

W. H. Hight

NATIONAL BUREAU OF STANDARDS REPORT

10 436

A SEARCH AND RESCUE SIMULATION MODEL FOR THE UNITED STATES COAST GUARD

Appendix B

PROGRAM LISTINGS FOR PROGRAMMER LEVEL DOCUMENTATION



U.S. DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS

NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards¹ was established by an act of Congress March 3, 1901. Today, in addition to serving as the Nation's central measurement laboratory, the Bureau is a principal focal point in the Federal Government for assuring maximum application of the physical and engineering sciences to the advancement of technology in industry and commerce. To this end the Bureau conducts research and provides central national services in four broad program areas. These are: (1) basic measurements and standards, (2) materials measurements and standards, (3) technological measurements and standards, and (4) transfer of technology.

The Bureau comprises the Institute for Basic Standards, the Institute for Materials Research, the Institute for Applied Technology, the Center for Radiation Research, the Center for Computer Sciences and Technology, and the Office for Information Programs.

THE INSTITUTE FOR BASIC STANDARDS provides the central basis within the United States of a complete and consistent system of physical measurement; coordinates that system with measurement systems of other nations; and furnishes essential services leading to accurate and uniform physical measurements throughout the Nation's scientific community, industry, and commerce. The Institute consists of an Office of Measurement Services and the following technical divisions:

Applied Mathematics—Electricity—Metrology—Mechanics—Heat—Atomic and Molecular Physics—Radio Physics²—Radio Engineering²—Time and Frequency²—Astrophysics²—Cryogenics.²

THE INSTITUTE FOR MATERIALS RESEARCH conducts materials research leading to improved methods of measurement standards, and data on the properties of well-characterized materials needed by industry, commerce, educational institutions, and Government; develops, produces, and distributes standard reference materials; relates the physical and chemical properties of materials to their behavior and their interaction with their environments; and provides advisory and research services to other Government agencies. The Institute consists of an Office of Standard Reference Materials and the following divisions:

Analytical Chemistry—Polymers—Metallurgy—Inorganic Materials—Physical Chemistry.

THE INSTITUTE FOR APPLIED TECHNOLOGY provides technical services to promote the use of available technology and to facilitate technological innovation in industry and Government; cooperates with public and private organizations in the development of technological standards, and test methodologies; and provides advisory and research services for Federal, state, and local government agencies. The Institute consists of the following technical divisions and offices:

Engineering Standards—Weights and Measures—Invention and Innovation—Vehicle Systems Research—Product Evaluation—Building Research—Instrument Shops—Measurement Engineering—Electronic Technology—Technical Analysis.

THE CENTER FOR RADIATION RESEARCH engages in research, measurement, and application of radiation to the solution of Bureau mission problems and the problems of other agencies and institutions. The Center consists of the following divisions:

Reactor Radiation—Linac Radiation—Nuclear Radiation—Applied Radiation.

THE CENTER FOR COMPUTER SCIENCES AND TECHNOLOGY conducts research and provides technical services designed to aid Government agencies in the selection, acquisition, and effective use of automatic data processing equipment; and serves as the principal focus for the development of Federal standards for automatic data processing equipment, techniques, and computer languages. The Center consists of the following offices and divisions:

Information Processing Standards—Computer Information—Computer Services—Systems Development—Information Processing Technology.

THE OFFICE FOR INFORMATION PROGRAMS promotes optimum dissemination and accessibility of scientific information generated within NBS and other agencies of the Federal government; promotes the development of the National Standard Reference Data System and a system of information analysis centers dealing with the broader aspects of the National Measurement System, and provides appropriate services to ensure that the NBS staff has optimum accessibility to the scientific information of the world. The Office consists of the following organizational units:

Office of Standard Reference Data—Clearinghouse for Federal Scientific and Technical Information³—Office of Technical Information and Publications—Library—Office of Public Information—Office of International Relations.

¹ Headquarters and Laboratories at Gaithersburg, Maryland, unless otherwise noted; mailing address Washington, D.C. 20234.

² Located at Boulder, Colorado 80302.

³ Located at 5285 Port Royal Road, Springfield, Virginia 22151.

NATIONAL BUREAU OF STANDARDS REPORT

NBS PROJECT

4314561

NBS REPORT

10 436

A SEARCH AND RESCUE SIMULATION MODEL FOR THE UNITED STATES COAST GUARD

Appendix B

PROGRAM LISTINGS FOR PROGRAMMER LEVEL DOCUMENTATION

IMPORTANT NOTICE

NATIONAL BUREAU OF STANDARDS
for use within the Government.
and review. For this reason, the
whole or in part, is not authorized
Bureau of Standards, Washington
the Report has been specifically

Approved for public release by the
Director of the National Institute of
Standards and Technology (NIST)
on October 9, 2015.

These accounting documents intended
subjected to additional evaluation
listing of this Report, either in
the Office of the Director, National
by the Government agency for which
copies for its own use.



U.S. DEPARTMENT OF COMMERCE

NATIONAL BUREAU OF STANDARDS

PREFACE

This volume is one of a series which documents a Search and Rescue Simulation Model for the United States Coast Guard. The material reported in this documentation was developed by an interdisciplinary team at the National Bureau of Standards with representation from the U.S. Coast Guard under MIPR Z-70099-0-01935.

The complete documentation is comprised of the following:

Volume I Executive Level Documentation

Volume II Analyst Level Documentation

Volume III Programmer Level Documentation for "PREPROCESSOR"

Volume IV Programmer Level Documentation for "OPSIM"

Volume V Programmer Level Documentation for "POSTPROCESSOR"

Appendix A Flow Charts for Programmer Level Documentation

Appendix B Program Listings for Programmer Level Documentation

The study was initially conducted under the supervision of Martin J. Aronoff; subsequently efforts were supervised by Richard T. Penn, Jr. Technical Project Leadership was supplied throughout the project by Stephen S. Karp. Other participants from the National Bureau of Standards Technical Analysis Division included the following:

Susan S. Chamberlin

Elizabeth E. Leyendecker

Linda K. Cummings

Marcia D. Maltese

Mary Jane Duberg

Patsy L.B. Saunders*

William Elliott, III

Wayne A. Steele

Walter G. Leight

Michael R. Vogt

Joel Levy

Arnold L. Weber

Valuable advice was received from Alan J. Goldman* and Prof. Gustave J. Rath of Northwestern University.

U.S. Coast Guard participants included:

Paul D'Zmura

Gerald L. Underwood

Thomas T. Matteson

Robert R. Wells

Support services were furnished by the following members of the NBS Technical Analysis Division:

Mary M. Abbott

Frances E. Jones

Theresa I. Conrad

Lucinda I. Farrell

* Staff members of the NBS Applied Mathematics Division

Appendix B

Table of Contents

	page
PREPRO	
MUTAPE	1
MAIN PROGRAM	1A
READ	7
WRITE	9
MUC130	11
MAIN PROGRAM	12
READ	16
WRITE	18
PCP	20
MAIN PROGRAM	20
READ	30
NEEDS	32
NUCASE	34
FIELD	37
RAND	41
DEMGEN	42
IO TABLE	42
MAIN PROGRAM	44
BOX	55
ADD	57
ADTIME	59
RAND	61
SELECT	62
HIST	64
MAIN PROGRAM	65
CONVER	69
OPSIM	71
DEFINITIONS	71
EVENTS	75
START	76
NVCRU	77
OPSIM	80
SRAS	84
CRES	85
VEC	89
OSET	90
RESAP	91
ROCA	94
MRAS	95
NOTIF	96

Table of Contents (con't.)

	page
TOW	97
DTD	98
SRCH	99
NOTE	101
SASS	102
SSS	104
READY	108
ARSCH	109
COMPL	111
SSET	112
FUEL	113
HOMEF	114
SNDBK	115
SRI SE	117
XSET	118
SERVE	119
DELAY	121
ARVSN	122
STATS	125
SRCHF	127
ONSCN	129
COVER	132
CHEKN	133
RETN	134
TERM	135
HOME	137
SAQ	138
QUEUE	140
EXQ	142
STNBY	147
SVQUE	151
WRECK	152
ENDSIM	154
DRIVE	157
REPORT SARSIM	159
REPORT GRPRES	162
REPORT HEADER	163
REPORT TITLE	164
REPORT RESULT	165
REPORT HEAD	166
EXCASE	167
REPORT DISTRIB	168
JUMPER	170

MUTAPE

This program is not completely parameterized. For each district run, one source card must be changed. In the main program, card number 8 includes the number for the district being exercised in columns 38 and 39. In the listing given the district is 3. For each run made this number must be set to equal the number of the district for which data are being processed.

MAIN PROGRAM

STORAGE USED (BLOCK, NAME, LENGTH)

0001 *CODE 001764
0000 *DATA 043317
0002 *BLANK 003427
0003 *READ 000011
0004 *WRITE 000011

EXTERNAL REFERENCES (BLOCK, NAME)

0005 CLOCKS
0006 NTRAN
0007 READ
0010 WRITE
0011 SOPEN2
0012 SRREL
0013 SSORT
0014 SSORT
0015 SOPEN3
0016 NWDUS
0017 NIOIS
0020 NIO2S
0021 NSTOPS
0022 NREXS
0023 NRBUS
0024 NABUS

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001	000063	10L	0000	043226	100F	0000	043231	101F	0000	043173	102F	0001	000151	11L
0001	000473	12L	0001	001433	13L	0001	001470	14L	0001	000055	142G	0001	001506	16L
0001	000126	165G	0001	001650	17L	0000	043235	171F	0001	000055	172L	0001	000143	174G
0001	001721	18L	0001	001756	19L	0001	000200	20L	0001	000213	221G	0001	000253	234G
0001	000421	24L	0001	000267	241G	0001	000316	254G	0001	000463	26L	0001	000362	266G
0001	000470	27L	0001	000477	30L	0001	000410	300G	0001	000501	301L	0001	000540	31L
0001	000442	313G	0001	000453	321G	0001	000603	33L	0001	000622	34L	0000	043202	341F
0001	000660	342L	0000	043214	343F	0001	000556	344G	0001	000701	344L	0001	000720	35L
0001	000567	351G	0001	000742	37L	0001	001005	40L	0001	001126	411L	0001	000714	416G
0001	001157	42L	0001	000725	425G	0001	001225	44L	0001	000766	441G	0001	001235	441L
0001	000775	447G	0001	001032	463G	0001	001346	47L	0001	001050	473G	0001	001355	48L
0001	001421	50L	0001	001105	506G	0001	001200	524G	0001	001211	531G	0001	001262	551G
0001	001311	563G	0001	001335	575G	0001	001402	615G	0001	001413	623G	0001	001477	647G
0001	001643	700G	0001	001710	725G	0000	043156	A3A	0000	043171	A4A	0000	043172	A4B
0000	043166	B	0000	001762	HUFF	0000	043165	8X	0000	043170	8Y	0000	043155	C
0000	000012	CASE	0004	000004	CF	0000	043140	CODE	0000	043152	CX	0000	043161	D
0000	000776	DATA	0004	000006	DP	0004	000005	DPN	0002	001000	D0MP	0000	043157	DX
0002	R 002000	E	0003	043145	I	0000	043154	IEND	0003	000001	IF	0000	043144	IFLAG

[illegible]

```

1* IMPLICIT INTEGER (A,B,C,D)
2* COMMON LOAD(512),DUMP(512),E,X0,X1,IQT,IONIT,IHDR(14),JHDR(768)
3* IPREC,JPREC,NPREC,MPREC
4* COMMON /HREAD/ IS,IF,LMT,INREX,KF,LPN,LP,ISTAT(2)
5* COMMON /BWRITE/ NS,NF,LIM,IOREX,CF,DPN,DP,NSTAT(2)
6* DIMENSION MJHDR(10),CASE(500),DATA(500),BUFF(1000),LONG(4000),
7* NX(3),NLGF(3),NUMB(3,4000)
8* DATA CODE /%000000%, MUHDR /% 03
9* ,
10* E=600.0
11* CALL CLOCKS(X0)
12* XO=X0+E
13* CALL NTRAN (7,2,256,LOAD(1),ISTAT(1))
14* IS=2
15* IF=257
16* LMT=482
17* INREX=0
18* NPREC=0
19* MPREC=0
20* CALL READ (<50)
21* LP=IF+32
22* NS=1
23* NF=1
24* LIM=226
25* IOREX=0
26* DP=NF
27* NX(1)=0
28* NX(2)=0
29* NX(3)=0
30* LX=0
31* LGFS=0
32* IFLAG=0
33* DO 9 I=1,32
34* 9 BUFF(I)=0
35* K00PFC=IHDR(2)
36* KU01ST=JHDR(2)
37* 10 IF (LOAD(LP)+EQ,0) GO TO 18
38* ISKIP=0
39* LGF=FLD(2,4,LOAD(LP))
40* CX=0
41* LPN=LP+LGF*32
42* IF (LGF,EQ,0) LPN=LPN+32
43* LAST=LPN-1
44* IEND=IF+255
45* IF (LAST,GT,IEND) LAST=IEND

```



```

00340 104* 31 IUNIT=1+R
00341 105* REWIND IUNIT
00342 106* 1A=NX(1)
00343 107* DO 32 J=1,IY
00346 108* CA=NUMB(1,J)
00347 109* READ (IUNIT) (CASE(C),C=1,CX)
00355 110* 32 CALL SRREL (CASE,CX)
00357 111* CALL SSORT
00360 112* 33 NX(1)=0
00361 113* NLGF(1)=0
00362 114* CALL SRRET (CASE,CX,S40)
00363 115* NUCASE=FLO(n,30,CASE(3))
00364 116* REWIND IUNIT
00365 117* GO TO 35
00366 118* 34 CALL SRRET (CASE,CX,S40)
00367 119* NUCASE=FLO(n,30,CASE(3))
00370 120* IF (NUCASE.FQ.KUCASE) GO TO 37
00372 121* NX(1)=NX(1)+1
00373 122* J=NX(1)
00374 123* IF (J.LE.4000) GO TO 342
00376 124* WRITE (6,341) I
00401 125* 341 FORMAT (1H,'MULTI-UNIT CASE DIMENSION EXCEEDED FOR YEAR ',11)
00402 126* STOP
00403 127* 342 NUMB(1,J)=BX
00404 128* IF (BX.LE.1000) GO TO 344
00406 129* WRITE (6,343) RUFF(3)
00411 130* 343 FORMAT (1H,'MULTI-UNIT CASE DIMENSION EXCEEDED FOR CASE ',A5)
00412 131* STOP
00413 132* 344 NLGF(1)=NLGF(1)+RUFF(1)
00414 133* WRITE (IUNIT) (BUFF(B),B=1,RX)
00422 134* 35 BUFF(1)=1
00423 135* BUFF(2)=0
00424 136* DO 36 J=3,13
00427 137* 36 BUFF(J)=CASE(J)
00431 138* FLD(0,12,BUFF(14))=FLO(n,12,CASE(14))
00432 139* KUCASE=NUCASE
00433 140* BX=32
00434 141* 37 BUFF(1)=BUFF(1)+FLD(2,4,CASE(1))
00435 142* IF (FLD(2,4,CASE(1)).EQ.0) BUFF(1)=BUFF(1)+1
00437 143* BUFF(2)=BUFF(2)+1
00440 144* DO 38 J=3,13
00443 145* 38 CASE(J)=0
00445 146* FLD(0,12,CASE(14))=0
00446 147* DO 39 J=1,CX
00451 148* BX=BX+1
00452 149* 39 BUFF(BX)=CASE(J)
00454 150* GO TO 34
00455 151* 40 NX(1)=NX(1)+1
00456 152* J=NX(1)
00457 153* NUMB(1,J)=BX
00460 154* NLGF(1)=NLGF(1)+BUFF(1)
00461 155* WRITE (IUNIT) (BUFF(B),B=1,RX)
00467 156* I=I+1
00470 157* IF (I.LE.3) GO TO 301
00472 158* DO 41 I=1,9
00475 159* DX=DX+1-I
00476 160* 41 DUMP(DX)=MUMDR(1)
00500 161* DUMP(DP+9)=(NLGF(1)+NLGF(2)+NLGF(3)+8)/8

```

```

00501 162* DUMP(OP+10)=BX(1)+BX(2)+BX(3)
00502 163* DUMP(OP+11)=BX(1)
00503 164* DUMP(OP+12)=BX(2)
00504 165* DUMP(OP+13)=BX(3)
00505 166* DO 410 I=15,32
00510 167* BX=OP+I-1
00511 168* 410 DUMP(DX)=0
00513 169* DP=OP+32
00514 170* I=1
00515 171* REWIND 39
00516 172* 411 CALL SDPEN3 ($42,$44,1000,47,47,6,0,0,1,3,31,12,0,0,2,46,13,36,0.
00516 173* 1 0.3,99999)
00517 174* 42 IUNIT=1+8
00520 175* REWIND IUNIT
00521 176* IQUIT=IDREX+20
00522 177* IX=NX(1)
00523 178* DO 43 J=1,IX
00526 179* BX=NUMB(I,J)
00527 180* READ (IUNIT) (BUFF(R),R=1,BX)
00535 181* 43 CALL SRREL (BUFF,BX)
00537 182* CALL SSDRT
00540 183* 44 CALL SRRET ,BUFF,BX,548)
00541 184* BY=BX
00542 185* BX=0
00543 186* 441 DPN=DP+32
00544 187* LAST=DPN-1
00545 188* IEND=NF+255
00546 189* IF (LAST.GT.IEND) LAST=IEND
00550 190* DO 45 J=DP, LAST
00553 191* BX=BX+1
00554 192* 45 DUMP(J)=BUFF(BX)
00556 193* IF (DPN.LT.LIM) GO TO 47
00560 194* IF (IOREX.LE.IQUIT) WRITE (6,100) (DUMP(D),D=NF,IEND)
00567 195* CALL WRITE
00570 196* DPN=DPN+CF
00571 197* IF (DPN.EQ.NF) GO TO 47
00573 198* LAST=DPN-1
00574 199* DO 46 J=NF, LAST
00577 200* BX=BX+1
00600 201* 46 DUMP(I)=BUFF(BX)
00602 202* 47 DP=DPN
00603 203* IF (RX.NE.BY) GO TO 441
00605 204* GO TO 44
00606 205* 48 I=I+1
00607 206* IF (I.LE.3) GO TO 411
00611 207* IF (OP.EQ.NF) GO TO 50
00613 208* IEND=NF+255
00614 209* DO 49 J=DP,IEND
00617 210* 49 DUMP(J)=0
00621 211* WRITE (6,100) (DUMP(D),D=NF,IEND)
00627 212* CALL WRITE
00630 213* 50 IQT=I
00631 214* CALL WRITE
00632 215* CALL NTRAN (8,9)
00633 216* STOP
00633 217* MPREC=0
00634 218* 13 CALL READ ($30)
00635 219* LPN=LPN+KF

```



```

00636 220* IF (LPN.EQ.1) GO TO 14
00640 221* IF (NPREC.EQ.1) LPN=LPN+32
00642 222* IF (15*IP,1,1-OR,1FLAG.EQ.1) GO TO 19
00644 223* GO TO 16
00645 224* 14 LAST=LPN-1
00646 225* DO 15 I=1,IAST
00651 226* CX=CX+1
00652 227* 15 CASE(CX)=LOAD(I)
00654 228* 16 A3A=FLO(D,6,LOAD(LP+2))
00655 229* IF (A3A.EQ.COE) GO TO 17
00657 230* I=FLO(2,4,LOAD(LP+3))
00660 231* J=FLO(32,4,LOAD(LP+2))
00661 232* A4A=I+J*10
00662 233* A4B=FLO(8,4,LOAD(LP+3))
00663 234* I=2
00664 235* IF (A4B.EQ.6*OR,(A4B.EQ.7*AND,A4A.L7)) I=1
00666 236* IF (A4B.EQ.9*OP,(A4B.EQ.A*AND,A4A.GE.7)) I=3
00670 237* IHOR(1)=IHOR(1)-1
00671 238* IHOR(1+1)=IHOR(1+1)-1
00672 239* NX(1)=NX(1)+1
00673 240* J=NX(1)
00674 241* NUMB(I,J)=CX
00675 242* IUNIT=I+8
00676 243* WRITE (IUNIT) (CASE(C),C=1,CX)
00704 244* 100 FORMAT (2(16(1X,A6)/))
00705 245* 101 FORMAT (1H+,T120,UNIT=,I6)
00706 246* GO TO 19
00707 247* 17 LX=LX+1
00710 248* IF (LX.LE.4000) GO TO 172
00712 249* WRITE (6,171) IHOR(2)
00715 250* 171 FORMAT (1H,UNIT CASE DIMENSION EXCEEDED FOR OPFAC ',A5)
00716 251* STOP
00717 252* 172 LONG(LX)=CX
00720 253* LGFS=LGFS+LGF
00721 254* IF (LGF.EQ.0) LGFS=LGFS+1
00723 255* WRITE (39) (CASE(C),C=1,CX)
00731 256* IF (1FLAG.EQ.1) GO TO 20
00733 257* GO TO 19
00734 258* 18 ISKIP=1
00735 259* LPN=LP+32
00736 260* IF (LPN.LT.LMT) GO TO 19
00740 261* IF (NPREC.EQ.IPREC) GO TO 20
00742 262* IF (NPREC.EQ.JPREC) GO TO 30
00744 263* CALL READ ($30)
00745 264* LPN=LPN-KF
00746 265* IF (NPREC.EQ.1) LPN=LPN+32
00750 266* 19 LP=LPN
00751 267* IFLAG=0
00752 268* GO TO 10
00753 269* END

```

END OF UNITVAC 1108 FORTRAN V COMPILATION. 0 •DIAGNOSTIC• MESSAGE(S)

```

PHASE 1 TIME = 1 SEC.
PHASE 2 TIME = 0 SEC.
PHASE 3 TIME = 1 SEC.
PHASE 4 TIME = 0 SEC.
PHASE 5 TIME = 1 SEC.
PHASE 6 TIME = 0 SEC.

```

SUBROUTINE READ ENTRY POINT 000326

STORAGE USED (BLOCK, NAME, LENGTH)

0001 *CODE 000337
 0000 *DATA 000045
 0002 *BLANK 003427
 0003 *READ 000011
 0004 *WRITE 000011

EXTERNAL REFERENCES (BLOCK, NAME)

0005 NTRAN
 0006 CLOCKS
 0007 NADUS
 0010 NI02\$
 0011 NSTOP\$
 0012 NERR4\$
 0013 NERR3\$

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001	000007	IL	0000	000012	100F	0001	000233	11L	0001	000241	12L	0001	00014	154G	
0001	000217	177G	0001	000021	2L	0001	000034	3L	0001	000062	4L	0001	000115	5L	
0001	000121	6L	0001	000204	8L	0001	000211	9L	0004	000004	CF	0000	000004	DERR	
0004	000006	DP	0004	000005	DPN	0002	001000	QUMP	0002	R	002000	E	0000	000006	I
0003	000001	IF	0002	002005	IHOR	0003	000003	INREX	0004	000003	IORFX	0002	003423	IPREC	
0002	002003	IQT	0003	000000	IS	0000	000001	ISKIP	0003	000007	ISTAT	0000	000002	IT	
0002	002004	IUNIT	0000	000000	IX	0000	000003	JF	0002	002023	JHOR	0002	003424	JPREC	
0000	000007	JX	0003	000004	KF	0004	000002	LIM	0003	000002	LMT	0002	000000	LOAD	
0003	000006	LP	0003	000005	LPN	0000	000010	LX	0002	003426	MPREC	0000	000005	NEXT	
0004	000001	NF	0000	000011	NPR	0002	003425	NPREC	0004	000000	NS	0004	000007	NSTAT	
0002	R	002001	X0	0002	R	002002	X1								

00101 1* SUBROUTINE READ (S)
 00103 2* IMPLICIT INTEGER (A,B,C,D)
 00104 3* COMMON LOAD(512),DUMP(512),E,X0,X1,IQT,IUNIT,IHDR(14),JHOR(768),
 00104 4* I IPREC,JPREC,NPREC,MPREC
 00105 5* COMMON /BREAD/ IS,IF,LMT,INREX,KF,LPN,LP,ISTAT(2)
 00106 6* COMMON /WRITE/ NS,NF,LIM,IORFX,CF,DPN,DP,NSTAT(2)
 00107 7* IF (MPREC.GT.0) GO TO 1
 00111 8* IX=1
 00112 9* ISKIP=0
 00113 10* I IT=3-IS
 00114 11* JF=258-IF
 00115 12* KF=1F+1F-2

```

00116 13* 2 IF (ISTAT(11)+1),2,6
00121 14* IF (ISTAT(11)+2),5,6
00124 15* DERR=1
00126 16* 3 CALL NTRAN (7,22)
00127 17* CALL NTRAN (7,7,-1)
00130 18* CALL NTRAN (7,2,256,LOAD(JF),ISTAT(11))
00133 19* 4 IF (ISTAT(11)+1),4,6
00136 20* IF (ISTAT(11)+2),5,6
00137 21* DERR=DERR+1
00141 22* IF (DERR.LT.4) GO TO 3
00142 23* NEXT=INREX+1
00145 24* WRITE (6,100) NEXT
00146 25* STOP
00147 26* 5 RETURN 1
00151 27* 6 IF (IQT.EQ.1) RETURN
00153 28* IF (MPREC.GT.0.OR.ISKIP.EQ.1) GO TO 9
00156 29* DO 7 1=1,256
00157 30* JX=1X+1-1
00160 31* LX=JF+1-1
00162 32* 7 JHOR(JX)=LOAD(LX)
00164 33* IF (IX.EQ.1) NPR=JHOR(1)
00165 34* NPR=NPR-1
00167 35* IF (NPR.EQ.0) GO TO 8
00170 36* 1X=1X+256
00171 37* GO TO 12
00172 38* 8 JPREC=JHOR(3)
00173 39* ISKIP=1
00174 40* GO TO 12
00176 41* 9 IF (NPREC.GT.0) GO TO 11
00201 42* DO 10 1=1,14
00202 43* LX=JF+1-1
00204 44* 10 IHOR(1)=LOAD(LX)
00205 45* IPREC=IHOR(10)
00206 46* 11 NPREC=NPREC+1
00207 47* MPREC=MPREC+1
00210 48* 12 INREX=INREX+1
00211 49* CALL NTRAN (7,2,256,LOAD(1F),ISTAT(15))
00212 50* IS=1T
00213 51* IF=JF
00214 52* LMT=708-LMT
00215 53* CALL CLOCKS(X1)
00217 54* IF (X1.GE.XD) RETURN 1
00221 55* IF (MPREC.EQ.0) GO TO 1
00222 56* RETURN
00223 57* 100 FORMAT (1H,'UNABLE TO READ RECORD ',16)
00223 58* ENO

```

ENO OF UNIVAC 1108 FORTRAN V COMPILATION. 0 *DIAGNOSTIC* MESSAGE(S)

```

PHASE 1 TIME = 0 SEC.
PHASE 2 TIME = 0 SEC.
PHASE 3 TIME = 0 SEC.
PHASE 4 TIME = 0 SEC.
PHASE 5 TIME = 0 SEC.
PHASE 6 TIME = 1 SEC.

```

TOTAL COMPILATION TIME = 1 SEC

SUBROUTINE WRITE ENTRY POINT 000170

STORAGE USED (BLOCK, NAME, LENGTH)

0001 *CODE 000174
0000 *DATA 000045
0002 *BLANK 003427
0003 BREAD 000011
0004 BWRITE 000011

EXTERNAL REFERENCES (BLOCK, NAME)

0005 NTRAN
0006 NHDUS
0007 NIO2S
0010 NSTOPS
0011 NERR3S

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001	000012 IL	0000	000004 IOOF	0000	000012 IOIF	0001	000025 2L	0001	000053 3L
0001	000106 4L	0001	000115 SL	0004	000004 CF	0000	000002 OERR	0004	000006 DP
0004	000005 QPN	0002	001000 DUMP	0002	R 002000 E	0003	000001 IF	0002	002005 IHOR
0003	000003 INREX	0004	000003 IOREX	0002	003423 IPREC	0002	002003 IQT	0003	000000 IS
0003	000007 ISTAT	0002	002004 IUNIT	0002	002023 JHDR	0002	003424 JPREC	0003	000004 KF
0004	000002 LIM	0003	000002 LMT	0002	000000 LOAD	0003	000006 LP	0003	000005 LPN
0000	000001 MF	0002	003426 MPREC	0000	000003 NEXT	0004	000001 NF	0002	003425 NPREC
0004	000000 NS	0004	000007 NSTAT	0000	000000 NT	0002	R 002001 XO	0002	R 002002 XI

00101	1*	SUBROUTINE WRITE
00103	2*	IMPLICIT INTEGER (A,B,C,D)
00104	3*	COMMON LOAD(512),DUMP(512),E,XD,X1,IQT,IUNIT,IHDR(14),JHDR(768),
00104	4*	1 IPREC,JPREC,NPREC,NPREC
00105	5*	COMMON /BREAD/ IS,IF,LMT,INREX,KF,LPN,LP,ISTAT(2)
00106	6*	COMMON /BWRITE/ NS,NF,LIM,IOREX,CF,DPN,DP,NSTAT(2)
00107	7*	NT=3-NS
00110	8*	MF=258-NF
00111	9*	CF=NF-NF-2
00112	10*	1 IF (NSTAT(NT)+1),1,5
00115	11*	IF (NSTAT(NT)+2),4,5
00120	12*	DERR=1
00121	13*	2 CALL NTRAN (8,22)
00122	14*	CALL NTRAN (8,7,-1)
00123	15*	CALL NTRAN (8,1,256,DUMP(MF),NSTAT(NT))
00124	16*	3 IF (NSTAT(NT)+1),3,5
00127	17*	IF (NSTAT(NT)+2),4,5

```

00132      DEPR=DEPR+1
00133      IF (DEPR*11.4) GO TO 2
00135      DEAT=IOREX+1
00136      WRITE (6,100) TEXT
00141      STOP
00142      4 WRITE (6,101)
00144      STOP
00145      5 IF (IOT.EQ.1) RETURN
00147      IOREX=IOREX+1
00150      CALL NTRAN (8,1,256,DUMP(QF),NSTAT(NS))
00151      NS=NT
00152      NF=NF
00153      LINE700-LIM
00154      RETURN
00155      100 FORMAT (1H.,UNABLE TO WRITE RECORD ',16)
00156      101 FORMAT (1H.,ATTEMPT TO WRITE PAST END OF TAPE')
00157      END

```

END OF UNIVAC 1100 FORTRAN V COMPILATION. 0 *DIAGNOSTIC* MESSAGE(S)

```

PHASE 1 TIME = 0 SEC.
PHASE 2 TIME = 0 SEC.
PHASE 3 TIME = 0 SEC.
PHASE 4 TIME = 0 SEC.
PHASE 5 TIME = 0 SEC.
PHASE 6 TIME = 1 SEC.

```

TOTAL COMPILATION TIME = 1 SEC

MUC130

Like MUTAPE, the source deck must be changed in this program to correspond to the data being processed. In the main program, card number 7 must state information for the C-130 station under consideration. The district number goes in columns 20 and 21, OPFAC number in columns 25 - 29, and OPFAC name in column 36 of card 7 through column 11 of card 8. In the listing given, the C-130 station is San Francisco, OPFAC number 20180 in district 12.

In Subroutine READ, the OPFAC number must be given in card 7, columns 21 through 25.

