1545.

The second change involves implementation of upper and lower bounds on the stock levels computation. If IMINSK = 1, the expected number of assets in the repair/resupply pipeline is used as a lower bound on the computed stock level. Also, in this case, no additional stock is allocated to a base after the expected backorders at that base is reduced below BOMIN/NBASES. These stock level limitation calculations are implemented in program lines 2571-2605 and 2952-2963.

```

1 *ISUM=IYHM/655/ONEIND.O(BCD,NOB0)
2 *ONEIND.S
3 C      NAME=NET, VERSION 3
4 C      COMPILED FILE IS MODNETR2/0BSINGLE
5 C
6 DIMENSION PH(23),OST(23),BRT(100),DRT(100),DS(15,100),
7 &      KNO(100),CNRNU(100),YMTNF(100),YMTNF(100),APF(100),
8 &      TBACK(20),BBACK(100,23,3),CCOST(3),WLRUD(100,3),ADBLT(23),
9 &      JSTCK(23),HRAWN(100),WLRUD(100,23,3),DELTA(1000),GN(25),
10 &      BAKO(20),BUDG(20),RATE(20),LENE(100),MONTHS(12),
11 &      CON(100),PLT(100),FO(100),YTOT(100),WNO(100)
12 &      TPNOB(28),TPILL(23),FWWT(23),PNOB(100,3),FILL(100,3)
13 C
14 REAL LERT,LAGRAN,LAGR(2)
15 C
16 INTEGER NDC(17),ID(100,6),ZACC(100),NSN(100,4),IDNSN(6)
17 C
18 DATA KEY/6H267,AQ/
19 DATA NEN/400*4H /
20 DATA LENS/106*4H /,MARK/6H /,KBLNK/4H /
21 DATA MONTHS/9H JAN ,5H FEB ,5H MAR ,5H APR ,5H MAY ,
22 &      5H JUN ,5H JUL ,5H AUG ,5H SEP ,5H OCT ,5H NOV ,5H DEC /
23 C
24 CHARACTER CARD*72,NAME
25 C
26 DIMENSION IDCD(9)
27 C
28 I97=1
29 I98=1
30 I99=1
31 INDC=0
32 PRD=3
33 INOJ=
34 NUNAB=100
35 MAXSTK=1000
36 CALL ISNT(I2NUNB,IDCD)
37 CALL DETA (I2ATN)
38 5 READ(4,6,ERR=9,END=9)NAME
39 IF(NAME,NE,"LAST ")GO TO 5
40 BACKSPACE 4
41 WRITE(6,7)IDATE,ISNUNB,IDCD
42 6 FORMAT(1X,A6)
43 7 FORMAT(1X,ONEIND ",1X,I6,1X,I5(1X,9A6)
44 8 FORMAT(1X,"LAST ")
45 WRITE(6,8)
46 9 CALL DETCH(4,ISTAT, )
47 IYR=IDATE/10000
48 IKK=IYR*10000
49 IHO=(I2ATN-IKK)/100
50 IKK=IKK+IHO*100
51 IDAY=I2ATN-IKK
52 IHO=MONTHS(IHO)

```

ONEIND

T 01 10-20-79 18.389 JS

```

53      C
54      C      DETERMINE TYPE OF INPUT
55      1 IF(I99.EQ.1.AND.I98.EQ.1.AND.I97.EQ.1)GO TO 400
56      READ(5,2)IC1,IC2,CARD
57      2 FORMAT(2E1,1X,A72)
58      IF(IC1.EQ.9)GO TO 4
59      WRITE(6,3)I99,I98,I97
60      3 FORMAT(5H1I99=,I2,5X,4HI98=,I2,5X,4HI97=,I2,10X,5HERROR/)
61      STOP
62      C
63      C      PROGRAM PARAMETERS-- 92,97,98,99
64      4 I9=0
65      GO TO(4,92,1,1,1,1,97,98,99),YC2
66      C
67      C      92-- HEADING/TITLE FOR TOP OF PAGE
68      92 DECODE(CARD,192)HDG
69      192 FORMAT(17A4)
70      GO TO 1
71      C
72      C      97-- FLYING HOURS & ORDER AND SHIP TIME PER BASE
73      C
74      97 I97=1
75      C
76      C      INPUT EQUAL BASE FLAG(YE@BAS) AND
77      C      NUMBER OF BASES(NBASES).
78      DECODE(CARD,197) NBASES
79      197 FORMAT(I3)
80      IF(NBASES.GT.23)GO TO 897
81      IB=1
82      LB=MINO(6,NBASES)
83      297 DECODE(CARD,197)IPH(I),OST(Y)FI=IB,LB)
84      397 FORMAT(13X,6(P6.0;P3.0))
85      IF(LB.EQ.NBASES)GOTO 597
86      IB=IB+8
87      LB=MINO(LB+6,NBASES)
88      READ(5,2)IC1,IC2,CARD
89      IF(IC1.EQ.9.AND.YC2.EQ.7)GO TO 297
90      496 WRITE(6,497)
91      497 FORMAT(28H0FLYING HOUR CARD INCOMPLETE)
92      STOP
93      597 FLYTOT=0
94      AOST=0
95      ADDRT = 0.0
96      OHIN = 1.E35
97      DO 697 K=1,NBASES
98      AOST=AOST+OST(K)
99      OHIN = AOHIN ( OST(K).OHIN )
100     697 FLYTOT=FLYTOT+PH(K)
101     DO 797 K=1,NBASES
102     ADDRT = ADDRT + OST(K) * PH(K)
103     797 PHWT(K)=PH(K)/FLYTOT
104     ADDRT = ADDRT / FLYTOT = OHIN

```

T 01 10-20-79 13.989 JS

```

57 512 FORMAT('0NOUT',I2,' IPRWT',I2,' IPRCH',I2,' IPRSO',I2)
58 WRITE(NOUT,515)
59 515 FORMAT(2ENOBASE PLY HRS OST )
60 DO 520 K=1,NBASES
61 520 WRITE(NOUT,525)K,PH(K),OST(K)
62 525 FORMAT(1X,I4,4F9.0)
63 530 WRITE(NOUT,535)
64 535 FORMAT(//6X,12NWORK PART,10X,
65 & 43HALC UNIT MEAN TIME(HRS) REMOVALS PER,3X,
66 & 43NRTE COND REPAIR TIME PROCUR LEAD OPN,3X,
67 & 63SYSTEM./
68 & 6X,CUNIT NUMBER","/NSW",4X,
69 & 43NCODE COST BETWEEN DEMANDS 100 PLY HRS,3X,
70 & 36NRATE RATE BASE DEPOT TIME(MONTHS) ," HA",4X,
71 & 8NPIPELISE//)
72 C
73 C-----BEGIN XITE CHANGES FOR JEOPAS
74 IF(JEOPAS.NE.1)GO TO 599
75 C
76 C SET PARAMETERS FOR EQUIL BASE CALCULATIONS
77 C
78 DO 598 N=1,NBASES
79 PHWT(N)=ABPH/PLYTOT
80 PH(K)=ABPH
81 OST(K)=EOST
82 598 CONTINUE
83 599 CONTINUE
84 C-----END OF JEOPAS CHANGES
85 NU=0
86 PIPCST=0
87 RATE(1)=0.
88 TOTBDD=0
89 BAKO(1)=0.
90 BUDG(1)=0.
91 BSOLST=1.E36
92 CHAX=1
93 CHIN=1.E80
94 BUDINC=0
95 C
96 600 NU=NU+1
97 IF(INOB.NE.0)GO TO 2580
98 615 READ(5,2,END=9999)IC1,IC2,CKR9
99 IF ( IC1.LE.5 ) GO TO 605
00 GO TO(11,605,605,605,605,605,605,615),IC1
01 C
02 9999 INOJ=1
03 IF(NU.EQ.1)GO TO 2580
04 NU=NU-1
05 GO TO 600
06 615 IF(IC2.EQ.1)GO TO 605
07 I9=1
08 NU=NU-1

```


T 01 10-20-79 17.389 JS

```

09      GO TO 2000
10      C
11      C      11-- BASIC DATA
12      C
13      11 IF(IC2.NB.1)GO TO 800
14      DECODE(CRRD,11)(ID(NU,I),I=1,6),CPERU(NU),
15      & YMTBF(NU),YNRYS(NU),CON(NU),NAPP,BRT(NU),DRT(NU),
16      & PLT(NU),ALC(NU),PC(NU)
17      111 FORMAT(6A4,3X,F7.0,F6.0,F6.2,F3.2,Y2,3F3.0,A1,AU)
*W 1282 INCOMPATIBLE V.9 FIELD IN 'P' SPECIFICATION
*W 1222 THE CHARACTERS PRECEDING ABOVE FORMAT ENFOR ARE ,F3.2.
18      DRT(NU) = DRT(NU) + ADDRT
19      YDUMMY=YNRYS(NU)
20      CDUMMY=CON(NU)
21      APP(NU)=MAX0(NAPP,1)
22      R=100.*APP(NU)/YMTBF(NU)
23      IF(YNRYS(NU).LE.0)GO TO 180
24      CON(NU)=CON(NU)*YNRYS(NU)
25      PNU=CON(NU)*PLT(NU)*30.*CFAE/YNRYS(NU)
26      DNU=PNU+(YNRYS(NU)-CON(NU))*DRT(NU)/YNRYS(NU)
27      GO TO 185
28      C
29      180 YNRYS(NU)=CON(NU)
30      PNU=600
31      DNU=600
32      185 PIPE=' '
33      C
34      DO 211 K=1,NBASES
35      TT=(1.+YNRYS(NU))*BRT(NU)+YNRYS(NU)*(DNU+OST(K))
36      PIPE=PIPE+TT*APP(NU)*PH(K)/YMTBF(NU)*30.)
37      211 CONTINUE
38      PIPCST=PEPCST+PIPE*CPERU(NU)
39      BAKO(1)=BAKO(1)+PIPE
40      C
41      C
42      700 WRITE(ROUT,705)NU,(ID(NU,I),I=1,6),ALC(NU),CPERU(NU),YMTBF(NU),
43      & R,YDUMMY,CDUMMY,BRT(NU),DRT(NU),PLT(NU),
44      & APP(NU),PIPE
45      705 FORMAT(1X,I3,1X,6A4,1X,A1,F6.0,F12.1,9X,F6.5,4X,2F5.2,86?0U
46      & 2F7.0,8X,F3.0,3X,F6.1)
47      C
48      CMAX=AMAX1(CMAX,CPERU(NU))
49      CHIN=AMIN1(CHIN,CPERU(NU))
50      ICHIN=0
51      PLT(NU)=PNU
52      DRT(NU)=DNU
53      BUDINC=BUDINC+CPERU(NU)
54      C
55      IF(NU.EQ.NUMAX)GO TO 2000
56      GO TO 600
57      C
58      800 IF(IC2.NB.6)GO TO 605