[illegible]

```

00207 05*
00210 00*
00213 01*
00214 02*
00215 03*
00216 04*
00217 05*
00220 06*
00221 07*
00222 08*
00223 09*
00224 10*
00225 11*
00226 12*
00227 13*
00230 14*
00231 15*
00232 16*
00233 17*
00234 18*
00236 19*
00237 20*
00240 21*
00241 22*
00244 23*
00246 24*
00247 25*
00250 26*
00252 27*
00253 28*
00255 29*
00256 30*
00260 31*
00261 32*
00263 33*
00265 34*
00267 35*
00270 36*
00271 37*
00272 38*
00273 39*
00274 40*
00276 41*
00300 42*
00301 43*
00302 44*
00303 45*
00304 46*
00312 47*
00314 48*
00315 49*
00316 50*
00320 51*
00322 52*
00323 53*
00324 54*
00325 55*
00326 56*
00327 57*

LAST=LP+1
GO TO 17,LP
NX=NX+1
DATA(0)=LOAD( )
LP=LP+1
1. IX=1
IF=24
K03=0
C130=0
GO TO 17
1. IF=FLD(8,4,DATA(IX))
J=FLD(2,4,DATA(IX))
K03=I+J*10-1
1. IY=IX
IZ=IX+18
17 IF=FLD(20,4,DATA(IY))
J=FLD(14,4,DATA(IY))
O1=I+J*10
IF (O1.EQ.60) GO TO 19
FLD(12,24,DATA(IY))=0
JY=IY+1
JZ=IY+8
DO 16 JX=JY,JL
1 DATA(JX)=0
GO TO 20
1. C130=1
2. IF (K03.EQ.0) GO TO 21
K03=K03-1
IF (K03.EQ.0) GO TO 22
IY=IY+9
IF (IY.LE.12) GO TO 17
21 IX=IX+32
IF (K03.GT.0) GO TO 16
IF (LGF.GT.1) GO TO 15
22 IF (C130.EQ.0) GO TO 23
I=FLD(2,4,DATA(4))
J=FLD(32,4,DATA(3))
A4A=I+J*10
A4B=FLD(8,4,DATA(4))
I=2
IF (A4B.EQ.6.OR.(A4B.EQ.7.AND.A4A.LT.7)) I=1
IF (A4B.EQ.9.OR.(A4B.EQ.8.AND.A4A.GE.7)) I=3
IX(I)=IX(I)+1
NX=NX+1
NWD5(NX)=OX
LGFS=LGFS+LGF
WRITE (36) (DATA(0),D=1,OX)
23 IF (IQUIT.EQ.1) GO TO 40
GO TO 10
30 LP=LP+32
IF (LPN.LT.LMT) GO TO 31
IF (NPREC.EQ.NPREC) GO TO 40
CALL READ ($50)
LPN=LP+KF
31 LP=LPN
GO TO 10
32 IQUIT=1
GO TO 14

```

```

00350 117*      + IF (IA.EQ.0) G TO 50
00352 118*      DP=IF
00353 119*      CHDR(10)=(LGFS+A)/3
00354 120*      CHDR(11)=IX
00355 121*      CHDR(12)=IX(1)
00356 122*      CHDR(13)=IX(2)
00357 123*      CHDR(14)=IX(3)
00358 124*      DO 41 I=1,32
00359 125*      DX=DP+I-1
00360 126*      IF (1.LE.14) J=DP(DX)=CHDR(I)
00361 127*      41 IF (1.51.14) DUMP(DX)=0
00362 128*      DP=DP+32
00363 129*      REWIND 36
00364 130*      DO 44 N=1,NX
00365 131*      DX=NWD5(N)
00366 132*      READ (36) (DATA(D),D=1,DX)
00367 133*      DP=DP+DX
00368 134*      LAST=DPN-1
00369 135*      IEND=NF+255
00370 136*      IF (LAST.GT.IEND) LAST=IEND
00371 137*      DX=0
00372 138*      DO 42 I=DP,LAST
00373 139*      DX=DX+1
00374 140*      DUMP(I)=DATA(DX)
00375 141*      IF (DPN.LI.LIW) GO TO 44
00376 142*      WRITE (6,100) (DUMP(D),D=NF,IEND)
00377 143*      100 FORMAT (2(16(1X,A6)/))
00378 144*      CALL WRITE
00379 145*      DPN=DPN-CF
00380 146*      IF (DPN.EQ.NF) GO TO 44
00381 147*      LAST=DPN-1
00382 148*      DO 43 I=NF,LAS;
00383 149*      DX=DX+1
00384 150*      43 DUMP(I)=DATA(DX)
00385 151*      44 DP=DPN
00386 152*      IF (DP.EQ.NF) GO TO 50
00387 153*      IEND=NF+255
00388 154*      DO 45 I=DP,IEND
00389 155*      45 DUMP(I)=0
00390 156*      WRITE (6,100) (DUMP(D),D=NF,IEND)
00391 157*      CALL WRITE
00392 158*      50 IGT=1
00393 159*      CALL WRITE
00394 160*      CALL NTRAN (8,0)
00395 161*      STOP
00396 162*      END

```

END OF UNIVAC 1103 FORTRAN V COMPILATION. 0 *DIAGNOSTIC* MESSAGE(S)

```

PHASE 1 TIME = 0 SEC.
PHASE 2 TIME = 0 SEC.
PHASE 3 TIME = 1 SEC.
PHASE 4 TIME = 0 SEC.
PHASE 5 TIME = 0 SEC.
PHASE 6 TIME = 1 SEC.

```

TOTAL COMPILATION TIME = 2 SEC

LET FOR K-AD-32-20
UNIVAC 1100 F) KIRAN V LEVEL 2206 013 F5013D
THIS COMPILATION WAS DONE ON 05 JUL 71 AT 04:37:47

[illegible]

at Turbulence JS-2 (BL-CR, AVE, Lr. vol 1)

0001	*CODE	000411
0000	*DATA	000054
0002	*BLANK	003427
0003	BREND	000011
0004	BWFILE	000011

EXTERNAL REFERENCES (BLOCK NAME)

0005	NIRAN
0006	CLOCKS
0007	NW008
0010	NW008
0011	NSIDP#
0012	NW008
0013	NER343
0014	NER338

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001	00017 1L	0000	00017 100F	0000	000014 101F	0001	00027 11L	0001	000303 12L
0001	00012 106	0001	000031 2L	0001	000232 2006	0001	000245 2116	0001	000044 3L
0001	000072 4L	0001	000125 5L	0001	000131 6L	0001	000220 8L	0001	000237 9L
0004	000005 4F	0000	000005 0ER	0004	000006 0P	0004	000005 0PA	0002	001000 0IMP
0002	002006 5	0000	000007 1	0003	000001 1F	0002	002003 1HDR	0003	000003 1NEXP
0000	000000 1PFAC	0004	000003 1OREX	0002	003423 1PREC	0002	003422 1OT	0003	000000 1S
0000	000002 1SKIP	0003	000007 1STAT	0000	000003 1T	0002	003421 1U01T	0000	000001 1X
0000	000013 1	0000	000004 1F	0002	002021 1HDR	0002	003424 1PREC	0000	000010 1X
0003	000004 1F	0004	000002 1IM	0003	000002 1VT	0002	000000 1OAD	0003	000006 1P
0003	000005 1PA	0000	000011 1X	0002	003426 1PREC	0000	000006 1NYT	0004	000001 1NF
0000	000012 1NR	0002	003425 1PREC	0004	000000 1S	0004	000007 1STAT	0002	002001 1X0
0000	002002 1X								

```

00101 1* SUBROUTINE READ (B)
00102 2* IMPLICIT INTEGER (A,B,C,D)
00103 3* COMMON LOAD(512),JUMP(512),EXX0,X1,THDR(14),JHDR(768),IUIJIT,IOT,
00104 4* IPREC,JPREC,NPREC,MPREC
00105 5* COMMON /BREAD/ IS,IF,LMT,INREX,KF,LPN,LP,ISTAT(2)
00106 6* COMMON /BWRITE/ NS,NF,LM,IOREX,CF,DPN,DPA,NSTAT(2)
00107 7* DATA IOPFAC /'20180 ' /
00108 8* IF (LUNIT.EQ.7.0).MPREC=61.0 GO TO 1
00109 9* IX=1
00110 10* ISKIP=0

```

```

00115 11* IF=5-15
00116 12* JF=256-IF
00117 13* KF=IF+IF=2
00120 14* IF (ISTAT(11)+1),2,5
00123 15* IF (ISTAT(11)+1),5,5
00126 16* DEORE=1
00127 17* CALL NTRAV (IUNIT,22)
00130 18* CALL NTRAV (IUNIT,7,-1)
00131 19* CALL NTRAV (IUNIT,256,LOAD(JF),ISTAT(11))
00132 20* IF (ISTAT(11)+1),4,5
00133 21* IF (ISTAT(11)+1),5,6
00140 22* DERREDEX+1
00141 23* IF (DERREDEX) GO TO 3
00143 24* NEXTINREX+1
00144 25* WRITE (6,100) *EXT
00147 26* STOP
00150 27* RETURN 1
00151 28* IF (IOT.EQ.1) RETURN
00153 29* IF (IUNIT.EQ.7) GO TO 12
00155 30* IF (MPREC.GT.0.0R.ISKIP.EQ.1) GO TO 9
00157 31* DO 7 I=1,256
00162 32* JX=IX+I-1
00163 33* LX=JF+I-1
00164 34* 7 JHOP(JX)=LOAD(LX)
00166 35* IF (IX.EQ.1) JPR=JHOP(1)
00170 36* JPR=JPR-1
00171 37* IF (JPR.EQ.0) GO TO 8
00173 38* IX=IX+256
00174 39* GO TO 12
00175 40* 8 ISKIP=1
00176 41* WRITE (6,101) (JHOP(J),J=1,JX)
00204 42* 101 FORMAT (2(16(1X,A6/)))
00205 43* GO TO 12
00206 44* 9 IF (MPREC.GT.0) GO TO 11
00210 45* DO 10 I=1,14
00213 46* LX=JF+I-1
00214 47* 10 JHOP(I)=LOAD(LX)
00216 48* JPRE=JHOP(10)
00217 49* IF (JHOP(2).EQ.10PFAC) GO TO 11
00221 50* CALL NTRAV (IUNIT,7,JPRE-1)
00222 51* GO TO 12
00223 52* 11 NPREC=JPRE+1
00224 53* JPRE=JPRE+1
00225 54* 12 INREX=INREX+1
00226 55* CALL NTRAV (IUNIT,256,LOAD(IF),ISTAT(15))
00227 56* IS=11
00230 57* IF=JF
00231 58* LMT=703-LMT
00232 59* CALL CLOCKS(X1)
00233 60* IF (X1.GE.X0) RETURN 1
00235 61* IF (IUNIT.EQ.9.AND.*JPRE.EQ.0) GO TO 1
00237 62* RETURN
00240 63* 100 FORMAT (1H *,*UNABLE TO READ RECORD *,I6)
00241 64* END

```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 0 *DIAGNOSTIC* MESSAGE(S)

PHASE 1 TIME = 0 SEC.
PHASE 2 TIME = 0 SEC.

ALL FOR WRITE*WRITE
UNIVAC 1105 FORTRAN V LEVEL 2205 010 050183
THIS COMPILATION WAS DONE ON 06 AUG 71 AT 04:37:48

SUBROUTINE WRITE ENTRY POINT 000170

STORAGE USED (BLOCK, NAME, LENGTH)

0001 *CODE 000174
0000 *DATA 000045
0002 *BLOCK 000427
0003 *BREAD 000011
0004 *WRITE 000011

EXTERNAL REFERENCES (BLOCK, NAME)

0005 NTRAN
0006 NWRITE
0007 NIOFS
0010 NSTOP
0011 NERFS

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001	000012	1L	0000	000004	10NF	0000	000012	101F	0001	000025	2L	0001	000053	3L
0001	000106	4L	0001	000115	5L	0004	000004	CF	0000	000002	DEPR	0004	000006	DP
0004	000005	DPN	0002	000100	DUMP	0002	R	002000	E	0003	000001	IF	002003	IHDR
0005	000003	I,REX	0004	000003	IOREX	0002	I	003423	IPREC	0002	003422	IQT	000000	IS
0005	000007	ISTAT	0002	0003421	IUNIT	0002	I	002021	JHDR	0002	003424	JPREC	000004	KF
0004	000002	LIM	0003	000002	LMT	0002	I	000000	LOAD	0003	000006	LP	000005	LPN
0000	000001	MF	0002	0003426	NPREC	0000	I	000003	NEXT	0004	000001	NF	003425	NPREC
0004	000000	N5	0004	000007	NSTAT	0000	I	000000	NT	0002	R	002001	X0	

```

00101 1* SUBROUTINE WRITE
00103 2* IMPLICIT INTEGER (A,B,C,D)
00104 3* COMMON LOAD(512),DUMP(512),E,X0,X1,IHDR(14),JHDR(768),IUNIT,IQT,
00104 4* 1,IPREC,JPREC,NPREC,NPREC
00105 5* COMMON /BREAD/ IS,IF,LMT,INREX,F,LPN,LP,ISTAT(2)
00106 6* COMMON /WRITE/ NS,NF,LIM,IOREX,CF,DPN,DP,NSTAT(2)
00107 7* NT=3-NS
00110 8* MF=258-NF
00111 9* CF=NF*NF-2
00112 10* 1 IF (NSTAT(NT)+1),1,5
00115 11* IF (NSTAT(NT)+2),4,5
00120 12* DERR=1
00121 13* 2 CALL NTRAN (8,22)
00122 14* CALL NTRAN (8,7,-1)
00123 15* CALL NTRAN (8,1,256,DUMP(MF),NSTAT(NT))
00124 16* 3 IF (NSTAT(NT)+1),3,5
00127 17* IF (NSTAT(NT)+2),4,5

```

```

00132 18* DEK=JERR+1
00133 19* IF (JERR.LT.4) GO TO 2
00135 20* NEXT=JERR+1
00136 21* WRITE (6,100) EXT
00137 22* STOP
00138 23* WRITE (6,101)
00139 24* STOP
00140 25* IF (JUT.EQ.1) RETURN
00141 26* JERR=JERR+1
00142 27* CALL NFRAN (B,1,255,DUMP(JF),JSTAT(NS))
00143 28* NSENT
00144 29* NF=WF
00145 30* LINE=768-LIM
00146 31* RETURN
00147 32* 100 FORMAT (1H,'UNABLE TO WRITE RECORD','I6)
00148 33* 101 FORMAT (1H,'ATTEMPT TO WRITE PAST END OF TAPE')
00149 34* END

END OF UNIVAC 1104 FORTRAN V COMPILATION. 0 *DIAGNOSTIC* MESSAGE(S)

PHASE 1 TIME = 0 SEC.
PHASE 2 TIME = 0 SEC.
PHASE 3 TIME = 1 SEC.
PHASE 4 TIME = 0 SEC.
PHASE 5 TIME = 0 SEC.
PHASE 6 TIME = 0 SEC.

TOTAL COMPILATION TIME = 1 SEC

```


ALL FOR PUP,POP
JNIVAC 1106 FOR PUP V LEVEL 2206 0013 550130
THIS COMPILED BY JOE G. H. JUL 71 AT 15:22:36

MAIN PROGRAM

STORAGE USED (BLOCK, NAME, LENGTH)

0001 *CODE 003504
0000 *DATA 025313
0002 *BLANK 022545

EXTERNAL REFERENCES (BLOCK, NAME)

0003 NIKAN
0004 REA
0005 NUCASE
0006 RAN
0007 REES
0010 NRJIS
0011 NI023
0012 NI013
0013 NI023
0014 COS
0015 TAN
0016 SIN
0017 ATA
0020 ASI
0021 NI033
0022 NSI0P5

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001	000051	1L	0001	009405	10L	0001	002362	1007G	025154	101F	0001	002411	1017G
0000	025157	1.2F	0000	025161	103F	0000	025172	134F	000300	105L	0000	025175	106F
0000	025160	1.7F	0001	002553	1070G	0001	002601	11L	000601	1107G	0001	002776	111L
0001	001057	112L	0001	001133	118L	0000	025177	115F	001163	116L	0001	002767	1160G
0001	001172	117L	0001	001175	118L	0001	001263	119L	001355	121L	0001	003123	1212G
0001	001365	122L	0001	003257	1272G	0001	001312	13L	003374	1360G	0001	001371	14L
0001	003435	1403G	0001	003443	1411G	0001	003453	1417G	003466	1427G	0001	000043	151G
0001	000055	140G	0001	001423	17L	0001	000202	2L	001450	20L	0001	003403	200L
0000	025220	211F	0000	025223	203F	0000	025242	205F	001474	21L	0001	001551	24L
0001	001553	25L	0001	002020	26L	0001	001773	269L	001777	261L	0001	002022	27L
0001	002027	24L	0001	002224	29L	0001	000210	3L	025245	300F	0000	025253	302F
0001	000343	313G	0001	000420	323G	0001	002240	35L	002243	351L	0001	000627	353G
0001	001006	377G	0001	000226	4L	0001	001141	424G	001206	450G	0001	001333	477G
0000	025152	498F	0000	025153	493F	0001	000303	5L	002246	50L	0000	025165	500F
0001	002275	511L	0001	002344	512L	0001	002347	52L	002356	520L	0001	001413	526G
0001	002436	54L	0001	002440	54L	0001	002454	55L	001501	557G	0001	002503	56L
0001	001530	576	0001	002531	60L	0001	001722	640G	001756	652G	0001	002005	666G
0001	003400	7L	0001	002533	70L	0001	002033	703G	002063	713G	0001	002644	72L
0001	002573	73L	0001	002614	74L	0001	002706	75L	002303	763G	0001	002333	773G
0001	002717	74L	0001	002743	80L	0001	003017	91L	003021	90L	0001	003153	92L

0000	020215	9 F	0001	003315	94L	0000	025015	9 F	0000	025211	06E	0001	003136	97L				
0000	020215	9 F	0001	003320	94L	0000	025070	A	0000	I	025130	A550	0002	I	002313	A1A		
0000	020275	A13	0002	I	002457	A3	0002	I	002472	A3A	0002	I	022645	A0LD	0000	I	025151	A4
0000	020299	A1A	0002	I	002491	A43	0002	I	013240	3	0000	I	025114	A0	0000	I	001136	0AC5AF
0000	020313	3 AT	0000	I	025115	3A	0002	I	002467	31A	0002	I	002470	B12A	0002	I	002471	B13
0000	020273	316	0002	I	022545	316A	0002	I	002074	3173	0002	I	002463	B3	0002	I	002463	B5
0000	020284	3	0002	I	002405	38	0002	I	002656	39	0000	I	025112	C45E5	0000	I	024556	C6ELTA
0000	010237	C6L60	0000	I	025150	C6A4MA	0002	I	015750	C4AX	0000	I	000004	C0DE	0000	I	025147	C7E
0000	020275	C1	0002	I	002557	C1A	0002	I	022546	C130	0002	I	002550	C5	0002	I	002641	C7A1
0000	020273	C7A2	0002	I	003005	C731	0002	I	003067	C732	0002	I	003315	C9	0000	I	025126	D
0000	017257	D1TA	0002	I	016735	DELTA	0000	I	025073	D1ST	0000	I	000023	DLAT	0000	I	000045	DLON
0000	017258	D1	0002	I	003377	D1	0002	I	011267	D11	0002	I	012253	D12	0002	I	013242	D13
0000	022547	D130	0002	I	004353	73	0002	I	005347	D4	0002	I	001324	D4FRST	0002	I	004333	D6
0000	025136	D5FRST	0000	I	025121	D6TOT	0002	I	007317	D8	0002	I	010303	D9	0000	R	025137	ENRTE
0000	000000	F	0000	R	000001	FC	0000	R	000002	FF	0000	I	025140	FRST	0000	I	025127	F5
0000	022035	G4MA	0002	I	002310	I	0002	I	017247	IF	0000	I	022527	IHDR	0000	I	025107	I1
0000	025116	L	0002	I	022363	I NEED	0002	I	017253	INREX	0002	I	017252	IPREC	0002	I	017245	I0T
0000	017246	L5	0002	I	001000	ISTAT	0000	I	025066	I3	0002	I	002311	J	0000	I	025110	JJ
0000	025122	K	0000	I	025113	K45E5	0002	I	025357	KF	0000	I	025123	KK	0000	I	025117	KOPFAC
0000	025141	LAST	0000	I	025074	LAT	0000	I	025142	LEAVE	0002	I	002312	LGF	0000	I	000155	LIST
0000	017250	LAT	0002	I	000000	LOAD	0000	I	025075	LONG	0002	I	017254	LP	0002	I	017255	LPN
0000	015751	LVTIME	0002	I	022362	M	0000	I	025145	MAXTOS	0000	I	000626	MEED	0002	I	022360	MFLAG
0000	025134	MILES	0002	I	022634	VINC1	0000	I	025130	MINO	0000	I	000034	MLAT	0000	I	000056	MLON
0000	022361	N	0002	I	015130	NEED	0000	I	025143	NITEHR	0002	I	022632	NN	0002	I	014226	NO
0000	017251	NPREC	0000	I	025111	NS	0000	I	025135	NSEV	0002	I	001002	OPFAC	0000	R	025072	P1
0000	000007	P0B	0000	R	000100	R	0006	R	000000	RAND	0002	R	003151	RC7A	0002	R	003233	RC7R
0000	0001																	

[illegible]

00113	18*	C
00113	19*	C
00113	20*	C

217

```

21* READ (5,40) C140
22* READ (9,42) Q140
23* FORMAT (A3)
24* +9 FORMAT (1J)
25* +9 FORMAT (1J)
26* READ (5,101) J ST, LAT, LONG, ZDET, XGT, YGT, XLON1, YLAT, XLON2, YLONG1,
27* 1 XDELTA, YDELTA
28* READ (316,F7.3,RI5)
29* READ (5,102) (G0A3(I),I=1,70)
30* FORMAT (20I4)
31* READ (5,102) (I1NEED(I),I=1,100)
32* 1 READ (5,107,EN=2) J
33* 107 FORMAT (I2)
34* I=J
35* IF (J.GT.9) I=J-10
36* READ (9,103) J,AT(I),MLAT(I),DLON(I),MLON(I),PROP(I)
37* FORMAT (2X,I3,I2),6X,F5.3)
38* CONVERT DISTRICT CENTROIDS TO RADIANS
39*
40* RLAT(I)=(J,AT(I)+MLAT(I)/60.0)*.017453
41* RLON(I)=(J,LON(I)+MLON(I)/60.0)*.017453
42*
43* CALCULATE DISTANCE FROM EACH DISTRICT TO DISTRICT ORIGIN(MILES)
44*
45* XDIST(I)=(LONG-DLON(I)*60+MLON(I))*ZETA/60
46* YDIST(I)=(J,AT(I)*60+MLAT(I))-LAT
47* WRITE (6,500) J,RLAT(I),DLAT(I),MLAT(I),DLON(I),MLON(I),MLON(I),
48* 1 PROP(I),XDIST(I),YDIST(I)
49* 500 FORMAT (1X,I2,2(F12.8,2I6),F7.3,2I6)
50* GO TO 1
51* ? IF=I
52* JJ=J
53* NS=1
54* READ (5,104,EN=4) LIST(NS),XSTA(NS),YSTA(NS)
55* 104 FORMAT (6X,I5,4X,2I6)
56* NS=NS+1
57* GO TO 3
58* 4 NS=NS-1
59* READ (5,106) B(1),B(2)
60* 106 FORMAT (2(0I2,1X))
61* CASES=0
62* KASES=0
63* BO=0
64* BA=0
65* IL=0
66* WFLAG=0
67* KOPFAC=1
68* STAMAX=0
69* IOT=0
70* CALL NTRAN(7,2,256,LOAD(1),ISTAT(1))
71* IS=2
72* IF=257
73* LMT=462
74* NPREC=0
75* I1REX=0
76* CALL READ (120 )
77* LP=IF+32
78* GO TO 5

```

```

00267 R-AD A CASE FROM CAR TAPE
00268
00269
00270
00271
00272
00273
00274
00275
00276
00277
00278
00279
00280
00281
00282
00283
00284
00285
00286
00287
00288
00289
00290
00291
00292
00293
00294
00295
00296
00297
00298
00299
00300
00301
00302
00303
00304
00305
00306
00307
00308
00309
00310
00311
00312
00313
00314
00315
00316
00317
00318
00319
00320
00321
00322
00323
00324
00325
00326
00327
00328
00329
00330
00331
00332
00333
00334
00335
00336
00337
00338
00339
00340
00341
00342
00343
00344
00345
00346
00347
00348
00349
00350
00351
00352
00353
00354
00355
00356
00357
00358
00359
00360
00361
00362
00363
00364
00365
00366
00367
00368
00369
00370
00371
00372
00373
00374
00375
00376
00377
00378
00379
00380
00381
00382
00383
00384
00385
00386
00387
00388
00389
00390
00391
00392
00393
00394
00395
00396
00397
00398
00399
00400
00401
00402
00403
00404
00405
00406
00407
00408
00409
00410
00411
00412
00413
00414
00415
00416
00417
00418
00419
00420
00421
00422
00423
00424
00425
00426
00427
00428
00429
00430
00431
00432
00433
00434
00435
00436
00437
00438
00439
00440
00441
00442
00443
00444
00445
00446
00447
00448
00449
00450
00451
00452
00453
00454
00455
00456
00457
00458
00459
00460
00461
00462
00463
00464
00465
00466
00467
00468
00469
00470
00471
00472
00473
00474
00475
00476
00477
00478
00479
00480
00481
00482
00483
00484
00485
00486
00487
00488
00489
00490
00491
00492
00493
00494
00495
00496
00497
00498
00499
00500
00501
00502
00503
00504
00505
00506
00507
00508
00509
00510
00511
00512
00513
00514
00515
00516
00517
00518
00519
00520
00521
00522
00523
00524
00525
00526
00527
00528
00529
00530
00531
00532
00533
00534
00535
00536
00537
00538
00539
00540
00541
00542
00543
00544
00545
00546
00547
00548
00549
00550
00551
00552
00553
00554
00555
00556
00557
00558
00559
00560
00561
00562
00563
00564
00565
00566
00567
00568
00569
00570
00571
00572
00573
00574
00575
00576
00577
00578
00579
00580
00581
00582
00583
00584
00585
00586
00587
00588
00589
00590
00591
00592
00593
00594
00595
00596
00597
00598
00599
00600
00601
00602
00603
00604
00605
00606
00607
00608
00609
00610
00611
00612
00613
00614
00615
00616
00617
00618
00619
00620
00621
00622
00623
00624
00625
00626
00627
00628
00629
00630
00631
00632
00633
00634
00635
00636
00637
00638
00639
00640
00641
00642
00643
00644
00645
00646
00647
00648
00649
00650
00651
00652
00653
00654
00655
00656
00657
00658
00659
00660
00661
00662
00663
00664
00665
00666
00667
00668
00669
00670
00671
00672
00673
00674
00675
00676
00677
00678
00679
00680
00681
00682
00683
00684
00685
00686
00687
00688
00689
00690
00691
00692
00693
00694
00695
00696
00697
00698
00699
00700
00701
00702
00703
00704
00705
00706
00707
00708
00709
00710
00711
00712
00713
00714
00715
00716
00717
00718
00719
00720
00721
00722
00723
00724
00725
00726
00727
00728
00729
00730
00731
00732
00733
00734
00735
00736
00737
00738
00739
00740
00741
00742
00743
00744
00745
00746
00747
00748
00749
00750
00751
00752
00753
00754
00755
00756
00757
00758
00759
00760
00761
00762
00763
00764
00765
00766
00767
00768
00769
00770
00771
00772
00773
00774
00775
00776
00777
00778
00779
00780
00781
00782
00783
00784
00785
00786
00787
00788
00789
00790
00791
00792
00793
00794
00795
00796
00797
00798
00799
00800
00801
00802
00803
00804
00805
00806
00807
00808
00809
00810
00811
00812
00813
00814
00815
00816
00817
00818
00819
00820
00821
00822
00823
00824
00825
00826
00827
00828
00829
00830
00831
00832
00833
00834
00835
00836
00837
00838
00839
00840
00841
00842
00843
00844
00845
00846
00847
00848
00849
00850
00851
00852
00853
00854
00855
00856
00857
00858
00859
00860
00861
00862
00863
00864
00865
00866
00867
00868
00869
00870
00871
00872
00873
00874
00875
00876
00877
00878
00879
00880
00881
00882
00883
00884
00885
00886
00887
00888
00889
00890
00891
00892
00893
00894
00895
00896
00897
00898
00899
00900
00901
00902
00903
00904
00905
00906
00907
00908
00909
00910
00911
00912
00913
00914
00915
00916
00917
00918
00919
00920
00921
00922
00923
00924
00925
00926
00927
00928
00929
00930
00931
00932
00933
00934
00935
00936
00937
00938
00939
00940
00941
00942
00943
00944
00945
00946
0094
```

```

00075 157*
00076 158*
00077 159*
00078 160*
00079 161*
00080 162*
00081 163*
00082 164*
00083 165*
00084 166*
00085 167*
00086 168*
00087 169*
00088 170*
00089 171*
00090 172*
00091 173*
00092 174*
00093 175*
00094 176*
00095 177*
00096 178*
00097 179*
00098 180*
00099 181*
00100 182*
00101 183*
00102 184*
00103 185*
00104 186*
00105 187*
00106 188*
00107 189*
00108 190*
00109 191*
00110 192*
00111 193*
00112 194*

*140=1
DO 112 J=2,11
IF (D.EQ.4.0N*.E9.5) GO TO 112
IF ((11.E*.JJ.AND.D.E9.2).OR.(11.NE.JJ.AND.D.E9.5)) GO TO 112
IF (ABS(R(I)).LT.A35(R(MI(J)))) *IND=0
112 CONTINUE
A1A(I)=MI*ND
IF (11.NE.JJ) A1A(I)=A1A(I)+10
IF (A1A(I).EQ.01ST) A13(I)=0
X=((LU*G-(C781(I)*60+C782(I))*-BETA)/60
Y=(C7A1(I)*60+C792(I))-LAT
GO TO 14
113 IF (CFLAG.EQ.1) GO TO 113
DO 114 J=1,NS
IF (A13(J).EQ.11ST(J)) GO TO 113
114 CONTINUE
WRITE (6,115) A13(I)
115 FORMAT (1H,'UNKNOWN' LOCATION FOR OPFAC',I6)
X=XPT
Y=YPT
GO TO 117
116 X=XSTA(J)+XDELTA
Y=YSTA(J)+YDELTA
117 IF (MFLAG.EQ.0) GO TO 14
GO TO 13
118 RN=RAND(0.8)
DO 119 D=1,11,-1
IF (D.EQ.4.0N*.E9.6) GO TO 119
IF ((11.E*.JJ.AND.D.E9.2).OR.(11.NE.JJ.AND.D.E9.5)) GO TO 119
IF (RN.LT.PROB(D)) ASSD=0
119 CONTINUE
A1A(I)=ASSD
IF (11.NE.JJ) A1A(I)=A1A(I)+10
IF (A1A(I).EQ.01ST) A13(I)=0
X=XD1ST(ASSD)
Y=YD1ST(ASSD)
GO TO 14
120 CONTINUE
IF (X.NE.999.999.999) GO TO 122
DO 120 J=1,NS
IF (A13(J).EQ.11ST(J)) GO TO 121
121 CONTINUE
WRITE (6,115) A13(I)
X=XPT
Y=YPT
GO TO 14
121 X=XSTA(J)+XDELTA
Y=YSTA(J)+YDELTA
GO TO 14
122 X=XX
Y=YY
C
C SET DATE AND TIME OF NOTIFICATION 'MINC1'
C
C 1+ MINC1=C1(1)
C
C CONVERT 'A13(I)' TO 'STAND', THE PRIMARY STATION
C

```

```

00021 190* IF (NFLAG.EQ.0.AND.CFLAG.EQ.0) GO TO 20
00022 190* IF (A13(1).EQ.0) OPFAC) GO TO 21
00023 197* NO 15 I=1,STAVAX
00024 195* IF (A13(1).E..OPFAC(I)) GO TO 17
00025 194* 1 CONTINUE
00026 200* GO TO 20
00027 201* 1/ STANO=I
00028 202* IF (KOPFAC.EQ.0.AND.OPFAC(STAVAX-1).EQ.0) STANO=STANO-1
00029 203* IF (A13(1).EQ. ) STANO=0
00030 204* KOPFAC=A13(1)
00031 205* GO TO 21
00032 206* 2/ IF (A13(1).EQ.0) OPFAC) GO TO 21
00033 207* STAVAX=STAVAX+1
00034 208* OPFAC(STAVAX)=A13(1)
00035 209* STANO=STAVAX
00036 210* IF (KOPFAC.EQ.0) STANO=STANO-1
00037 211* IF (A13(1).EQ.0) STANO=0
00038 212* KOPFAC=A13(1)
00039 213* 21 DO 25 I=1,CMAX
00040 214* IF (C5(I).EQ.5 OR.C5(I).EQ.6) GO TO 25
00041 215* IF (NO(I).EQ.0) GO TO 25
00042 216* 3/ CALCULATE SEARCH ILES *TSM*
00043 217* K=NO(I)
00044 218* DO 24 J=1,K
00045 219* IF (J1(I,J).GE.70) GO TO 24
00046 220* IF (J2(I,J).EQ.0) GO TO 24
00047 221* D6TOT=D6TOT+D6(I,J)
00048 222* BOAT=D1(I,J)
00049 223* TSM=TSM+((J2(I,J)*SOA3(BOAT))
00050 224* 24 CONTINUE
00051 225* 25 CONTINUE
00052 226* TSM=(TSM*9)/10
00053 227* 3/ CALCULATE MILES OFF-SHORE, *MILES*
00054 228* ZMILE=B16
00055 229* IF (B16A.EQ.5) ZMILE=0
00056 230* IF (B16.EQ.1.AND.B16.EQ.6) ZMILE=.35
00057 231* IF (B16.EQ.7.OR.B16.EQ.8) ZMILE=.3
00058 232* IF (B16.EQ.9) ZMILE=.8
00059 233* IF (B16.EQ.999) ZMILE=.95
00060 234* MILES=ZMILE*10
00061 235* 4/ CONVERT SEVERITY *NSEV*
00062 236* NSEV=B17B
00063 237* IF (B17B.EQ.3) NSEV=4
00064 238* IF (B17B.EQ.4) NSEV=3
00065 239* IF (NSEV.LT.1) NSEV=1
00066 240* IF (NSEV.GT.5) NSEV=5
00067 241* IF (D6TOT.GT.5) GO TO 50
00068 242* 5/ SORT SEARCH CASE
00069 243* 51=0
00070 244* 52=1
00071 245*
00072 246*
00073 247*
00074 248*
00075 249*
00076 250*
00077 251*
00078 252*