```

T 01 10-20-79 11.989 3

```

59      J=NU+1
60      DECODE(CGRD,805)IDNSN
61      805 FORMAT(6E4)
62      DO 806 I = 1,6
63      806 IF(IDNSN(I) .NE. ID(J,I))GO TO 605
64      DECODE(CGRD,810)(NSN(J,I),I=1E4)
65      810 FORMAT(25X,4E4)
66      WRITE(MONT,815)(NSN(J,I),I=1,6)
67      815 FORMAT(19X,4E4)
68      GO TO 805
69      C
70      2000 IF(MU.EQ.0)GO TO 500
71      TOTBUD=PEPCST*BSTART
72      WRITE(MONT,2002)PEPCST,BSTART,TOTBUD
73      2002 FORMAT(20HOTAL PIPELINE      ,F12.17
74      &      20H      TIMES BSTART =,F19.8/20X,2(6H-4m---)/
75      &      20H STARTING BUDGET =,F12.8//)
76      WRITE(MONT,2005)BAKO(1)
77      2005 FORMAT(10X EXPECTED BASE BACKORDERS WITH ZERO STOCK =,F9.8)
78      CAVG=2.*BUDINC/LOAT(MU)
79      BUDINC=AMIN1(BUDINC*.5,PBINC*PEPCST)
80      BUDINC=AMAX1(CAVG,BUDINC)
81      DO 2010 K=1,MU
82      AA=PRD*APP(J)/(YHTBF(J)*30.6)
83      UDD=0.
84      DO 2040 N=1,NBASES
85      2040 UDD=UDD+PH(K)*AA*YNRTS(J)
86      RDD=UDD*BETA1
87      PDD=PRD/(BETA*ORT(J)+BPRD)
88      BACKD=RDD*(1.0-PDD)/PDD
89      DEV=SQRT(BACKD/PDD)
90      IDEV=DEV
91      MNIGN=BACKD+1.5*DEV+1.5
92      MNIGN=MAX0(MNIGN,12)
93      IF(MNIGN.LE.MAXSTK) GO TO 2060
94      WRITE(8,2050) J,BACKD,MAXSTK-5
95      2050 FORMAT(10X DEPOT PIPELINE FOR LRU =,I6,  ,F8.2,
96      &      10X  DEPOT MAX STOCK SET TO ,I6)
97      MNIGN=MAXSTK
98      2060 CONTINUE
99      NLOW=BACKD-DEV
100     NLOW=MAX0(NLOW,MAXSTK-IDEV)
101     NLOW=MIN0(NLOW,1)
102     C-----v-----w-----x-----y-----z-----RENT CHANGES
103     C
104     C      IF ZHINK=1, SET MIN DEPOT STOCK LEVEL EQUAL TO
105     C      THE EXPECTED PIPELINE
106     IF(INISH.NE.1) GO TO 2007
107     IBACKD=BACKD+.5
108     NLOW=MAX0(NLOW,IBACKD)
109     MNIGN=MIN0(NLOW,MNIGN)
110     2007 CONTINUE

```

BT 01 11-20-79 11.389 JS

```

311 C
312 CALL ADENSE(DD,RDD,BACKD7MLOW,MHIGH,DELTA)
313 C
314 C IF INHOK=1, SET MHIGH TO FIRST LEVEL WITH
315 C A DEPT DELAY FACTOR(DELTA)← 0.0001
316 C
317 IF(INHOK.NE.1) GO TO 2069
318 DO 2066 I=MLOW,MHIGH
319 IF(DELTA(I).LT.0.0001) GO TO 2066
320 2064 CONTINUE
321 GO TO 2069
322 C RESET MHIGH
323 C
324 2066 MHIGH=I
325 MHIGH=MAX0(MLOW+1,MHIGH)
326 2069 CONTINUE
327 C-----SND LINE CHANGES
328 C
329 MLIH=MHIGH-MLOW
330 MRANGE(J)=MLIH
331 DO 2070 I=1,MLIH
332 2070 DS(I,J)=DELTA(MLOW+I-1)
333 2010 KENO(J)=MLOW
334 EPS=1.E-30
335 KSAFE=0
336 NPTS=1
337 2020 LAGR(1)= -1.0/CHAX
338 C ENTER WITH INCREMENTED BUDGET.
339 2030 LAGR(2)=0.
340 TCOST(1)=0
341 TCOST(2)=0
342 DO 2180 NBIS=1,NBIS
343 LAGRAN=(LAGR(1)+LAGR(2)) * 0.5
344 GN(KBIS)=LAGRAN
345 TCOST(3)=0
346 DO 2150 J=1,NU
347 CLAGRA=LAGRAN*CBERU(J)
348 LCRIT=CLAGRA+1.
349 WHIN=9999.
350 RR=(1.+YPTS(J))*BRT(J)
351 AA=PRD*APP(J)/(YHTBP(J)+30.0)
352 MLIH=MRANGE(J)
353 DO 2130 I=1,MLIH
354 W=0
355 DO 2110 K=1,NBASES
356 TT=RR * YPTS(J)*(OST(K)+DS[I,K]*BRT(J))
357 PE=SPRD/(BETA*TT+SPRD)
358 RE=FN(K)*AA*BETA1
359 RI=RE-1.
360 P1=1.-PE
361 HE=PE*RE
362 BACKN=RE*P1/PE

```

T 01 10-20-79 10.989 S

```

63      NCUN=HE
64      STCKM=0
65      C-----LINE CHANGES
66      C          IF MINIMUM STOCK FLAG (I@INSK)=1, SET MINSTK
67      C          EQUAL TO THE EXPECTED PIPELINE
68      BBOLIN=@ONIN/FLOAT(NBASES)
69      MINSTK=0
70      IF(I@INSK.EQ.1)MINSTK=BACKM+ 0.5
71      2090 CONTINUE
72      IF(STCKM.LT.MINSTK) GO TO 2098
73      IF(NCUN.GE.LCRITY) GO TO 2100
74      C
75      IF(BACKM.LT.BBOLIN)GO TO 2106
76      C
77      C-----END LINE CHANGES
78      C
79      IF(HE.ELEPS)GOTO 2100
80      2093 CONTINUE
81      STCKM=STCKM+1.0
82      HE=(STCKM+1)*P1*HE/STCKM
83      BACKM=BACKM+NCUN-1.
84      NCUN=NCUN+HE
85      GO TO 2090
86      2100 CONTINUE
87      W=W+BACKM=CLAGRA*STCKM
88      JSTCK(N)=STCKM
89      TBACK(N)=BACKM
90      TPNOB(N)=NCUN
91      TPILL(N)=NCUN-HE
92      2110 CONTINUE
93      X@HO=X@HO(J)+I@HO=2
94      W=W-LA@RA@N+C@PERU(J)*X@HO
95      IF(W.GE.WMIN) GO TO 2130
96      WMIN=W
97      WLRUB(J,3)=X@HO
98      PNOB(J,3)=0.
99      FILL(J,3)=0.
100     DO 2120 N=1,NBASES
101     WLRUB(J,N,3)=JSTCK(N)
102     BBACK(J,N,3)=TBACK(N)
103     PNOB(J,3)=PNOB(J,3)+TPNOB/N*FRHT(N)
104     FILL(J,3)=FILL(J,3)+TPILL/N*FRHT(N)
105     2120 CONTINUE
106     2130 CONTINUE
107     LSUR=WRND(J,3)
108     DO 2140 N=1,NBASES
109     2140 LSUR=LSUR+WLRUB(J,N,3)
110     TCOST(3)=TCOST(3)+FLOAT(LSUR)*C@PERU(J)
111     2150 BOWTINUE
112     IF(ICMEN.GT.0)GO TO 2155
113     IF(TCOST(1).GT.0)GO TO 2155
114     LAGR(1)=1/CMIN

```

BT 01 10-20-79 10.289 JS

```

415      ICHIN=1
416      GO TO 2080
417      2155 CONTINUE
418      L=1
419      IF(TCOST(3),ST.TOTBUD) L=2
420      LAGR(L)=LAGRAN
421      DO 2170 J=1,NU
422      WLRUD(J,L)=WLRUD(J,3)
423      PNOBO(J,L)=PNOBO(J,3)
424      FILL (J,L)=FILL (J,3)
425      DO 2160 N=1,NBASES
426      WLRUN(J,N,L)=WLRUN(J,K,3)
427      2160 BBACK(J,N,L)=BBACK(J,K,3)
428      2170 CONTINUE
429      TCOST(L)=TCOST(3)
430      2180 CONTINUE
431      IF(IWRITE,GT,3)WRITE(ROUT,2190) (CH(X),X=1,NBIS)
432      2190 FORMAT(13H0MULTIPLIERS=/1X,40B(2,3))
433      L=1
434      IF(TCOST(2),LE,0.)GOTO 2280
435      IF(TCOST(1),LE,0. .OR. TCOST(N)-TOTBUD:LT,TOTBUD=TCOST(1)) L=2
436      2200 WRITE(ROUT,2210)
437      C
438      2210 FORMAT(///,95(6H *****))
439      WRITE(ROUT,2220)
440      2220 FORMAT(5X,47HSUMMARY OF REQUIREMENTS, DISTRIBUTION AND COSTS)
441      WRITE(ROUT,2230)
442      2230 FORMAT(15(6H *****))
443      WRITE(ROUT,2240)
444      2240 FORMAT(6X,4HUNIT,4X,11HPART NUMBER,5X,3HALC,3X,4HCOST,
445      & 3X,5HTOTAL,2X,5HDEPOT,2X,3HROW,JX,5HBASES)
446      BMIN=0
447      CT=0
448      DO 2245 J=1,NU
449      LTOT(J)=WLRUD(J,L)
450      DO 2246 N=1,NBASES
451      BMIN=BMIN+BBACK(J,K,L)
452      2244 LTOT(J)=LTOT(J)+WLRUN(J,K,L)
453      2245 CT=CT+LTOT(J)*CPRU(J)
454      IF(CT,LE,BDD(BPTS))GOTO 2370
455      BSO=(BMIN-SAKO(BPTS))/(CT*BDD(BPTS))
456      IF(BSO,LT,BBOLST)GOTO 2370
457      C
458      2248 IF(ISO,NE,0)WRITE(ISO,2250)BSO,ISD(1,1)FISD(4,2),NU,BMIN,CT
459      C
460      2250 FORMAT(1X,3H88 ,E12,5,2A4,I3,E12,8,F12:07
461      DO 2280 J=1,NU
462      CH=FLOAT(LTOT(J))*CPRU(J)
463      KC=(FLOAT(WLRUD(J,L))*.999)/WPT(0)/DET(J)
464      KD=WLRUD(J,L)-KC
465      TTFNB=0
466      TTFIL=0

```

T 01 10-20-79 10.569 JS

```

67      XHTBF=0.0
68      IF(IBSO.NE.0)WRITE(IBSO,2270)(ID(J,I),I=1,6),(NSH(J,I)KI*1X4),
69      &      ALC(J),CH,LTOT(J),KD,KC,
70      &      (NLRUB(J,K,L),K=1,NBASES)
71      2270 FORMAT(1X,3H11,6A4,4A4,A1,F9.0,14,2E3,25I3)
72      C
73      2280 WRITE(NOUT,2290)J,(ID(J,I),I=1,6),ALC(J),CH,LTOT(J),KD,KC,
74      &      (NLRUB(J,K,L),K=1,NBASES)
75      2290 FORMAT(1X,23,2X,6A4,12,A1,12,F9.0,17,2I6,23I3)
76      C
77      WRITE(NOUT,2300) CT
78      2300 FORMAT(//,38X,16H***** TOTAL COST,F12.0)
79      WRITE(NOUT,2310) BMIN
80      2310 FORMAT(38X,16H***** BACKORDERS,F12.6)
81      WRITE(NOUT,2320)BSO
82      2320 FORMAT(38X,17H***** BUY SUP OBJ,E12.5)
83      DO 4221 J=1,NU
84      XHTBF = XHTBF + 1./YHTBF(J)
85      TTPNB=BNOBO(J,L)*1./YHTBF(J) + TTPNB
86      TTPIL=PIAL(J,L)/YHTBF(J)+TTPIL
87      4221 CONTINUE
88      WRITE(NOUT,4226)TTPNB/XHTBF:TTPIL/XHTBF*100.
89      4226 FORMAT(38X,17H***** PROB O BAKK,FS.4/
90      &      38X,17H***** FILL RATE =FPA.6)
91      IF(IERRT.GT.1)WRITE(NOUT,2321)
92      &      (J,PNOBO(J,L),FILL(J,L),J=1,NU)
93      2321 FORMAT('0LNU PROBABILITY OF ENO BACKORDERS FILL RATE')
94      &      10d(1X,I3,45X,F5.3,15X,F5.3A) )
95      IF(IERRT.LT.6)GOTO 2328
96      DO 2328 J=1,NU
97      JJ=NLRUB(J,L)-KNO(J)+2
98      2322 PDSO(J)=NSH(JJ,J)
99      WRITE(NOUT,2324)(J,PDSO(J),J=1,NU)
100     2324 FORMAT(/9 PROBABILITY OF BESO8 STOCKOUT FOR EACH ITEM'A
101     &      8(13(25,F5.2)/))
102     2328 CONTINUE
103     C
104     IF(NPTS.EQ.2)GOTO 2330
105     IF((CT*BUDG(NPTS))/(BAKO(NPTS)-BMIN).LT.
106     &      (BODS(NPTS)-BUDG(NPTS-1))/(BAKO(NPTS-1)-BAKO(NPTS)))GOTO 2340
107     2330 NPTS=NPTS+1
108     KSAH=0
109     2340 BUDG(NPTS)=CT
110     BAKO(NPTS)=BMIN
111     RATE(NPTS)=TTPIL/XHTBF
112     BSOLST=BSO
113     CALL PTIME(ITIME)
114     IF(IERRT.GT.1)WRITE(NOUT,2350)ITIME
115     2350 FORMAT('OBLASSED COMPUTE TIME',I9,11H MILLISECS.)
116     IF(NPTS.EQ.5)GOTO 2360
117     IF(LAGE(L).EQ.BUSTOP)GO TO 2390
118     IF(NPTS.EQ.2)GOTO 2390