```


[illegible]


```

00761 311* LAST=0
00762 312* DO 512 I=1,CVAX
00763 313* IF (C5(I).EQ.5*10.0R.C5(I).EQ.0) GO TO 512
00764 314* IF (C6(I).EQ.0) GO TO 512
00771 315* K=J(I)
00772 316* DO 51 J=1,K
00773 317* IF (C6(I,J).LI.70) GO TO 52
00774 318* 51 CONTINUE
01001 319* 512 CONTINUE
01003 320* GO TO 520
01004 321* 52 FIRST=4(I,J)
01005 322* LAST=FIRST+16(I,J)
01006 323* DO 54 I=1,CVAX
01011 324* IF (C5(I).EQ.5*10.0R.C5(I).EQ.0) GO TO 54
01013 325* IF (C6(I).EQ.0) GO TO 54
01015 326* K=H(I)
01019 327* DO 55 J=1,K
01021 328* IF (C6(I,J).GE.70) GO TO 53
01023 329* IF (C6(I,J).LI.FIRST) FIRST=4(I,J)
01025 330* LEAVE=D4(I,J)+16(I,J)
01026 331* IF (LEAVE.GT.LAST) LAST=LEAVE
01030 332* 55 CONTINUE
01032 333* 53 CONTINUE
01034 334* TE=LAST-FIRST
01035 335* IF (C6TOT.GT.140) GO TO 55
01037 336* NITEHR=0
01040 337* GO TO 56
01041 338* 55 IF (TE.LE.240) NITEHR=TE-140
01043 339* IF (TE.GT.240) NITEHR=TE/2.4
01045 340* Z=D6TOT
01046 341* S1=Z/(TE-NITEH)+.99
01047 342* IF (S1.GT.10) S1=10
01051 343* DO 52=0
01052 344* GO TO 28
01052 345* C CALCULATE 'DELTA(I)' AND 'GAMMA'
01052 346* C
01053 347* C
01053 348* 70 MAXTOS=0
01054 349* SUMTOS=0
01055 350* TE=0
01056 351* CFE=0
01057 352* IF (N.EQ.0) GO TO 75
01061 353* I=SAVEI(1)
01062 354* J=SAVEJ(1)
01063 355* FIRST=J4FIRST(I,J)
01064 356* LAST=LVTIME(I,J)
01065 357* IF (N.EQ.1) GO TO 74
01067 358* DO 71 K=2,N
01072 359* I=SAVEI(K)
01073 360* J=SAVEJ(K)
01074 361* IF (J4FIRST(I,J).LT.FIRST) FIRST=J4FIRST(I,J)
01076 362* IF (LVTIME(I,J).GT.LAST) LAST=LVTIME(I,J)
01100 363* 71 CONTINUE
01102 364* 74 TE=LAST-FIRST
01103 365* CIE=TE
01104 366* IF (TE.LT.0) TE=0
01106 367* DO 75 K=1,N
01111 368* I=SAVEI(K)

```

```

01112 357* J=JAVEJ(K)
01113 370* IF (IE.NE.TOS(1)) GO TO 72
01114 371* DELTA(K)=6
01115 372* CDELTA(K)=0
01116 373* GO TO 75
01117 374* 73 DELTA(K)=(100*(.04F35T(I,J)-F35T(I,J)+F35T(I,J)))/(TE-TOS(K))
01118 375* CDELTA(K)=DELTA(K)
01119 376* IF (DELTA(K).GT.99) DELTA(K)=99
01120 377* IF (DELTA(K).LT.0) DELTA(K)=0
01121 378* IF (TOS(K).GT.MAXTOS) MAXTOS=TOS(K)
01122 379* SU-TOS=SUMTOS+TOS(K)
01123 380* 74 CONTINUE
01124 381* 75 NN=1
01125 382* IF (SUMTOS.NE.MAXTOS) GO TO 78
01126 383* GAMMA=0
01127 384* CGAMMA=0
01128 385* GO TO 80
01129 386* 7 GAMMA=(100*(TE-MAXTOS))/(SUMTOS-MAXTOS)
01130 387* CGAMMA=GAMMA
01131 388* IF (GAMMA.GT.99) GAMMA=99
01132 389* IF (GAMMA.LT.0) GAMMA=0
01133 390* IF (.EQ.0) GO TO 90
01134 391* NN=NN+1
01135 392* TOS(NN)=0
01136 393* DELTA(NN)=0
01137 394* NEED(NN)=NEED(1)
01138 395* IF (.EQ.1) GO TO 90
01139 396* DO 81 I=2,N
01140 397* IF (NEED(I).EQ.10) GO TO 81
01141 398* IF (NEED(NN).EQ.10.OR.NEED(NN).EQ.19.OR.NEED(1).EQ.17)
01142 399* 1 NEED(NN)=NEED(I)
01143 400* 81 CONTINUE
01144 401* 90 A4=A4A*10+A43
01145 402* IF (B13.EQ.50) B13=56
01146 403* IF (B13.EQ.51) B13=101
01147 404* IF (B13.EQ.52) B13=201
01148 405* IF (B13.GE.50.AND.B13.LE.99) B13=0
01149 406* IF (NN.NE.0) GO TO 97
01150 407* IF (S1.GT.0.04.S2.GT.0) GO TO 95
01151 408* KASES=KASES-1
01152 409* 90=90+1
01153 410* IF (30.GT.100) GO TO 5
01154 411* DO 91 J=1,DX
01155 412* 9X=9X+1
01156 413* BUCASE(3X)=DATA(3)
01157 414* 91 CONTINUE
01158 415* GO TO 5
01159 416* 92 NEED(1)=0
01160 417* TOS(1)=0
01161 418* DELTA(1)=0
01162 419* IF (STAVO.LE.7.OR.A1B(1).EQ.0170)
01163 420* 1WRITE (6,95) (IA(1),STAVO,03,AP,MINC1,03,05,06,08,09,B10,B12A,
01164 421* 2 013,NSEV
01165 422* 95 FORMAT (1H,06I6,112,7I6)
01166 423* IF (STAVO.LE.7.OR.A1B(1).EQ.0130)
01167 424* 1WRITE (6,96) (N1,N1W,GAMMA,TE,51,52,TSM,MILES,X,Y,CGAMMA,CIF
01168 425* 96 FORMAT (1H,13I6)
01169 426* IF (STAVO.GT.7.AND.A1B(1).NE.0170) GO TO 90
01170 427*

```



```

00109 7* 2 D6(50,10),D8(50,10),D9(50,10),I1(50,10),I2(50,10),CFLAG,B(2),
00110 8* 3 D13(50,10),NOT(50),SAVEI(200),SAVEJ(200),NEED(200),TOS(200),CMAX,
00111 9* 4 LVTIME(50,10),DELTA(200),IOT,IS,IF,LMT,NPREC,IPREC,INREV,LP,
00112 10* 5 LPA,DX,DATA(1-100),KF,MFLAG,N,M,INEED(100),IHDR(14),B16A,C130,D130
00113 11* COMMON C5(50),IN,STAND,WINDCL,GAMMA,TE,TSM,S1,S2,ZMILE,X,Y,A3OLD
00114 12* DATA BLANK/,
00115 13* 1 I1=3-15
00116 14* JF=258-IF
00117 15* KF=IF+IF-2
00118 16* 2 IF (ISTAT(I1)+1),2,5
00119 17* IF (ISTAT(I1)+2),5,5
00120 18* DERR=1
00121 19* 3 CALL NTRAV (7,22)
00122 20* CALL NTRAV (7,7,-1)
00123 21* CALL NTRAV (7,2,255,LOAD(JF),ISTAT(I1))
00124 22* 4 IF (ISTAT(I1)+1),4,5
00125 23* IF (ISTAT(I1)+2),5,5
00126 24* DERR=DERR+1
00127 25* IF (DERR,LT,4) GO TO 3
00128 26* NEXT=INREX+1
00129 27* WRITE (6,100) EXT
00130 28* 100 FORMAT (1H,UNABLE TO READ RECORD ,I6)
00131 29* STOP
00132 30* 5 RETURN 1
00133 31* 6 IF (IOT.EQ,1) RETURN
00134 32* 10 IF (NPREC.GT,0) GO TO 12
00135 33* DO 11 I=1,14
00136 34* LX=JF+I-1
00137 35* 11 IHDR(I)=LOAD(LX)
00138 36* IPREC=IHDR(10)
00139 37* MFLAG=0
00140 38* CFLAG=0
00141 39* IF (IHDR(2).EQ,BLANK) MFLAG=1
00142 40* IF (IHDR(2).EQ,C130) CFLAG=1
00143 41* 12 NPREC=NPREC+1
00144 42* 13 INREX=INREX+1
00145 43* CALL NTRAV (7,2,255,LOAD(IF),ISTAT(JS))
00146 44* IS=IF
00147 45* IF=JF
00148 46* LMT=703-LMT
00149 47* RETURN
00150 48* END

```

END OF UNIVAC 1108 FORTAN V COMPILATION. 0 *DIAGNOSTIC* MESSAGE(S)

```

PHASE 1 TIME = 0 SEC.
PHASE 2 TIME = 1 SEC.
PHASE 3 TIME = 0 SEC.
PHASE 4 TIME = 0 SEC.
PHASE 5 TIME = 0 SEC.
PHASE 6 TIME = 0 SEC.

```

TOTAL COMPILATION TIME = 1 SEC

SUBROUTINE NEEDS ENTRY POINT 000356

STORAGE USED (BLOCK, NAME, LENGTH)

0001 *CODE 000374
0000 *DATA 000014
0002 *BLANK 022646

EXTERNAL REFERENCES (BLOCK, NAME)

0003 NER334

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001	000236	1L	0001	000312	1L	0001	000250	12L	0001	000346	13L	0001	000351	14L
0001	000122	2L	0001	000127	5L	0001	000224	7L	0001	000231	8L	0002	I	002313
0002	I	002375	A18	0002	I	002457	A3	0002	I	002472	A3A	0002	I	002460
0002	I	002461	A48	0002	I	013240	B	0002	I	002467	B10	0002	I	002471
0002	I	002473	B16	0002	I	002545	P15A	0002	I	002474	B178	0002	I	002463
0002	I	002484	B5	0002	I	002465	B8	0002	I	002466	B3	0002	I	002450
0002	I	002475	C1	0002	I	002557	C1A	0002	I	002546	C130	0002	I	015750
0002	I	002723	C7A2	0002	I	003005	C7A1	0002	I	002546	C130	0002	I	002641
0002	I	017257	JATA	0002	I	016735	DELTA	0002	I	003067	C732	0002	I	002641
0002	I	012253	J12	0002	I	013242	J13	0002	I	003377	C9	0002	I	000000
0002	I	001324	U4FIRST	0002	I	006333	D6	0002	I	003377	D1	0002	I	011267
0002	I	002310	I	0002	I	017247	IF	0002	I	003377	D1	0002	I	005347
0002	I	017255	I0REC	0002	I	017245	I0T	0002	I	004363	D3	0002	I	005347
0002	I	022357	KF	0002	I	002312	L6F	0002	I	010303	D9	0002	I	002635
0002	I	017255	LPN	0002	I	015751	LVT14E	0002	I	022363	INFED	0002	I	017253
0002	I	022351	N	0002	I	015130	NEED	0002	I	001000	ISTAT	0002	I	002311
0002	I	001002	OPFAC	0002	R	003151	RC7A	0002	I	000000	LOAD	0002	I	017254
0002	I	001160	S/A3	0002	I	002633	STA10	0002	I	022360	MFLAG	0002	I	022334
0002	I	015440	TOS	0002	I	022637	TSM	0002	R	014226	NO	0002	I	017251
				0002	I	022643	X	0002	I	014310	SAVEI	0002	I	014620
								0002	I	022641	S2	0002	I	022636
										022644	Y	0002	R	022642
														ZMILF

00101	1*	SUBROUTINE NEEDS
00103	2*	IMPLICIT INTEGER (A,B,C,D,E,G,H,I,S,T,X,Y)
00104	3*	COMMON LOAD(512),ISTAT(2),OPFAC(110),SOA3(100),O4ERST(50,10),I,J
00105	4*	COMMON L6F,A1A(50),A1J(50),A3,A4A,A4B,33,B5,B6,B9,B10,H12A,R13
00106	5*	COMMON A3A,R16,B17,C1(50),C1A(50),C7A1(50),C7A2(50),C731(50),
00106	6*	1 C742(50),RC7A(50),RC73(50),C9(50),J1(50,10),J3(50,10),J4(50,10),
00106	7*	2 J6(50,10),J8(50,10),J9(50,10),J11(50,10),J12(50,10),CFLA6,B(2),
00106	8*	3 J13(50,10),NO(50),SAVEI(200),SAVEJ(200),NFED(200),TOS(200),CMAJ,
00106	9*	4 LVTINE(50,10),DELTA(200),I0T,I0,IF,LWT,NPREC,IPREC,INPREC,LP,
00106	10*	5 LPN,X,JATA(1500),KF,4FLA6,N,Y,INFED(100),IHUR(114),B16A,C130,D130
00107	11*	COMMON C5(50),R,STAN0,MINC1,GAMMA,TE,TSM,S1,S2,ZMILE,X,Y,A30LN
00110	12*	Z=C1(1,J)/10

```

00111 13*
00112 14*
00113 15*
00114 16*
00115 17*
00116 18*
00117 19*
00118 20*
00119 21*
00120 22*
00121 23*
00122 24*
00123 25*
00124 26*
00125 27*
00126 28*
00127 29*
00128 30*
00129 31*
00130 32*
00131 33*
00132 34*
00133 35*
00134 36*
00135 37*
00136 38*
00137 39*
00138 40*
00139 41*
00140 42*
00141 43*
00142 44*
00143 45*
00144 46*
00145 47*
00146 48*
00147 49*
00148 50*
00149 51*
00150 52*
00151 53*
00152 54*
00153 55*
00154 56*
00155 57*
00156 58*
00157 59*
00158 60*
00159 61*
00160 62*
00161 63*
00162 64*
00163 65*
00164 66*
00165 67*
00166 68*
00167 69*
00168 70*
00169 71*
00170 72*
00171 73*
00172 74*
00173 75*
00174 76*
00175 77*

IF (D12(I,J).EQ.0) GO TO 1
IF (D12(I,J).E.0) GO TO 5
IF (D12(I,J).E.63.OR.D12(I,J).EQ.65.OR.D12(I,J).EQ.66) GO TO 2
IF (D12(I,J).E.64.OR.D12(I,J).EQ.63.OR.D12(I,J).EQ.65) GO TO 2
IF (D12(I,J).E.66.OR.D12(I,J).EQ.60) GO TO 2
IF (D12(I,J).EQ.76.OR.D12(I,J).EQ.76) GO TO 2
D=D12(I,J)
NEED(N)=1/NEED(N)
RETURN
C NEED(N)=19
RETURN
5 IF (D12(I,J).E.0) GO TO 8
IF (D12(I,J).EQ.63.OR.D12(I,J).EQ.65) GO TO 7
IF (D12(I,J).EQ.66.OR.D12(I,J).EQ.69.OR.D12(I,J).EQ.76) GO TO 7
D=D12(I,J)
NEED(N)=1/NEED(N)
IF (D12(I,J).EQ.68.AND.B13.6E.60.AND.B13.LT.70) /NEED(N)=11
RETURN
7 NEED(N)=15
RETURN
3 NEED(N)=14
RETURN
10 IF (D12(I,J).E.0) GO TO 12
D=D12(I,J)
NEED(N)=1/NEED(N)
RETURN
12 IF (D12(I,J).EQ.0) GO TO 14
IF (D12(I,J).EQ.63.OR.D12(I,J).EQ.65) GO TO 13
IF (D12(I,J).EQ.66.OR.D12(I,J).EQ.69.OR.D12(I,J).EQ.76) GO TO 13
D=D12(I,J)
11 NEED(N)=1/NEED(N)
IF (B13.6E.60.AND.B13.LT.70.AND.D.EQ.68) NEED(N)=17
RETURN
13 D=68
GO TO 11
14 NEED(N)=16
RETURN
END

```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 0 *DIAGNOSTIC* MESSAGE(S)

```

PHASE 1 TIME = 1 SEC.
PHASE 2 TIME = 0 SEC.
PHASE 3 TIME = 0 SEC.
PHASE 4 TIME = 0 SEC.
PHASE 5 TIME = 0 SEC.
PHASE 6 TIME = 1 SEC.

```

TOTAL COMPILATION TIME = 2 SEC

JIT FOR HUCASE, HUCATE
 UNIVAC 1105 FORTRAN V LEVEL 2205 01-15 F50183
 THIS COMPILATION WAS DONE ON 03 JUL 71 AT 13:22:44

0A JUL 71

15:22:44.177

SUBROUTINE HUCASE ENTRY POINT 000342

STORAGE USED (BLOCK, NAME, LENGTH)

0001 *CODE 000352
 0000 *DATA 000031
 0002 *BLANK 022640

EXTERNAL REFERENCES (BLOCK, NAME)

0003 READ
 0004 FIELD
 0005 FIELD3
 0006 FIELD2
 0007 NERR46
 0010 NERR36

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001	000054	1276	0001	000315	199L	0001	000322	240L	0001	000000	201L	0001	000032	205L
0001	000114	213L	0001	000154	214L	0001	000171	215L	0001	000175	214L	0001	000222	217L
0001	000237	220L	0001	000267	221L	0001	000272	240L	0001	000133	250L	0002	I 002313	A1A
0002	I 002375	A1B	0002	I 002457	A3	0002	I 002472	A3A	0002	I 022645	A30LD	0002	I 002460	A4A
0002	I 002461	A4B	0002	I 013240	A	0002	I 002474	B17A	0002	I 002470	B12A	0002	I 002471	B13
0002	I 002473	B16	0002	I 022545	B16A	0002	I 002474	B17A	0002	I 002462	B3	0002	I 002463	B5
0002	I 002464	B5	0002	I 002465	B8	0002	I 002466	B9	0002	I 013237	CFLAG	0002	I 015750	CMAA
0002	I 002475	C1	0002	I 002557	C1A	0002	I 022546	C130	0002	I 022550	C5	0002	I 002641	C7A1
0002	I 002723	C7A2	0002	I 003005	C731	0002	I 003067	C732	0002	I 003315	C9	0002	I 017257	DATA
0002	I 016735	DELTA	0002	I 017256	DX	0002	I 000005	D2	0002	I 003377	D1	0002	I 011267	D11
0002	I 012253	D12	0002	I 013242	D13	0002	I 022547	D130	0002	I 004363	D3	0002	I 005347	D4
0002	I 001324	D4FIRST	0002	I 000333	D6	0002	I 007317	D8	0002	I 010303	D9	0002	I 022635	GAMMA
0002	I 002310	I	0000	I 000003	IEAD	0002	I 017247	IF	0002	I 022527	IHDR	0002	I 022363	INEEN
0002	I 017453	IJEX	0000	I 000001	IPARTS	0002	I 017252	IPREC	0002	I 017245	IQT	0002	I 017246	IS
0002	I 001000	ISTAT	0000	I 000017	IX	0000	I 000013	I7	0000	I 000011	I1	0000	I 000012	I2
0002	I 002311	J	0000	I 000004	K	0000	I 000016	KD3	0002	I 022357	KF	0000	I 000002	LAST
0002	I 002312	LGF	0002	I 017250	LAT	0002	I 000008	L7A3	0002	I 017254	LP	0002	I 017255	LPN
0002	I 015751	LTIME	0002	I 022362	M	0002	I 022360	MFLAG	0002	I 022634	MINC1	0002	I 022361	N
0000	I 000006	NXX	0002	I 015130	NEED	0000	I 000000	NL3F	0002	I 022632	N4	0002	I 014226	N0
0002	I 017251	NPREC	0002	I 001002	OPFAC	0002	R 003151	RC7A	0002	R 003233	RC7B	0002	I 014310	SAVEI
0002	I 014620	SAVEJ	0002	I 001160	SOA3	0002	I 022633	STANO	0002	I 022640	S1	0002	I 022641	S2
0002	I 022636	TE	0002	I 015440	T05	0002	I 022637	T54	0002	I 022643	X	0002	I 022644	Y
0002	R 022642	ZFILE												

00101 1* SUBROUTINE HUCASE (\$)
 00103 2* IMPLICIT INTEGER (A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y)
 00104 3* COMMON LOAD(S1),ISTAT(2),OPFAC(110),SOA3(100),D4FIRST(50,10),I,J

[illegible]

00105	4
00106	5
00107	6
00108	7
00109	8
00110	9
00111	10
00112	11
00113	12
00114	13
00115	14
00116	15
00117	16
00118	17
00119	18
00120	19
00121	20
00122	21
00123	22
00124	23
00125	24
00126	25
00127	26
00128	27
00129	28
00130	29
00131	30
00132	31
00133	32
00134	33
00135	34
00136	35
00137	36
00138	37
00139	38
00140	39
00141	40
00142	41
00143	42
00144	43
00145	44
00146	45
00147	46
00148	47
00149	48
00150	49
00151	50
00152	51
00153	52
00154	53
00155	54
00156	55
00157	56
00158	57
00159	58
00160	59
00161	60
00162	61
00163	62
00164	63
00165	64
00166	65
00167	66
00168	67
00169	68
00170	69
00171	70
00172	71
00173	72
00174	73
00175	74
00176	75
00177	76
00178	77
00179	78
00180	79
00181	80
00182	81
00183	82
00184	83
00185	84
00186	85
00187	86
00188	87
00189	88
00190	89
00191	90
00192	91
00193	92
00194	93
00195	94
00196	95
00197	96
00198	97
00199	98
00200	99

00214	02*	IF (LPI.LI.L4I) GO TO 021	
00216	03*	IF (NPREC.LO.IPREC) NPREC=0	
00220	04*	CALL READ(J200)	
00221	05*	LPNELP1=KF	
00222	06*	IF (NPREC.LO.I) LPNELP1*2	
00224	07*	221 LPNELP1	
00225	08*	GO TO 201	
00226	09*	240 NO(I)=J	
00227	70*	CMAX=I	
00230	71*	IF (MFLAG.EQ.0) GO TO 193	
00232	72*	IF (I.EQ.IPARTS) GO TO 179	
00234	73*	I=I+1	
00235	74*	J=0	
00236	75*	DX=NDX	
00237	76*	CALL FIELD2	
00240	77*	GO TO 250	
00241	78*	194 DX=DZ	
00242	79*	RETURN	
00243	80*	200 RETURN 1	
00244	81*	END	
		END OF UNIVAC 1108 FORTRAN V COMPILATION.	0 *DIAGNOSTIC* MESSAGE(S)
		PHASE 1 TIME = 0 SEC.	
		PHASE 2 TIME = 0 SEC.	
		PHASE 3 TIME = 0 SEC.	
		PHASE 4 TIME = 0 SEC.	
		PHASE 5 TIME = 1 SEC.	
		PHASE 6 TIME = 0 SEC.	
		TOTAL COMPILATION TIME = 1 SEC	

SUBROUTINE FIELD ENTRY POINT 001362

FIELD ENTRY POINT 001365

FIELD ENTRY POINT 001370

STORAGE USED (BLOCK, NAME, LENGTH)

0001 *CODE 001373
0000 *DATA 000054
0002 *BLANK 022646

EXTERNAL REFERENCES (BLOCK, NAME)

0003 NEXT35

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001	000247	1L	0001	001015	10L	0001	000251	2L	0001	000314	3L	0001	000316	4L
0001	000362	5L	0001	000364	6L	0001	000430	7L	0001	000432	8L	0002	002313	A1A
0002	002375	A1B	0002	002457	A3	0002	002472	A3A	0002	022645	A3OLD	0002	002460	A4A
0002	002461	A4B	0002	0013240	B	0000	000002	BLANK	0002	002467	B1C	0002	002470	B12A
0002	002471	B13	0002	002473	B16	0002	002545	B16A	0002	002474	B17B	0002	002462	B3
0002	002463	B5	0002	002464	B6	0002	002465	B9	0002	002466	B9	0002	013237	CFLAG
0002	015750	C1AX	0002	002475	C1	0002	002557	C1A	0002	022546	C130	0002	022550	C5
0002	002541	C7A1	0002	002723	C7A2	0002	003005	C731	0002	003067	C7A2	0002	003315	C9
0002	017257	D1A1	0002	0016735	DELTA	0002	017256	DX	0002	003377	D1	0002	011267	D11
0002	012253	D12	0002	0013242	D13	0002	022547	D130	0002	004363	D3	0000	000013	D3A
0000	000014	D3B	0000	000015	D3C	0002	005347	D4	0000	000016	D4A	0000	000017	D4B
0000	000020	D4C	0002	001324	D4FIRST	0002	006333	D5	0002	007317	DA	0002	010303	D9
0002	022555	GAMMA	0002	002310	I	0002	017247	IF	0002	022527	IHDR	0002	023363	INDED
0002	017253	I1REX	0002	017252	I1PREC	0002	017245	I1T	0002	017246	I5	0002	001000	I5TAT
0000	000003	I1	0000	000004	I2	0000	000005	I3	0000	000006	I4	0000	000007	I5
0000	000010	I5	0000	000011	I7	0000	000012	I8	0002	002311	J	0002	022357	KF
0002	002312	L1F	0002	017250	LMT	0002	000000	LOAD	0002	017254	LP	0002	017255	LPN
0002	015751	L1TIME	0002	022362	M	0002	022360	NFLAG	0002	022634	MINC1	0002	023361	N
0000	000001	N1	0002	015130	NEED	0000	000000	NK	0002	022632	NN	0002	014226	NO
0002	017251	N1PREC	0002	001002	OFFAC	0002	R 003151	RC7A	0002	R 003233	RC7B	0002	014310	SAVEI
0002	014620	SAVEJ	0002	001160	SOA3	0002	022633	STAT0	0002	002640	S1	0002	022641	S2
0002	022536	T	0002	0015440	T05	0002	022637	T5M	0002	022643	X	0002	022644	Y
0002	R 022542	Z1ILE												

00101 1* SUBROUTINE FIELD
00103 2* IMPLICIT INTEGER (A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z)
00104 3* CO-4004 LOAD(512),L1STAT(2),OFFAC(110),SOA3(100),D4FIRST(50,10),I,J
00105 4* COMMON LGF,ALA(50),A13(50),A3,A4A,A4B,A3,B5,B6,B9,B10,B12A,B13

```

00100 5* 004000 A5A*310,R17*CI(50),C1A(*,0),C7A1(50),C7A2(50),C7A1(50),
00106 0* 1 C7*2(50),C7A(50),RC7*3(50),C8(*,0),C1(50,10),C3(50,10),C9(50,10),
00106 7* 2 D5(50,10),D8(50,10),D3(50,10),D4(50,10),D11(50,10),D12(50,10),CEL*5(3),
00106 9* 3 D13(50,10),D9(50,10),SAVEI(200),SAVEJ(200),NEED(200),TOS(200),C*AX,
00106 9* 4 L*TIME(50,10),ZELIA(200),I01,I*,IF,LMT,IPREC,IPREC,IPREC,LP,
00106 10* 5 LP*IP*AX,DAT*1(500),K*F,MEL*3(10),J*IF*FED(100),I*OP(14),R16A,C130*130
00107 11* 6 COMPO C5(50),N,STAR,MI,C1,G*AVA,IE,ISM,51,52,Z*ILE,X,Y,ASOLD
00110 12* 7 DATA NK/*,00000K/*,04/*,000ANA/*,BLANK/*,00000 */
00114 13* 8 A3A=FLD(0,0,DATA(DX+2))
00115 14* 9 I1=FLD(8,4,DATA(DX+2))
00116 15* 10 I2=FLD(11,4,DATA(DX+2))
00117 16* 11 I3=FLD(20,4,DATA(DX+2))
00120 17* 12 I4=FLD(26,4,DATA(DX+2))
00121 18* 13 A3=I1*100+I2*10+I3*10+I4
00122 19* 14 A3OLD=I2*100+I3*10+I4
00123 20* 15 I1=FLD(32,4,DATA(DX+2))
00124 21* 16 I2=FLD(2,4,DATA(DX+3))
00125 22* 17 A4=I1*10+I2
00126 23* 18 A4B=FLD(8,4,DATA(DX+3))
00127 24* 19 I1=FLD(26,4,DATA(DX+5))
00130 25* 20 I2=FLD(32,4,DATA(DX+5))
00131 26* 21 I3=FLD(2,4,DATA(DX+6))
00132 27* 22 I4=FLD(4,4,DATA(DX+6))
00133 28* 23 B5=I1*100+I2*100 +I3*10+I4
00134 29* 24 IF (35,GT,4095) B5=4095
00136 30* 25 I1=FLD(2,4,DATA(DX+7))
00137 31* 26 I2=FLD(8,4,DATA(DX+7))
00140 32* 27 I3=FLD(14,4,DATA(DX+7))
00141 33* 28 I4=FLD(20,4,DATA(DX+7))
00142 34* 29 I5=FLD(26,4,DATA(DX+7))
00143 35* 30 I6=FLD(32,4,DATA(DX+7))
00144 36* 31 I7=FLD(2,4,DATA(DX+8))
00145 37* 32 I8=FLD(8,4,DATA(DX+8))
00146 38* 33 B5=I1*100000+I2*100000+I3*100000+I4*10000+I5*1000+I6*100+I7*10
00146 39* 34 1+I8
00147 40* 35 IF (35,GT,130001) B5=130001
00151 41* 36 B6=FLD(12,12,DATA(DX+8))
00152 42* 37 IF (B6,EQ,NK,0),B6,EQ,NA,0R,B6,FQ,BLANK) GO TO 1
00154 43* 38 I1=FLD(14,4,DATA(DX+8))
00155 44* 39 I2=FLD(20,4,DATA(DX+8))
00156 45* 40 B6=I1*10+I2
00157 46* 41 GO TO 2
00160 47* 42 1 B6=99
00161 48* 43 B6=FLD(0,12,DATA(DX+9))
00162 49* 44 IF (B6,EQ,NK,0),B6,EQ,NA,0R,B6,FQ,BLANK) GO TO 3
00164 50* 45 I1=FLD(2,4,DATA(DX+9))
00165 51* 46 I2=FLD(8,4,DATA(DX+9))
00166 52* 47 B8=I1*10+I2
00167 53* 48 GO TO 4
00170 54* 49 3 B8=1
00171 55* 50 B9=FLD(12,12,DATA(DX+9))
00172 56* 51 IF (B9,EQ,NK,0),B9,EQ,NA,0R,B9,FQ,BLANK) GO TO 5
00174 57* 52 I1=FLD(14,4,DATA(DX+9))
00175 58* 53 I2=FLD(20,4,DATA(DX+9))
00176 59* 54 B9=I1*10+I2
00177 60* 55 GO TO 6
00180 61* 56 5 B9=1
00201 62* 6 B10=FLD(24,12,DATA(DX+9))

```

```

00202 00* IF (CX).EQ.0, 3, 10, 2, 4, 0, 10, 5, 3, 2, 0, 6, 7, 7
00203 04* I1=FLD(20,4,DATA(IX+9))
00205 05* I2=FLD(32,4,DATA(CX+9))
00206 06* I10=I1+I0+I2
00207 07* GO TO 8
00210 08* /
00211 09* 8
00212 70* I1=FLD(8,4,DATA(IX+10))
00213 71* I2=FLD(14,4,DATA(CX+10))
00214 72* I12A=I1+I0+I2
00215 73* I1=FLD(26,4,DATA(CX+10))
00216 74* I2=FLD(32,4,DATA(CX+10))
00217 75* I13=I1+I0+I2
00220 76* I10A=FLD(18,0,ATA(CX+11))
00221 77* I1=FLD(20,4,DATA(CX+11))
00222 78* I2=FLD(26,4,DATA(CX+11))
00223 79* I3=FLD(32,4,DATA(CX+11))
00224 80* I16=I1+I0+I2+I3
00225 81* I17=FLD(14,4,DATA(CX+12))
00227 82* IF (MFLAG.EQ.1) CX=CX+32
00230 83* ENTRY FIELD2
00231 94* I1=FLD(8,4,DATA(CX))
00232 95* I2=FLD(14,4,DATA(CX))
00233 96* I1A(I)=I1+I0+I2
00234 97* I1=FLD(20,4,DATA(CX))
00235 98* I2=FLD(26,4,DATA(CX))
00236 99* I3=FLD(32,4,DATA(CX))
00237 90* I4=FLD(2,4,DATA(CX+1))
00240 91* I5=FLD(8,4,DATA(CX+1))
00241 92* I18(I)=I1+I0+I2+I3+I0+I4+I5
00242 93* I1=FLD(14,4,DATA(CX+13))
00243 94* I2=FLD(20,4,DATA(CX+13))
00244 95* I3=FLD(26,4,DATA(CX+13))
00245 96* I4=FLD(32,4,DATA(CX+13))
00246 97* I5=FLD(2,4,DATA(CX+14))
00247 98* I6=FLD(8,4,DATA(CX+14))
00250 99* C1(I)=I1+I0+I2+I3+I0+I4+I5+I6
00251 100* C1A(I)=(I1+I0+I2)*240+I3+I0+I4+I5+I6+I6
00252 101* C5(I)=FLD(24,6,DATA(CX+15))
00253 102* I1=FLD(32,4,DATA(CX+16))
00254 103* I2=FLD(2,4,DATA(CX+17))
00255 104* I3=FLD(8,4,DATA(CX+17))
00256 105* I4=FLD(14,4,DATA(CX+17))
00257 106* C7A1(I)=I1+I0+I2
00260 107* C7A2(I)=I3+I0+I4
00261 108* I1=FLD(26,4,DATA(CX+17))
00262 109* I2=FLD(32,4,DATA(CX+17))
00263 110* I3=FLD(2,4,DATA(CX+18))
00264 111* I4=FLD(8,4,DATA(CX+18))
00265 112* I5=FLD(14,4,DATA(CX+18))
00266 113* C7B1(I)=I1+I0+I2+I0+I3
00267 114* C7A2(I)=I4+I0+I5
00270 115* I1=FLD(8,4,DATA(CX+19))
00271 116* I2=FLD(14,4,DATA(CX+19))
00272 117* I3=FLD(20,4,DATA(CX+19))
00273 118* I4=FLD(26,4,DATA(CX+19))
00274 119* C9(I)=I1+I0+I2+I0+I3+I4
00275 120* LGF=FLD(2,4,DATA(CX))
RETURN

```

```

00276      ENTRY FIELD3
00277      IF (FLD(12,24,ATA(XX)).NE.0) G 10 10
00301      J=J-1
00302      RETURN
00303      10
00304      I1=FLD(14,4,DATA(XX))
00305      I2=FLD(20,4,DATA(XX))
00306      O1(I,J)=I1*10+I2
00307      I1=FLD(26,4,DATA(XX+1))
00310      I2=FLD(32,4,DATA(XX+1))
00311      O3A=I1*10+I2
00312      I1=FLD(2,4,DATA(XX+2))
00313      I2=FLD(8,4,DATA(XX+2))
00314      O3B=I1*10+I2
00315      I1=FLD(14,4,DATA(XX+2))
00316      I2=FLD(20,4,DATA(XX+2))
00317      O3C=I1*10+I2
00320      O3(I,J)=O3A*24 +O3B*10+O3C/6
00321      IF (O3(I,J).EQ.0) O3(I,J)=C1A(I)
00322      I1=FLD(26,4,DATA(XX+2))
00323      I2=FLD(32,4,DATA(XX+2))
00324      O4A=I1*10+I2
00325      I1=FLD(2,4,DATA(XX+3))
00326      I2=FLD(8,4,DATA(XX+3))
00327      O4B=I1*10+I2
00330      I1=FLD(14,4,DATA(XX+3))
00331      I2=FLD(20,4,DATA(XX+3))
00332      O4C=I1*10+I2
00333      O4(I,J)=O4A*24+O4B*10+O4C/6
00334      IF (O4(I,J).EQ.0) O4(I,J)=D3(I,J)
00336      I1=FLD(14,4,DATA(XX+4))
00337      I2=FLD(20,4,DATA(XX+4))
00340      I3=FLD(26,4,DATA(XX+4))
00341      I4=FLD(32,4,DATA(XX+4))
00342      O6(I,J)=I1*1000+I2*100+I3*10+I4
00343      I1=FLD(32,4,DATA(XX+5))
00344      I2=FLD(2,4,DATA(XX+6))
00345      I3=FLD(8,4,DATA(XX+6))
00346      I4=FLD(14,4,DATA(XX+6))
00347      I1=FLD(20,4,DATA(XX+6))
00350      I2=FLD(26,4,DATA(XX+6))
00351      O9(I,J)=I1*10+I2
00352      I1=FLD(8,4,DATA(XX+7))
00353      I2=FLD(14,4,DATA(XX+7))
00354      O11(I,J)=I1*10+I2
00355      I1=FLD(20,4,DATA(XX+7))
00356      I2=FLD(26,4,DATA(XX+7))
00357      O12(I,J)=I1*10+I2
00360      I1=FLD(32,4,DATA(XX+7))
00361      I2=FLD(2,4,DATA(XX+8))
00362      O13(I,J)=I1*10+I2
00363      RETURN
00364      END

```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 0 *DIAGNOSTIC* MESSAGE(S)

PHASE 1 TIME = 1 SEC.
 PHASE 2 TIME = 0 SEC.
 PHASE 3 TIME = 1 SEC.

all FOR DEMONSTRATION
UNIVAC 1103 FORTRAN V LEVEL 2200 0018 250122
THIS COMPILATION WAS DONE ON 01 JUL 71 AT 17:04:23

08 JUN 71

18:08:23.703

MAIN PROGRAM

STORAGE USED (BLOCK, NAME, LENGTH)

0001 *CODE 003352
0000 *DATA 005176
0002 *BLANK 116563
0003 BIA 002733
0004 AIA 000357
0005 SEAL 000001

EXTERNAL REFERENCES (BLOCK, NAME)

0006 MELVIN
0007 NTRAN
0010 SELECT
0011 BOX
0012 ADD
0013 RAND
0014 ADTIME
0015 EXIT
0016 NRDJ\$
0017 NI01\$
0020 NI02\$
0021 NWDJ\$
0022 NR00\$
0023 NSTOP\$
0024 AL0G

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001	002703	100L	0001	003127	101L	0001	002454	102L	0001	001541	10216	0001	002456	103L
0001	001550	1030G	0001	001555	1036G	0001	002536	104L	0001	001605	1055G	0001	002415	106L
0001	001017	1064G	0001	002525	107L	0001	001637	1075G	0001	002607	110L	0001	002535	111L
0001	001720	1113G	0001	002606	114L	0001	002204	115L	0001	001745	116L	0001	002230	118L
0001	000012	120G	0001	002266	1217G	0001	000024	127G	0001	002430	1274G	0001	002515	1313G
0001	002537	1326G	0001	002555	1340G	0001	002572	1350G	0001	002573	1353G	0001	002616	1366G
0001	002045	1402G	0001	002646	1404G	0001	002743	1447G	0001	002761	1451G	0001	002773	1467G
0001	003005	1475G	0001	001075	15L	0001	003046	1511G	0001	003062	1521G	0001	000074	153G
0001	003156	1566G	0001	003174	1575G	0001	003207	1603G	0001	003217	1612G	0001	003232	1620G
0001	003245	1631G	0001	003251	1635G	0001	000116	164G	0001	003272	1645G	0001	003304	1656G
0001	003320	1665G	0001	003326	1672G	0001	003340	1702G	0001	000140	173G	0001	000140	176G
0001	000056	20L	0001	000145	202G	0001	001533	21L	0001	002270	222L	0001	000174	224G
0001	002304	224L	0001	000212	235G	0001	000224	244G	0001	000224	246G	0001	000667	25L
0001	000224	250G	0001	003276	255L	0001	003330	260L	0001	000245	266G	0001	002677	27L
0001	002413	28L	0001	001437	29L	0001	000265	301G	0001	000270	304G	0001	000312	312G
0001	000321	320G	0001	000406	346G	0001	000417	355G	0001	000426	363G	0001	000442	372G
0001	000375	4L	0001	000453	402G	0001	000464	412G	0001	000542	442G	0001	000600	456G
0001	000647	457G	0000	004561	500F	0000	004601	501F	0000	004603	502F	0000	004604	503F

```

00101 1*  PARAMETER  IP=24,N1=45,IXB=4500,NX3X=1000,NH=20,NCP=250,N5=75  *WS 2/7
00101 2*  ,N1=19,N4=48  *WS 2/71
00103 3*  INTEGER  VLW1X(NS,NV),STAT1,TTOS,CAS2DA(NS),TOTAL(NV),AGPAG8(NS)  *WS 2/71
00103 4*  ,CAL(12,31,4),TEMP(NV)
00104 5*  REAL  LAMBDA(N,NP)
00105 6*  DATA  AGRAGB /9,9,NA,9,4,4,4,4,3,2,3,9,9,9,NA,1,4,1,2,NA,NA,17,2,6,20*WS 2/71
00105 7*  ,13,15,34,29,NA,NA,NA,NA,NA,2,NA,2,27,37,1,NA,4,31,3,NA  *WS 2/71
00105 8*  ,16,23,28,6,42,25,47,4,5,35,40,46,10,30,43,33,1,2,11,3,9,38,NA  *WS 2/71
00105 9*  ,41,14,35,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA/  *WS 2/71
00107 10*  DIMENSION  ISWIT(10),IHOL(NH),LIST(2,2,2),DAYS(8),CBOX(9),NC(9,NP)
00107 11*  ,IADR(8,MX3),NADR(4,NX3),CPEP(NP),NCASES(NP),ICASE(ICP)
00107 12*  ,ISPPS(3),ISPP(3)
00107 13*  ,X(2)
00107 14*  ,ZNDAYS(8),NEWCAS(4,NP),XNLAW3(8,NP)
00107 15*  ,NCASE(NP),NEED(NV),TOS(NN),DELTA(NN)
00110 16*  COMMON/IX/  ISWIF,N5PPS,NSDPE,DELTA,IHOL,ILIST,LAMBDA,ITYPE,
00111 17*  ITSCED,ITSCED0,N5PP,TOHOLS,IDAY,TIME,DEL
00111 18*  COMMON/SEAL/  NUPPER
00112 19*

```

```

00113 C          MODRUM, IADR, NADR, NCASES, CPER, CDDV, ICASE
00114 CALL RELVIN
00115 LSFSEN=1
00116
00117 C      READ IN PARAMETERS DEFINING THE RUN
00118 C      ISWIT(1)=1--ISWIT(2)=0--CASELOAD SAME AS THAT OF INPUT TAGE
00119 C      ISWIT(2)=1 --CASELOAD CHANGES BY PERCENT
00120 C      ISWIT(1)=3 --CASES ARE INCREASED BY OPFAC AND CASE PARAMETER CRITERIA
00121 C
00122 C      READ(5,500) (ISWIT(I), I=1,10)
00123 C      FORMAT(10I1)
00124 C      WRITE(6,610) (ISWIT(I), I=1,10)
00125 C      FORMAT(10I1,20X, 'THE VALUES OF THE SWITCH ARRAY ARE', /1X,10I1)
00126 C      IF (ISWIT(2).EQ.1) READ(5,511) PER
00127 C      FORMAT(5F4.0)
00128 C      PER=PER/100.
00129 C      IF (ISWIT(10).EQ.1) READ(5,512) MAXCA
00130 C      FORMAT(15)
00131 C      IF (ISWIT(1).EQ.3) READ(5,516) MODRUM, (CPER(I), I=1, MODRUM)
00132 C      FORMAT(12,12(F5.2))
00133 C
00134 C      READ IN NO. OF PEAK PERIODS AND NO. OF YEARS IN CALENDAR
00135 C      START AND END WEEK OF EACH PEAK PERIOD
00136 C
00137 C      READ(5,501) NPP, IYEARS, (ISWPP(I), I=1, NPP)
00138 C      FORMAT(2I2,20I3)
00139 C      DO 1 IYR=1, IYEARS
00140 C      DO 1 IDAY=1, 31
00141 C      READ(5,502) (CAL(MO, IDAY, IYR), MO=1, 12)
00142 C      1 CONTINUE
00143 C      FORMAT(12I6)
00144 C
00145 C      READ SCENARIO LIMITS START HOUR START DAY END HOUR END DAY
00146 C
00147 C      READ(5,503) ISCSH, ISCSO, ISCEH, ISCED
00148 C      FORMAT(6I4)
00149 C      READ(5,506) NSPP, (NSPPS(I), I=1, NSPP)
00150 C      FORMAT(12,10I3)
00151 C
00152 C      READ TABLE OF HOLIDAYS
00153 C
00154 C      READ(5,504) NOHOLS, (IHOL(I), I=1, NOHOLS)
00155 C      FORMAT(12,20I3)
00156 C
00157 C      READ BOX IDENTIFICATION TABLE
00158 C
00159 C      READ(5,505) ((ILIST(I,J,K), I=1,2, J=1,2, K=1,2)
00160 C      FORMAT(8I1))
00161 C
00162 C      READ RANDOM NUMBER SEEDS
00163 C
00164 C      READ(5,509) B(1), B(2)
00165 C      FORMAT(012,1X,012)
00166 C      WRITE(6,609) B
00167 C      FORMAT(1, 'THE ORIGINAL RANDOM NUMBER SEED WAS ', 012, 012)
00168 C      READ(5,512) NSCEN
00169 C
00170 C      CALCULATE NO. OF DAYS IN EACH BOX USING THE CALENDAR
00171 C
00172 C
00173 C
00174 C
00175 C
00176 C
00177 C

```

*WS 2/71
*WS 2/71
*WS 3/71

```

00273 76* IF (ISAT(I),EQ,1) GO TO 63
00279 79* DO 2 I=1,DAYS
00300 80* DO 3 M=1,16
00303 81* K=0*8
00306 82* IF (K,GT,16) K=0*8
00307 83* IF (K,GT,16) K=0*8
00311 84* DO 4 IDAY=1,61
00314 85* IWK=CAL(K,IDAY,1)
00315 86* IF (IWK,EQ,0) GO TO 1
00317 87* DO 5 KK=1,NUPP
00322 88* IF (IWK,GE,ISAPP(KK),AND,IWK,LE,IFWPP(KK)) GO TO 6
00324 89* 5 CONTINUE
00326 90* J=5
00327 91* IF (MOD(IWK,2),NE,0) J=7
00331 92* DAYS(J)=DAYS(J)+1
00332 93* GO TO 4
00333 94* 6 J=1
00334 95* IF (MOD(IWK,2),NE,0) J=3
00336 96* DAYS(J)=DAYS(J)+1
00337 97* 4 CONTINUE
00341 98* 3 CONTINUE
00343 99* 2 CONTINUE
00345 100* DO 7 I=2,8,2
00350 101* DAYS(I)=DAYS(I-1)
00351 102* 7 CONTINUE
00353 103* WRITE(6,510) (DAYS(I),I=1,8)
00361 104* 510 FORMAT(2X,8F5.0)
00361 105* C
00361 106* C INITIALIZE COUNTERS
00361 107* C
00362 108* DO 10 I=1,8
00365 109* CBOX(I)=0.
00366 110* IUNIT=I+9
00367 111* CALL INTRAN(IUNIT,10)
00370 112* IADR(I,1)=1
00371 113* DO 11 J=1,24
00374 114* NC(I,J)=0
00375 115* 11 CONTINUE
00377 116* 10 CONTINUE
00401 117* DO 18 I=1,4
00404 118* NADR(I,1)=1
00405 119* 13 CONTINUE
00405 120* C
00405 121* C READ IN FIRST 28 CASE PARAMETERS
00407 122* NCASIN=0
00410 123* 8 READ(7) (ICASE(I),I=1,29)
00416 124* NCASIN=NCASIN+1
00417 125* IWRITE=1
00420 126* IF (ICASE(1),EQ,99) GO TO 35
00422 127* NEXTRA=(ICASE(J5)-1)*3
00423 128* IF (ICASE(15),EQ,0) NEXTRA=0
00425 129* NUPPER=29+NEXTRA-1
00426 130* IF (ICASE(15),GE,35) WRITE(6,9000) NCASIN,NUPPER,ICASE(1),ICASE(15)
00435 131* 9000 FORMAT(4I10)
00436 132* IF (NEXTRA,EQ,0) GO TO 9
00436 133* C *****
00436 134* C CASE CONTAINS MORE THAN 28 WORDS READ IN REMAINDER
00436 135* C *****

```

```
00440      READ (7) (ICASE(I),I=2,NUPPER)
00446      IF (ISWIT(1).EQ.1) GO TO 60
00450      CALL SELECT
00451      60) IWRITE=1
00451      C *****
00451      C SUBROUTINE BOX DETERMINES BOX AND HOUR FOR OCCURANCE TIME OF CASE
00451      C *****
00452      50) CALL BOX(IBOX,NT)
00452      C *****
00452      C INCREMENT CASE COUNTER FOR THE BOX AND HOUR
00452      C *****
00453      NC(IBOX,NT)=NC(IBOX,NT)+1
00454      IUNIT=IBOX+9
00455      DO 62 I=1,IWRITE
00455      C *****
00455      C INCREMENT BOX CASE COUNTER
00455      C *****
00460      CBOX(IBOX)=CBOX(IBOX)+1.
00461      K=CHOK(IBOX)
00461      C *****
00461      C *****
00461      C DETERMINE STARTING DRUM ADDRESS FOR NEXT CASE IN THIS BOX
00461      C *****
00462      IADR(IBOX,K)=IADR(IBOX,K)+NUPPER
00462      C *****
00462      C WRITE CASE ONTO DRUM IN ITS APPROPRIATE BOX
00462      C *****
00463      CALL NTRAN(IUNIT,1,NUPPER,ICASE(1),L)
00464      IF (ICASE(15).LE.0) WRITE(6,9004) (ICASE(M),M=1,NUPPER)
00473      9004 FORMAT(/, 'BAD CASE -- 0 NEEDS ',/(10I10))
00474      20 IF (L+1) 25,20,
00477      62 CONTINUE
00501      GO TO 3
00502      25 WRITE(6,507)
00504      507 FORMAT(2X,'ERROR IN NTRAN TRANSMISSION')
00505      WRITE(6,555) (CBOX(I),I=1,8)
00513      WRITE(6,560) IUNIT,L,IBOX,NT
00521      560 FORMAT(1X,4I6)
00521      STOP
00523      85 IF (ISWIT(1).EQ.3) CALL ADD
00523      C *****
00523      C REWIND ALL DRUM UNITS
00523      C *****
00525      DO 73 I=1,8
00530      IUNIT=I+9
00531      CALL NTRAN(IUNIT,10)
00532      73 CONTINUE
00532      C *****
00532      C COMPUTE HISTORICAL LAMBDAS FOR ALL BOXES AND TIMES
00532      C *****
00532      C *****
00534      WRITE(6,615)
00536      615 FORMAT(/, 20X,'HISTORICAL MEAN HOURLY ARRIVAL RATES',/)
00537      DO 86 I=1,8
00542      DO 87 J=1,24
00545      LAMBDAS(I,J)=NC(I,J)/DAYS(I)
00546      87 CONTINUE
00550      WRITE(6,550) (LAM3DAS(I,J),J=1,24)
00556      86 CONTINUE
```

*WS 2/71
*WS 2/71

10

```

194* 000560
195* IF (ISWIT(3).EQ.1) READ(5,551) ((LAMBDA(I,J),J=1,24),I=1,8)
196* IF(ISWIT(3).EQ.1) WRITE(6,615)
197*
198* 651 FORMAT(12F5.3)
199* IF(ISWIT(2).EQ.0)GOTo15
200*
201* C *****
202* C HISTORICAL CASE LOAD IS TO BE INCREASED BY PER PERCENT
203* C THIS IS EQUIVALENT TO MULTIPLYING THE LAMBDA'S BY (1+PER)
204* DO 16 I=1,8
205* C *****
206* DO 17 J=1,24
207* LAMBDA(I,J)=LAMBDA(I,J)*(1.+PER)
208*
209* 17 CONTINUE
210* 16 CONTINUE
211* DO 300 I=1,8
212* WRITE(6,550) (LAMBDA(I,J),J=1,24)
213*
214* 300 CONTINUE
215* DO 1000 ISCN=1,NSCN
216* WRITE(6,9005)ISCN
217*
218* 9005 FORMAT('1',///25X,'THE FOLLOWING IS VERSION',I3,' OF THIS SCENARIO.
219* 1',////)
220* DO321IBOX=1,8
221* ZNDAYS(IBOX)=0
222* DO320JTIME=1,NP
223* NEWCAS(IBOX,JTIME)=0
224*
225* 320 CONTINUE
226* DO330I=1,NV
227* TOTAL(I)=0
228* TEMP(I)=0
229* DO329STATN=1,LSSTSN
230* VLDMTX(STATN,I)=0
231*
232* 329 CONTINUE
233* IF(ISWIT(3).EQ.1) GO TO 80
234*
235* 550 FORMAT(1X,24F5.2)
236* WRITE (6,620)
237*
238* 620 FORMAT(//1X,'NUMBER OF CASES IN EACH BOX',//)
239* WRITE(6,555) (CBOX(I),I=1,8)
240*
241* 555 FORMAT(1X,8F7.0)
242* DO 72 I=1,8
243* IUNIT=I+9
244* CALL INTRAN(IUNIT,10)
245*
246* 72 CONTINUE
247* IDNO=3
248* IDAY=0
249* IHR=0
250* WRITE(6,580) IDNO,IDAY,IHR,IMIN
251*
252* 580 FORMAT(13,I4,I3,I2)
253* IDNO=1
254* ITIME=ISCSH
255* ITIME=ISCSH
256* IDAY=ISCSJ
257* INUMB=0
258*
259* 83 INUMB=INUM4,3+1
260* I1=2
261* DO 113 J=1,NSPP
262* IF(IDAY,6E,NSPPS(J).AND.IDAY.LE.NSPPE(J)) I1=1

```

```

00740 252* 113 CONTINUE
00750 253* 7) I2=1
00751 254* K=MOD(IJAY,7)
00752 255* IF(A.EQ.0.OR.A.EQ.6) I2=2
00754 256* DO 84 J=1,MODLS
00757 257* IF(IJAY.E.,IHL(J)) GO TO 81
00761 258* 84 CONTINUE
00763 259* GO TO 83
00764 260*
00765 261* 81 I2=2
00766 262* 83 I3=1
00770 263* IF(ITIME.LT.0600.OR.ITIME.GE.2000) I3=2
00771 264* IBOX=ILIST(I1,I2,I3)
00773 265* IF(I5WIT(3).EQ.1) GO TO 29
00774 266* IT1=CB0X(IBOX)
00776 267* IF(IT1.EQ.0) DEL=60.
00777 268* IF(IT1.EQ.0) GO TO 75
00780 269* IUNIT=IBOX+9
00781 270* CALL NTRAN(IUNIT,10)
00782 271* JTIME=(ITIME/100)+1
00783 272* NEWCAS(IBOX,JTIME)=NEWCAS(IBOX,JTIME)+1
00784 273* XLAMB=LAMBDA(IBOX,JTIME)
00785 274* RN=RAND(0,8)
00786 275* IF(I5WIT(3).EQ.1) GO TO 116
00787 276* INUM=CB0X(IBOX)*RN+1
00788 277* NLOCK=IADR(IBOX,INUM+1)-IADR(IBOX,INUM)
00789 278* NSPACE=IADR(IBOX,INUM)-1
00790 279* CALL NTRAN(IUNIT,6,NSPACE)
00791 280* CALL NTRAN(IUNIT,2,NBLOCK,ICASE(1),L)
00792 281* 21 IF(L+1) 25,21,
00793 282* DO 94 I=1,3
00794 283* NCASE(I)=ICASE(I)
00795 284* 94 CONTINUE
00796 285* NCASE(4)=ICASE(6)
00797 286* DO 96 I=5,15
00798 287* NCASE(I)=ICASE(I+3)
00799 288* 96 CONTINUE
00800 289* DO 97 I=16,21
00801 290* NCASE(I)=ICASE(I+4)
00802 291* 97 CONTINUE
00803 292* NCASE(22)=IBOX
00804 293* NCASE(23)=ICASE(7)
00805 294* IHL=ITIME/100
00806 295* IMIN=ITIME-(IHL*100)
00807 296* WRITE(8,535) IDAY,IDAY,IHL,IMIN,NCASE(I),I=1,12)
00808 297* 585 FORMAT(I3,I4,I3,I2,I3,I1I5)
00809 298* WRITE(8,590) (NCASE(I),I=13,23)
00810 299* IF(ICASE(15).EQ.0) GO TO 116
00811 300* IL=((NBLOCK-28)/3)+1
00812 301* TTOS=0
00813 302* DO 93 I=1,IL
00814 303* J=26*(I-1)*3
00815 304* NEED(I)=ICASE(J)
00816 305* TOS(I)=(FLOAT(ICASE(J+1)))/240.
00817 306* TTOS=TTOS+ICASE(J+1)
00818 307* DELTA(I)=FLOAT(ICASE(J+2))/100.
00819 308* 93 CONTINUE
00820 309* KL=IL
00821 310* IF(ICASE(17).GT.0.AND.IL.GT.1) KL=IL-1

```

*WS 2/71

*WS 2/71


```

01111 3104
01121 311*
01122 312*
01130 313*
01131 314*
01132 315*
01133 316*
01135 317*
01136 318*
01137 319*
01140 320*
01142 321*
01144 322*
01146 323*
01147 324*
01150 325*
01151 326*
01152 327*
01153 328*
01154 329*
01156 330*
01157 331*
01161 332*
01162 333*
01163 334*
01164 335*
01166 336*
01167 337*
01170 338*
01171 339*
01172 340*
01173 341*
01174 342*
01175 343*
01177 344*
01200 345*
01201 346*
01203 347*
01204 348*
01205 349*
01206 350*
01207 351*
01211 352*
01212 353*
01213 354*
01214 355*
01215 356*
01216 357*
01221 358*
01222 359*
01224 360*
01226 361*
01227 362*
01230 363*
01231 364*
01233 365*
01235 366*
01237 367*

      WRITE(8,995) (NED(I),TOS(I),DELTA(I),I=1,NL)
59) FORALL(8(1),F8(4),F4(2))
      IF(FL-LF,IL) WRITE(996) NED(1L),TOS(1L),DELTA(1L)
59) FORALL(101,110)
110) NCAS=JCA5+1
      STATN=ICASE(2)
      IF(STATN,0) STATN=75
      LSTATN=MAX(LSTATN,STATN)
      VLDMTX(STATN,1)=VLDMTX(STATN,1)+1
      VLDMTX(STATN,2)=VLDMTX(STATN,2)+ICASE(15)
      IF(ICASE(20),GT,0) VLDMTX(STATN,3)=VLDMTX(STATN,3)+1
      IF(ICASE(21),GT,0) VLDMTX(STATN,4)=VLDMTX(STATN,4)+1
      IF(ICASE(17),GT,0) VLDMTX(STATN,5)=VLDMTX(STATN,5)+1
      VLDMTX(STATN,6)=VLDMTX(STATN,6)+ITOS
      VLDMTX(STATN,7)=VLDMTX(STATN,7)+ICASE(22)
      NPRIOR=ICASE(14)+7
      VLDMTX(STATN,NPRIOR)=VLDMTX(STATN,NPRIOR)+1
      CAS2DA(STATN)=CAS2DA(STATN)+1
      NDAY=IDAY
      IF(ABS(XLAMR).LE,1.E-5) GOTOT6
      DEL=(LOG(RR)/-(XLAMB3))*60.
      IF(ISWIT(8),EQ,1) GO TO 115
      IDELTA=DEL
      DELTE=(3./XLAMB3)*60.
      IDELTE=DELTE
      IF(IDELTA,GT,IDELTE) DEL=DELTE
75) IDELTA=DEL
      ZMIN=(TIME/100)*60+AMOD(TIME,100.)
      TMIN=ZMIN+DEL
      TIME=(IFX(TAIN)/60)*100+AMOD(TMIN,60.)
      ITIME=TIME
      GO TO 116
76) DEL=61.
      IF(ISWIT(6),EQ,1) GO TO 115
      GO TO 75
115) IDELTA=DEL
      IF(IDELTA,LT,60) GO TO 75
      DEL=60.
      IDELTA=DEL
      CALL ADTIME
118) CALL NTRAP(IJUIT,10)
      IF(ITIME,GT,2400) GO TO 83
      GO TO 82
83) JDAY=ITIME/2400
      TIME=AMOD(TIME,2400.)
      ITIME=TIME
      IDAY=IDAY+JDAY
      DO225 STATN=1,N5
      NC2D=0
222) IF(NC2D,EQ,CAS2DA(STATN)) GOTOT224
      IF(NC2D,EQ,6) GOTOT224
      NC2D=NC2D+1
      GOTOT22
224) CAS2DA(STATN)=0
225) VLDMTX(STATN,NC2D+1)=VLDMTX(STATN,NC2D+13)+1
82) IF(ISWIT(10),EQ,1) GO TO 91
      IF(IDAY,GT,ISCED) GO TO 90
      IF(IDAY,EQ,ISCED.AND,ITIME,GT,ISCEN) GO TO 92

```

```

      *WS 2/71
      *WS 2/71
      *WS 2/71
      *WS 2/71
      *WS 2/71
      *WS 2/71
      *WS 2/71
      *WS 2/71
      *WS 2/71
      *WS 2/71
      *WS 2/71
      *WS 2/71

```

```

01241 369*      GO TO 34
01242 369*      92 IDAY=IDAY+1
01243 370*      91 IF(NCAS .LT. MAXCA) GO TO 48
01244 371*      90 IF(ISWIT(3).EQ.1) GO TO 28
01247 372*      IDJ=2
01250 373*      I HOUR=0
01251 374*      IMIN=10
01252 375*      WRITE(8,550) IDNO,IDAY,I HOUR,IMIN
01260 376*      WRITE(6,580) IDNO,IDAY,I HOUR,IMIN
01266 377*      CALL INTRA(8,9)
01267 378*      28 I=-1
01270 379*      105 I=I+1
01271 380*      IF(I.GT.NDAY) GO TO 111
01273 381*      DO 112 J=1,NSPP
01276 382*      IF(I.GE.NSPPS(J).AND.I.LE.NSPPE(J)) GO TO 102
01300 383*      112 CONTINUE
01302 384*      J=5
01303 385*      GO TO 103
01304 386*      102 J=1
01305 387*      103 K=MOD(I,7)
01306 388*      IF(K.EQ.0.OR.K.EQ.6) J=J+2
01310 389*      IF(K.EQ.0.OR.K.EQ.6) GO TO 104
01312 390*      DO 105 K=1,NOHOLS
01315 391*      IF(I.EQ.IHOL(K)) GO TO 107
01317 392*      105 CONTINUE
01321 393*      GO TO 104
01322 394*      107 J=J+2
01323 395*      104 ZNDAYS(J)=ZNDAYS(J)+1
01324 396*      GO TO 106
01325 397*      111 DO 108 J=1,7,2
01330 398*      ZNDAYS(J+1)=ZNDAYS(J)
01331 399*      103 CONTINUE
01333 400*      WRITE(6,625)
01335 401*      625 FORMAT(//1X,'NUMBER OF SCENARIO DAYS IN EACH OF THE 8 BOXES')
01336 402*      WRITE(6,610) (ZNDAYS(J),J=1,8)
01344 403*      WRITE(6,630)
01346 404*      630 FORMAT(//1X,'HOURLY MEAN ARRIVAL RATE OF GENERATED CASES')
01347 405*      DO 109 I=1,8
01352 406*      DO 110 J=1,24
01355 407*      IF(ZNDAYS(I).LE.1.E-3)GOTO114
01357 408*      XNLAMB(I,J)=NEWCAS(I,J)/ZNDAYS(I)
01360 409*      GOTO110
01361 410*      114 XNLAMB(I,J)=0.0
01362 411*      110 CONTINUE
01364 412*      WRITE(6,550) (XNLAMB(I,J),J=1,24)
01372 413*      109 CONTINUE
01374 414*      552 FORMAT(1X,I6)
01375 415*      WRITE(6,640)
01377 416*      640 FORMAT(//1X,'NUMBER OF CASES IN EACH BOX BY HOUR')
01400 417*      WRITE(6,553) ((NEWCAS(I,J),J=1,24),I=1,8)
01411 418*      553 FORMAT(1X,24I4)
01412 419*      IF(ISWIT(9).EQ.0) GO TO 101
01414 420*      IF(I$CEN.EQ.1)GOTO27
01416 421*      CALLNTRAN(8,8,-2)
01417 422*      CALLNTRAN(8,8,1)
01420 423*      GOTO100
01421 424*      27 CALLNTRAN(8,10)
01422 425*      100 READ(8,530) IDNO,IDAY,I HOUR,IMIN

```

*WS 3/71

*WS 3/71
*WS 3/71*WS 4/71
*WS 3/71
*WS 4/71

*WS 2/71

*WS 2/71
*WS 2/71*WS 3/71
*WS 3/71
*WS 3/71
*WS 3/71


```

01071 484* DO250KELIV
01074 485* 250 TEMP(K)=0
01076 486* 200 CONTINUE
01700 437* 1000 WRITE(6,9003)TOTAL
01707 488* CALLEAIT
01710 469* END

```

```

*WS 2/71
*WS 2/71
*WS 2/71
*WS 2/71

```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 0 *DIAGNOSTIC* MESSAGE(S)

```

PHASE 1 TIME = 2 SEC.
PHASE 2 TIME = 0 SEC.
PHASE 3 TIME = 1 SEC.
PHASE 4 TIME = 0 SEC.
PHASE 5 TIME = 1 SEC.
PHASE 6 TIME = 2 SEC.