```

BT 01 10-20-79 10.389 JS

```

519      C
520      2360 TOTBUD=TOTBUD+BUDINC*FLOAT(NBMSBS)
521          LAGR(1)=MIN1(LAGR(1),LAGR(2))
522          GOTO 2030
523      2370 KSAME=KSAME+1
524          BUDINC=BUDINC+BUBINC
525          WRITE(NOUT,2190)(GM(I),I=1,NBYS)
526          IF(KSAME.LT.5)GOTO 2360
527          WRITE(NOUT,2380)
528      2380 FORMAT(31HNO PROGRAM LOOPING WITH SAME COST )
529      C-----TIME CHANGE
530          IF(NPTS.GE.2) GO TO 2390
531      C-----END CHANGE
532          GOTO 500
533      C
534      C
535      2390 WRITE(NOUT,2400)(NDG(I),I=1,10),BAKO(2),BAKO(2),BUDG(2)
536          & ,RATE(2)
537      2400 FORMAT(1E1,14X,29HPLOT OF BACKORDERS VS. BUDGET,5X,10AS/
538          & 11H BACKORDERS,72X,10HBACKORDERS,5X,6HBUDGET,2X,6HPZLARIS/
539          & 1X,F10.4,3H -X,69X,F8.4,F13.0,2X,F8.4)
540          IF(IPNCH.EQ.0)GO TO 2430
541      C-----TIME CHANGE-----STATEMENTS MOVED HERE FROM
542      C FOLLOWING STATEMENT 2005
543          IF(IPNCH.NE.0)WRITE(IPNCH,2420)(ID(1,I),I=1,2),BUDG(1),BAKO(2)
544          & ,RATE(1)
545      C-----END CHANGE
546          DO 2410 I=2,NPTS
547      2410 WRITE(IPNCH,2420)ID(1,I),ID(1,2),BUDG(I),BAKO(I),RATE(3)
548      2420 FORMAT(1X,A4,A2,F11.0,2F10.6)
549      2430 LAST=1
550          NP=2
551          DL=(BAKO(2)-BAKO(NPTS))/49.6
552          DK=(BUDG(NPTS)-BUDG(2))/99.6
553          DO 2520 I=3,NPTS
554          NEXT=(BAKO(2)-BAKO(I))/DL + 075
555          NB=NEXT-EAST+1
556          LAST=NEXT
557          IF(NB.LT.1)GOTO 2470
558          IF(NB.GT.70)GOTO 2520
559          DO 2460 J=1,NB
560          WRITE(NOUT,2440)
561      2440 FORMAT(12X,1H=)
562          IF(NP.GE.NPTS)GOTO 2460
563          NP=NP+1
564          WRITE(NOUT,2450)BAKO(NP),BUDG(NP),RATE(NP)
565      2450 FORMAT(1E+02,2X,F8.4,F13.0,2X,F8.6)
566      2460 CONTINUE
567      2470 NK=(BUDG(I)-BUDG(2))/DK + 1.5
568          LINE(NK)=MARK
569          IF(NB.LT.0)GOTO 2490
570          WRITE(NOUT,2480)BAKO(I),LINE

```

01 10-20-79 10.989 5

```

71 2480 FORMAT(1X,F10.4,2X =,100A1,4H)
72 LINE(NR)@KBLNK
73 GOTO 2510
74 2490 WRITE(ROUT,2490) LINE
75 2500 FORMAT(1X=.12X,100A1)
76 LINE(NR)@KBLNK
77 2510 IF(NR.NE.NPTS)GOTO 2520
78 NP=NR+1
79 WRITE(ROUT,2510)BAKO(NP),SUBGZNPY,NAME(NP)
80 2520 CONTINUE
81 WRITE(ROUT,2530)
82 2530 FORMAT(1X.10(10H.....IT/32X,1NX,5(19X,1NX))
83 B2=DK*80.0 + BUDG(2)
84 B3=DK*60.0 + BUDG(2)
85 B4=DK*80.0 + BUDG(2)
86 B5=DK*60.0 + BUDG(2)
87 WRITE(ROUT,2540)BUDG(2),B2,B3,B4,B5,BUDG(NPTS)
88 2540 FORMAT(7H BUDGET,F10.0,5F20.0)
89 WRITE(ROUT,2550)DL,DK
90 2550 FORMAT(12HOLINE VALUE,F7.5Y50,13HCOLUMN VALUE+,F8.1)
91 IF(IPNCH.NE.0)WRITE(IPNCH,2560)
92 2560 FORMAT(1H1,48X)
93 GO TO 500
94 C
95 2580 WRITE(ROUT,2590)
96 2590 FORMAT(12HEND OF DATA)
97 STOP
98 END
**W 1470 EQUALITY OR NON-EQUALITY COMPARISON MAY NOT BE MEANINGFUL IN LOGICAL IF EX
**W 7 MEMORY EXPANDED. USE $LIMITS OR $ORIG= OPTION FOR NEXT RUN

```


Subroutine: OVHSTL

Function:

This routine computes overhaul stock levels using a 14-day supply rule.

Calling Parameters:

IQTR = The quarter number for which stock levels are to be
 computed.
JJ = The item number for which stock levels are to be
 computed.
IOVSTL = The computed overhaul stock level.

Description:

This routine first computes the total number of depot reparable generations for item number JJ which will be generated during the 4 quarter interval beginning with quarter IQTR. It then sets the computed overhaul stock level IOVSTL equal to a 14-day supply based upon the computed demand rate.

01 10-20-79 13.871

```

1      SUBROUTINE OVHSTL(IQTR,JJ,IQVSTL)
2      COMMON/IDRGN/IDRGN(16,40)
3      COMMON/TW2/TW2(20)
4      C
5      IDBUG=INT(6)
6      C
7      C      COMPUTE OVERHAUL STOCK LEVEL.
8      C
9      TDD=0.0
10     IS=IQTR
11     IF(IS.GT.13) IS=13
12     IE=IS+3
13     DO 100 I=IS,IE
14     TDD=TDD+IDRGN(I,JJ)
15     100 CONTINUE
16     C-----OYST IS SET TO 18.0 -----
17     OYST=18.0
18     IOVSTL=(TDD/365.0)*OYST +0.5
19     C
20     IF(IDBUG.LE.1)WRITE(6,103)JJ,IQTR,IS,IE,TDD,OYST,
21     IOVSTL
22     103 FORMATT/5---OYHSTL=ITEN=9,X2," OTR=Y,I3," IS=,I3,
23     IE=,I3," TDD=,F10.08
24     & OYST=,25.0," IOVSTL=,I5}
25     C
26     RETURN
27     END

```

OVHSTL

Subroutine: SAVDAT

Function:

This program saves item description data on random file 08. This information is subsequently retrieved by program GETDAT using a sequence number assigned by program SAVDAT.

Description:

This program is called prior to a ONEIND levels computation.

The program reads MOD-METRIC input records from file 05, and assigns a sequence number ISEQ to each incoming data card. It then writes each input card to the random file 08 using ISEQ as an index for the random file. The index ISEQ is then inserted in to the part number field of the input card, and the modified input record is written as output to file 09. This process continues until all records in the input file had been read, stored, and modified.

Outputs to file 09 are subsequently used as input to the ONEIND levels computation program. Later in the job processing stream, the sequence number ISEQ is used to retrieve information from the random file 08, and to relate it to computed stock levels. For additional discussions of this process, see the description of program GETDAT.

01 10-20-79 14.896 JS

```

1 *BRUN=MAIN9/002/SAVDAT.O(BCD,NO007
2 *SAVDAT.S
3 C THIS PROGRAM SAVE ICD=1 DATA ON RANDOM FILE "08".
4 C THIS INFORMATION IS THEN RETRIEVED BY PROGRAM
5 C GETSAT.S FOLLOWING GETSSO. THE SEQUENCE NUMBER
6 C "ISSO" IS USED AS ONE DATA RETRIEVAL KEY.
7 C
8 C
9 CHARACTER ICARD*72,ICARD1*10,ICARD2*50
10 C
11 C SET RANDOM FILE RECORD SIZE (SEE P. 6-33 OF FORTRAN
12 C MANUAL FOR DETAILS)
13 C
14 CALL NANSZ100,127
15 C
16 IREC=0
17 C
18 10 READ(5,23,END=900)ICARD
19 IREC=IREC+1
20 23 FORMAT(578)
21 C
22 DECODE(ICRD,93)IC1,IC2,ICRD1,ICARD2
23 93 FORMAT(2I1,1X1A10,3X,A50)
24 C
25 RECORD ICARD2 DATA ON FILE "08" FOR RECALL AFTER
26 C GETSSO ROUTINE
27 C
28 ISSO=ISSO+1
29 WRITE(6,1220)ICARD
30 C
31 IF(IC2.NE.1) GO TO 50
32 IF(IC1.NE.9) GO TO 50
33 C
34 C INSERT SEQUENCE NUMBER IN PART NUMBER FIELD.
35 C AND OUTPUT THE RECORD. THIS NUMBER WILL BE
36 C USED TO RETRIEVE ONE ICARD DATA AFTER
37 C GETSSO.
38 C
39 WRITE(9,43)IC1,IC2,ICARD1,ISSO,ICARD2
40 43 FORMAT(2I1,1X1A10,1I3,A50)
41 GO TO 40
42 C
43 C OUTPUT RECORD AS READ
44 C
45 50 CONTINUE
46 WRITE(9,23)ICARD
47 C
48 80 CONTINUE
49 GO TO 40
50 C
51 C END OF FILE REACHED, STOP PROGRAM
52 C

```

SAVDAT

01 10-20-79 14.898 CS

3 900 CONTINUE
4 WRITE(6,903) "====SAYDAT==== END OF FILE REACHED AFTER",
5 " PROCESSING " ,INED," INPUT CARDS"
6 903 FORMAT(I)
7 STOP
8 END

P 01 10-20-79 10.206 28

EXT DATE 08-16-78 USE 3/1

	ELAPSED TIME (SECT)	BYTES/ MINUTE
REHEAD	.18	
USE 1	.79	36766
USE 2	.00	
USE 4	.08	39871
USE 5	.30	11267
TOTAL	.67	5126
AL TIME	.70	

THE WERE NO DIAGNOSTICS IN ABOVE COMPILATION
5K WORDS WERE USED FOR THIS COMPILATION

Subroutine: SPNDMS

Function:

This TimeSharing-Program is used to generate the Job Control Language statements required to utilize the Recoverable Item Management Evaluator System.

Description:

This program is used to simplify the preparation of run request for the RIME system. For a detailed discussion of the functions of this routine, see the User Instructions Chapter of Volume I.

```
*$RUN=$SPNDMS.0(ASCII)
```

```
*SPNDMS.S
```

```
DIMENSION ISTAT(2)
```

```
CHARACTER CARD*72
```

```
CHARACTER SIMCAR*50
```

```
CHARACTER ANS*1
```

```
DIMENSION RP(10,13)
```

```
DATA RP/1.,0.,0.,1.,0.,0.,0.,0.,0.,0.,
```

```
2 3.,1.,0.,3.,1.,0., 0.,0.,0.,0.,
3 1.,0.,1.,1.,0.,1., 0.,0.,0.,0.,
4 1.,0.,0.,1.,0.,0., 1.,0.001,1.,0.001,
5 3.,1.,1.,3.,1.,1., 1.,0.001,1.,0.001,
6 2.,0.,0.,2.,0.,0., 0.,0.,0.,0.,
7 2.,0.,0.,2.,0.,0., 1.,0.01,1.,0.01,
8 4.,0.,0.,4.,0.,0., 0.,0.,1.,0.001,
9 2.,0.,0.,3.,1.,1., 0.,0.,1.,0.001,
10 2.,0.,0.,3.,1.,1., 1.,0.01,1.,0.001,
11 4.,0.,0.,1.,0.,0., 0.,0.,1.,0.001,
12 2.,0.,0.,1.,0.,0., 0.,0.,1.,0.001,
13 2.,0.,0.,2.,0.,0., 1.,0.01,1.,0.001/
```

```
RP(J,I)=J-TH PARAMETER FOR INVENTORY POLICY I.
```

```
1 CONTINUE
```

```
SIMCAR="$;TAPE:07,X9D,,70053,,EXOGF"
```

```
READ OUTPUT CONTROLS
```

```
PRINT,"AC? PUNCH? SIMULATE? MORE CORE? (1=YES, 0=NO)"
```

```
IMICRO=0
```

```
READ,IAC,IPNCH,ISIMU,ICORE
```

```
5 IF(ISIMU.NE.1) GO TO 10
```

```
PRINT," IS --",SIMCAR
```

```
PRINT," OK FOR EXOGFILE? (Y=YES)"
```

```
READ,ANS
```

```
IF(ANS.EQ."Y")GO TO 10
```

```
PRINT," INPUT EXOGFILE CONTROL CARD"
```

```
READ(5,3)SIMCAR
```

```
3 FORMAT(A50)
```

```
PRINT," CARD WAS READ AS--"
```

```
PRINT,"====",SIMCAR
```

```
PRINT," IS CARD OK?(Y OR N)"
```

```
READ,ANS
```

```
IF(ANS.NE."Y") GO TO 5
```

```
10 CONTINUE
```

```
READ SIMULATION PARAMETERS
```

```
16 CONTINUE
```

```
PRINT 13," NFGRP, NLGRP,INQTR,NDHIS, CPU-LIMIT"
```

```
READ(5,13) NFGRP, NLGRP,INQTR,NDHIS,ICPULM
```

```
IF(ICPULM.GE.10) GO TO 17
```

```
PRINT," CPU-LIMIT MUST BE =>10"
```

```
GO TO 16
```

```
17 CONTINUE
```

```
ITOT=NDHIS+INQTR
```

```
IF(ITOT.LE.16) GO TO 19
```

```
SPNDMS
```


PRINT, "NUMISTINGWK CANNOT EXCEED 10. STOP RUN"
STOP

67

19 CONTINUE

-----CREATE A TEMPORARY FILE 09-----

30 CONTINUE

CALL CREATE(9,1,0,ISTAT(1),)
IF(ISTAT(1).NE.0)GO TO 9000

PRINT 11

11 FORMAT(///)

read ident of run desired

PRINT 13,"IDENT?"

READ(5,13)I

set run parameters

IDENT=I

IMETH=RP(1,I)

IEQBAS=RP(2,I)

ICOST=RP(3,I)

KMETH=RP(4,I)

KEQBAS=RP(5,I)

KCOST=RP(6,I)

IMINSKI=RP(7,I)

BOMINI=RP(8,I)

KMINSK=RP(9,I)

BOMINK=RP(10,I)

READ MANAGEMENT METHOD CODES

PRINT 13," IDENT IMETH IEQBAS ICOST KMETH KEQBAS KCOST"

PRINT 14, IDENT, IMETH,IEQBAS,ICOST,KMETH,KEQBAS,KCOST

13 FORMAT(V)

14 FORMAT(I7,6I7)

PRINT 13," IMINSK BOMINI KMINSK BOMINK"

PRINT 15,IMINSK,BOMINI,KMINSK,BOMINK

15 FORMAT(I7, F10.3,I7,F10.3)

-----BUILD BATCH CARD DECK-----

IF(IAC.EQ.1) WRITE(9,1013)

1013 FORMAT("010##NORM,R(AC)")

IF(IAC.NE.1) WRITE(9,1014)

1014 FORMAT("010##NORM")

WRITE(9,1023) IDENT,NFGRP,NLGRP,IMETH,IEQBAS,ICOST,

KMETH,KEQBAS,KCOST

,IMINSK,BOMINI,KMINSK,BOMINK

1023 FORMAT("011\$:IDENT:WF1462,XRS/DEMMY-----",

I2,"***",I2,"*",I2,"*", "---",6I2/

"012\$:NOTE:-----",2(I2,"-",F6.4,"=="))

IF(ICPULM.LT.100)WRITE(9,1026)ICPULM,ICPULM

IF(ICPULM.GE.100)WRITE(9,1027)ICPULM,ICPULM

1026 FORMAT(

"018\$:LIMITS:",I2,"",9K"/

"020\$:PARAM:",I2/

"023\$:SELECTA:RIME/GO/DMSGN.E1")

1027 FORMAT(

"018\$:LIMITS:",I2,"",9K"/

```

004      *023*:SELECTA:RIME/GO/DMSGN.E1*)
10C
20C      WRITE DMSGN1 PARAMETERS TO FILE 09
30C
40      WRITE(9,33)IDENT, IMETH,IEQBAS,ICOST,KMETH,KEQBAS,KCOST
50 33  FORMAT('025*',8I4)
60      WRITE(9,34)IMINSK,BOMINI,KMINSK,BOMINK
70 34  FORMAT('026*',2(I5,F10.5))
80      WRITE(9,23)NFGRP, NLGRP,INQTR,NDHIS
90 23  FORMAT('027*',8I5)
00C
10C-----OBTRAIN MOD-METRIC PARAMETERS FROM RIME/DMS.CTL
20C
30      WRITE(9,37)
40 37  FORMAT('030*:SELECTA:RIME/DMS.CTL')
50C
60C      OBTAIN FLYING PROGRAMS FROM RIME/F111FH.D
70C
80      IF(NFGRP.LE.13)WRITE(9,38)
90 38  FORMAT('032*:SELECTA:RIME/F111FH.D')
00      IF(NFGRP.GT.13)WRITE(9,39)
10 39  FORMAT('032*:SELECTA:RIME/F15FH.D')
20C
30C-----GENERATE REMAINDER OF JCL
40C
50      WRITE(9,1008)
60 1008 FORMAT('035*:SELECTA:RIME/GO/DMSGN.E2')
70C
80      WRITE(9,1018) IMETH
90 1018 FORMAT('040*:NOTE:***** INITIAL PROV--IMETH=',I2)
00      WRITE(9,1028) IMETH
10 1028 FORMAT('050*:SELECTA:RIME/GO/INPRV.E',I1)
20      WRITE(9,1038) KMETH
30 1038 FORMAT('060*:NOTE:***** REPLENISHMENT--KMETH=',I2)
40      WRITE(9,1043)KMETH
50 1043 FORMAT('070*:SELECTA:RIME/GO/REPLN.E',I1)
60      WRITE(9,1053)
70 1053 FORMAT('080*:NOTE:*****SORT LEVELS')
80      WRITE(9,1063)
90 1063 FORMAT('090*:SELECTA:RIME/GO/SORTL.E1')
00      IF(IPNCH.EQ.1) WRITE(9,1213)
10 1213 FORMAT('1200*:NOTE:*****PUNCH LEVELS',
20      &/, '1210*:SELECTA:RIME/GO/PUNCH.E1')
30C
40      IF(ISIMU.NE.1)GO TO 1400
50      IF(ICORE.EQ.0) WRITE(9,1313)SIMCAR
60 1313 FORMAT('1300*:NOTE:*****RUN RIMSIM SIMULATION'/
70      '1310*:SELECTA:RIME/GO/RISMS.E2'/
80      '1320',A50)
90      IF(ICORE.NE.0)WRITE(9,1314)SIMCAR
00 1314  FORMAT('1300*:NOTE:*****RUN RIMSIM BIG MODEL'/
10      '1310*:SELECTA:RIME/GO/RISMS.E4'/
20      '1320',A50)
30C
40 1400 CONTINUE
50C
60      IF(IMICRO.EQ.1)WRITE(9,1414)
70 1414 FORMAT('1400*:NOTE:*****COPY P1 FOR MICROFICHE'/
80      &'1410*:SELECTA:RIME/GO/MICRO.E1')
90      WRITE(9,9003)
00 9003 FORMAT('100*:ENDJOB')
10C
20C-----SPAWN THE BATCH COMPILE-----
30C
40      )
50      ENDFILE 09

```

```

) REWIND 09
) CALL CALLSS("RUN 09#", "CDIN")
)C
)C-----DETACH THE TEMPORARY FILE-----
)C
) CALL DETACH(09, ISTAT, )
)C
)C-----CONTINUE ? ? ?
)C
) PRINT 13, "CONTINUE?(Y OR N?)"
) READ(5, 13)ANS
) IF(ANS.EQ."Y") GO TO 30
)C
) PRINT, " DO ANOTHER GROUP?(Y OR N)"
) READ(5, 13)ANS
) IF(ANS.EQ."Y")GO TO 1
)C
) STOP
)C
)C
) 9000 CONTINUE
) PRINT, "THIS PROGRAM USES FILE 09 AS A TEMPORARY FILE, BUT"
) PRINT, "FILE 09 ALREADY EXISTS."
) PRINT, " DO A --REMO CLEARFILES--"
) STOP
) END

```

Subroutine: .TWOIND

Function:

The program TWOIND implements the MOD-METRIC two-indenture stock level computation model.

Description:

This program implements the MOD-METRIC two-indenture stock level computation model. Input requirements and computation logic for this routine is described in AFLCP 57-13, and will not be discussed further here. The RIME version of this routine is identical with the AFLCP 57-13 version with two exceptions. First, subroutine INPUT has been modified to read in values for the stock level computation variables IMINSK and BOMIN from the "99" input record. The value of the MOD-METRIC variable BSTOP is then set to the input value for BOMIN whenever the bounds flag IMINSK equals 1. Otherwise, no change to BSTOP is made. These changes are implemented in program lines 3061-3102.

The second RIME modification is implemented in subroutine OUTPUT. In this routine, if the bounds flag IMINSK equals 1, the stock levels computed by TWOIND are compared to the expected number of assets in the repair/resupply pipeline. If a stock level computed by TWOIND is below this value, the corresponding stock level is reset to the expected number of assets in the pipeline. On the other hand, if IMINSK is not equal to 1, no bounds calculations are performed.