```

TOTAL COMPILATION TIME = 6 SEC

311 FOR BOX,BOX
UNIVAC 1108 FORTRAN V LEVEL 2206 0018 F5014P
THIS COMPILATION WAS DONE ON 09 JUN 71 AT 1:08:29

SUBROUTINE BOX ENTRY POINT 000153

STORAGE USED (BLOCK, NAME, LENGTH)

0001 *CODE 000176
0000 *DATA 000024
0002 *BLANK 116563
0003 BIX 002733

EXTERNAL REFERENCES (BLOCK, NAME)

0004 NW0J\$
0005 NI02\$
0006 NERR3\$

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001	000035	1176	0001	000070	2L	0001	000101	3L	0000	000007	4F	0003	I	000000	CAL
0002	R	116161	CBOX	0002	R	116131	CPER	0000	I	000005	I	0002	I	116171	ICASE
0000	I	000002	ID	0003	I	002725	IEWPP	0000	I	000001	IM	0003	I	002720	ISWPP
0000	I	000004	IWK	0000	I	000000	IY	0000	I	000006	J	0002	I	106241	NADR
0002	I	000000	NODRUM	0003	I	002732	QPP								NCASES

00101	1*	SUBROUTINE BOX(I BOX,NT)													
00103	2*	PARAMETER NP=24,NN=45,MXBX=4500,NXBY=1000,NH=20,NCP=250,NS=75													*WS 2/71
00103	3*	1													*WS 2/71
00104	4*	INTEGER CAL(12,31,4)													
00105	5*	DIMENSION IADR(8,MXBX),IADR(4,NXBX),CPER(NP),CBOX(4),NCASES(NP)													
00105	6*	DIMENSION ICASE(NCP)													
00107	7*	COMMON/BIK/													
00110	8*	COMMON NODRUM,IADR,IADR,NCASES,CBOX,CBOX,ICASE													
00111	9*	IY=MOD(ICASE(4),10)-5													
00112	10*	IM=ICASE(4)/10													
00113	11*	ID=ICASE(5)/10000													
00114	12*	IT=MOD(ICASE(5),10000)													
00115	13*	IWK=CAL(IY,ID,IY)													
00116	14*	DO 1 I=1,QPP													
00121	15*	IF(IWK*GE,ISWPP(I),AND,IWK*LE,IEWPP(I)) GO TO 2													
00123	16*	1 CONTINUE													
00125	17*	J=5													
00126	18*	IF(MOD(IWK,2).NE.0) J=7													
00130	19*	GO TO 3													
00131	20*	2 J=1													
00132	21*	IF(MOD(IWK,2).NE.0) J=3													
00134	22*	3 IF(IT,LT,0800.OR,IT,GE,2000) J=J+1													
00136	23*	IT=(IT/100)+1													

```
00137 24*      1BOX=0
00140 25*      IF(1BOX.GT.0) WRITE(6,4) 1BOX,1Y,1X,1D,1I,1F,1M
00152 26*      + FORMAT(1X,7I5)
00153 27*      RETURN
00154 28*      END
```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 0 *DIAGNOSTIC* MESSAGE(S)

```
PHASE 1 TIME = 0 SEC.
PHASE 2 TIME = 0 SEC.
PHASE 3 TIME = 0 SEC.
PHASE 4 TIME = 0 SEC.
PHASE 5 TIME = 0 SEC.
PHASE 6 TIME = 0 SEC.
```

TOTAL COMPILATION TIME = 0 SEC

AIR FOR ADD:ALL
 UNIVAC 1100 FORTRAN V LEVEL 2005 0018 250103
 THIS COMPIATION WAS COMPLETED ON JUN 71 AT 11:00:40

SUBROUTINE ADD
ENTRY POINT 000156

STORAGE USED (BLOCK, NAME, LENGTH)

0001	*CODE	000104
0000	*DATA	000042
0002	*BLANK	110563

EXTERNAL REFERENCES (BLOK, 1941)

0003	NTR4N
0004	RAN0J
0005	BOX
0006	NER33\$

 STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME) |

0001	000004	107G	0001	000034	117G	0000	R	000007	B	0002	I	000001	J	0000	I	000001	K	0000	I	000017	K	0000	I	000001	LAUNIT	0002	R	116161	CROX	0000	I	000013	LA	0000	I	000013	CPFR
0000	R	000002	CPERC	0000	I	000000	I	0002	I	000001	JA	0000	I	000005	J	0000	I	000003	NCASE	0000	I	000004	NJM	0000	I	000020	NIPPER	0002	R	116101	NCASES	0000	I	000013	LA		
0000	I	116171	ICASE	0000	I	000016	IUNIT	0000	I	000005	J	0000	I	000010	NCASIO	0000	I	000003	NCASE	0002	I	000004	NJM	0000	I	000020	NIPPER	0002	R	116101	NCASES	0000	I	000013	LA		
0002	I	106241	NADR	0000	I	000011	IBLOCK	0000	I	000015	NT	0000	I	000015	NT	0000	I	000004	NJM	0002	I	000004	NJM	0000	I	000020	NIPPER	0002	R	116101	NCASES	0000	I	000013	LA		
0002	I	000000	NODRUM	0000	I	000012	USPACE	0000	I	000015	NT	0000	I	000015	NT	0000	I	000004	NJM	0002	I	000004	NJM	0000	I	000020	NIPPER	0002	R	116101	NCASES	0000	I	000013	LA		
0000	R	000000	NODR	0000	R	000006	RN	0000	R	000006	RN	0000	R	000006	RN	0000	R	000006	RN	0002	I	000004	NJM	0000	I	000020	NIPPER	0002	R	116101	NCASES	0000	I	000013	LA		

```

00101 1* SUBROUTINE ADD
00103 2* PARAMETER IP=24, NN=45, NX3X=4500, NXP=1000, NH=20, NCP=250, NS=75
00104 3* , NV=18
00104 4* DIMENSION IADR(8,4*BX), IADR(4,NXBX), CPER(NP), CROX(N), NCASES(NP)
00104 5* , ICASE(NCP)
00105 6* COMMON NORDUM, IADR, IADP, NCASES, CPER, CROX, ICASE
00106 7* DO 1 I=1, JORDUM
00110 8* IADP(I)=I+17
00112 9* CALL NTRAI(IADP(I),10)
00113 10* CPER=CPER(I)
00114 11* NCASE=ICASES(I)
00115 12* NUM=NCASE*CPERC
00116 13* DO 2 J=1, NUM
00121 14* RN=Rand(0,3)
00122 15* NCANO=RN*NCASE+1
00123 16* NBLOCK=IADR(IADP(I),ICANO+1)-IADR(IADP(I),NCANO)
00124 17* NSPACE=IADR(IADP(I),JCANO)-J
00125 18* CALL NTRAI(IADP(I),6,NSPACE)
00126 19* CALL NTRAI(IADP(I),2,NBLOCK,ICASE(1),L)
00127 20* CALL BOX(IBOX,NT)
00130 21* IADP(I)=IBOX+9
00131 22* CROX(IBOX)=CROX(IBOX)+1

```

```

00132 23*      KECROX(1:30X)
00133 24*      IADR(130X,K+1)=IADR(130X,K)+NBLOCK
00134 25*      CALL NFRAN(UNIT,1,'UPPER',ICASF(1),L)
00135 26*      CALL NFRAN(IADR(1),1)
00136 27*      2 CONTINUE
00140 28*      1 CONTINUE
00142 29*      RETURN
00143 30*      END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.      0 *DIAGNOSTIC* MESSAGE(S)

PHASE 1 TIME = 0 SEC.
PHASE 2 TIME = 0 SEC.
PHASE 3 TIME = 0 SEC.
PHASE 4 TIME = 0 SEC.
PHASE 5 TIME = 0 SEC.
PHASE 6 TIME = 0 SEC.

```

TOTAL COMPILATION TIME = 0 SEC

SUBROUTINE ADTIME ENTRY P IAT 00042

STORAGE USED (BLOCK, NAME, LENGTH)

0001 *CODE 000407
0000 *DATA 000043
0002 *BLANK 000000
0003 AIX 000367

EXTERNAL REFERENCES (BLOCK, NAME)

0004 RAND
0005 ALG5
0006 NERR35

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001	000007	1L	0001	000303	10L	0001	000121	131G	0001	000165	144G	0001	000107	2L
0001	000175	7L	0001	000177	AL	0001	000314	9L	0003	R	000360	B	0003	R
0000	I	000014	I3OX	0003	I	000364	IDAY	0003	I	000021	IHOL	0003	I	000045
0000	I	000002	ISWIT	0003	I	000356	ISCEH	0003	I	000000	ISWIT	0003	I	000045
0000	I	000007	11	0000	I	000011	I2	0000	I	000010	J	0000	I	000006
0000	I	000015	JTIME	0000	I	000012	K	0000	I	000000	KDAY	0000	I	000003
0003	I	000363	I3HOLS	0003	I	000362	NSPP	0003	I	000015	NSPPE	0004	R	000000
0000	R	000016	KI	0003	R	000365	TIME	0000	R	000005	TMIN	0000	R	000004
0000	R	000001	ZTIME											

00101	1*	SUBROUTINE ADTIME												
00103	2*	PARAMETER IP=24, JN=45, NMAX=4500, NABX=1000, NH=20, NCP=250, NS=75												
00103	3*	J												
00104	4*	REAL LAMBDA(6,NSP)												
00105	5*	DIMENSION ISWIT(10),NSPPS(3),NSPPE(3),THOL(NH),ILIST(2,2,2),B(2)												
00106	6*	COMMON/AXI/ ISWIT,NSPPS,NSPPE,DELTA,IHOL,ILIST,LAMBDA,ITIME,												
00106	7*	ISCEH,ISCEH2,NSPP,NOHOLS,IDAY,TIME,DEL												
00107	8*	KDAY=IDAY												
00110	9*	ZTIME=TIME												
00111	10*	ISWIT=ITIME												
00112	11*	N=0												
00113	12*	I=JN+1												
00114	13*	ZMIN=(ITIME/100)*60+AMOD(TIME,100.)												
00115	14*	TMIN=ZMIN+DEL												
00116	15*	TIME=(IFIX(TMIN)/60)*100+AMOD(TMIN,60.)												
00117	16*	ITIME=TIME												
00120	17*	IF(ITIME.LE.2400) GO TO 2												
00122	18*	JDAY=TIME/2400												
00123	19*	TIME=AMOD(TIME,2400.)												

*WS 2/71
*WS 2/71

```

00124 20* ITIME=TIME
00125 KDAY=KDAY+JDAY
00126 ? I1=2
00127 I1=2
00128 I1=2
00130 DO 5 J=1,45PP
00133 IF(KDAY.GE.NSPPS(J).AND.<KDAY.LE.NSPPE(J)) I1=1
00135 5 CONTINUE
00137 I2=1
00140 K=MOD(KDAY,7)
00141 IF(K.EQ.0.OR.K.EQ.6) I2=2
00143 DO 6 J=1,NOHOLS
00146 31* IF(KDAY.EQ.IHOL(J)) GO TO 7
00150 6 CONTINUE
00152 GO TO 8
00153 7 I2=2
00154 8 I3=1
00155 IF(ITIME.LT.0800.OR.ITIME.GE.2000) I3=2
00157 IBOX=LIST(I1,I2,I3)
00160 JTIME=(ITIME/100)+1
00161 RN=Rand(0,1)
00162 40* XLAMB=LAMBDA(IBOX,JTIME)
00163 *DIAGNOSTIC* THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00165 41* IF(XLAMB.EQ.0.) GO TO 10
00166 42* DEL=(LOG(RN)/-XLAMB)*60.
00167 IDelta=DEL
00168 43* IF(IDelta.LT.60) GO TO 9
00169 44* DEL=60.
00171 45* IDelta=DEL
00172 46* GO TO 1
00173 47* IDelta=DEL
00174 48* GO TO 1
00175 49* IDelta=DEL
00176 50* GO TO 1
00177 51* 9 ZTIME=(IOLDTM/100)*60+AMOD(ZTIME,100.)
00200 TMIN=ZTIME+N*60+DEL
00201 52* TIME=(IFIX(TMIN)/60)*100+AMOD(TMIN,60.)
00202 53* ITIME=TIME
00203 54* RETURN
00204 55* END
00204 56*

```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 1 *DIAGNOSTIC* MESSAGE(S)

```

PHASE 1 TIME = 0 SEC.
PHASE 2 TIME = 0 SEC.
PHASE 3 TIME = 0 SEC.
PHASE 4 TIME = 0 SEC.
PHASE 5 TIME = 1 SEC.
PHASE 6 TIME = 0 SEC.

```

TOTAL COMPILATION TIME = 1 SEC

BIT FOR SELECT, SELECT

UNIVAC 1108 FORTRAN V LEVEL 2206 0018 F5018P

THIS COMPILATION WAS DONE ON 06 JUN 71 AT 10:08:43

18:08:33.120

09 JUN 71

SUBROUTINE SELECT ENTRY POINT 000073

STORAGE USED (BLOCK, NAME, LENGTH)

0001 *CODE 000076
 0000 *DATA 000026
 0002 *BLANK 110563
 0003 SEAL 000001

EXTERNAL REFERENCES (BLOCK, NAME)

0004 NTRAN
 0005 NWDJ\$
 0006 NI02\$
 0007 NSI0P\$
 0010 NERR3\$

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001	000051	1L	0001	000045	20L	0001	000054	25L	0000	000004	507F	0002	R	116161	CROX
0002	R	116131	CPER	0002	I	000001	IADR	0000	I	000001	IAUNIT	0002	I	116171	ICASE
0000	I	000002	K	0000	I	000003	L	0002	I	106241	NADR	0002	I	116101	NCASES
0003	I	000000	NUPPER									0002	I	000000	NODRUM

```

00101 1* SUBROUTINE SELECT
00103 2* PARAMETER NP=24,NN=45,MXBX=4500,NBXX=1000,NH=20,NCP=250,NS=75
00103 3* 1,NV=18 *WS 2/71
00104 4* COMMON/SEAL/ NUPPER *WS 2/71
00105 5* DIMENSION IADR(8,MXBX),NADR(4,NBXX),CPER(NP),CBOX(A),NCASES(NP)
00105 6* 1,ICASE(NCP)
00106 7* COMMON NODRUM,IADR,NADR,NCASES,CPER,CBOX,ICASE
00106 8* C
00106 9* C THESE ARE USER FURNISHED STATEMENTS
00106 10* C
00107 11* IDRUM=0
00110 12* IF(ICASE(14).GT.3) IDRUM=1
00112 13* IF(ICASE(17).GT.0) IDRUM=2
00114 14* IF(IDRUM.EQ.0) GO TO 1
00116 15* IAUNIT=17+IDRUM
00117 16* NCASES(IDRUM)=NCASES(IDRUM)+1
00120 17* KNCASES(IDRUM)+1
00121 18* NADR(IDRUM,K)=NADR(IDRUM,K-1)+NUPPER
00122 19* CALL NTRAN(IAUNIT,1,NUPPER,ICASE(1),L)
00123 20* 20 IF(L+1) 25,20
00126 21* 1 RETURN
00127 22* 25 WRITE(6,507)

```

00131 23* 507 FORMAT (2A,*,ERROR IN DATA TRANSMISSION IN SELECT,*)
 00132 24* STOP
 00133 25* END

-END OF UNIVAC 1108 FORTRAN V COMPILATION. 0 *DIAGNOSTIC* MESSAGE(S)

PHASE 1 TIME = 0 SEC.
 PHASE 2 TIME = 0 SEC.
 PHASE 3 TIME = 0 SEC.
 PHASE 4 TIME = 0 SEC.
 PHASE 5 TIME = 0 SEC.
 PHASE 6 TIME = 0 SEC.

TOTAL COMPILATION TIME = 0 SEC

HIST

Certain information pertinent to the specific simulation being carried out must be indicated in the first two source cards of HIST. In card 1, NMO is set equal to the number of months the simulation will be exercised. In card 2, IMO is set to the number of the month the simulation begins (January = 1, etc.), IYR is the last digit of the year it begins and ISTANO is the station number of the C-130 station which would respond to any cases involving C-130's in the simulation.

11 FOR 11:12:00.150
UNIVAC 1100 FORTRAN V LEVEL 2206 0015 F50100
THIS COMPILATION WAS DONE ON 07 JUL 71 AT 10:11:00

MAIN PROGRAM

STORAGE USED (BLOCK, NAME, LENGTH)

0001 *CODE 000773
0000 *DATA 001104
0002 *BLANK 000000
0003 COW/ 001757

EXTERNAL REFERENCES (BLOCK, NAME)

0004 INTRIN
0005 CONVER
0006 NEW IF
0007 NI023
0010 INR013
0011 NI013
0012 NST013

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

BLOCK	TYPE	RELATIVE LOCATION	NAME
0001	000237	1 L	0001
0001	000534	122L	0001
0001	000326	15L	0001
0001	000356	254G	0001
0001	000307	335G	0001
0001	000723	436G	0001
0001	000742	500L	0001
0000	001053	602F	0000
0001	000056	7L	0000
0000	001022	13AY	0000
0000	001025	14OW	0000
0000	001012	15POINT	0000
0000	000314	15WIT	0000
0003	001751	11	0003
0000	001006	J	0000
0000	001020	N BLOCK	0000
0003	001750	N UPPER	0003

00101 1* PARAMETER NM02=
00103 2* DIMENSION IM0(NM02)/8/,YR(11)/8/,ISTANO(11)/81/
00107 3* DIMENSION ICASE(500),NCASE(500),ISTART(100),IEND(100),IS,IT(100),
00107 4* INTIME(100),NCOUNT(100),JWO(12)
00110 5* COWON/COW/ ICASE,NCASE,NUPPER,I1,I2,I3,I4,KAP,IKAP
00111 6* DATA J40/31,28,31,30,31,30,31,31,30,31,30,31,30,31/
00113 7* KAPEU

```

00114 b*
00115 9*
00120 10*
00121 11*
00123 12*
00124 13*
00125 14*
00126 15*
00127 16*
00135 17*
00136 18*
00141 19*
00142 20*
00143 21*
00145 22*
00147 23*
00150 24*
00151 25*
00152 26*
00153 27*
00154 28*
00155 29*
00156 30*
00164 31*
00166 32*
00167 33*
00170 34*
00173 35*
00175 36*
00177 37*
00200 38*
00201 39*
00203 40*
00205 41*
00207 42*
00211 43*
00212 44*
00214 45*
00215 46*
00216 47*
00217 48*
00221 49*
00223 50*
00224 51*
00226 52*
00227 53*
00231 54*
00237 55*
00241 56*
00242 57*
00245 58*
00247 59*
00250 60*
00251 61*
00254 62*
00255 63*
00256 64*
00257 65*

IKAP=0
DO 1 I=1,100
  COUNT(I)=0
  ISWAT(I)=0
  I1=5
  I2=0
  I3=0
  I4=0
  WRITE(8,601) I,I2,I3,I4
  I2END=0
  DO 11 I=1,IMO
    J=IMG(I)
    I2END=I2END+J*(J)
    IF (IMO(I).EQ.2.AND.IYR(1).EQ.8) I2END=I2END+1
  11 CONTINUE
  I1=1
  CALL NTRAN(35,I1)
  IOPFAC=0
  IFLAG=0
  IDNO=1
  IPOINT=1
  IACCEPT=0
  READ (7) (ICASE(I),I=1,29)
  IF (ICASE(1).EQ.99) GO TO 100
  IMONTH=ICASE(4)/10
  IYEAR=ICASE(4)-IMONTH*10
  DO 43 I=1,NMO
    IF (IMONTH.EQ.IMO(I).AND.IYEAR.EQ.IYR(1)) GO TO 44
  43 CONTINUE
  IACCEPT=1
  GO TO 60
  44 IF (IFLAG.EQ.2) GO TO 60
  IF (IFLAG.EQ.1) GO TO 50
  IF (ICASE(3)/1000.NE.2) IFLAG=1
  45 IF (IOPFAC.EQ.ICASE(2).AND.IFLAG.EQ.0) GO TO 60
  IOPFAC=ICASE(2)
  IF (IDNO.NE.1) IEND(IDNO-1)=IPOINT-1
  ISTART(IDNO)=IPOINT
  IDNO=IDNO+1
  GO TO 60
  50 IF (ICASE(2).EQ.0.OR.ICASE(2).EQ.ISTANO(1)) IFLAG=2
  IF (IFLAG.EQ.2) GO TO 45
  60 NEXTRA=(ICASE(15)-1)*3
  IF (ICASE(15).EQ.0) NEXTRA=0
  NUPPER=29+NEXTRA-1
  IF (NEXTRA.EQ.0) GO TO 10
  READ(7) (ICASE(I),I=29,NUPPER)
  IF (IACCEPT.EQ.1) GO TO 2
  CALL NTRAN(35,I1,28,ICASE(1),ISTAT)
  3 IF (ISTAT+1) 500,3,
  IF (NEXTRA.EQ.0) GO TO 15
  NBLOCK=NUPPER-28
  CALL NTRAN(35,I1,NBLOCK,ICASE(29),ISTAT)
  4 IF (ISTAT+1) 500,4,
  15 IPOINT=IPOINT+NUPPER
  NCOUNT(IDNO-1)=NCOUNT(IDNO-1)+1
  GO TO 2
  100 IDNO=IDNO-1

```



```

00250 DO*
00251 07*
00271 08*
00277 09*
00300 70*
00301 71*
00302 72*
00310 73*
00311 74*
00314 75*
00316 76*
00317 77*
00320 78*
00321 79*
00324 80*
00325 81*
00326 82*
00327 83*
00330 84*
00331 85*
00332 86*
00334 87*
00337 88*
00340 89*
00342 90*
00344 91*
00352 92*
00353 93*
00354 94*
00355 95*
00356 96*
00360 97*
00361 98*
00364 99*
00366 100*
00370 101*
00371 102*
00372 103*
00374 104*
00400 105*
00401 106*
00402 107*
00403 108*
00404 109*
00407 110*
00410 111*
00412 112*
00413 113*
00415 114*
00416 115*
00417 116*
00422 117*
00423 118*
00424 119*
00425 120*
00433 121*
00435 122*
00440 123*

IE4(I, IDNO)=INPUT(I)-1
WRITE(6,605) 1, NO, (I, START(I), IEND(I), I=1, IDNO)
WRITE(6,606) (NCOUNT(I), I=1, IDNO)
605 FORMAT(1X,39I3)
606 FORMAT(14, (16I7))
110 CALL NTRAN(35,10)
72* WRITE(6,605) (I, SWIT(I), I=1, IDNO)
607 FORMAT(110,39I2)
DO 120 I=1, IDNO
IF (ISWIT(I).EQ.1) GO TO 120
K=ISTART(I)-1
CALL NTRAN(35,6,K)
CALL NTRAN(35,2,28, ICASE(1), ISTAT)
5 IF (ISTAT+1) 500,5,
IMONTH=ICASE(4)/10
IYEAR=ICASE(4)-IMONTH*10
IDAY=ICASE(5)/10000
ITIME=ICASE(5)-IDAY*10000
NTIME(I)=IYEAR*10**8+IMONTH*10**6+IDAY*10**4+ITIME
CALL NTRAN(35,10)
120 CONTINUE
DO 121 I=1, IDNO
JNO=I
IF (ISWIT(I).NE.1) GO TO 122
121 CONTINUE
609 WRITE(6,606) (ISWIT(I), I=1, IDNO)
610 FORMAT(110, 'ERROR IN ISWIT ARRAY',/(20I4))
STOP
122 ILOW=NTIME(JNO)
IS=JNO+1
IF (IS.GT.IDNO) IS=IDNO
INDEX=JNO
DO 125 I=IS, IDNO
IF (ISWIT(I).EQ.1) GO TO 125
IF (NTIME(I).GE.ILOW) GO TO 125
ILOW=NTIME(I)
INDEX=I
125 CONTINUE
WRITE(6,605) INDEX, ILOW
605 FORMAT(21I10)
K=ISTART(INDEX)-1
CALL NTRAN(35,6,K)
CALL NTRAN(35,2,28, ICASE(1), ISTAT)
9 IF (ISTAT+1) 500,6,
NEXTTRA=(ICASE(15)-1)*3
IF (ICASE(15).EQ.0) NEXTTRA=0
NUPPER=29*NEXTTRA-1
IF (NEXTTRA.EQ.0) GO TO 130
NBLOCK=NUPPER-28
CALL NTRAN(35,2, NBLOCK, ICASE(29), ISTAT)
7 IF (ISTAT+1) 500,7,
130 CALL CONVER
ISTART(INDEX)=ISTART(INDEX)+NUPPER
NCOUNT(INDEX)=NCOUNT(INDEX)-1
WRITE(6,606) (NCOUNT(I), I=1, IDNO)
IF (ISTART(INDEX).GE.IEND(INDEX)) ISWIT(INDEX)=1
DO 135 I=1, IDNO
IF (ISWIT(I).EQ.0) GO TO 110

```

```

00442 124* 13* CONTINUE
00443 125* 11=2
00444 126* 12=12*10
00445 127* 13=0
00446 128* 14=5
00447 129* WRITE(3,601) 11,12,13,14
00448 130* 601 FORMAT(13,I4,I7,I2)
00449 131* WRITE(6,701) 11,12,13,14
00450 132* 701 FORMAT(1H0,4I4)
00451 133* GO TO 140
00452 134* 500 WRITE(6,602)
00453 135* 602 FORMAT(110,'ERROR IN NTRAY TRANSMISSION')
00454 136* STOP
00455 137* 140 END

```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 0 *DIAGNOSTIC* MESSAGE(S)

```

PHASE 1 TIME = 0 SEC.
PHASE 2 TIME = 0 SEC.
PHASE 3 TIME = 1 SEC.
PHASE 4 TIME = 0 SEC.
PHASE 5 TIME = 0 SEC.
PHASE 6 TIME = 1 SEC.

```

TOTAL COMPILATION TIME = 2 SEC

SUBROUTINE CONVER ENTRY POINT 000340

STORAGE USED (BLOCK, NAME, LENGTH)

0001 *CODE 000351
 0000 *DATA 000542
 0002 *BLANK 000000
 0003 CONV 001757

EXTERNAL REFERENCES (BLOCK, NAME)

0004 NW0J\$
 0005 NI01\$
 0006 NI02\$
 0007 NERR3\$

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001	000002	106G	0001	000011	115G	0001	000016	123G	0001	000141	142G	0001	000153	171G
0001	000172	202G	0001	000250	217G	0001	000315	244G	0001	000300	5L	0000	000474	590F
0000	000463	505F	0000	000467	590F	0000	000471	595F	0001	000275	6L	0000	000506	600F
0001	000307	7L	0000	R	000310	DELTA	0000	I	000454	I	000000	ICASE	0003	I
0000	I	000460	1L	0000	I	000455	IPREV	0003	I	001751	I1	0003	I	000456
0003	I	001753	I3	0003	I	001754	I4	0000	I	000457	I6	0003	I	001755
0000	I	000462	KL	0003	I	000764	NCASE	0000	I	000000	NEED	0003	I	000144
														TOS

00101 1* SUBROUTINE CONVER
 00103 2* COMMON/CONV/ ICASE,NCASE,NUPPER,I1,I2,I3,I4,KAP,IKAP
 00104 3* DIMENSION ICASE(500),NCASE(500),NEED(100),TOS(100),DELTA(100)
 00105 4* DO 1 I=1,3
 00110 5* NCASE(I)=ICASE(I)
 00111 6* 1 CONTINUE
 00113 7* NCASE(4)=ICASE(6)
 00114 8* DO 2 I=5,15
 00117 9* NCASE(I)=ICASE(I+3)
 00120 10* 2 CONTINUE
 00122 11* DO 3 I=16,21
 00125 12* NCASE(I)=ICASE(I+4)
 00126 13* 3 CONTINUE
 00130 14* NCASE(22)=0
 00131 15* NCASE(23)=ICASE(7)
 00132 16* I2=ICASE(5)/1000
 00133 17* IF(I2.LT.1.OR.I2.GT.31) GO TO 5
 00135 18* IF(ICASE(4).EQ.IPREV.AND.I2.LT.I2PREV) GO TO 7
 00137 19* IF(I2.LT.KAP) IKAP=IKAP+KAP
 00141 20* KAP=I2

```

00142 21* I2=I2+IKAP-1
00143 22* I2PREV=I2
00144 23* IPREV=ICASE(4)
00145 24* I=ICASE(5)-(ICASE(5)/10000)*10000
00146 25* I3=I0/100
00147 26* I4=I0-I3*100
00150 27* IF(I3.GT.23) 5 TO 5
00152 28* IF(I4.GT.59) 60 TO 5
00154 29* WRITE(8,505) I1,I2,I3,I4,(ICASE(I),I=1,12)
00156 30* FORMAT(13,14,13,12,13,11,5)
00157 31* WRITE(8,590) (ICASE(I),I=13,23)
00175 32* FORMAT(10,15,110)
00176 33* IF(ICASE(15).EQ.0) GO TO 6
00200 34* IL=((NUPPER-28)/3)+1
00201 35* DO 4 I=1,IL
00204 36* J=26+(I-1)*3
00205 37* NEED(I)=ICASE(J)
00206 38* TOS(I)=(FLOAT(ICASE(J+1)))/240.
00207 39* DELTA(I)=FLOAT(ICASE(J+2))/100.
00210 40* CONTINUE
00212 41* KL=IL
00213 42* IF(ICASE(17).GT.0.AND.IL.GT.1) KL=IL-1
00215 43* WRITE(8,595) (NEED(I),TOS(I),DELTA(I),I=1,KL)
00225 44* FORMAT(6,12,F0.4,F4.2))
00226 45* IF(KL.LT.IL) WRITE(8,595) NEED(IL),TOS(IL),DELTA(IL)
00234 46* RETURN
00235 47* WRITE(6,560) ICASE(5)
00240 48* FORMAT(110,'ERROR IN DATA CASE REJECTED-DAY AND TIME =',I8)
00241 49* GO TO 6
00242 50* WRITE(6,600) (ICASE(I),I=1,28)
00250 51* FORMAT(1X,14I9/)
00251 52* GO TO 6
00252 53* END

```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 0 *DIAGNOSTIC* MESSAGE(S)

```

PHASE 1 TIME = 0 SEC.
PHASE 2 TIME = 0 SEC.
PHASE 3 TIME = 0 SEC.
PHASE 4 TIME = 0 SEC.
PHASE 5 TIME = 1 SEC.
PHASE 6 TIME = 0 SEC.

```

TOTAL COMPILATION TIME = 1 SEC

	+ T CASE #8A2	T ITOL	012/4	1	1 RES	E /	RQUED *
000001		T MM	021/4	1	2COST	1 / 2 F	EXCSO *
000002		T NN	022/4	1	3E1AT	1 / 2 F	L1STQ *
000003		T XC	031/2*	F	4TVEC	1 / 2 F	NSET1 *
000004		T YC	2R1/2*	F	5IFLT	1 / 2 F	SRHS1 *
000005		T POB	041/3	1	61B	1 / 4 1	CCUE
000006		T UTYPE	042/3	1	7PRIOR	1 / 4 1	CSET
000007		T L	043/3	1	8NCASE	1 / 2 1	TSET
000008		T SIS	051/4*	1	9PR	1 / 2 F	PSET
000009		T S2S	052/4*	1	10UTIL	1 / 2 F	
000010		T STATN	053/4	1	11TUTL	1 / 2 F	
000011		T AIR	054/4	1	12SRQUE	1 / 2 1	
000012		T OST	061/2	F	13PTSET	1 / 2 1	
000013		T RESA	063/4	1	14STSET	1 / 2 1	
000014		T SWELL	064/4	1	15PCSET	1 / 2 1	
000015		T NEED	071/4	1	16SCSET	1 / 2 1	
000016		T PRI	072/4	1	17PPSET	1 / 2 1	
000017		T FLG	073/4	1	18SPSET	1 / 2 1	
000018		T SIGNL	074/4	1	19XR	1 / 2*	
000019		T PCQUE	081/2	1	20YR	1 / 2*	
000020		T SCQUE	082/2	1	21STN	1 / 4 1	
000021		T FPR	111/4	1	22TYPE	1 / 4 1	
000022		T REA	112/4	1	127DVEC	1 / 2 F	
000023		T IDLOC	113/4	1		/	
000024		T WIND	114/4	1		/	
000025		T NOINT	121/4	1		/	
000026		T LOC	122/4	1		/	
000027		T COUNT	123/4	1	23RST	E /	
000028		T WAIT	124/4	1	24AUT	1 / 2 F	
000029		T VIS	131/4	1	25END	1 / 2 F*	
000030		T I52	132/4*	1	26COSTD	1 2 F*	
000031		T OCCUR	27	F	27RCOST	1 / 4 1*	
000032		T NOCAS	141/2	1	28SOA1	1 / 2 F*	
000033		T OFSHR	142/2	F	29SOA2	1 / 2 F*	
000034		T FNSET	151/2	1	30SOA3	1 / 2 F*	
000035		T LNSET	152/2	1	31SLIM	1 / 4 1*	
000036		T FSRHS	161/2	1	32TF	1 / 2 F*	
000037		T LSRHS	162/2	1	33MR	1 / 2 F*	
000038		T GAMMA	171/2	F	120VOP	1 4 1*	
000039		T SEXCS	172/2	1	1128SQTAG	1 / 4 1*	
000040		T XCX	181/2*	F	129DLAY	1 2 F*	
000041		T YCY	182/2*	F		/	
000042		T TING	25	F		/	
000043		T TOUE	132/2	F	34STA	E /	
000044		T TSM	221/2	F	35ACS	2 / 4 1*	
000045		T TOUE	1222/2	F	36ADJS	2 / 4 1*	
000046		T COSTC	231/2	F	37CUT	2 / 4 1*	
000047		T TAILT	232/2	F	3AXS	1 / 2*	
000048		T TINT	282/2	F	39YS	1 / 2*	
000049		T STING	24	F	40AVGTW	1 / 2 F	
000050		T TSVC	212/2	F	41CREW	1 / 4 1	
000051		T NOUE	211/4	1	42FALL	1 / 4 1	
000052		T CNRES	212/4	1	43FALL2	1 / 4 1	
000053		T VALUE	26	1	44FALL3	1 / 2 1	
000054							

000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097
000098
000099
000100
000101
000102
000103
000104
000105
000106
000107
000108
000109
000110
000111
000112

T IT04 311/3 I 45NCAS 1 /4 I
T BOX 312/3 I 46NEEDS 1 /2 I
T SYTAG011/4 I 47MINTR 1 /4 I
T OPFAC313/3 I 48ASTRY 1 /4 I
T DMERT32 F 49REST 2 /4 I*
N COMP 031/4 I 50SHIFT 2 /4 I*
N NUMR032/4 I 51UNPHO 1 /4 I
N CAS 041/2 I 52USE 1 /2 F
N SNSET042/2 I 54USHF 2 /2 F
N XHAND051/2*F13DMRT 1 2 F
N YHAND052/2*F132GRP 1 4 I
N KRES 064/4 I133VCTR 1 /2 F
N DELTA032/2 F134TATOL 1 /2*F
/ 53NEADS E /
/ 55AVUS 1 /2 F
/ 56TME 1 /2 F
/ 57TOTME 1 /2 F
N TS 031/2 F /
N SSRHS032/2 I /
N SM 041/2 F /
N ESAC 042/2 I 58TOLER E /
N SDAY 051/4 I 59TOL 1 /2 F*
N SFLAG052/4 I 60TOLS 1 /2 F*
N RSRC 052/2 I /
N SLIST0A1/2 I /
N SASG 064/4 I /
/ 61CPRL E /
/ 62CAP 1 / I*
T HCREW011/2 I /
T XDEST021/2*F /
T YDEST022/2*F 63MSK E /
T IDEV 031/2 I 64MASK 1 / I*
T ACASE032/2 I /
T DEP 06 F /
T F1TON042/2 I /
T XI 051/2*F 65PRTM E /
T YI 052/2*F 66PRYSM 2 /2 F*
T TOW 111/4 I /
T MFLG 112/4 I /
T TFLT 07 F /
T TARVL041/2 F 68PATRL E
T RLS OR F 69IPAT 1 4 I*
T ROS 12 F /
T TOWSP112/2 F 70AVUTO 0 / F
/ 71CL 0 / I*
72COSTO 0 / I*
73D1ST 0 / I*
74EPSLN 0 / F*
/ 75FCQUE 0 / I
/ 76FCSET 0 / I
/ 77FEXCS 0 / I
N LRES 0 1/2 I 78FPSET 0 / I
N LFLG 032/2 I 79FRQUE 0 / I
N LNATED4 / I 80FTSET 0 / I
N CASN003 / I 81HO 0 / F*
N RESN004 / I 82IDELT 0 / I*
N NCMPL03 / I 831FDAY 0 / I*
N RCMPL04 / I 84IMO 0 / I

* N NOTIF8
* N ARSCH4
* N ARVSN4
* N COMPL4

* T FLT 82

000113	+N FUEL 4	N MFUFLO3 /	T RSISA 0 /	I 1
000114	+	N KFUFLO4 /	T RAKKA 0 /	F
000115	+N HOF 4	/	R7LCQUE 0 /	I 1
000116	+	/	R8LCSET 0 /	I 1
000117	+N HOPEF4	N NHOP 03 /	I R9LEXCS 0 /	I 1
000118	+	N PHOM 04 /	T 90LPSET 0 /	I 1
000119	+N NUCRU4	/	91LRQUE 0 /	I 1
000120	+	/	92LTSET 0 /	I 1
000121	+N ONSCN4	/	93MEANW 0 /	F
000122	+	/	94NRRCO 0 /	I 1
000123	+N SNDRK4	N NSDRK03 /	I 95NBRC5 0 /	I 1
000124	+	N RSDRK04 /	I 96NBREA 0 /	I 1
000125	+N SRISF2	/	97NBREA 0 /	I 1
000126	+	/	98NBREC 0 /	I 1
000127	+N SSET 4	N ETON 03 /	I 99NWD 0 /	I 1
000128	+	N SER 04 /	I100NWE 0 /	I 1
000129	+N STNBY4	N STAT 03 /	I101PDC1 0 /	F*
000130	+	/	I02PDC2 0 /	F*
000131	+N XSET 2	/	I03PRON 0 /	F*
000132	+	/	I04PRUP 0 /	F*
000133	+N READY4	N NREAD031/2	I105PSHET 0 /	I 1
000134	+	N RREAD032/2	I106RAP 0 /	I*
000135	+	N FREAD04	I107RISE 0 /	F*
000136	+N DELAY4	/	I0ASET 0 /	F*
000137	+	/	I09SNEED 0 /	I 1
000138	+N CHEKN4	/	I10S2PRI 0 /	I*
000139	+	/	I11THOOK 0 /	F*
000140	+	/	I12TLAST 0 /	F
000141	+	/	I13TOSBY 0 /	I 1
000142	+	/	I14TOTIN 0 /	I 1
000143	+	/	I15TSPI 0 /	F*
000144	+	/	I16TSP2 0 /	F*
000145	+	/	I17TUMPR 0 /	I*
000146	+	/	I18XRK 0 /	F*
000147	+	/	I19FLIST 0 /	I 1
000148	+	/	I20LLIST 0 /	I 1
000149	+	/	I21LIMIT 0 /	I 1
000150	+	/	I22KOUNT 0 /	I 1
000151	+	/	I23SCNVT E	I*
000152	+	/	I24NSN 1 /4	I*
000153	+	/	I25STAPE 0 /	I*
000154	+	/	I26TPRI 0 /	I*
000155	+	/	I30TCHEK 0 /	F*
000156	+	/	I35MEAND 0 /	F
000157	+	/	I36MEANV 0 /	F
000158	+	/	I37MNTMT 0 /	F*
000159	+	/	I38GROUP E	I*
000160	+	/	I39AVDRT 1 2	F
000161	+	/	I40CS 1 2	I
000162	+	/	I41FL1 1 2	I
000163	+	/	I42FL2 1 2	I
000164	+	/	I43FL3 1 2	I
000165	+	/	I44INTRP 1 2	I
000166	+	/	I45NDS 1 2	I
000167	+	/	I46NONPR 1 2	I
000168	+	/	I47NDSB 1 2	I
000169	+	/	I48TMTAV 1 2	F
000170	+	/	I49TVAVG 1 2	F


```
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025

EVENTS
  3 EXOGENOUS
    OPSTM (1)
    ENDSIM (2)
    START (3)
  18 ENDOGENOUS
    ARSCH
    ARVSN
    CHEKN
    COMPL
    DELAY
    FUEL
    HOME
    HOMEF
    NOTF
    NOTIF
    NUCRU
    ONSCN
    READY
    SNDRK
    SRISE
    SSET
    STNBY
    XSET

END
```

```

000001      EXOGENOUS EVENT START
000002      LET TLIM = (10500.-10.*FLOAT(NRES))-19.*FLOAT(NSTA)-6.*FLOAT(NRST)
000003      * -6.*FLOAT(NGROUP)-5.*FLOAT(NDSTRO)/50.
000004      LET LIMIT = TLIM
000005      LET M = 0
000006      DO, FOR EACH STA I
000007      DO, FOR EACH RST J
000008      DO, FOR K=1) (RST(I,J)),
000009      LET M = M + 1
000010      LET STN(M) = I
000011      LET TYPE(M) = J
000012      LET XR(M) = XS(I)
000013      LET YR(M) = YS(I)
000014      LOOP
000015      LOOP
000016      LOOP
000017      IF (M)EQ(NRES), GO TO I=0
000018      WRITE ON TAPE 6, M,NRES
000019      FORMAT (' ERROR IN DATA  M = ',15,' NRES = ',15)
000020      STOP
000021      100 LET ISA = 6-IFDAY
000022      IF (ISA)LS(0), LET ISA = 6
000023      LET PSHT = 1
000024      IF IFDAY GE 6, LET PSHT = NWD + 1
000025      LET IMO = MOD(ISA+2,7)
000026      IF HOLID(1) EQ -1, GO TO 200
000027      DO, FOR EACH HLDY I
000028      IF (HOLID(1))EQ(0), LET PSHT = NWD + 1
000029      LOOP
000030      200 CREATE NUCRU
000031      CAUSE NUCRU AT TME(PSHT)
000032      LET TLAST = 0.
000033      CREATE XSET
000034      CAUSE XSET AT TIME + SET
000035      CREATE SRISE
000036      CAUSE SRISE AT TIME + RISE - XR
000037      RETURN
000038      END

```

```

000001      ENDogenous ELEMENT UCURU
000002      5 LET TOTIME(PSHIFT) = TOTIME(PSHIFT) + TIME-TLAST
000003      GO TO 20, FOR EACH PFS IRS
000004      IF EIAT(IRS) EQ 0, GO TO 10
000005      IF (TFLT(1FL/IRS))EQ(0,0), GO TO 10
000006      IF TFLT(1FLT/IRS) GR TLAST, GO TO 6
000007      LET TUTIL(IRS) = TUTIL(IRS) + TIME - TLAST
000008      GO TO 10
000009      6 LET TUTIL(IRS) = TUTIL(IRS) + TIME - TFLT(1FLT(IRS))
000010      10 LET UTIL(IRS) = UTIL(IRS) + TUTIL(IRS)
000011      LET USHFT(STN(IRS),PSHFT) = USHFT(STN(IRS),PSHFT) + TUTIL(IRS)
000012      LET CLCT = CLCT + TUTIL(IRS)
000013      LET TUTIL(IRS) = 0.
000014      20 LOOP
000015      LET JJ = 0
000016      IF PSHFT EQ NWD, LET JJ = 1
000017      IF PSHFT EQ RWD+RWF, LET JJ = 2
000018      IF JJ EQ 0, GO TO 28
000019      LET MEEN(JJ) = MEEN(JJ) + CLCT
000020      LET CNTR(JJ) = CNTR(JJ) + 1
000021      LET STDEV(JJ) = STDEV(JJ) + (CLCT*CLCT)
000022      LET X = (CLCT*100,0)/FLOAT(NRES)
000023      IF X GR 0.5, GO TO 200
000024      LET CAT41(JJ) = CAT41(JJ) + 1
000025      GO TO 210
000026      200 IF X GR 1.0, GO TO 201
000027      LET CAT62(JJ) = CAT62(JJ) + 1
000028      GO TO 210
000029      201 IF X GR 2.0, GO TO 202
000030      LET CAT63(JJ) = CAT63(JJ) + 1
000031      GO TO 210
000032      202 IF X GR 3.0, GO TO 203
000033      LET CAT64(JJ) = CAT64(JJ) + 1
000034      GO TO 210
000035      203 IF X GR 4.0, GO TO 204
000036      LET CAT65(JJ) = CAT65(JJ) + 1
000037      GO TO 210
000038      204 IF X GR 5.0, GO TO 205
000039      LET CAT66(JJ) = CAT66(JJ) + 1
000040      GO TO 210
000041      205 IF X GR 10.0, GO TO 206
000042      LET CAT67(JJ) = CAT67(JJ) + 1
000043      GO TO 210
000044      206 LET CAT68(JJ) = CAT68(JJ) + 1
000045      210 LET CLCT = 0.0
000046      28 LET I = 0
000047      LET IDAY = DPART(TIME)
000048      IF HOLID(I) EQ -1, GO TO 29
000049      NO, FOR EACH HLDY J
000050      IF IPAY EQ HOLID(J), LET I = 1
000051      LOOP
000052      29 LET IDAY = IDAY - (IDAY/J)*7
000053      IF PSHFT NE RWD, GO TO 40
000054      IF I EQ 1, GO TO 50

```

10170

```

000055 IF IDAY EQ ISA, GO TO 30
000056 LET PSHFT = 00 + 1
000057 GO TO 70
000058 30 LET PSHFT = 1
000059 GO TO 70
000060 40 IF PSHFT EQ NWD+NUE, GO TO 50
000061 LET PSHFT = PSHFT + 1
000062 GO TO 70
000063 50 IF I EQ 1, GO TO 55
000064 IF IDAY EQ 1ND, GO TO 60
000065 IF IDAY EQ ISA, GO TO 55
000066 LET ISU = 6
000067 IF 1ND GR 0, LET ISU = 1ND - 1
000068 IF IDAY NE ISU, GO TO 60
000069 55 LET PSHFT = NWD + 1
000070 GO TO 70
000071 60 LET PSHFT = 1
000072 70 LFT ND = FLOAT(OPART(TIME))
000073 CAUSE NJCRU AT 00 + TIME(PSHFT)
000074 LET TLAST = TIME
000075 IF CQUE IS EMPTY, RETURN
000076 DO TO 900, FOR EACH ICNN IN CQUE
000077 IF (FLG(ICNN))EQ(2), GO TO 700
000078 IF (FLG(ICNN))EQ(3), GO TO 800
000079 CALL CRES(ICNN,NEED(ICNN),*KNT,*II)
000080 CALL VEC(ICNN)
000081 CALL OSFT(ICNN,KNT)
000082 DO TO 650, FOR EACH IR IN RQUE
000083 IF (IR(IR))NE(0), GO TO 450
000084 IF (E1AT(IR))EQ(0,0), GO TO 610
000085 IF (HCREW(E1T(IR))EQ(0), GO TO 630
000086 GO TO 650
000087 410 CALL ROCA(ICNN,IR,*III)
000088 IF (III)EQ(1), GO TO 630
000089 650 LOOP
000090 GO TO 900
000091 630 LET TQUE(ICNN) = TQUE(ICNN) + (TIME-TINS(ICNN))
000092 IF (SIGNL(ICNN))EQ(1),LFT TINT(ICNN)=TINT(ICNN)+(TIME-TINS(ICNS)),
000093 REMOVE ICNN FROM CQUE
000094 CALL SERVE(ICNN,E,IR)
000095 GO TO 900
000096 LET CASE = CAS(ICNN)
000097 CALL CRES(CASE,NEED(ICNN),*KNT,*II)
000098 CALL VEC(CASE)
000099 CALL OSFT(CASE,KNT)
000100 GO TO 750, FOR EACH IR IN RQUE
000101 IF (IR(IR))NE(0), GO TO 750
000102 IF (E1AT(IR))EQ(0,0), GO TO 710
000103 IF (HCREW(E1T(IR))EQ(0), GO TO 730
000104 GO TO 750
000105 710 CALL ROCA(CASE,IR,*III)
000106 IF (III)EQ(1), GO TO 730
000107 LOOP
000108 GO TO 900
000109 730 CALL SVQUE(ICNN,IR)
000110 GO TO 900
000111 800 LFT CASE = PSAC(ICNN)
000112 CALL SASS(CASE,SWT(ICNN),ICNN,*LJK)

```

```

101000
10200
10210
10220
10230
10240
10250

```

```

10260
10290
10300
10310
10320

```

```

000113 GO TO 850, FOR EACH I, I, 0,000
000114 IF (I*(IR)) EQ 0, GO TO 850
000115 IF (I*(IR)) EQ 0, GO TO 81
000116 IF (I*(IR)) EQ 0, GO TO 81
000117 GO TO 850
000118 CALL R0C(CASE,IR,IJK)
000119 IF (IJK) EQ 0, GO TO 830
000120 GO TO 850
000121 IF (I*(IR)) EQ 0, GO TO 870
000122 LET X = AINT(TIME) + RISE
000123 IF (TIME) EQ 0, GO TO 870
000124 LET X = AINT(TIME) + SET
000125 IF (TIME) EQ 0, GO TO 870
000126 LET DVEC(IR) = TVEC(IR)
000127 IF (I*(IR)) EQ 0,0, LET DVEC(IR) = TVEC(IR) + DLAY(TYPE(IR))
000128 IF (I*(IR)) EQ 0,0, LET DVEC(IR) = TVEC(IR) + DLAY(TYPE(IR))
000129 LET X = AINT(TIME) + RISE + 1,0
000130 IF (TIME) EQ 0,0, LET DVEC(IR) = TVEC(IR) + DLAY(TYPE(IR))
000131 IF (I*(IR)) EQ 0,0, LET DVEC(IR) = TVEC(IR) + DLAY(TYPE(IR))
000132 LOOP
000133 GO TO 900
000134 LET I = SIGNL(ICNN)
000135 LET SIGNL(ICNN) = 2
000136 LET JIND = 2
000137 LET KIND = 2
000138 DO TO 872, FOR EACH J IN SRHS(CASE)
000139 IF SIGNL(J) EQ 0, LET JIND = 1
000140 IF SIGNL(J) EQ 1, LET KIND = 1
000141 LOOP
000142 IF (I*(IR)) EQ 0, GO TO 878
000143 IF (JIND) EQ 1, GO TO 877
000144 IF (KIND) EQ 1, GO TO 877
000145 LET TQUC(CASE) = TQUC(CASE) + (TIME - TIND(CASE))
000146 GO TO 877
000147 IF (KIND) EQ 1, GO TO 877
000148 IF (JIND) EQ 2, GO TO 866
000149 LET TINT(CASE) = TINT(CASE) + (TIME - STIND(CASE))
000150 GO TO 877
000151 LET TQUC(CASE) = TQUC(CASE) + (TIME - TIND(CASE))
000152 LET TINT(CASE) = TINT(CASE) + (TIME - STIND(CASE))
000153 LET RSRG(ICNN) = I,0
000154 LET SASG(ICNN) = 1
000155 REMOVE ICNN FROM CASE
000156 CALL SSS(ICNN,IR,J)
000157 LOOP
000158 RETURN
000159 END

```

```

000001      EXPENSES = E - E1 - OPSI
000002      C
000003      C OPSIM CREATES AN ENTITY CASE, READS IN CASE ATTRIBUTES FROM THE
000004      C DEMAND TABLE, CHECKS FOR IMPOSSIBLE COMBINATIONS OF CASE ATTRIBUTES,
000005      C AND DETERMINES WHICH SUBROUTINE TO CALL.
000006      C
000007      SAVE EVENT CARD
000008      LET KOUNT = KOUNT + 1
000009      LET DUM = 9999.0
000010      CREATE CASE
000011      LET NRKCS = NRKCS + 1
000012      READ IDLOC(CASE), STAT(CASE), NOCAS(CASE), FOR(CASE), AIR(CASE),
000013      * WIND(CASE), SWELL(CASE), VIS(CASE), GTYPE(CASE), L(CASE), FPRI(CASE),
000014      * FORMAT (13,1115)
000015      READ MIN(CASE), MM(CASE), IDUM, SIS(CASE), S2S(CASE), JDUU, LDUM, IX,
000016      * IY, BOX(CASE), VALUE(CASE)
000017      FORMAT (1015,110)
000018      IF PRIOT EQ 0, GO TO 4
000019      WRITE ON TAPE 6, CASE, TIME
000020      FORMAT (' CASE CREATED IS ,14.9 AT TIME ',M3.2,2)
000021      WRITE ON TAPE 6, IDLOC(CASE), STAT(CASE), NOCAS(CASE), AIR(CASE),
000022      * WIND(CASE), SWELL(CASE), VIS(CASE), GTYPE(CASE), FPRI(CASE), N, MIN(CASE),
000023      * MM(CASE), SIS(CASE), S2S(CASE), JDUU, LDUM, IX, IY
000024      FORMAT (13,13,15,414,613,415)
000025      IF MM(CASE) GR 2, LET MM(CASE) = 2
000026      LET OFFAC(CASE) = STAT(CASE)
000027      LET XCX(CASE) = FLOAT(IX)
000028      LET YCY(CASE) = FLOAT(IY)
000029      LET XC(CASE) = XCX(CASE)
000030      LET YC(CASE) = YCY(CASE)
000031      IF (STAT(CASE))EQ(0), GO TO 3
000032      LET STAT(CASE) = NSN(STAT(CASE))
000033      IF (STAT(CASE))EQ(0), GO TO 3
000034      IF (XS(STAT(CASE)))NE(0.0), GO TO 10
000035      IF (YS(STAT(CASE)))NE(0.0), GO TO 10
000036      DO TO 5, FOR EACH STA I, WITH (PSIN(I))EQ(1)
000037      IF (XS(I))NE(0.0), GO TO 1
000038      IF (YS(I))NE(0.0), GO TO 1
000039      GO TO 5
000040      1 LET X = XS(I) - XC(CASE)
000041      LET Y = YS(I) - YC(CASE)
000042      LET D = SQRT(X*X+Y*Y)
000043      IF (D)LE(DUM), GO TO 5
000044      LET DUM = D
000045      LET STAT(CASE) = I
000046      5 LOOP
000047      GO TO 20
000048      10 IF (PSIN(STAT(CASE)))EQ(0), GO TO 3
000049      LET X = XS(STAT(CASE)) - XC(CASE)
000050      LET Y = YS(STAT(CASE)) - YC(CASE)
000051      LET DUM = SQRT(X*X+Y*Y)
000052      LET JJ = STAT(CASE)
000053      IF (ADJUS(STAT(CASE),JJ))EQ(0), GO TO 20
000054      DO TO 15, FOR I = 1 (ADJUS(STAT(CASE)))

```



```

000055 IF (X$ADJUST(STAT(CASE),1))NE(0.0), GO TO 2
000056 IF (Y$ADJUST(STAT(CASE),1))NE(0.0), GO TO 2
000057 GO TO 15
000058
000059 2 IF (P$TIME(X$ADJUST(STAT(CASE),1))EQ(0)), GO TO 15
000060 LET X = X$ADJUST(STAT(CASE),1) - X(CASE)
000061 LET Y = Y$ADJUST(STAT(CASE),1) - Y(CASE)
000062 LET D = SORT(X*Y+Y)
000063 IF (D)GE(DUM), GO TO 15
000064 LET DUM = D
000065 LET JJ = ADJUST(STAT(CASE),1)
000066
000067 15 LOOP
000068 LET STAT(CASE) = JJ
000069 LET GAMMA(CASE) = FLOAT(DUM)/100.0
000070 LET TSM(CASE) = FLOAT(DUM)
000071 LET OFSHR(CASE) = FLOAT(DUM)/10.0
000072
000073 C INITIALIZE CASE ATTRIBUTES
000074 C
000075 LET COST(CASE) = 0.0
000076 LET FLG(CASE) = 0
000077 IF (N)EQ(1), LET FLG(CASE) = 1
000078 LET NEED(CASE) = 0
000079 LET ITOL(CASE) = 2
000080 LET IMAT(CASE) = 0
000081 LET LOC(CASE) = 0
000082 LET NOTIF(CASE) = 0
000083 LET NQUE(CASE) = 0
000084 LET CHRES(CASE) = 0
000085 LET OCCUR(CASE) = TIME
000086 LET PRI(CASE) = FPR(CASE)
000087 LET REA(CASE) = 2
000088 LET RESA(CASE) = 0
000089 LET SYTAG(CASE) = 0
000090 LET TINT(CASE) = 0.0
000091 LET TQUE(CASE) = 0.0
000092 LET TQUEL(CASE) = 0.0
000093 LET TSVC(CASE) = 0.0
000094 LET TS2(CASE) = 0
000095 IF (N)LE(1), GO TO 500
000096 IF (N)GR(1), GO TO 300
000097 IF (MM(CASE))GR(1), GO TO 200
000098
000099 READ IN NEED OF SINGLE RESOURCE CASE.
000100 C
000101 READ NEED(CASE),OST(CASE)
000102 FORMAT (I2,D1.4)
000103 IF PRTO EQ 0, GO TO 500
000104 WRITE ON TAPE 6, NEED(CASE),OST(CASE),OST(CASE)
000105 FORMAT (' NEED ',I2,' OST ',D1.4,' CR ',M3.2.2)
000106 GO TO 500
000107
000108 C CREATE NOTIFS FOR MULTI-RESOURCE NEEDS; READ IN NEED(1), OST(1),
000109 C DELTA(1) FOR EACH NOTIF.
000110 C
000111 300 IF (N(N(CASE))EQ(0)), GO TO 400
000112 LET INI = N
000113 IF (MM(CASE))GR(1), LET INI = N - 1
000114 DO, FOR I = (1)INI

```

```

000113 CREATE NOTIF
000114 STORE CASE TO CAS(NOTIF)
000115 LET FLS(NOTIF) = 2
000116 LET PRF(NOTIF) = PFI(CASE)
000117 LET RESA(NOTIF) = 0
000118 LET XRES(NOTIF) = 0
000119 LET NUMR(NOTIF) = 0
000120 LET SIGNL(NOTIF) = 0
000121 FILE NOTIF IN NSET(CASE)
000122 LOOP
000123 READ NEFD(1),OST(1),DELTA(1), FOR EACH I IN NSET(CASE)
000124 FORMAT 6(12,D1.4,D1.2)
000125 IF PRFOT EQ 0, GO TO 777
000126 DO, FOR EACH I IN NSFT(CASE)
000127 WRITE ON TAPE 6, NEFD(1),OST(1),DELTA(1)
000128 FORMAT (' NFD',12,' OST',M3.2+2,' DELTA',D1.2)
000129 LOOP
000130 777 IF (MMM(CASE))LE(0), GO TO 500
000131 400 READ ITOL(CASE)
000132 FORMAT (12)
000133 IF PRFOT EQ 0, GO TO 880
000134 WRITE ON TAPE 6, ITOL(CASE)
000135 FORMAT (' ITOL',12)
000136
000137 C
000138 C CHECK FOR IMPOSSIBLE COMBINATIONS OF CASE ATTRIBUTES.
000139 C
000140 888 IF ITOL(CASE) NE 17, GO TO 500
000141 IF NNN(CASE) EQ 0, GO TO 500
000142 200 IF NSET(CASE) IS EMPTY, GO TO 210
000143 REMOVE FIRST NOTIF FROM NSET(CASE)
000144 DESTROY NOTIF
000145 GO TO 200
000146 210 LET ITOL(CASE) = 6
000147 FILE CASE IN EXCS
000148 RETURN
000149 500 IF KOUNT LS LIMIT, GO TO 501
000150 IF LIMIT NE 0, WRITE ON TAPE 6, TIME,LIMIT
000151 FORMAT (' A R N I N G AT TIME',M4.2+2,' THERE ARE AT LEAST,
000152 **, CASES IN THE SYSTEM.', NO MORF CASES WILL OCCUR. THOSE ALRE
000153 ADY IN THE SYSTEM WILL RUN TO COMPLETION.',)
000154 LET LIMIT = 0
000155 602 IF NSET(CASE) IS EMPTY, GO TO 503
000156 REMOVE FIRST NOTIF FROM NSET(CASE)
000157 DESTROY NOTIF
000158 GO TO 502
000159 503 DESTROY CASE
000160 RETURN
000161 501 LET I = NNN(CASE)+MM(CASE)+SIS(CASE)+S2S(CASE)
000162 IF (I)NE(0), GO TO 30
000163 GO TO 700
000164 30 IF (SIS(CASE))EQ(0), GO TO 31
000165 IF (S2S(CASE))EQ(0), GO TO 31
000166 GO TO 700
000167 31 LET I = NNN(CASE)+MM(CASE)+SIS(CASE)
000168 IF (I)NE(0), GO TO 600
000169 LET ITOL(CASE) = 6
000170 FILE CASE IN EXCS
000171 IF (VLE(1), GO TO 701

```



```

000001 SUBROUTINE SRAS(CASE)
000002 C
000003 C SUBROUTINE SRAS SERVES AS A DRIVER FOR SINGLE RESOURCE CASES.
000004 C
000005 CALL CREP(CASE,REQ(CASE),*KMT,*II)
000006 IF II NE 0, GO TO 20
000007 LET ITOL(CASE) = 3
000008 LET FAIL(STAT(CASE)) = FAIL(STAT(CASE)) + 1
000009 FILE CASE IN EXCS
000010 RETURN
000011 20 IF KMT NE 0, GO TO 30
000012 LET ITOL(CASE) = 4
000013 LET FAIL2(STAT(CASE)) = FAIL2(STAT(CASE)) + 1
000014 FILE CASE IN EXCS
000015 RETURN
000016 30 CALL VEC(CASE)
000017 CALL OSPT(CASE,KMT)
000018 CALL RESAP(CASE,RAP,*INFS)
000019 IF (INFS)EQ(0), GO TO 100
000020 CALL SERVE(CASE,0,INFS)
000021 50 RETURN
000022 100 IF (NEED(CASE))EQ(17), GO TO 200
000023 LET TING(CASE) = TIME
000024 IF (RFAIL(CASE))EQ(2), LET REA(CASE) = 1
000025 LET NQUE(CASE) = NQUE(CASE) + 1
000026 LET SIGNAL(CASE) = 0
000027 FILE CASE IN QUE
000028 RETURN
000029 200 LET ITOL(CASE) = 5
000030 FILE CASE IN EXCS
000031 RETURN
000032 END

```

```

000001 SUBROUTINE PRE(CASE, I, K, TO, II)
000002 C
000003 C SUBROUTINE PRE DETERMINES CAPABLE RESOURCE TYPES AND DETERMINES
000004 C ALL STATIONS FROM WHICH RESOURCES CAN BE OBTAINED. IT THEN
000005 C PLACES ALL CAPABLE RESOURCES MEETING THE TYPE AND STATION
000006 C REQUIREMENTS INTO A LIFE SET--RQUE.
000007 C
000008 DIMENSION IPAY1(20)
000009 DIMENSION IPAY2(30)
000010 LET J = 0
000011 LET KOUTR = 0
000012 IF RQUE IS EMPTY, GO TO 5
000013 REMOVE FIRST IAS FROM RQUE
000014 IF RQUE IS NOT EMPTY, GO TO 10
000015 5 IF (SWELL(CASE))GE(5), GO TO 20
000016 LET IPAY1(2)=28
000017 GO TO 30
000018 20 IF (SWELL(CASE))GE(10), GO TO 21
000019 LET IPAY1(2)=29
000020 GO TO 30
000021 21 IF (SWELL(CASE))GE(20), GO TO 22
000022 LET IPAY1(2)=30
000023 GO TO 30
000024 22 LET IPAY1(2)=31
000025 30 IF (WIND(CASE))LS(60), GO TO 31
000026 LET IPAY1(3)=33
000027 GO TO 32
000028 31 LET IPAY1(3)=32
000029 32 IF (VIS(CASE))SR(0), GO TO 40
000030 LET IPAY1(4)=34
000031 GO TO 50
000032 40 LET IPAY1(4)=35
000033 60 IF (AIR(CASE))LS(20), GO TO 61
000034 LET IPAY1(6)=37
000035 GO TO 62
000036 61 LET IPAY1(6)=36
000037 62 IF (OFSHR(CASE))GE(15), GO TO 70
000038 LET IPAY1(5)=38
000039 GO TO 80
000040 70 IF (OFSHR(CASE))GE(5,0), GO TO 71
000041 LET IPAY1(5)=39
000042 GO TO 80
000043 71 IF (OFSHR(CASE))GE(10,0), GO TO 72
000044 LET IPAY1(5)=40
000045 GO TO 30
000046 72 IF (OFSHR(CASE))GE(20,0), GO TO 73
000047 LET IPAY1(5)=41
000048 GO TO 30
000049 73 IF (OFSHR(CASE))GE(5,0), GO TO 74
000050 LET IPAY1(5)=42
000051 GO TO 80
000052 74 LET IPAY1(5)=43
000053 60 IF (INDREQ(15), GO TO 81
000054 IF (INDREQ(14), GO TO 9.