No other changes to the program TWOIND were made.

01 10-26-79 18,450

```

1 SUBROUTINE INPUT
2 C-----
3 COMMON/LIMITS/INTNSK(20000)
4 C-----
5 COMMON /XOPT/ NORT?RUMOD,INLCOM,SEASBY,ENSTIN,OUTOR
6 LOGICAL RUMOD,INLCOM?YHXSK,ENSTIN,OUTOR
7
8 C
9 COMMON /XBSZ/ PLTMIN,PLTMAX,XTONIX,XIONAX,PRRSON
10 DATA PLTMIN /1,235/, PLTMAX /1,235/, XTONIX 9,23,500/,
11 XIONAX /33,500/
12 C
13 COMMON /BASE/ NBASIS,MBUS,MBPLYOT,OST,SEASRDATA1,PRD,SPD,
14 EBREV,PBINC,START,STGLD,STWVW,HOSTOP,OSTON
15 DIMENSION PH(30),OST(30),IBREV(30)
16
17 C
18 COMMON /LRUS/ CLRU,CHTAD,YNRS,RTLRU,RTSRU,PLR,ENNOBK
19
20 C
21 COMMON /SRUS/ CSRUCOM,CSRW,HRDDBY,XHDD,INTS,RTSRU,DTOR,SPRT,
22 ENNOBK,PCFRT
23 DIMENSION CSRU(45),XHTB(65),XRTB(45),SRUSU(45),DTORU(65),
24 SPLT(45),INFORK(45),PCTRW(45)
25
26 C
27 COMMON /CHAB/ HDB,IBAT,INO,ITR,IDLRU,INRSH,RSR,RSRVA,SC,SKLC
28 & IDEND,TCT,ENT
29 CHARACTER HDB*62,INDB,IDLRU*24,IBRU*24(45),ROR*46,SPR*26(45)
30 & ALCT,ALCT1(45),EDERD(9),KBT*6,CTCT*6(3),ENH*6(3)
31 DATA KEY / 6HH-TWH /
32
33 C
34 COMMON /MOUT/ TARGET,MOUT,WHIS,XPRNT,ZRUCH,ITELB
35 DATA TARGET / 0,0 /
36
37 C
38 COMMON /LCOM/ PSGLI,PSGLT
39 DIMENSION PSGLI(45)
40
41 C
42 DIMENSION TRAC(30),ENHVR(30),SACK(30),SW(65),REV(3),
43 & STKL(80),STACH(30),ENH(30),
44 & ADLX(30),H(30),P(30),B(30),
45 & HAV(30),HAY(30),PAV(30),SAV(30),
46 & RBSAV(30),
47 & FLSAV(30),
48 & ABSAV(30),T(30),TRAV(30)
49
50 C
51 C
52 C
53 CHARACTER CARD*77,IS*28,ISPR*28,EDSPEN*28
54 & BLANK16*16,HTBD*16,HTBR*16,IOSTRS,IRRUVE*3
55 DATA BLANK16 / 16R /
56 DATA ISLWS / 18Q /, ISLWH 7 1R /
57 DATA HTBR / 18HBTWHEN ENOVKLS /, HTBD / 16H BETWEEN ENHDS /
58 DATA IOST / 5R OST /, IARBU / 5RARBU /
59
60 C
61 DIMENSION SPR(45),SUC(45),SPSD(45),SSD(45),CON(45),
62 & SPO(45)

```

TWOIND/INPUT

1 02 10626-79 1R,250

```

13      INTEGER OSL(8),OSL(8),LSP(8),LSP(48,8)
14      C
15      REAL CERR2,LCPIT,LACHIN,LACHMAX
16      C
17      DATA IFF,INBO,REOJ,INWSTU,ISUCRU /870/
18      DATA NODT,ISUBT,IPHON,IPXLD 7 6, 870 /
19      DATA NETA,REETA,ERRBO,STOLD,STKOP,CPAG,PHINC
20      / 3.0 1.011111, 23.0 1.0% 0.0% 1.0% 0.05 /
21      DATA PRD,MAXTRU / 30,285 /
22      DIMENSION OPTST(5),TESTIN(5)
23      DATA OPTST / 6,NOHND% UNXNCON, UNXNBSN, UNXNCSN, ENDSTOCK /
24      LOGICAL OPTION(2)
25      EQUIVALENCE ( OPTION(1),ROUNDD )
26      DATA SBANKS / 68 /
27      DATA HTLEV / 0 /, HOURS / 100 /
28      C *****
29      C
30      C          DETERMINE TYPE OF INPUT
31      IF ( NSBUS,GT,0 ) GO TO 3
32      C
33      1 IF ( XFOJ,GT,0 ) GO TO 1090
34      READ (5,2)PHO=999,ERRBO=0,IC1,IC2,CARD
35      2 FORMAT(2X1)139A77)
36      IF ( NETERU,LE,0 ) GO TO 3
37      IF ( IC1,LE,0 .OR. IC2,GT,9 ) GO TO 1
38      IF ( IC1,GT,9 .AND. IC2,LE,9 ) GO TO 1
39      3 GO TO 140230,30640759200576,007901,IC1
40      C
41      C          PROGRAM PARAMETERS= 92.98498699
42      C
43      90 IF ( INWSTU,GE,1 ) GO TO 1000
44      IF(IC1,GT,0)GO TO 704
45      C          907- PROGRAM CONTROL OPTIONS CARD
46      C
47      DECODE(CARD,929) ( NWSWT(XOPT),IOPF=1,NOFF )
48      929 FORMAT(12I8,1X)
49      DO 930 IOPF = 1,NOFF
50      930 OPTION(IOPF) = ,TABLE:
51      KOPT = 0
52      IZHOPT = 0
53      DO 931 IOPF = 1,NOFF
54      IF ( TESTIN(IOPF)=SBANKS ) 931,932,931
55      C
56      931 DO 932 JOPT = 1,NOFF
57      IF ( TESTIN(IOPF)=OPTST(JOPT) ) 933,932,931
58      932 OPTION(JOPT) = ,TABLE:
59      WRITE(6,940) OPTST(JOPT)
60      KOPT = KOPT + 1
61      GO TO 934
62      933 CONTINUE
63      C
64      IF ( ZEROPT,LE,0.0 ) WRITE(6,940)

```

02 10426-79 18,450

```

5      IERRPT = IERRPT + 1
6      WRITE(8,948) TESTIN(IERRPT)
7      934 CONTINUE
8      C
9      935 IF ( IERRPT * IERRPT ) 936-936-938
10     936 WRITE(8,937)
11     937 FORMAT(" *****/" "***** BLANK PROGRAM CONTROL CARD".
12     & " * ENCOUNTERED"/" ***** STANDARD OPTIONS SERV/
13     & " *****//")
14     GO TO 942
15     938 WRITE(8,940)
16     IF ( IERRPT ) HERR = HERR
17     IF ( IERRPT ) IERR = IERR + 1
18     IF ( IERRPT ) IERR = IERR + 1000
19     LOFF = 0
20     DO 939 IERR = 1, IERRPT
21     IF ( .NOT. OPTION(IERR) ) GO TO 939
22     WRITE(8,948) OPTION(IERR)
23     LOFF = LOFF + 1
24     TESTIN(LOFF) = OPTTEST(IERR)
25     939 CONTINUE
26     940 FORMAT("NO *****/NO INVALID/ON PROGRAM/ON CONTROL/
27     & " ON OPTIONS/")
28     941 FORMAT(5X,0)
29     944 FORMAT("PROGRAM/ON CONTROL/ON OPTIONS/")
30     942 GO TO 1
31     901 GO TO (91,92,93,94,95,96,97,98,99),IC2
32
33     C
34     C          91-- DATA FILE CONTROL CARD
35     C
36     91 GO TO 1
37
38     C
39     C          92-- HEADING/TITLE FOR TOP OF PAGERECORDS
40     C
41     92 DECODE(CARD,192)NDS
42     192 FORMAT(A89)
43     GO TO 1
44
45     C
46     93 DECODE(CARD,93339) VARGET
47     93339 FORMAT(F12,0)
48     94 GO TO 1
49     95 GO TO 1
50     96 GO TO 1
51
52     C
53     C          97-- FLYING HOURS AND UNDER/SHIP TIME PER BASE
54     C
55     97 I97=1
56     DECODE(CARD,197)NBASES
57     197 FORMAT(I3)
58     IF(NBASES.GT.30)GO TO 4070
59
60     C
61     DO 975 IBS = 1,NBASES

```

T 02 10026-79 10,150

```

57 975 IBERE(INS) = IBS
58
59 B
60 IB=1
61 LB=MINO(6,UBASES)
62 297 DECODE(CARD,397)(PH(Z),OST(Z),YU(Z),LB)
63 397 FORMAT(IZ)0(26.0,P3,0))
64 IF(LB,UB,UBASES) GO TO 497
65 IB=IB+6
66 LB=MINO(IB+6,UBASES)
67 READ(UB2)IC1,IC2,CERR
68 IF(IC1)IB,0.02,IC2,UB,(V) GO TO 1080
69 GO TO 297

```

```

70 C
71 497 FLYTOT=0
72 ADDR = 0.0
73 ONIN = FLYMAX
74 ONAX = FLYMIN
75 DO 597 IBS=1,UBASES
76 ONIN = MIN( ( OST(IBS),ONIN )
77 ONAX = MAX( ( OST(IBS),ONAX )
78 597 FLYTOT=FLYTOT+PH(IBS)
79 IF ( ONIN,UB,ONAX ) GO TO 1
80 DO 598 IBS = 1,UBASES
81 598 ADDR = ADDR + OST(IBS) * PH(IBS)
82 ADDR = ADDR / FLYTOT
83 ADDR = ADDR - ONIN

```

```

84 C
85 C
86 C
87 C
88 DETERMINE MAX # BASES WITH BENE FLYING HONEY

```

```

89 NBASES = 1
90 IF ( NBASES,LT,2 ) GO TO 1
91 NBASE = NBASES = 1
92 DO 2052 NBS = 1,NBASE
93 NBASE = 1
94 NBASE = NBS + 1
95 DO 2056 NBS = NBASE,NBASES
96 IF ( PH(NBS)PH(NBS) ) 2057,2053,2057
97 2053 IF ( OST(NBS)-OST(NBS) ) 2057,2055,2057
98 2054 IF ( IBERE(NBS)NBS ) 2056,2055,2056
99 2055 IBERE(NBS) = NBS
100 2056 NBASE = NBASE + 1
101 2057 CONTINUE
102 2058 CONTINUE
103 NBASES = MAX( NBASES,NBASE )
104 2059 CONTINUE
105 GO TO 1