```

```

000055 LET IRAY1(I)=1.0
000056 GO TO 100
000057 41 IF (LCASE)GE(30), GO TO 32
000058 LET IRAY1(I)=15
000059 GO TO 100
000060 32 IF (LCASE)GE(65), GO TO 33
000061 LET IRAY1(I)=20
000062 GO TO 100
000063 33 IF (LCASE)GE(100), GO TO 34
000064 LET IRAY1(I)=21
000065 GO TO 100
000066 34 IF (LCASE)GE(200), GO TO 35
000067 LET IRAY1(I)=22
000068 GO TO 100
000069 35 LET IRAY1(I)=23
000070 GO TO 100
000071 90 IF (POR(CASE))GE(5), GO TO 91
000072 LET IRAY1(I)=16
000073 GO TO 100
000074 91 IF (POR(CASE))GE(10), GO TO 92
000075 LET IRAY1(I)=24
000076 GO TO 100
000077 92 IF (POR(CASE))GE(15), GO TO 93
000078 LET IRAY1(I)=25
000079 GO TO 100
000080 93 IF (POR(CASE))GE(25), GO TO 94
000081 LET IRAY1(I)=26
000082 GO TO 100
000083 94 LET IRAY1(I)=27
000084 100 LET IAND = CAP(IRAY1(I))
000085 DO TO 101, FOR I=(2)(6)
000086 STORE AND(CAP(IRAY1(I)),IAND) IN IAND
000087 101 LOOP
000088 IF (IAND)GR(0), GO TO 102
000089 LET KONTK = 0
000090 LET II = 0
000091 RETURN
000092 102 LET K=0
000093 GO TO 103, FOR I=(1)(NRST)
000094 STORE A+D(IAND,MASK(I)) IN ITEMP
000095 IF (ITEMP)LE(0), GO TO 103
000096 LET K=K+1
000097 LET IRAY1(K)=I
000098 103 LOOP
000099 LET II=1
000100 IF (S1(CASE))GR(0), GO TO 210
000101 IF (MM(CASE)+MM(CASE))GR(0), GO TO 210
000102 GO TO (210,210,210,201,201,210), MAP
000103 201 LET IRAY2(I) = STATN(CASE)
000104 LET J=J+1
000105 DO TO 202, FOR I=(1)(NACS(STATN(CASE)))
000106 IF (ACS(STATN(CASE),I))GR(0), GO TO 204
000107 LET J=J+1
000108 LET IRAY2(J) = ACS(STATN(CASE),I)
000109 202 LOOP
000110 204 DO TO 203, FOR I=(1)(NOUT(STATN(CASE)))
000111 IF (OUT(STATN(CASE),I))GR(0), GO TO 205
000112 LET J=J+1

```

```

000113 LET IPAY2(J) = CUT(STATN(CASE),I)
000114 LOOP
000115 GO TO 250
000116 LET IPAY2(J) = STATN(CASE)
000117 LET J=J+1
000118 DO TO 211, FOR I=(1)(NADJUS(STATN(CASE)))
000119 IF (ADJUS(STATN(CASE),I))EQ(0), GO TO 219
000120 LET J=J+1
000121 LET IPAY2(J) = ADJUS(STATN(CASE),I)
000122 LOOP
000123 DO TO 212, FOR I=(1)(NACS(STATN(CASE)))
000124 IF (ACS(STATN(CASE),I))EQ(0), GO TO 206
000125 LET J=J+1
000126 LET IPAY2(J) = ACS(STATN(CASE),I)
000127 LOOP
000128 DO TO 213, FOR I=(1)(NCUT(STAT (CASE)))
000129 IF (CUT(STATN(CASE),I))EQ(0), GO TO 207
000130 LET J=J+1
000131 LET IPAY2(J) = CUT(STATN(CASE),I)
000132 LOOP
000133 DO TO 214, FOR I=(1)(NADJUS(STATN(CASE)))
000134 IF (ADJUS(STATN(CASE),I))EQ(0), GO TO 250
000135 GO TO 215, FOR LL = (1)(NACS(ADJUS(STATN(CASE),)))
000136 IF (ACS(ADJUS(STATN(CASE),I),LL))EQ(0), GO TO 206
000137 LET NO = ACS(ADJUS(STATN(CASE),I),LL)
000138 LET ITAG=0
000139 DO TO 216, FOR M=(1)(J)
000140 IF (NONE(IPAY2(M)), GO TO 216
000141 LET ITAG=1
000142 LOOP
000143 IF (ITAG)EQ(1), GO TO 215
000144 LET J=J+1
000145 LET IPAY2(J)=NO
000146 LOOP
000147 DO TO 217, FOR LL=(1)(NCUT(ADJUS(STATN(CASE),I)))
000148 IF (CUT(ADJUS(STATN(CASE),I),LL))EQ(0), GO TO 214
000149 LET NO = CUT(ADJUS(STATN(CASE),I),LL)
000150 LET ITAG=0
000151 DO TO 218, FOR M=(1)(J)
000152 IF (NONE(IPAY2(M)), GO TO 218
000153 LET ITAG=1
000154 LOOP
000155 IF (ITAG)EQ(1), GO TO 217
000156 LET J=J+1
000157 LET IPAY2(J)=NO
000158 LOOP
000159 DO TO 260, FOR EACH RES I
000160 GO TO 270, FOR LL=(1)(J)
000161 IF (STAT(I))EQ(IPAY2(LL)), GO TO 280
000162 LOOP
000163 GO TO 260
000164 DO TO 290, FOR LL=(1)(K)
000165 IF (TYPE(I))EQ(IPAY1(LL)), GO TO 291
000166 LOOP
000167 GO TO 260
000168 FILE 1 IN 'AGE
000169 LET KOUT=KOUT+1
000170

```


000171
000172
000173

260 1979
58104
F00

```

000001 SUBROUTINE EC(CASE)
000002 C
000003 C
000004 C
000005 SUBROUTINE EC CALCULATES TIME-TO-VECTOR FOR EACH RES IN ROW.
000006
000007 DO TO 100, FOR EACH IRS OF ROW
000008 IF (E1A(IRS))EQ(0.0), GO TO 15
000009 IF (DEP(IFLT(IRS)))NE(0.0), GO TO 20
000010 LET X = XR(IRS)-XC(CASE)
000011 LET Y = YR(IRS)-YC(CASE)
000012 LET D = SQR(X*X+Y*Y)
000013 IF (SHELL(CASE))GRSLIM(TYPE(IRS)), GO TO 30
000014 LET TVEC(IRS) = D/SOA1(TYPE(IRS))
000015 GO TO 100
000016 30 LET TVEC(IRS) = D/SOA2(TYPE(IRS))
000017 GO TO 100
000018 20 LET X = XR(IRS)-XDEST(IFLT(IRS))
000019 LET Y = YR(IRS)-YDEST(IFLT(IRS))
000020 LET D = SQR(X*X+Y*Y)
000021 IF (IR(IRS))NE(1), GO TO 60
000022 IF (SHELL(CASE))GRSLIM(TYPE(IRS)), GO TO 61
000023 LET C = (TIME-DEP(IFLT(IRS)))*SOA1(TYPE(IRS))
000024 GO TO 70
000025 61 LET C = (TIME-DEP(IFLT(IRS)))*SOA2(TYPE(IRS))
000026 GO TO 70
000027 60 IF (TOW(IFLT(IRS)))EQ(0), GO TO 40
000028 LET C = TIME - DEP(IFLT(IRS))
000029 IF (C)LE(0.0), GO TO 65
000030 LET C = (TIME-DEP(IFLT(IRS)))*TOWSP(IFLT(IRS))
000031 GO TO 70
000032 65 LET C = 0.0
000033 GO TO 70
000034 40 IF (SHELL(CASE)(IFLT(IRS)))GRSLIM(TYPE(IRS)), GO TO 50
000035 LET C = (TIME-DEP(IFLT(IRS)))*SOA1(TYPE(IRS))
000036 GO TO 70
000037 50 LET C = (TIME-DEP(IFLT(IRS)))*SOA2(TYPE(IRS))
000038 GO TO 70
000039 70 IF (C)GE(0), GO TO 80
000040 LET X1(IFLT(IRS)) = XR(IRS)-((C*X)/D)
000041 LET Y1(IFLT(IRS)) = YR(IRS)-((C*Y)/D)
000042 GO TO 90
000043 80 LET X1(IFLT(IRS)) = XDEST(IFLT(IRS))
000044 LET Y1(IFLT(IRS)) = YDEST(IFLT(IRS))
000045 LET X = X1(IFLT(IRS))-XC(CASE)
000046 LET Y = Y1(IFLT(IRS))-YC(CASE)
000047 LET D = SQR(X*X+Y*Y)
000048 IF (SHELL(CASE))GRSLIM(TYPE(IRS)), GO TO 95
000049 LET TVEC(IRS) = D/SOA1(TYPE(IRS))
000050 GO TO 100
000051 95 LET TVEC(IRS) = D/SOA2(TYPE(IRS))
000052 GO TO 100
000053 REPEAT 10
000054 RETURN
000055 END

```

```

000001 SUBROUTINE TSET(CASE,KOCTR)
000002 C
000003 C OSET RANKS THE CAPABLE RESOURCES, WITH TVEC(IRS) LESS THAN OR
000004 C EQUAL TO TOL(I), IN INCREASING COST AND RANKS THE REMAINING
000005 C RESOURCES IN INCREASING TVEC(IRS). THE TAG SETS ARE THEN COMPILED
000006 C INTO A SINGLE LIST.
000007 C
000008 DO TO 100, FOR I=(1)(KOCTR)
000009 REMOVE FIRST IRS FROM RQUE
000010 LET TVEC(IRS) = TVEC(IRS)
000011 IF (TOL(IRS))EQ(0.0), LET TVEC(IRS)=TVEC(IRS)+DELAY/TYPE(IRS))
000012 IF (TVEC(IRS))LE(TOL(PRI(CASE))), GO TO 20
000013 FILE IRS IN TSET
000014 GO TO 100
000015
000016 20 IF (COST(IRS))GT(0), GO TO 30
000017 LET COST(IRS)=COST(TYPE(IRS))*TVEC(IRS)
000018 FILE IRS IN CSET
000019 GO TO 100
000020
000021 30 LET COST(IRS) = PCOST(TYPE(IRS))
000022 FILE IRS IN CSET
000023 100 LOOP
000024
000025 35 IF CSET IS EMPTY, GO TO 36
000026 REMOVE FIRST IRS FROM CSET
000027 FILE IRS IN RQUE
000028 GO TO 35
000029
000030 36 IF TSET IS EMPTY, GO TO 37
000031 REMOVE FIRST IRS FROM TSET
000032 FILE IRS IN RQUE
000033 GO TO 36
000034
000035 37 DO TO 40, FOR EACH III IN RQUE
000036 LET COST(III) = COST(TYPE(III))*TVEC(III)
000037 40 LOOP
000038 RETURN
000039 END

```

16


```

000113 DO TO 511, FOR I=(1)ADJUS(STATN(CASE)))
000114 IF (STN(I,RS))EQ(ADJUS(STATN(CASE),1)), GO TO 550
000115 DO TO 512, FOR K=(1)NAC(ADJUS(STATN(CASE),1)))
000116 IF (STN(I,RS))EQ(AC(ADJUS(STATN(CASE),1),K)), GO TO 650
000117 LOOP
000118 DO TO 613, FOR K=(1)ECUT(ADJUS(STATN(CASE),1)))
000119 IF (STN(I,RS))EQ(CUT(ADJUS(STATN(CASE),1),K)), GO TO 550
000120 LOOP
000121 LOOP
000122 GO TO 510
000123 IF (PRIOR(I,PS))GE(PRI(C,SE)), GO TO 510
000124 IF (PRIOR(I,PS))GE(IDUM), GO TO 510
000125 LET IDUM = PRIOR(I,RS)
000126 LET JRS = IRS
000127 LOOP
000128 IF (IDUM)NE(100), GO TO 480
000129 GO TO (400,400,900,900,900,900), IRAP
000130 C
000131 C
000132 C
000133 LET IDUM = 100
000134 DO TO 710, FOR EACH IRS ON RQUE
000135 IF (IB(I,RS))EQ(0), GO TO 710
000136 IF (HCREW(I,FLT(I,RS)))GR(0), GO TO 710
000137 IF (PRIOR(I,PS))GE(PRI(CASE)), GO TO 710
000138 IF (PRIOR(I,RS))GE(IDUM), GO TO 710
000139 LET IDUM = PRIOR(I,RS)
000140 LET JRS = IRS
000141 LOOP
000142 IF (IDUM)NE(100), GO TO 480
000143 GO TO (900,900,400,800,900,900), IRAP
000144 C
000145 C
000146 C
000147 C
000148 C
000149 LET JRS = 0
000150 RETURN
000151 WRITE ON TAPE 6, NAC(CASE)
000152 FORMAT (' SOMETHING JOLLY WFL WENT HAYWIRE WHEN CASE ',IR,
000153 1 * ENTERED THE SYSTEM.')
000154 RETURN
000155 END
000156

```

```

000001 SUBROUTINE POCA(CASE,IRS,I1)
000002 C
000003 C SURROUTINE POCA DETERMINES THE OPERATIONAL STATUS AND THE
000004 C AVAILABILITY OF A CRE4 FOR A RESOURCE.
000005 C
000006 IF (RANDMLF(MR(TYPE(IRS))), GO TO 10
000007 LET I1 = 2
000008 RETURN
000009 10 LET K = SHIFT(STN(IRS),PSHFT)-BCRE#(STN(IRS))
000010 IF (CL)GE(0), GO TO 20
000011 IF (K)GR(0), GO TO 15
000012 LET I1 = 2
000013 RETURN
000014 20 IF (K)LE(CL), GO TO 30
000015 15 LET I1 = 1
000016 LET BCRE#(STN(IRS))=BCRE#(STN(IRS))+1
000017 RETURN
000018 30 IF (K)LS(CL), GO TO 50
000019 IF (CL)EQ(0), GO TO 50
000020 LET I1 = 1
000021 LET BCRE#(STN(IRS))=BCRE#(STN(IRS))+1
000022 GO TO 40
000023 50 LET I1 = 2
000024 40 IF (SYTAG(CASE))FQ(1), GO TO 60
000025 LET SYTAG(CASE) = 1
000026 CREATE STNBY
000027 LET NSTAY(STN(IRS)) = NSTAY(STN(IRS)) + 1
000028 LET TOSBY = TOSBY + 1
000029 STORE STN(IRS) IN STAT(STNBY)
000030 CAUSE STNBY AT TIME + (2.0/24.0)
000031 60 RETURN
000032 END

```

```

000001 SUBROUTINE MRAS(CASE)
000002 IF MM(CASE) EQ 0, GO TO 50
000003 LET STOS = 0.
000004 LET XTOS = 0.
000005 DO, FOR EACH NOTIF IN NSET(CASE)
000006 IF OST(NOTIF) GR XTOS, LET XTOS = OST(NOTIF)
000007 LET STOS = STOS + OST(NOTIF)
000008 LOOP
000009 LET TF = (1.0-GAMMA(CASE))*XTOS + GAMMA(CASE)*STOS
000010 DO, FOR EACH NOTIF IN NSET(CASE)
000011 CAUSE NOTIF AT DELTA(NOTIF)*(TE-OST(NOTIF)) + TIME
000012 LET NUMBR(NOTIF) = 0
000013 LET COMP(NOTIF) = 0
000014 LOOP
000015 50 IF MM(CASE) NE 0, CALL TOW(CASE)
000016 RETURN
000017 END OF MRAS

```

```

200020
20010
20020
20030
20040
20050
20060
20070
20080
20090
20100
20110
20120
20130
20140
20150
20160

```



```

000001      ENDOGENOUS EVENT NOTIF
000002      LET JRAP = RAP
000003      IF RAP EQ 4, LET JRAP = 7
000004      IF RAP EQ 5, LET JRAP = 6
000005      7 CALL CRES(CAS(NOTIF),NEED(NOTIF),*KONTR,*11)
000006      IF(11)EQ(0), GO TO 33
000007      IF (KONTR)EQ(0), GO TO 87
000008      IF KONTR GR 1, GO TO 15
000009      DO TO 10, FOR EACH IRS OF RQUE
000010      IF 1R(IRS) LT 1, GO TO 10
000011      IF MFLG(IFLT(IRS)) NE 9, GO TO 10
000012      IF ACASE(IFLT(IRS)) NE CAS(NOTIF), GO TO 10
000013      LET KRES(FITN(IFLT(IRS))) = 0
000014      LET TVFC(IRS) = 0.0
000015      LET IRES = IHS
000016      GO TO 18
000017      10 LOOP
000018      GO TO 15
000019      33 LET FAIL1(STATN(CAS(NOTIF)))=FAIL1(STATN(CAS(NOTIF)))+1
000020      LET ITOL(CAS(NOTIF)) = 3
000021      GO TO 89
000022      87 LET FAIL2(STATN(CAS(NOTIF)))=FAIL2(STATN(CAS(NOTIF)))+1
000023      LET ITOL(CAS(NOTIF)) = 4
000024      89 LET COMP(NOTIF) = 4
000025      CALL WRECK(NOTIF)
000026      RETURN
000027      15 CALL VEC(CAS(NOTIF))
000028      CALL OSET(CAS(NOTIF),KONTR)
000029      CALL RESAP(CAS(NOTIF),JRAP,*IRES)
000030      18 LET KRES(NOTIF) = IRES
000031      IF (IRES)EQ(0), GO TO 11
000032      LET COMP(NOTIF) = 2
000033      CALL SERVE(CAS(NOTIF),NOTIF,IRES)
000034      RETURN
000035      11 LET M=0
000036      CALL QUEUE(NOTIF,M)
000037      RETURN
000038      END OF NOTIF

```

```

000001 SUBROUTINE TQ%(CASE)
000002 CALL DTQ(CASE,XD,YD,DA)
000003 LET GAP = D/2.0
000004 IF HO LS GAP, LET GAP = HO
000005 LET M = MM(CASE)
000006 DO TO 50, FOR I=(1) (M)
000007 CREATE NOTIF
000008 FILE NOTIF IN NSET(CASE)
000009 LET NEF(NOTIF) = ITO%(CASE)
000010 LET PRI(NOTIF) = PRI(CASE)
000011 LET CAS(NOTIF) = CASE
000012 LET NUMR(NOTIF) = 1
000013 LET RESA(NOTIF) = 0
000014 LET KRES(NOTIF) = 0
000015 LET FLG(NOTIF) = 2
000016 LET COMP(NOTIF) = 5
000017 LET SIGNL(NOTIF) = 2
000018 IF I NE M, GO TO 40
000019 60 LET XHAND(NOTIF) = XD
000020 LET YHAND(NOTIF) = YD
000021 GO TO 45
000022 40 IF D LE 0.0, GO TO 60
000023 LET YHAND(NOTIF) = YO + ,GAP*(YC(CASE)-YD))/D
000024 LET XHAND(NOTIF) = XD + (GAP*(XC(CASE)-XD))/D
000025 45 IF NNN(CASE) GR 0, GO TO 50
000026 IF I EQ 2, GO TO 50
000027 LET COMP(NOTIF) = 0
000028 CAUSE NOTIF AT TIME
000029 50 LOOP
000030 RETURN
000031 END OF TQ%

```

25000
25010
25012
25013
25020
25030
25040
25042
25050
25060
25070
25080
25090
25100
25110
25120

25130

25150
25160

25180
25190
25200
25210
25220
25230
25240
25250

```

000001 SURROUTINE DTG(CASE,XD,Y,ND)
000002 IF NPATRL LE 0, GO TO 10
000003 DO TO 5, FOR I = (1)(NPATRL)
000004 IF STATN(CASE) NE IPAT(I), GO TO 5
000005 LET XD = XS(ADJUS(STATN(CASE),1))
000006 LET YD = YS(ADJUS(STATN(CASE),1))
000007 GO TO 3
000008 5 LOOP
000009 10 LET XD = XS(STATN(CASE))
000010 LET YD = YS(STATN(CASE))
000011 3 LET D = SQRT((XC(CASE)-XD)**2 + (YC(CASE)-YD)**2)
000012 RETURN
000013 END OF DTG
300007
300010
300020
300030
300040
300050
300060
300070
300080
300090
300240
300250
300260

```

```

000001 SUBROUTINE SRCH(CASE)
000002 C
000003 C SUBROUTINE SRCH CREATES AN EVENT NOTICE ENTITY (NOT.) FOR EACH
000004 C SEARCH NEED. SRCH THEN ATTEMPTS TO ASSIGN THE FIRST RESOURCE TO
000005 C COMPLETE SM(I).
000006 C
000007 DO TO 10, FOR I=(1)(SIS,CASE)
000008 CREATE NOTE
000009 LET SM(NOTE) = PRISM(SIS(CASE),I)*TSM(CASE)
000010 LET PRI(NOTE) = PRI(CASE)
000011 LET FLG(NOTE) = 3
000012 LET SDAY(NOTE) = 1
000013 LET SASG(NOTE) = 0
000014 LET ESAC(NOTE) = CASE
000015 LET SFLAG(NOTE) = 7
000016 LET SIGNAL(NOTE) = 2
000017 LET RESA(NOTE) = 0
000018 LET RSRC(NOTE) = 0
000019 FILE NOTE IN SRHS(CASE)
000020 10 LOOP
000021 LET X = AINT(TIME)
000022 LET SN = X + RISE
000023 IF (TIME)LS(SN-XRX), GO TO 100
000024 LET SN = X + SET
000025 IF (TIME)LS(SN), GO TO 500
000026 LET K = 0
000027 DO TO 190, FOR EACH I IN SRHS(CASE)
000028 IF (K)NE(0), GO TO 150
000029 CALL SASS(CASE,SM(I),I,IJK)
000030 IF (IJK)NE(0), GO TO 300
000031 LET JRAP = RAP
000032 IF (RAP)EQ(4), LET JRAP = 2
000033 IF (RAP)EQ(5), LET JRAP = 6
000034 CALL RESAP(CASE,JRAP,IP)
000035 IF (IR)EQ(0), GO TO 110
000036 LET SFLAG(I) = 1
000037 LET RSRC(I) = IR
000038 LET SASG(I) = 1
000039 CALL SSS(I,IR,I)
000040 GO TO 120
000041 110 FILE I IN LIST
000042 120 LET K = K + 1
000043 GO TO 190
000044 150 FILE I IN LIST
000045 190 LOOP
000046 RETURN
000047 200 LET K = 0
000048 DO TO 290, FOR EACH I IN SRHS(CASE)
000049 IF (K)NE(0), GO TO 250
000050 CALL SASS(CASE,SM(I),I,IJK)
000051 IF (IJK)NE(0), GO TO 300
000052 LET JRAP = RAP
000053 IF (RAP)EQ(4), LET JRAP = 2
000054 IF (RAP)EQ(5), LET JRAP = 6

```

```

000055 CALL RESAP(CASE,JRAP,I,I)
000056 IF (IR)EQ(0), GO TO 216
000057 LET SFLAG(I) = 1
000058 LET SAS3(I) = 1
000059 LET RSRC(I) = IR
000060 CALL SSS(I,IR,I)
000061 GO TO 220
000062
216 LET SIGN(I) = 0
000063 LET TIME(ESAC(I)) = TIME
000064 LET NOJE(ESAC(I)) = NOJE(ESAC(I)) + 1
000065 LET REA(ESAC(I)) = 1
000066 FILE I IN QUEUE
000067 LET K = K + 1
000068 GO TO 290
000069
250 CAUSE NOTE CALLED I AT TIME + FPSLN
290 LOOP
000070
RETURN
000071
300 REMOVE FIRST NOTE FROM SRHS(CASE)
000072 DESTROY NOTE
000073 IF SRHS(CASE) IS NOT EMPTY, GO TO 300
000074 IF (IJK)EQ(1), GO TO 310
000075 IF (IJK)EQ(2), GO TO 320
000076 IF (IJK)EQ(3), GO TO 320
000077 LET ITOL(CASE) = 3
000078 LET FAIL1(STATN(CASE)) = FAIL1(STATN(CASE)) + 1
000079 GO TO 330
000080
320 LET ITOL(CASE) = 4
000081 LET FAIL2(STATN(CASE)) = FAIL2(STATN(CASE)) + 1
000082
330 FILE CASE IN EXCS
000083 RETURN
000084 END
000085

```

```

000001      C      ENDOGENOUS EVENT NOTE
000002      C
000003      C      ENDOGENOUS EVENT NOTE OCCURS AT TIME+EPSLN FOR SEARCH NEEDS,
000004      C      SM(1) WHERE I=(2)(51): NOTE IS CAUSED ONLY IF CASE OCCURS DURING
000005      C      THE DAY
000006      C
000007      LET X = AINT (TIME)
000008      LET SN = X + RISE
000009      IF (TIME)LS(SN-XRX), GO TO 500
000010      LET SN = X + SET
000011      IF (TIME)GE(SN), GO TO 500
000012      CALL SASS(ESAC(NOTE),SN(NOTE),NOTE,*IJK)
000013      LET JRAP = RAP
000014      IF (RAP)EQ(4), LET JRAP = 2
000015      IF (RAP)EQ(5), LET JRAP = 6
000016      CALL RESAP(ESAC(NOTE),JRAP,*IR)
000017      IF (IR)EQ(0), GO TO 300
000018      LET SASG(NOTE) = 1
000019      LET PSRC(NOTE) = IP
000020      LET I = 0
000021      DO TO 200, FOR EACH LL IN SRHS(ESAC(NOTE))
000022      IF (SFLAG(LL))EQ(1), LET I = 1
000023      200 LOOP
000024      IF (I)EQ(0), LET SFLAG(NOTE) = 1
000025      IF (SFLAG(NOTE))EQ(1), GO TO 210
000026      CALL SSS(NOTE,IR,2)
000027      GO TO 220
000028      210 CALL SSS(NOTE,IR,1)
000029      RETURN
000030      220 LET I = 0
000031      DO TO 305, FOR EACH LL IN SRHS(ESAC(NOTE))
000032      IF (SIGNL(LL))NE(2), LET I = 1
000033      305 LOOP
000034      IF (I)EQ(0), LET TING(ESAC(NOTE)) = TIME
000035      LET NQUE(ESAC(NOTE)) = NQUE(ESAC(NOTE)) + 1
000036      IF (REA(ESAC(NOTE))EQ(2), LET REA(ESAC(NOTE)) = 1
000037      LET SIGNL(NOTE) = 0
000038      FILE NOTE IN CQUE
000039      RETURN
000040      500 FILE NOTE IN LIST
000041      RETURN
000042      END

```

```

000001 SUBROUTINE SASS(CASE,X,LF5,NOTE,IOC)
000002 C
000003 C
000004 C
000005 C
000006 C
000007 C
000008 C
000009 C
000010 C
000011 C
000012 C
000013 C
000014 C
000015 C
000016 C
000017 C
000018 C
000019 C
000020 C
000021 C
000022 C
000023 C
000024 C
000025 C
000026 C
000027 C
000028 C
000029 C
000030 C
000031 C
000032 C
000033 C
000034 C
000035 C
000036 C
000037 C
000038 C
000039 C
000040 C
000041 C
000042 C
000043 C
000044 C
000045 C
000046 C
000047 C
000048 C
000049 C
000050 C
000051 C
000052 C
000053 C
000054 C

SUBROUTINE SASS(CASE,X,LF5,NOTE,IOC)
SUBROUTINE SASS MAY BE CALLED FROM THE SERVICE ROUTINE WHEN
S(2)=2 OR FROM THE SEARCH ROUTINE. IT FINDS AN ORDERED SET OF
RESOURCES TO SERVE THE SEARCH CASE AND PLACES THEM IN RQUE.

LEFT IDC = 0
CALL CRES(CASE,IP,KWT,10)
IF (IJINE(0), GO TO 5
LEFT IDC = 1
RETURN
5 IF (KNT)NE(0), GO TO 6
LEFT IDC = 2
RETURN
6 CALL VEC(CASE)
LEFT SNS = AINT(TIME) + SET
IF (TIME)GE(SNS), LET SNS = SNS + 1.0
IF (TIME+TOLS(PRI(CASE)))GR(SNS), GO TO 7
LET TOLRS = TOLS(PRI(CASE))
GO TO 8
7 LEFT TOLRS = SNS - TIME
CALCULATE PR FOR EACH RESOURCE IN RQUE
8 DO TO 9, FOR EACH IRS IN RQUE
LET X = XS(STN(IRS))-XC(CASE)
LET Y = YS(STN(IRS))-YC(CASE)
LET D = SQRT(X*X+Y*Y)
IF (SMELL(CASE))GR(SIM(TYPE(IRS))),GO TO 100
LET X = D/50A1(TYPE(IRS))
GO TO 101
100 LET X = 0/50A2(TYPE(IRS))
101 LET Y = END(TYPE(IRS))-(2.0*X)
IF (YLE(0.0), GO TO 150
LET D = END(TYPE(IRS)) + IF(TYPE(IRS))
LET STOLR = TOLRS
IF (VOP(TYPE(IRS)))EQ(0), LET STOLR = TOLS(PRI(CASE))
LET SNS = AINT(STOLR/D)
LEFT ANS = STOLR - (D*SNS) - X
IF (ANS)GE(0.0), GO TO 102
LEFT ANS = 0.0
GO TO 103
102 IF (ANS)LE(Y), GO TO 103
LEFT ANS = Y
103 LET ANS = (Y*SNS) + ANS
IF (ANS)LS(0.0), GO TO 150
IF (XMLE)LE(0.0), GO TO 150
LET PR(IRS) = (50A3(TYPE(IRS))/XMLE)*ANS
GO TO 9
150 LEFT PR(IRS) = 0.0
9 LOOP
DO, FOR EACH IRS IN RQUE
LET TST = XMLE/50A3(TYPE(IRS))
LET COST(IRS) = COSTD(TYPE(IRS))*(TVFC(IRS)+TST)

```

```

000055 LET DVFC(IRS) = TVFC(IRS)
000056 IF (ELAT(IRS))LE(0.0), LET DVFC(IRS)=TVFC(IRS)+ULAY(TYPE(IRS))
000057 LOOP
000058 IF (S25(CASE))EQ(2), GO TO 20
000059 IF (SDAY(NOTE))EQ(1), LET PERC = PDC1
000060 IF (SDAY(NOTE))GR(1), LET PERC = PDC2
000061 GO TO 21
000062
20 LET PERC = 1.0
000063 IF RQUE IS EMPTY, GO TO 31
000064 REMOVE FIRST IRS FROM RQUE
000065 IF (PR(IRS))GR(0.0), GO TO 40
000066 LET X = XS(STN(IRS))-XC(CASE)
000067 LET Y = YS(STN(IRS))-YC(CASE)
000068 LET D = SQRT(X*X+Y*Y)
000069 IF (SWELL(CASE))GR(SLIN(TYPE(IRS))), GO TO 25
000070 LET X = D/SDA1(TYPE(IRS))
000071 GO TO 24
000072
25 LET X = D/SDA2(TYPE(IRS))
000073
26 IF (END(TYPF(IRS))-(2.0*X))LE(0.0), GO TO 21
000074 FILE IRS IN TSET
000075 GO TO 21
000076
40 IF (PR(IRS))GE(PERC), GO TO 50
000077 FILE IRS IN PSET
000078 GO TO 21
000079
50 FILE IRS IN CSET
000080 GO TO 21
000081
31 IF CSET IS EMPTY, GO TO 60
000082 REMOVE FIRST IRS FROM CSET
000083 FILE IRS IN RQUE
000084 GO TO 31
000085
60 IF PSET IS EMPTY, GO TO 70
000086 REMOVE FIRST IRS FROM PSET
000087 FILE IRS IN RQUE
000088 GO TO 60
000089
70 IF TSET IS EMPTY, GO TO 80
000090 REMOVE FIRST IRS FROM TSET
000091 FILE IRS IN RQUE
000092 GO TO 70
000093
80 IF RQUE IS EMPTY, LET IDC = 3
000094 RETURN
000095 END