```

```

106 C
107 C
108 C
109 C
110 98-b OUTPUT SPECIFICATIONS

```


Z 01 10626-79 1R1350

```

59 YNATE=INHOURE*APP/YNTSD
60 WRITE (HOUR,421)NDOP,IBAY,IMOCITY
61 421 FORMAT(1H1A69/50X,12NDATE OF RUN ,13,A5,2H19;12)
62 WRITE(ROUT;521)NDOP,IBAY,IBSTOP,IBBASES,CYAC,IBENC,IMJNSK,BBONIN
63 521 FORMAT(6H0515=,1375X,8HIBETA,7F5.3,5X,
64 8 8HIBSTOP,7F4.2/ 7H 848930;1375X,8HCFAC=,7F3.
65 8 2X,8HIBINC=,7F5.3, 7HIMSK =",16," BONIN =",7F6.4)
66 WRITE(ROUT; 324)HOUR,IBPNT,IBPCH,IBFILE
67 324 FORMAT(8H0515=,1227 7HIBPNT="12," 7HIBPCH="12," 7HIBSO=",12)
68 IF(CYAC;LE,0)WRITE(ROUT;929)
69 929 FORMAT(9552,750 CONSTRUCTION STOCK COMPUTED"/)
70 WRITE(ROUT;621) IORT
71 621 FORMAT(15H0515 72X NRS ,6X,A5)
72 DO 721 IBS=1,IBBASES
73 WRITE(ROUT;824) IBS,PR(IBS),OUT(IBS)
74 IF ( XBASEX ) WRITE(ROUT;722) XNRS(IBS)
75 722 FORMAT(1H+126X,13)
76 721 CONTINUE
77 821 FORMAT(1X;14,2F9.0)
78 WRITE(ROUT;622) FLYTOT
79 822 FORMAT(8X;7H----- / OH TOTAL . 88.0)
80 WRITE(ROUT;221) NTSD,INOURS
81 221 FORMAT(1X;16HWORK PART NO.,6X,
82 8 6SHALC UNIT MEAN TIME(HRS) REMOVAL PER,3X,
83 8 40HRTS COND REPAIR TIME PROCUR LEAD QFA,/
84 8 5X,11HUNIT NSN,12X
85 8 14HCODE COST,2X,82031678H FLY NRS,3X,
86 8 40HRTS RATE BASE DEPT TIME(MONTHS) WEA )
87 WRITE(ROUT;320)IDLEU,SEC,CLRU,INTSD,TRATEXNRS,
88 8 COND,DETERU,DELEU,PLT,APP
89 320 FORMAT(5H0LRU ,A24,2H19,0,F12.1,9X,F6.5,9X,2H5,2,
90 8 F6,0,2F7.0,8X,F3,0/HN SRU)
91 IF(YNATS;LE,0.)GO TO 322
92 CONL=CONL*YNTS
93 PR=CONL*PR*30,*CYAC/YNTS
94 DNTLRU = DNTLRU + APP
95 DNTLRU=PLT*(YNTS-CONL)+DNTLRU/YNTS
96 GO TO 823
97 322 YNATS=CONL
98 PLT=60,
99 DNTLRU=60,
100 323 YNTSD=INTSD/APP
101 GO TO 1
102 GO TO 1
103 GO TO 1
104 GO TO 1
105
106 8
107 8
108 25 DECODE(CARD;125)OSL,APP
109 125 FORMAT(25X)4I2,4X;4E369X;8Y1)
110 IF ( DSTOCK ) DECODE(CARD;124) LNWOK

```

25-4 LRU PHASE PROVISIONING AND RELEVERE SCHEDULE

P 01 10-26-79 18.250

```

11      124 FORMAT(5HX,I3)
12      GO TO 1
13
14      C
15      26-7 LRU FEDERAL STOCK # & UNIQUE DISTRT
16
17      26 DECODE(CARD,126)IDFEN,NSR,NSRNU
18      124 FORMAT(A24,1X,A16,2X,PA,1)
19      IF(IDLRU,NE,IDFEN)GO TO 1
20      IF(NSRNEW,LT,0)NSRNEW=1
21      WRITE(ROUT,226)FSM
22      226 FORMAT(13X,A16/)
23      GO TO 1
24      27 GO TO 1
25      28 GO TO 1
26      29 GO TO 1
27
28      C
29      30 GO TO(31,32,33,34,35,36,37,38,39)XC2
30
31      C
32      31-4 SRU BASIC DATA
33
34      31 IF(1SRU,LT,1)GO TO 1050
35      IF ( 127,LE,0 ) GO TO 1040
36      NSRU=NSRU+1
37      IF(NSRU>GE,MAXSRU)GO TO 1050
38      J=J+1
39      1SRU=1
40      DECODE(CARD,131)IDSRU(NSRU),BUC(NSRU),SPSD(NSRU),SSD(NSRU),
41      & CNU(NSRU),XNTD(NSRU),XNRU(NSRU),CON(NSRU),NAPP;
42      & DTRU(NSRU),DTRU(NSRU),SPC(NSRU),SPLC(NSRU),SPO(NSRU)
43      131 FORMAT(A24,3A17,7,0,6,7,PA,2,9,8,12,3,3,0,A1,A4)
44      *W 1282 INCOMPATIBLE W.D FIELD IN PI SPECIFICATION
45      *W 1222 THE CHARACTERS PRECEDING ABOVE FORMAT ERROR ARE .P&T.
46      C *****
47      C *****
48      SAPP=MAXO(NAPP,1)
49      XRATE=ENOUS*SAPP*ENR/XNTD(NSRU)
50      IF ( 10,GE,M ) XRATE = 0.0
51      PGLI(NSRU) = XRATE / XRATE*(1-XRTE)
52      FSUM=FSUM+XRATE
53      WRITE(ROUT,231)J,ISSU(NSRU),XSLC(NSRU),CNU(NSRU),
54      & XNTD(NSRU),XRATE,XRTE(NSRU),CON(NSRU),DTRU(NSRU),
55      & DTRU(NSRU),SPLC(NSRU),SKD
56      231 FORMAT(1X,I2,2X,A24,A1,P9.0,P12,4,9X,P6,5,4X,2,2,2,
57      & P6,0,2,7,0,8X,P3,0 )
58      XNTD(NSRU)=XNTD(NSRU)/(ENR+SAPP)
59      IF(XNRU(NSRU),LE,0,1)GO TO 232
60      CON(NSRU)=CON(NSRU)*XRTE(NSRU)
61      SPLC(NSRU)=CON(NSRU)*SPL(NSRU)*30,4*PAC/XNRU(NSRU)
62      DTRU(NSRU) = DTRU(NSRU) + ADDRT
63      DTRU(NSRU)=SPL(NSRU)+(XRTE(NSRU)*CON(NSRU))
64      & " DTRU(NSRU)/XRTE(NSRU)
65      GO TO 1

```

```

13 1055 FORMAT(2YHOLBU HAS MORE THAN 65 SEU'S,3X,A77)
14     NXTLBU = 1
15     GO TO 1
16 1060 WRITE(ROUT,1065)CARD
17 1065 FORMAT(4YHOMISSING LBU "21",3X,A77)
18     GO TO 1
19 1070 WRITE(0,1075)NBASES
20 1075 FORMAT(4H1, "NBASES CANNOT EXCEED 90--97 CARD IS IN
21 & ERROR",10X,"NBASES",7(2Y3)
22     SWOP
23 1080 WRITE(0,1085)
24 1085 FORMAT(4M12) " INPUT DATA FILE SMPTZ"
25 1090 SWOP

```

C
C
C
C

EDIT THE DATA

```

2000 CONTINUE
2010 PSGLT = PSUM / XRATE*(100/PYNTS)
     WRITE(ROUT,2010)PSUM
2010 FORMAT(60X,EN-----*2760X,F0.5)
     IF(PSUM,GT,YDATE * (1+(XNRTS)))WRITE(ROUT,2015)
2015 FORMAT(7Y6(4H***)?7 WARNING - SUV BASE REMOVALS MORE THAN",1X,
&     "LBU BASE REMOVALS",4X,"(4H***)X/)
2025 IF(DPTRU,LT,.5)WRITE(ROUT,2030)
     IF(BPTRU,LT,.5)WRITE(ROUT,2030)
     IF(CLAV,LE,100)WRITE(ROUT,2030)
     IF(XMTSD,LT,40)WRITE(ROUT,2030)
     IF(PLT,LE,1)AND(ENST,GT,0)AND(CFAC,GT,0)WRITE(ROUT,2030)
2030 FORMAT(7,1X,5(6H.....), "LBU DATA QUESTIONABLE",5(6H.....)/)
     DO 2040 IRU = 1,NSRUB
     IF(DPTRU(IRU),LT,.5)AND(XNRTS(IRU),GT,90)WRITE(ROUT,2045)IRU
     IF(BPTRU(IRU),LT,.5)AND(XNRTS(IRU),LT,40)WRITE(ROUT,2045)IRU
     IF(CSRU(IRU),LT,25)WRITE(ROUT,2045)IRU
     IF(XMTSD(IRU),LT,20)WRITE(ROUT,2045)IRU
2040 CONTINUE
2045 FORMAT(7,1X,5(6H.....), "SRUB",13Y " DATA QUESTIONABLE",
&     5(6H.....)/)

```

S

```

CSET = CSRU(1)
IF ( NSOUNO ) CSET = CSRU
CALL INITAL
RETURN
END

```

7 MEMORY EXPANDED, USE SLIMITS OR CORRE OPTION FOR NEXT RUN

```

61      232 XNRTS(NSRUS)*CON(NSRUS)
62      SPLT(NSRUS)=60.
63      DNTSRU(NSRUS)=60.
64      GO TO 1
65      32 GO TO 1
66      33 GO TO 1
67      34 GO TO 1

```

C
C
C

35- SRU PHASE PROVISIONING & DELIVERY SCHEDULE

```

71      35 DECODE(CARD,295)ID
72      295 FORMAT(A24)
73      IF(ID,NE,IDSRU(NSRUS))GO TO 1
74      DECODE(CARD,135)(OBS(NSRUS,X),XT(138),(OFF(NSRUS,X),I=1,8)
75      135 FORMAT(25X,4I2,4X,4E22.12,8I4)
76      IF ( BSTOCK ) DECODE(CARD,136) INVRK(NSRUS),BCTPR(NSRUS)
77      134 FORMAT(5X)I2,FS,01
78      GO TO 1

```

C
C
C

36- SRU FEDERAL STOCK NUMBER

```

82      34 DECODE(CARD,136)IDFNS
83      136 FORMAT(A24)
84      IF(IDFNS,NE,IDSRU(NSRUS))GO TO 1
85      DECODE(CARD,236)STFR(NSRUS)
86      236 FORMAT(25X,A16)
87      IF ( SDR(NSRUS),NE,BLANK16 ) WRITE(ROUT,839) SFR(NSRUS)
88      334 FORMAT(13X)A16)
89      GO TO 1
90      37 GO TO 1
91      38 GO TO 1
92      39 GO TO 1

```

C
C
C

40(N1)= MODE OF X SRU(GO TO SRU DECODE)

```

96      40 NMODE = NMODE + 1
97      GO TO 80
98      50 GO TO 1
99      60 GO TO 1
00      70 GO TO 1
01      80 GO TO 1
02      999 IEQJ = 1
03      1000 IF ( I97.LE.0 ) GO TO 1010
04      I97=0
05      GO TO 1000
06      1010 WRITE(8,1020)
07      1020 FORMAT(' MISSING 97 CARD - ABORT JOB')
08      STOP
09      1030 WRITE(8,1030)
10      1040 FORMAT('NONFLYING NOUN CARD INCOMPLETE')
11      STOP
12      1050 WRITE(ROUT,1055)CARD

```

IF 01 10628-79 1R, 159

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
46
47
48
49
50
51
52

```

SUBROUTINE OUTPUT
COMMON /IOBT/ NORT, ZCOMM, INLCOM, XBASEX, ENGINE, DSTOCK
LOGICAL BOUND, INLCOM, XBASEX, ENGINE, DSTOCK
COMMON /XSYS/ PLTMIN, EXTMAX, X1OMIN, X1OMAX, X2ERSLON

C
COMMON /XBASE/ HDG, ZURE, INO, IUR, IDIRU, IBSRU, FSN, SFSN, ALC, SALC
PIDCRB, TCT, TMT
CHARACTER HDG*69, INOR9, IDIRU*24, IBSRU*24(85), FSN*16, SFSN*16(45),
ALC*3, SALC*1(45), ZPERD(9), KEY*6(8), TCT*6(3), ZHW*6(3)

C
COMMON /BASE/ NBASES, NBASES, FN, FLYTOT, OST, BETA, RBETA1, PRD, BPSD,
KBSRP, RBTDC, BSTART, BSTOP, BSWNEW, BOSTOP, BSTOP
DIMENSION BNC(30), CBT(80), IBSRP(30)

C
COMMON /LRUS/ CLRUS, ENTBD, YNBS, BRZLRU, DTRRU, BRT, LNRORK

C
COMMON /SRUB/ CSRUB, ONEN, CSRT, NMODS, XNTRP, NRTS, DTRRU, DTRRU, BPLT,
ENRORK, PCFRET
DIMENSION DSRU(45), XNTRP(45), YNRTS(45), BRZSRU(45), DTRRU(45),
BPLT(45), ENRORK(45), PCFRET(45)

C
COMMON /SAVE/ ADE, ZELE, NZ, PE, TZ, TZ, XPHON, BZSRUB

C
COMMON /STCK/ DSTCK, BSWCK, RENOV, BBACK, BBRK
INTEGER DSTCK(45, 3), BSWCK(45, 30, 3)
DIMENSION BBACK(45, 30, 3), RENOV(30), BBACK(30)

C
DIMENSION RDZ(30), FSLZ(30), NZ(30), NZ(30), BZ(30), TZ(30),
XPHON(30, 10)

C
COMMON /NUMB/ LTOT, BLBU, NLRUS, NLRUD, NRSU, NRSUB, NRSUR, NRSUR, MAXSRK,
NSET, BRUNX, SETMAX, MAXXRU, NTRBD, NLRIN, BPSON, NBASES, LRUS, LRPT
INTEGER NLRUD(30), NRSUB(45, 30), NRSUR(45), NRSUR(45), LRUS(2),
LRPT(3)

C
COMMON /NOUT/ TARGST, NOUT, NRTS, XPRNT, XPRCH, XFILE

C
COMMON /BOUT/ NRTS, INBDS, BUDS, TCOB, BAKC, BATE, BBOZ, BMIN, XMIN,
PBZHEW, BOLD, BMIN, BBACK
DIMENSION BUDC(30), BBOZ(3), BAKC(30), BATE(30), BBACK(45)

C
COMMON /XRUB/ DELTA, NRSUB, NLOW, RD, RD, BNEAN, FDB, RDD, BNEANM
DIMENSION DELTA(1000), RDD(45), RDD(45), BNEAN(10)

C
COMMON /LCOM/ PSGLZ, BBLT
C-----
COMMON /SERIES/ YDRP, YVPR, YVPR(10), ZDRP(45),
KPRZ(15, 10)
COMMON /SERIES/ INMIN, BNEANM
DIMENSION NRSUB(45), YVPR(45, 10), NLRUB(10)

```

TWOIND/OUTPUT

01 10-26-79 12,259

```

53 C-----
54 DIMENSION PSGLT(45)
55 C
56 DIMENSION CTS(45),NCT(19)
57 DIMENSION DELAY(45),CASH(30)
58 C OUTPUT THE SOLUTION FOR THIS BUDGET.
59 IF ( IPRINT,LT,0 ) GOTO INPUT
60 5000 LTOT=NERUD
61 DO 5010 IBS = 1,NBASES
62 5010 LTOT=LTOT+NLNUB(IBS)
63 5020 CT=FLOAT(LTOT)*CLRH
64 CYL=CT
65 LC=(FLOAT(NLRUD)+.25)*PLT/DRTLRU
66 LB=NLRUD-LC
67 NUMB=NSRUS+1
68 DO 5025 IRU = 1,NSRUS
69 NSRUT(IRU)=NSRUB(IRU)
70 DO 5025 IBS = 1,NBASES
71 5024 NSRUT(IRU)=NSRUB(IRU)+NSRUB(IRU,IBS)
72 CTS(IRU)=CSRU(IRU)*FLOAT(NSRUT(IRU))
73 5025 CT=CT+CTS(IRU)
74 IF ( CT,LE,BUDG(NPTS) ) GO TO 5000
75 BSOZ=(XMIN-BAK0(NPTS))/(CT-BUDG(NPTS))
76 IF ( NPTS,LE,1 ) GO TO 5028
77 BSOZ1 = (XMIN-BAK0(NPTS-1))/(CT-BUDG(NPTS-1))
78 IF ( BSOZ1,LT,BSOZ ) GO TO 5028
79 WRITE(ROUT,5026)
80 5026 FORMAT(/7X,100('H')/72X,'ANOVE SOLUTION NONSCAPE -',
81 & 7 POINT NOT SAVED FOR GRAPH/CONVERT/'/7X,100('H'))
82 NPTS = NPTS + 1
83 IF ( IFILE,LE,0 ) GO TO 5028
84 BACKSPACE IFILE
85 DO 5027 IF = 1,NSRUS
86 5027 BACKSPACE IFILE
87 BACKSPACE IFILE
88 C
89 5028 IF ( IFILE,GT,0 )
90 & WRITE(IFILE,5030)BSOZ,IDLRU,NUMB,XMIN,CT
91 5030 FORMAT(1X,BHS ,212,22AS,23,F12,82F12,0)
92 C-----
93 NLRUD=NERUD
94 DO 20 IRU=1,NSRUS
95 NSRUD(IRU)=NSRUB(IRU)
96 DO 15 IBS=1,NBASES
97 NSRUB(IRU,IBS)=NSRUB(IRU,IBS)
98 NLRUB(IBS)=NLRUD(IRU)
99 15 CONTINUE
100 20 CONTINUE
101 C-----
102 C-----
103 IF(IIMINSE,LE,0) GO TO 90
104 IDPIPE=ZDPIPE + 0.5

```

07 01 10628-79 1A,259

```

105 IF (NLRUB;RE,EDPIFE) GO TO 30
106 NLRUB=EDPIFE
107 KC=(NLRUB(NLRUB)+.99)*RLT/DRTLRU
108 ED=NLRUB*LC
109 IF (LD;LE;0)LD=0
110 30 CONTINUE
111 DO 40 IBS=1,NBASES
112 EDPIFE=EDPIFE(IBS) + 0.5
113 IF (NLRUB(IBS).LT.EDPIFE)NLRUB(IBS)=EDPIFE
114 40 CONTINUE
115 C---*****CHECK SUB BOUNDS---
116 DO 50 IBS=1,NBASES
117 EDPIFE=EDPIFE(IBS) + 0.5
118 IF (NLRUB(IBS).LT.EDPIFE)NLRUB(IBS)=EDPIFE
119 DO 70 IBS=1,NBASES
120 EDPIFE=EDPIFE(IBS) + 0.5
121 IF (NLRUB(IBS).LT.EDPIFE) NLRUB(IBS)=EDPIFE
122 70 CONTINUE
123 50 CONTINUE
124 90 CONTINUE
125 C---*****END OF LEVELS CHECK---
126 C
127 WRITE(ROUT,5040)IDBS,INO,IYR
128 5040 FORMAT(XX/1X,12(7H****) /10X.
129 & 8H SUMMARY OF REQUIREMENTS, DISTRIBUTIONS AND COSTS //
130 & 8X,12H WORK PART:14X,3HALC,13X,32H REQUIREMENTS
131 & 8X,5H RATE :12,15,2218,22/1X,18H UNIT NUMBER,9X.
132 & 40H CORE COST,6X,20H TOTAL COST COND BASES )
133 C
134 WRITE(ROUT,5050) IRLNR,RLC,CRL,LTOP,LD,LC
135 & (NLRUB(IBS);IBS=2,NBASES)
136 5050 FORMAT(5HOLRU ,A24,A10R17.6,17,2I6,2X,23I8/62X,23I3)
137 C
138 IF (IYRLE.GT.0)WRITE(IFILE,5060)IDBRU,FSN,ALC,CRL,LTOP,
139 & LD,LC,(NLRUB(IBS);IBS=1,NBASES)
140 5060 FORMAT(1X)2H21 ,A24,A10.6,1.79.0,18,2I3,30E3)
141 WRITE(ROUT,5070)
142 5070 FORMAT(4H SUB)
143 DO 5100 IBS=1,NBASES
144 KC=(ED*EDPIFE(NLRUB(IBS))+.99)*RLT(IBS)/DRTLRU(IBS)
145 KC=NLRUB(IBS)*KC
146 C
147 IF (IYRLE.GT.0)
148 & WRITE(IFILE,5080)IDBRU(IBS),FSN(IBS),ALC(IBS),CRL(IBS);
149 & NLRUB(IBS),KC,KC,(NLRUB(IBS);IBS=1,NBASES)
150 5080 FORMAT(1X)2H21 ,A24,A10.6,1.79.0,18,2I3,30E3)
151 C
152 IF (IDBRU.GT.9) WRITE(ROUT,5105) XBACK(IBS)
153 5105 FORMAT(1X,10X,2HNO,10.6)
154 5100 WRITE(ROUT,5110) IBS,IDBRU(IBS),ALC(IBS)CRL(IBS),NLRUB(IBS),
155 & KC,KC,(NLRUB(IBS);IBS=1,NBASES)
156 5110 FORMAT(1X)2.2X,A24,A10.6,1.79.0,17,2I6,2X,23E3/62X,23I3)

```


: 01 10-26-79 10:259

```

57      C
58      JCT = OCT
59      DO 5120 I = 1, N
60      KP ( JCT ) 5120, 5120, 9917
61      5117 ICT = OCT * ( JCT/1000 ) * 1000
62      DO 5118 J = 1, 3
63      KCT(4+J) = ICT * (JCT/10) * 10
64      5118 ICT = ICT / 10
65      JCT = OCT / 1000
66      IP ( I64 ) 5120, 5120, 9919
67      5119 FMT(2) = ICT(I=4)
68      GO TO 5122
69      5120 WRITE(ROUT, 5121) KCT
70      5121 FORMAT(//40X, '10H*** TOTAL COST $,11X,2X1)
71      GO TO 5128
72      5122 WRITE(ROUT, FMT) KCT
73      5128 CONTINUE
74      5129 WRITE(ROUT, 5130) XNMX
75      5130 FORMAT(40X, '10H*** RECORDERS #,214:6)
76      WZEROB = 0.200
77      FFLT=0.0
78      DO 5132 IBS = 1, NBSBS
79      FFLT=FFLT+FILE(IBS)*RHH(2BS)
80      WZEROB = WZEROB + XZEROB(IBS,1)*RH(2BS)
81      5132 CONTINUE
82      WZEROB = 100.0 * WZEROB / FFLTOT
83      FFLT=FFLT/FLYTOT
84      WRITE(ROUT, 5133) 100.0*FFLT
85      5133 FORMAT(40X, '10H*** BILL RATE X ,214:2)
86      WRITE(ROUT, 5134) WZEROB
87      5134 FORMAT(40X, '10H*** HEADY RATE X ,214:2)
88      WRITE(ROUT, 5140) BDBE
89      5140 FORMAT(40X, '10H*** BUY SUPP ORG ,214:5)
90      IF(IPRNT,LE,0) GO TO 5205
91      C
92      C      GENERATE CONDITIONAL SRU WAIT TIMES FOR BCGH INPUT
93      C
94      IF ( .NOT.INLCOM ) GO TO 5169
95      C
96      DO 5142 IRU = 1, NRSUB
97      CALL ABSENZ( PDD(IRU),GRD(IRU),DNBKNR(IRU) )
98      5141 DELAY(IRU) = DELTA(INSRUB(IRU)) * DRTSBD(IRU)
99      C
100     DO 5148 IBS = 1, NBSBS
101     CADZ( IBS ) = 0.0
102     DO 5146 IRU = 1, NRSUB
103     TEJ = (17-INPTS(IRU)) * RTSRU(IRU)
104     + XNRTS(IRU) * (DST( IBS ) * DBDLAY(IRU) )
105     PX = WBD(IRU) * PDD(IRU)
106     P4 = 1. - PDD(IRU)
107     R4 = RBD(IRU) - 1.
108     YCVH = 0.0

```

18 02 10028-79 1R1359

```

109      STK = 0:0
110
111      C
112      5142 RCUN = RCUN + PX
113      IF ( SURTOT, NSRUB(ZRUZTSS) ) GO TO 5148
114      STK + STK + 1:0
115      PX = PX + 71 * (ZRUZTSS) / STK
116      GO TO 5142
117
118      C
119      5148 CONTINUE
120      5143 CADZ(ZSS) = CADZ(ZSS) + RSOLZ(ZSU) * (1.07CUN) * T13
121
122      C
123      CADZ(ZSS) = CADZ(ZSS) / RSOLZ
124
125      5148 CONTINUE
126
127      C
128      5149 CONTINUE
129
130      C
131      DO 5160 NRE = 2,10
132      DO 5160 ZSS = 1, NBASES
133      K=NLZUB(ZSS)+1
134      NZ(ZSS)=(X+RE(ZSS)+1)*(1-PE(ZSS))*NZ(ZSS)/X
135      KPROB(ZSS)=NZ(ZSS)
136      NSRUB(ZSS)=NLZUB(ZSS)+1
137
138      5150 CONTINUE
139
140      5160 CONTINUE
141      WRITE(ROUT,5170)
142
143      5170 FORMAT('XZX,1816H*V****)X26H ***X** PROBABILITY OF X,
144      & 80H BACKORDERS AT THE POINT IN THE ***V**V** FALL,
145      & 84H 12H WAIT TIME EXPECTED TOTAL )
146      WRITE(ROUT,5180)(ZSS,1:9)
147
148      5180 FORMAT('12H BASE 100.948H X0.11)X4Z0
149      & 44H RATE FOR NEEDED SRU 2RU 220UPPLY TMS,
150      & /9H ***-42.10(6H 0000),247H 0000)1H(22H+000-4) )
151
152      IF ( INLCOM ) WRITE(ROUT,5185)
153
154      5185 FORMAT('99X14H* (GIVEN WAIT )
155      DO 5200 ZSS = 1, NBASES
156      WRITE(ROUT,5190) ZSS, XPROP(ZSS, ZSS), ZSS=1,10), REZ(ZSS),
157      & ADZ(ZSS) * T2(ZSS)
158
159      IF ( INLCOM ) WRITE(ROUT,5195) CADZ(ZSS)
160
161      5195 FORMAT('1H+19QX, 1H(22H(1.24R) )
162
163      5190 FORMAT('1XZ5.3X, 1075.212775.2130219X.W902)
164
165      5200 CONTINUE
166      CALL PTIME(TTIME)
167      WRITE(ROUT,5202) TTIME
168
169      5202 FORMAT('0ELARSED CONPUTE TIME ='.Z9,' N121SECS:')
170
171      5205 NRTS=NRTS+1
172      BAKO(NRTS)=KRNIN
173      BUDG(NRTS)=CF
174      BATE(NRTS)=PILT
175
176      C-----RESET LEVELS TO THE INPUT VALUES
177
178      5800 CONTINUE
179      NSRUB=NLZUB
180      DO 5820 ZSU=1, NSRUB

```

01 10626-79 1R,259

```
01      NSRUD(IRU)=NSRUD(IRU)
02      DO 5025 IBS=1,NBASIS
03          NSRUB(IRU,IBS)=NSRUB(IRU,IBS)
04          NLRUB(IBS)=NLRUB(IBS)
05      5015 CONTINUE
06      5020 CONTINUE
07      -----
08      RETURN
09      END
```

01 10-20-79 10.809 13

*BUDGET=LINE/ORD/COMBIN.O(BCD,NOBY
 *COMBIN.S

C COMBENS
 C COMPILED FILE IS MODMETRIC/ORDORD

C DIMENSION BUD(200,20),BACK(200,20),NSV(400,43),NRP(200)X
 4 NL(200),TD(200,6),SL(200)
 CHARACTER STAR*3/3H**/,BLANK3/3H /,LNU*6,LINS*3(200)
 CHARACTER JBLANK*1/1H /,JPLNGT1
 CHARACTER NAME*6,JLRU*6
 DIMENSION TDORD(9)

C
 CALL ISNT(XNUMB,ICND)
 CALL DATE(IGATE)
 10 READ(4,20,ERR=9,END=9)NAME
 IF(NAME.EQ."LAST ")GO TO 16
 BACKSPACE 4
 WRITE(6,80)IGATE,ISNUMB,ICND
 20 FORMAT(12,A6)
 30 FORMAT(1X,COMBIN?,1X,I6,1X,X5(1X,9A6)
 40 FORMAT(1X,"LAST ")
 WRITE(6,80)
 9 CALL DETECH(6,ISTAT, J

C
 MAXINT=200
 NOUT=6
 NSW=1

C-----LINE CHANGES

BMIN=0.
 BMAX=1.810

C-----END CHANGES

50 FORMAT(2F10,0)
 IF(BMAX.EQ.0.)BMAX=1.810
 WRITE(NOUT,60)BMIN,BMAX
 60 FORMAT(1XMIN BUDGET="F11,0,58,"MAX BUDGET="F11,0)
 WRITE(NOUT,70)
 70 FORMAT(1XNO,24X,4NUNIT,4X,6NBUDGET,6X,10NBACKORDERS/
 8 22NODATA \$GINT SBT NO. 4)

ISBT=1
 IPT=0
 DO 80 I=1,80
 80 LINE(I)=BLANK
 90 READ(5,100,END=200)JPLAG,JLNUFBUDGET,BACKOR
 100 FORMAT(A1,A6,F11,0,F14,4)
 IF(JPLAG.EQ.1)GO TO 160
 LRU=0LRU
 IF(IPT.EQ.0)GO TO 110
 IF(BDDSET.LE.BUD(ISBT,IPT).OR.BACKOR.GE.BACK(ISBT,IPT))GO TO 140
 108 IF(IPT.LE.2)GO TO 110
 S1=(BACK(ISBT,IPT)-BACK(IANB,IPT)) /
 (BUD(ISBT,IPT)-BUD(IANB,IPT))
 8 S2=(BACK(ISBT,IPT)-BACKOR)/(BUDGET-BUD(ISBT,IPT))

COMBIN

01 10-20-79 1.505 28

```

      IF(SZ.LT.51)GOTO 110
      WRITE(NOUT,155)LRU,BUD(ISET,IPT),BACK(ISET,IPT)
155  FORMAT(60X,"NON-CONVEX PT SKIPPED ",A6,F10.0,F14.4)
      IPT=IPT-1
      GOTO 108
110  IPT=IPT+1
      BUD(ISET,IPT)=NBUDGET
      BACK(ISET,IPT)=NBCKOR
      DECODE(LRU,150)(ID(ISET,J),J=0,6)
130  FORMAT(60X)
      GOTO 90
140  WRITE(NOUT,150)LRU,BUDGET,BKCKOR
150  FORMAT(60X,"ERROR ",A6,F10.0,F14.4)
      GOTO 90
160  NPPLRU(ISET)*IPT
      WRITE(NOUT,160)(X,LRU,BUD(ISET,X),BACK(ISET,X),I=1,IPT)
120  FORMAT(24X,I5,1XIA6,F10.0,F14.4)
      IF(IPT.GT.1)GO TO 180
      KSW=0
      WRITE(NOUT,170)
170  FORMAT(160X," ERROR ,MORE THAN 1 POINT REQUIRED ")
180  ISET+ISET+1
      IF(ISET.GT.MAXSET)GOTO 370
      WRITE(NOUT,180)ISET
190  FORMAT(190X"DATA POINT SET NO.",I3)
      IPT=0
      GOTO 90
200  WRITE(NOUT,200)
210  FORMAT(120X"NO OF DATA)
      IF(KSW.EQ.1)GO TO 230
      WRITE(NOUT,220)
220  FORMAT(24X"NO ERRORS IN INPUT PINE)
      WRITE(NOUT,220)
      STOP
230  NLRU+ISET-1
      NA=1
      NST=32
240  NST=MIN0(NST,NLRU)
      TCOST=0.
      TBACK=0.
      WRITE(NOUT,250)
250  FORMAT(181,30X,15X"WORK UNIT CODES)
      DO 26 J=1,6
260  WRITE(NOUT,270)(ID(I,J),I*NX,NST)
270  FORMAT(38X,38(2X,A1))
      DO 28 I=1,NLRU
      NL(I)=1
      SL(I)=(BACK(I,1)-BACK(I,2)) / (BUD(I,2)-BUD(I,1))
      TBACK=TBACK+BACK(I,1)
280  TCOST=TCOST+BUD(I,1)
      WRITE(NOUT,280)TCOST,TBACK
290  FORMAT(28X BUDGET      BACKORERR      B50/

```

4 INO, P10.0, P10.4, P20X, 10XIN, 10XINL POINTS)

300 ISW=0

IF(TCONT,GT,BNAX)GOTO 360

BBO=-1.

DO 310 N=1,NLRN

IF(NZ(N)GE,NPPLBU(I))GOTO 310

IF(SL(N)LE,BBO)GOTO 310

ISW=1

BBO=SL(N)

N=N

310 CONTINUE

IF(ISW,EQ,0)GOTO 360

KOLD=N

NL(N)=SL(N)+1

KNEW=N

IF(KNEW,GT,NPPLBU(N))GOTO 320

NL(N)=09

GOTO 380

320 SL(N)=BACK(N,KNEW)-BACK(N,KNEW+1)/(BUD(N,KNEW+1)-BUD(N,KNEW))

330 TCOST=TCOST-BUD(N,KOLD)+BUD(N,KNEW)

TBACK=TBACK-BACK(N,KOLD)+BACK(N,KNEW)

IF(TCONT,LT,BNIN)GOTO 300

LINE(N)=NST

WRITE(ROUT,360)(LINE(I),I=NX,NST)

340 FORMAT(3X,3A3)

LINE(N)=BLANK

WRITE(ROUT,380) TCOST,TBACK,BBO,(NL(I),I=NA,NST)

350 FORMAT(1E,P10.0,P10.4,B12I5,3I3)

GOTO 300

360 IF(NST,GE,NLRN)STOP

NA=NST+1

NST=NA+31

GOTO 280

370 WRITE(ROUT,380)

380 FORMAT(20HODIMENSION TOO SMALL)

STOP

END

1470 EQUALITY OR NON-EQUALITY COMPARISON MAY NOT BE MEANINGFUL IN LOGICAL IF EXP

REFERENCES

1. Demmy, W. Steven, RIME: The Recoverable Item Management Evaluator, Volume I: Model Description, TR-80-01, Decision Systems, 3575 Charlene Drive, Dayton, Ohio 45432, May 1980, 153 pp.
2. Demmy, W. Steven, RIME: The Recoverable Item Management Evaluation Volume II, Program Listings and Narratives, TR-80-02, Decision Systems, 3575 Charlene Drive, Dayton, Ohio 45432, May 1980, 298 pp.
3. Demmy, W. Steven, An Empirical Evaluation of Proposed Stockage Policies for Recoverable Item Management, TR-80-03, Decision Systems, 3575 Charlene Drive, Dayton, Ohio 45432, May 1980, 173 pp.
4. Demmy, W. Steven and Victor J. Presutti, Jr., Multi-Echelon Inventory Theory in the Air Force Logistics Command, Working Paper 76-3011-27, Department of Management, Wright State University, Dayton, Ohio 45435.
5. Muckstadt, John A., "A Model for a Multi-Item, Multi-Echelon, Multi-Indenture Inventory System", Management Science, v20, n4, December, 1973, pp. 472-481.