```


104,

```

000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097
000098
000099
000100
000101
000102
000103
000104
000105
000106
000107
000108
000109
000110
000111
000112

45 CANCEL DELAY CALLED IDEV(IFLT(IRES))
   DESTROY DELAY CALLED IDEV(IFLT(IRES))
   GO TO 41
46 CANCEL READY CALLED IDEV(IFLT(IRES))
   DESTROY READY CALLED IDEV(IFLT(IRES))
41 LET IR(IRES) = 0
   LET MFLG(IFLT(IRES)) = 0
   CALL EXQ(IRFS,**IVAR,
   RETURN
44 WRITE ON TAPE 6, IRES,MFLG(IFLT(IRES))
   FORMAT (' RESOURCE *,214,* WENT HAYWIRE AT STATEMENT 40 IN SSS*,
   RETURN
C
C
C
C IF IR(IRES)=1, DEST NE 0.0 AND RESOURCE WILL NOT BE USED.
C
C
45 GO TO (70,73,72,71,71,71,71,71,71,71,71,71,71),MFLG(IFLT(IRES))
70 CANCEL ARVSN CALLED IDEV(IFLT(IRES))
   DESTROY ARVSN CALLED IDEV(IFLT(IRES))
   GO TO 46
72 CANCEL ARSCH CALLED IDEV(IFLT(IRES))
   DESTROY ARSCH CALLED IDEV(IFLT(IRES))
   GO TO 46
73 CANCEL ONSCH CALLED IDEV(IFLT(IRES))
   DESTROY ONSCH CALLED IDEV(IFLT(IRES))
46 LET IR(IRES) = 0
   LET XR(IRES) = XI(IFLT(IRES))
   LET YR(IRES) = YI(IFLT(IRES))
   LET XDEST(IFLT(IRES)) = 0.0
   LET YDEST(IFLT(IRES)) = 0.0
   LET MFLG(IFLT(IRES)) = 0
   CALL EXQ(IRFS,**IVAR)
50 RETURN
71 WRITE ON TAPE 6, IRES, MFLG(IFLT(IRES))
   FORMAT (' RESOURCE *,214,* WENT HAYWIRE AT STATEMENT 45 IN SSS *,
   RETURN
C
C
C UPDATING ALL NECESSARY RES/FLT ATTRIBUTES.
C
C
200 IF (ETAT(IRES))NE(0.0), GO TO 300
   CREATE FLT CALLED IFLT(IRES)
   LET ETAT(IRES) = 1.0
   LET IFLT(IFLT(IRES)) = 0.0
   LET TOX(IFLT(IRES)) = 0
   LET HCREW(IFLT(IRES)) = 0
   LET IR(IRES) = 1
   LET NCASE(IRES) = NCASE(IRES) + 1
   LET PRIOR(IRES) = PRI(NOTE)
   LET CNRES(ESAC(NOTE)) = CNRES(ESAC(NOTE)) + 1
   LET COSTC(ESAC(NOTE)) = COSTC(ESAC(NOTE)) + COST(IR,S)
   LET ACASE(IFLT(IRES)) = ESAC(NOTE)
   LET DEP(IFLT(IRES)) = 0.0
   LET FITON(IFLT(IRES)) = NOTE
   LET XDEST(IFLT(IRES)) = 0.0
   LET YDEST(IFLT(IRES)) = 0.0
   CREATE READY CALLED IDEV(IFLT(IRES))
   LET NREADY(IDEV(IFLT(IRES))) = NOTE
   LET RREADY(IDEV(IFLT(IRES))) = IRES
   LET RREADY(IDEV(IFLT(IRES))) = KLK

```

```

000113 LET MFLG(IFLT(IRES)) = 12
000114 CAUSE READY CALLED IDEV(IFLT(IRES)) AT TIME+DELAY(TYPE(IRES))
000115 LET NEEDS(ISTN(IRES)) = %FEDS(ISTN(IRES)) + 1
000116 RETURN
000117 300 IF (IR(IRES))NE(0), GO TO 400
000118 IF (MFLG(IFLT(IRES)))NE(9), GO TO 301
000119 IF (MFLG(IFLT(IRES)))NE( 7), GO TO 40
000120 CANCEL HOME CALLED IDEV(IFLT(IRES))
000121 DESTROY HOME CALLED IDEV(IFLT(IRES))
000122 GO TO 301
000123 400 WRITE ON TAPE 6, IRES,MFLG(IFLT(IRES))
000124 FORMAT ( ' RESOURCE ',214,' WENT HAYWIRE AT STATEMENT 300 IN SSS',)
000125 301 LET XR(IRES) = XI(IFLT(IRES))
000126 LET YR(IRES) = XI(IFLT(IRES))
000127 GO TO 600
000128 IF (DEP(IFLT(IRES)))NE(0,0), GO TO 500
000129 IF (MFLG(IFLT(IRES)))NE(9), GO TO 401
000130 IF (MFLG(IFLT(IRES)))NE( 2), GO TO 90
000131 CANCEL ONSCN CALLED IDEV(IFLT(IRES))
000132 DESTROY ONSCN CALLED IDEV(IFLT(IRES))
000133 GO TO 401
000134 90 IF (MFLG(IFLT(IRES)))NE( 4), GO TO 91
000135 CANCEL COMPL CALLED IDEV(IFLT(IRES))
000136 DESTROY COMPL CALLED IDEV(IFLT(IRES))
000137 GO TO 401
000138 91 IF (MFLG(IFLT(IRES)))NE( 5), GO TO 92
000139 CANCEL SSET CALLED IDEV(IFLT(IRES))
000140 DESTROY SSET CALLED IDEV(IFLT(IRES))
000141 GO TO 401
000142 92 IF (MFLG(IFLT(IRES)))NE( 6), GO TO 93
000143 CANCEL FUEL CALLED IDEV(IFLT(IRES))
000144 DESTROY FUEL CALLED IDEV(IFLT(IRES))
000145 GO TO 401
000146 93 IF (MFLG(IFLT(IRES)))NE(11), GO TO 94
000147 CANCEL DELAY CALLED IDEV(IFLT(IRES))
000148 DESTROY DELAY CALLED IDEV(IFLT(IRES))
000149 GO TO 401
000150 94 IF (MFLG(IFLT(IRES)))NE(12), GO TO 95
000151 CANCEL READY CALLED IDEV(IFLT(IRES))
000152 DESTROY READY CALLED IDEV(IFLT(IRES))
000153 GO TO 401
000154 95 IF (MFLG(IFLT(IRES)))NE(13), GO TO 96
000155 CANCEL CHEKN CALLED IDEV(IFLT(IRES))
000156 DESTROY CHEKN CALLED IDEV(IFLT(IRES))
000157 GO TO 401
000158 96 WRITE ON TAPE 6, IRES,MFLG(IFLT(IRES))
000159 FORMAT ( ' RESOURCE ',214,' WENT HAYWIRE AT STATEMENT 400 IN SSS',)
000160 401 IF IWAIT(ACASE(IFLT(IRES))) EQ IRES, LET IWAIT(ACASE(IFLT(IRES)))
000161 = 0
000162 GO TO 600
000163 500 IF (MFLG(IFLT(IRES)))NE(9), GO TO 501
000164 IF (MFLG(IFLT(IRES)))NE( 1), GO TO 100
000165 CANCEL ARVSN CALLED IDEV(IFLT(IRES))
000166 DESTROY ARVSN CALLED IDEV(IFLT(IRES))
000167 GO TO 501
000168 100 IF (MFLG(IFLT(IRES)))NE( 3), GO TO 101
000169 CANCEL ARSCH CALLED IDEV(IFLT(IRES))
000170 DESTROY ARSCH CALLED IDEV(IFLT(IRES))

```

```

000171 GO TO 501
000172 IF (MFLG(IFLT(IRES))INE(2), GO TO 102
000173 CANCEL ONSCH CALLED IDEV(IFLT(IRES))
000174 DESTROY ONSCH CALLED IDEV(IFLT(IRES))
000175 GO TO 501
000176
000177 WRITE ON TAPE 6, IPES,MFLG(IFLT(IRES))
000178 FORMAT (' RESOURCE *214,* FENT HAYTYPE AT STATEMENT 500 IN SSS')
000179 LET XR(IRES) = XI(IFLT(IRES))
000180 LET YR(IRES) = YI(IFLT(IRES))
000181 LET IR(IRES) = 1
000182 LET NCASE(IRES) = NCASE(IRES) + 1
000183 LET PRIOR(IRES) = PRI(NOTE)
000184 LET CNRES(ESAC(NOTE)) = CNRES(ESAC(NOTE)) + 1
000185 LET COSTC(ESAC(NOTE)) = COSTC(ESAC(NOTE)) + COST(IRES)
000186 LET ACASE(IFLT(IRES)) = FSAC(NOTE)
000187 LET DEP(IFLT(IRES)) = TIME
000188 LET FITON(IFLT(IRES)) = NOTE
000189 LET XDEST(IFLT(IRES)) = XC(ESAC(NOTE))
000190 LET YDEST(IFLT(IRES)) = YC(ESAC(NOTE))
000191
000192 CREATE AND CAUSE ARSCH FOR SEARCH NEED.
000193
000194 CREATE ARSCH CALLED IDEV(IFLT(IRES))
000195 LET LNOTE(IDEV(IFLT(IRES))) = NOTE
000196 LET LRES(IDEV(IFLT(IRES))) = IRES
000197 LET LFLG(IDEV(IFLT(IRES))) = KLK
000198 LET MFLG(IFLT(IRES)) = 3
000199 LET ROS(IFLT(IRES)) = TIME + TVEC(IRES)
000200 CAUSE ARSCH CALLED IDEV(IFLT(IRES)) AT TIME + TVEC(IRES)
000201 LET NEEDS(STN(IRES)) = NEEDS(STN(IRES)) + 1
000202 RETURN
000203 END

```

```

000001      ENDOGENOUS EVENT READY
000002      C
000003      C
000004      C
000005      C
000006      C
000007      C
000008      C
000009      C
000010      C
000011      C
000012      C
000013      C
000014      C
000015      C
000016      C
000017      C
000018      C
000019      C
000020      C
000021      C
000022      C
000023      C
000024      C
000025      C
000026      C
000027      C
000028      C
000029      C
000030      C

      STORE NREADY(READY) IN NOTE
      STORE RREADY(READY) IN IRS
      STORE FREADY(READY) IN LFG
      DESTROY READY
      LET X = XC(ESAC(NOTE)) - XP(IRS)
      LET Y = YC(ESAC(NOTE)) - YP(IRS)
      LET D = SORT(X*Y*Y)
      IF (SHELL(ESAC(NOTE)))GE(SLIM(TYPE(IRS))), GO TO 10
      LET TVEC(IRS) = D/5001TYPE(IRS)
      GO TO 20
10      LET TVEC(IRS) = D/5002TYPE(IRS)
20      LET DEP(IFLT(IRS)) = TIME
      LET TFLT(IFLT(IRS)) = TIME
      LET MFLG(IFLT(IRS)) = 3
      LET ROS(IFLT(IRS)) = TIME + TVEC(IRS)
      LET XDEST(IFLT(IRS)) = XC(ESAC(NOTE))
      LET YDEST(IFLT(IRS)) = YC(ESAC(NOTE))
      CREATE ARSCH CALLED IDEV(IFLT(IRS))
      LET LNOTE(IDEV(IFLT(IRS))) = NOTE
      LET LRES(IDEV(IFLT(IRS))) = IRS
      LET LFLG(IDEV(IFLT(IRS))) = LFG
      CAUSE ARSCH CALLED IDEV(IFLT(IRS)) AT TIME + TVEC(IRS)
      RETURN
      END

```

```

000001      ENDOGENOUS EVENT ARSCH
000002      C
000003      C ARSCH OCCURS WHEN A RESOURCE ARRIVES ON SCENE TO SERVE A SEARCH
000004      C NEED.
000005      C
000006      STORE LNOTE(ARSCH) IN NOTE
000007      STORE LRES(ARSCH) IN IRS
000008      LET LFG = LFLG(ARSCH)
000009      DESTROY ARSCH
000010      IF (RESA(ESAC(NOTE)))EQ(0), LET RESA(ESAC(NOTE)) = IRS
000011      IF (RESA(NOTE))EQ(0), LET RESA(NOTE) = IRS
000012      IF (ITOL(ESAC(NOTE)))NE(2), GO TO 201
000013      LET X = 0.0
000014      LET I = 0
000015      DO, FOR EACH JJ IN SRHS(ESAC(NOTE))
000016      IF (SIGNL(JJ))NE(2), LET I = 1
000017      LOOP
000018      IF (I)EQ(1), LET X = TIME - TIME(ESAC(NOTE))
000019      LET TQF1(ESAC(NOTE)) = TQF(ESAC(NOTE)) + X
000020      LET NCAS(STN(IRS)) = NCAS(STN(IRS)) + 1
000021      LET TWAIT(ESAC(NOTE)) = TIME - OCCUR(ESAC(NOTE))
000022      LET AVGTW(STN(IRS)) = AVGTW(STN(IRS)) + TWAIT(ESAC(NOTE))
000023      CALL STATSESAC(NOTE),IRS
000024      IF (TWAIT(ESAC(NOTE)))GT(TOL(PRI(ESAC(NOTE)))), GO TO 10
000025      LET ITOL(ESAC(NOTE)) = 1
000026      GO TO 201
000027      10 LET ITOL(ESAC(NOTE)) = 0
000028      LET FAIL3(STN(IRS)) = FAIL3(STN(IRS)) + 1
000029      LET XR(IRS) = XC(ESAC(NOTE))
000030      LET YR(IRS) = YC(ESAC(NOTE))
000031      LET XDEST(IFLT(IRS)) = 0.0
000032      LET YDEST(IFLT(IRS)) = 0.0
000033      LET DEP(IFLT(IRS)) = 0.0
000034      LET ROS(IFLT(IRS)) = TIME
000035      LET XVEC = XS(STN(IRS)) - XC(ESAC(NOTE))
000036      LET YVEC = YS(STN(IRS)) - YC(ESAC(NOTE))
000037      LET VFCTR = SQRT(XVEC*XVEC+YVEC*YVEC)
000038      IF (SMELL(ESAC(NOTE)))GR(SLIMTYPE(IRS)), GO TO 60
000039      LET VFCTR = VECTR/50A1(TYPE(IRS))
000040      GO TO 61
000041      60 LET VECTR = VECTR/50A2(TYPE(IRS))
000042      LET X = END(TYPE(IRS))-(2.0*VECTR)
000043      IF (X)GE(SM(NOTE)/50A3(TYPE(IRS))), GO TO 300
000044      LET TS(NOTE) = X
000045      IF (VOP(TYPE(IRS)))EQ(0), GO TO 100
000046      LET X = AINT(TIME)
000047      LET X = X + SET
000048      IF (X=TIME)LE(0.0), LET X = X + 1.0
000049      IF (TS(NOTE))GR(X-TIME), GO TO 400
000050      CREATE FUEL CALLED IDEV(IFLT(IRS))
000051      LET NFUEL(IDEV(IFLT(IRS))) = NOTE
000052      LET RFUEL(IDEV(IFLT(IRS))) = 1.5
000053      LET MFUG(IFLT(IRS)) = 6
000054      CAUSE FUEL CALLED IDEV(IFLT(IRS)) AT TIME + TS(NOTE)

```

```

000055
000056 LET TS(NOTE) = S8(NOTE)/50A3(TYPE(IRS))
000057 IF (LFS)EQ(1), GO TO 350
000058 IF (VOP(TYPE(IRS)))EQ(0), GO TO 350
000059 LET X = AINT(TIME)
000060 LET X = X + SFT
000061 IF (X-TIME)E(0.0), LET X = X + 1.0
000062 IF (TS(NOTE))GR(X-TIME), GO TO 400
000063 CREATE COMPL CALLED IDEV(IFLT(IRS))
000064 LET MELG(IFLT(IRS)) = 4
000065 LET NCPL(IDEV(IFLT(IRS))) = NOTE
000066 LET RCPL(IDEV(IFLT(IRS))) = IRS
000067 CAUSE COMPL CALLED IDEV(IFLT(IRS)) AT TIME + TS(NOTE)
000068 RETURN
000069 CREATE SSET CALLED IDEV(IFLT(IRS))
000070 LET TS(NOTE) = X - TIME
000071 LET MELG(IFLT(IRS)) = 5
000072 LET ETON(IDEV(IFLT(IRS))) = NOTE
000073 LET SER(IDEV(IFLT(IRS))) = IRS
000074 CAUSE SSET CALLED IDEV(IFLT(IRS)) AT X + 1.0/1440.0
000075 RETURN
000076 END

```

```

000001      ENDOGENOUS EVENT COMPL
000002      C
000003      C      COMPL OCCURS WHEN A LONG SEARCH NEED HAS BEEN COMPLETED.
000004      C
000005      STORE NCMP(LCOMPL) IN NOTE
000006      STORE RCMPL(COMPL) IN IRS
000007      DESTROY COMPL
000008      LET RSR(CNOTE) = 0
000009      LET SIS(ESAC(NOTE)) = SIS(ESAC(NOTE)) - 1
000010      LET IRT(IRS) = 0
000011      LET MFLG(FLUT(IRS)) = 9
000012      LET IVAR = 0
000013      CALL EXQ(IRS,*IVAR)
000014      IF (SIS(ESAC(NOTE)))>0(1), GO TO 200
000015      RETURN
000016      200 LET SIS(ESAC(NOTE)) = -1
000017      LET KJI = MIN(ESAC(NOTE)) + MIN(ESAC(NOTE))
000018      IF (KJI)<0(1), GO TO 400
000019      IF (KJI)>0(1), GO TO 300
000020      CALL SRAS(ESAC(NOTE))
000021      RETURN
000022      300 CALL MRAS(ESAC(NOTE))
000023      RETURN
000024      400 CALL TERM(ESAC(NOTE))
000025      RETURN
000026      END

```



```

000001
C
000002
C
000003
C
000004
C
000005
C
000006
C
000007
C
000008
C
000009
C
000010
C
000011
C
000012
C
000013
C
000014
C
000015
C
000016
C
000017
C
000018
C
END

```

```

000001      ENDGENOUS EVENT FUEL
000002      C
000003      C FUEL OCCURS WHEN A RESOURCE MUST RETURN HOME TO REFUEL. SM(NOTE)
000004      C IS UPDATED AND THE ENDOGENOUS EVENT HOMEF IS CREATED AND CAUSED.
000005      C
000006      STORE NFUEL(FUEL) IN NOTE
000007      STORE RFUEL(FUEL) IN IRS
000008      DESTROY FUEL
000009      LET SM(NOTE) = SM(NOTE) - (TS(NOTE)*SOA3(TYPE(IRS)),
000010      LET X = XR(IRS) - XS(ST(IRS))
000011      LET Y = YR(IRS) - YS(STN(IRS))
000012      LET X = SQRT(X*X+Y*Y)
000013      IF (SMELL(ESAC(NOTE)))GO(SLIMTYPE(IRS)), GO TO 20
000014      LET TVEC(IRS) = X/SOA1(TYPE(IRS))
000015      GO TO 30
000016      20 LET TVEC(IRS) = X/SOA2(TYPE(IRS))
000017      30 LET DEP(IFLT(IRS)) = TIME
000018      LET HCREW(IFLT(IRS)) = 2
000019      LET XDEST(IFLT(IRS)) = XS(STN(IRS))
000020      LET YDEST(IFLT(IRS)) = YS(STN(IRS))
000021      CREATE HOMEF CALLED IDEV(IFLT(IRS))
000022      LET NHOM(IDEV(IFLT(IRS))) = NOTE
000023      LET RHOM(IDEV(IFLT(IRS))) = IRS
000024      LET MFLG(IFLT(IRS)) = 8
000025      CAUSE HOMEF CALLED IDEV(IFLT(IRS)) AT TIME + TVEC(IRS)
000026      RETURN
000027      END

```

```

000001      ENDOGENOUS EVENT HOMEF
000002      C
000003      C HOMEF OCCURS WHEN A RESOURCE ARRIVES AT ITS HOME STATION TO
000004      C REFUEL. ALL ATTRIBUTES ARE THEN UPDATED.
000005      C
000006      STORE NHOM(HOMEF) IN NOTE
000007      STORE RHOM(HOMEF) IN IRS
000008      DESTROY HOMEF
000009      LET XRI(IRS) = XS(STN(IRS))
000010      LET YRI(IRS) = YS(STN(IRS))
000011      LET DEPIFLT(IRS) = 0.0
000012      LET XDEST(IFLT(IRS)) = 0.0
000013      LET YDEST(IFLT(IRS)) = 0.0
000014      CREATE SNDRY CALLED IDEV(IFLT(IRS))
000015      STORE NOTE IN NSDBK(IDEV(IFLT(IRS)))
000016      STORE IRS IN RSDBK(IDEV(IFLT(IRS)))
000017      LET MFLG(IFLT(IRS)) = 10
000018      CAUSE SNDRK CALLED IDEV(IFLT(IRS)) AT TIME + TF(TYPE(IRS))
000019      RETURN
000020      END

```

```

000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054

      ENDOGENOUS EVENT SNORK
C
C      SNORK OCCURS AT THE END OF REFUELING TIME OF A RESOURCE SERVING
C      A LONG SEARCH NEED: SUBROUTINE SASS IS CALLED TO OBTAIN AN ORDERED
C      SET OF RESOURCES; THE BEST IDLE RESOURCE IS CHOSEN TO SERVE THE
C      SEARCH NEED.
C
      STORE NSORK(SNDBK) IN NOTE
      STORE RSORK(SNDBK) IN IRS
      DESTROY SNORK
      LET RSRC(NOTE) = 0
      LET IR(IRS) = 0
      LET EIAT(IRS) = 0.0
      LET TEMP = TFLT(IFLT(IRS))
      IF (TEMP)LS(TLAST), LET TEMP = TLAST
      LET TUTIL(IRS) = TUTIL(IRS) + TIME - TEMP
      DESTROY FLT CALLED IFLT(IRS)
      LET BCRFW(STN(IRS)) = BCRFW(STN(IRS)) - 1
      CALL SASS(ESAC(NOTE),SM(NOTE),NOTE,*KJI)
      GO TO 100, FOR EACH IR IN ROUE
      IF (IR(IRS)GR(0), GO TO 100
      IF (EIAT(IRS)EQ(0.0), GO TO 10
      IF (HCREW(IFLT(IRS))EQ(0), GO TO 20
      GO TO 100
10  CALL ROCA(ESAC(NOTE),IR,*11)
      IF (11)EQ(1), GO TO 20
      GO TO 100
20  LET OVEC(IR) = TVEC(IR)
      IF (EIAT(IRS)EQ(0.0), LET OVEC(IR) = TVEC(IR) + OLAY(TYPE(IRS))
      IF (IR)EQ(IRS), LET OVEC(IR) = TVEC(IR)
      LET SN = AINT(TIME) + RISE - XRK
      IF (TIME)GE(SN), GO TO 1
      IF (TIME+OVEC(IRS))GE(AINT(TIME)+RISE), GO TO 200
      IF (EIAT(IRS)EQ(0.0), LET BCREW(STN(IRS))=BCRFW(STN(IRS))-1
      GO TO 100
1  LET SN = AINT(TIME) + SET
      IF (TIME)GE(SN), GO TO 2
      IF (OVEC(IRS))LS(SN-TIME), GO TO 200
      LET SN = AINT(TIME) + RISE + 1.0
      IF (TIME+OVEC(IRS))GE(SN), GO TO 200
      IF (EIAT(IRS)EQ(0.0), LET BCREW(STN(IRS))=BCREW(STN(IRS))-1
      GO TO 100
2  LET SN = AINT(TIME) + RISE + 1.0
      IF (TIME+OVFC(IRS))GE(SN), GO TO 200
      IF (EIAT(IRS)EQ(0.0), LET BCREW(STN(IRS))=BCREW(STN(IRS))-1
      GO TO 100
200 IF (IR)EQ(IRS), GO TO 300
      LET RSRC(NOTE) = IR
      LET SASG(NOTE) = 1
      CALL SSS(NOTE,IR,3)
      RETURN
300 CREATE FLT CALLED IFLT(IRS)
      STORE ESAC(NOTE) IN ACASE(IFLT(IRS))
      LET DEP(IFLT(IRS)) = TIME

```

```

000055 STORE NOTE IN F100(IFLT(IRS))
000056 LET HCRF(IFLT(IRS)) = F
000057 LET MELG(IFLT(IRS)) = 3
000058 LET ROS(IFLT(IRS)) = TIME + TVEC(IRS)
000059 LET FELT(IFLT(IRS)) = TIME
000060 LET TOA(IFLT(IRS)) = 0
000061 LET XDEST(IFLT(IRS)) = VC(ESAC(NOTE))
000062 LET YDEST(IFLT(IRS)) = VC(ESAC(NOTE))
000063 LET EIAT(IRS) = 1.0
000064 LET IR(IRS) = 1
000065 LET RSRG(NOTE) = IRS
000066 LET SASG(NOTE) = 1
000067 CREATE ARSCH CALLED IDEV(IFLT(IRS))
000068 STORE NOTF IN LNOTE(IDEV(IFLT(IRS)))
000069 STORE IRS IN LRES(IDEV(IFLT(IRS)))
000070 LET LFLG(IDEV(IFLT(IRS))) = 3
000071 CAUSE ARSCH CALLED IDEV(IFLT(IRS)) AT TIME + TVEC(IRS)
000072 RETURN
000073 LOOP
000074 FILE NOTE IN LIST
000075 RETURN
000076 END

```

100

```

000001      ENDogenous EVENT SRISE
000002      C
000003      C SRISE OCCURS AT SUNRISE-XXX; THE SUNRISE LIST IS EXAMINED AND
000004      C RESOURCES ARE VECTORED TO ARRIVE ON THE SEARCH SCENE AT SUNRISE
000005      C OR SOON THEREAFTER.
000006      C
000007      11 IF LIST IS EMPTY, GO TO 999
000008      REMOVE FIRST NOTE FROM LIST
000009      LET I = 0
000010      DO, FOR EACH II IN SRHS(ESAC(NOTE))
000011      IF (SASG(II))EQ(1), LET I = 1
000012      LOOP
000013      IF (I)EQ(0), GO TO 10
000014      DO, FOR EACH II IN SRHS(ESAC(NOTE))
000015      LET SDAY(II) = SDAY(II) + 1
000016      LET SASG(II) = 0
000017      LOOP
000018      10 CALL SASS(ESAC(NOTE),SM(NOTE),NOTE,•IC)
000019      LET JRAP = RAP
000020      IF (RAP)EQ(4), LET JRAP = 2
000021      IF (RAP)EQ(5), LET JRAP = 6
000022      CALL RESAR(ESAC(NOTE),JRAP,•IRES)
000023      IF (IRES)NE(0), GO TO 20
000024      LET NQUE(ESAC(NOTE)) = NQUE(ESAC(NOTE)) + 1
000025      IF (REA(ESAC(NOTE)))EQ(2), LET REA(ESAC(NOTE)) = 1
000026      LET I = 0
000027      DO, FOR EVERY JJ IN SRHS(ESAC(NOTE))
000028      IF (ISIGNL(JJ))NE(2), LET I = 1
000029      LOOP
000030      IF (I)EQ(0), LET TINQ(ESAC(NOTE)) = TIME
000031      LET SIGNL(NOTE) = 0
000032      15 FILE NOTE IN CQUE
000033      GO TO 11
000034      20 LET SASG(NOTE) = 1
000035      LET RSRC(NOTE) = IRES
000036      LET II = 0
000037      DO TO 30, FOR EACH I IN SRHS(ESAC(NOTE))
000038      IF SFLAG(I) EQ 1, LET II = 1
000039      LOOP
000040      IF (II)EQ(0), LET SFLAG(NOTE) = 1
000041      IF (SFLAG(NOTE))EQ(1), GO TO 40
000042      CALL SSS(NOTE,IRES,2)
000043      GO TO 11
000044      40 CALL SSS(NOTE,IRES,1)
000045      GO TO 11
000046      999 CAUSE SRISE AT TIME + ).0
000047      RETURN
000048      END

```

```

000001      ENDOGENOUS EVENT XSET
000002      C
000003      C XSET OCCURS AT SUNSET EACH DAY. IT PLACES QUEUED *LANG* SEARCH
000004      C CASES IN THE SUNRISE LIST--FIFO SET LIST
000005      C
000006      IF QUEUE IS EMPTY, GO TO 20
000007      DO TO 10, FOR EACH I IN QUEUE
000008      IF (FLG(I))NE(3), GO TO 10
000009      REMOVE I FROM QUEUE
000010      LET IK = SIGNL(I)
000011      LET SIGNL(I) = 2
000012      LET JIND = 2
000013      LET KIND = 2
000014      DO, FOR EACH J IN SRHS(PSAC(I))
000015      IF (SIGNL(J))EQ(0), LET JIND = 1
000016      IF (SIGNL(J))EQ(1), LET KIND = 1
000017      LOOP
000018      IF (IK)EQ(1), GO TO 50
000019      IF (JIND)EQ(1), GO TO 5
000020      IF (KIND)EQ(1), GO TO 5
000021      LET TQUE(ESAC(I)) = TQUE(ESAC(I)) + (TIME-TING(ESAC(I)))
000022      GO TO 5
000023      50 IF (KIND)EQ(1), GO TO 5
000024      IF (JIND)EQ(2), GO TO 60
000025      LET TINT(ESAC(I)) = TINT(ESAC(I)) + (TIME-STING(ESAC(I)))
000026      GO TO 5
000027      60 LET TQUE(ESAC(I)) = TQUE(ESAC(I)) + (TIME-TING(ESAC(I)))
000028      LET TINT(ESAC(I)) = TINT(ESAC(I)) + (TIME-STING(ESAC(I)))
000029      5 FILE I IN LIST
000030      10 LOOP
000031      20 CAUSE XSET AT TIME + 1.0
000032      RETURN
000033      END

```

```

000001 SUBROUTINE SERVE (CASE,I,OTF,IRS)
000002 LET COSTC(CASE) = COSTC(CASE) + COST(IRS)
000003 LET CNRES(CASE) = CNRES(CASE) + 1
000004 LET NEEDS(STN(IRS)) = NEEDS(STN(IRS)) + 1
000005 LET NCASE(IRS) = NCASE(IRS) + 1
000006 158 LET IB(IRS) = 1
000007 IF INOTF NE 0, GO TO 50
000008 LET PRIOR(IRS) = PRI(CASE)
000009 GO TO 60
000010 50 LET PRIOR(IRS) = PRI(INOTF)
000011 LET KRES(INOTF) = IRS
000012 60 IF EIAT(IRS) NE 0, GO TO (100,102,104,106,108,110,112,114,140,
000013 • 140,116,118,120),MFLG(IFLT(IRS))
000014 LET FIAT(IRS) = 99999.
000015 CREATE FLT CALLED IFLT(IRS)
000016 LET IFLT(IFLT(IRS)) = 0.0
000017 LET Y0(IFLT(IRS)) = 0
000018 LET HCREW(IFLT(IRS)) = 0
000019 LET ACASE(IFLT(IRS)) = CASE
000020 LET FITON(IFLT(IRS)) = INOTF
000021 LET ROS(IFLT(IRS)) = TIME + TVEC(IRS) + DLAY(TYPE(IRS))
000022 IF INOTF NE 0, LET RLS(IFLT(IRS)) = ROS(IFLT(IRS)) + OST(INOTF)
000023 IF INOTF EQ 0, LET RLS(IFLT(IRS)) = ROS(IFLT(IRS)) + OST(CASE)
000024 IF IRS EQ -152(CASE), LET RLS(IFLT(IRS)) = ROS(IFLT(IRS)) +
000025 • TSM(CASE)/SDA3(TYPE(IRS))
000026 LET DEP(IFLT(IRS)) = 0.
000027 CREATE DELAY CALLED IDEV,IFLT(IRS))
000028 STORE CASE IN CASNO(IDEV(IFLT(IRS)))
000029 STORE IRS IN RESNO(IDEV(IFLT(IRS)))
000030 CAUSE DELAY CALLED IDEV(IFLT(IRS)) AT TIME + DLAY(TYPE(IRS))
000031 LET MFLG(IFLT(IRS)) = 11
000032 RETURN
000033 100 CANCEL ARVSN CALLED IDEV,IFLT(IRS))
000034 LET XR(IRS) = XI(IFLT(IRS))
000035 LET YR(IRS) = YI(IFLT(IRS))
000036 GO TO 145
000037 102 CANCEL ONSCN CALLED IDEV(IFLT(IRS))
000038 DESTROY ONSCN CALLED IDEV(IFLT(IRS))
000039 GO TO 140
000040 104 CANCEL ARSCH CALLED IDEV(IFLT(IRS))
000041 DESTROY ARSCH CALLED IDEV(IFLT(IRS))
000042 LET XR(IRS) = XI(IFLT(IRS))
000043 LET YR(IRS) = YI(IFLT(IRS))
000044 GO TO 140
000045 106 CANCEL COMPL CALLED IDEV(IFLT(IRS))
000046 DESTROY COMPL CALLED IDEV(IFLT(IRS))
000047 GO TO 140
000048 108 CANCEL SSET CALLED IDEV(IFLT(IRS))
000049 DESTROY SSET CALLED IDEV(IFLT(IRS))
000050 GO TO 140
000051 110 CANCEL FUEL CALLED IDEV(IFLT(IRS))
000052 DESTROY FUEL CALLED IDEV(IFLT(IRS))
000053 GO TO 140
000054 112 CANCEL HOME CALLED IDEV(IFLT(IRS))

```

851100
 850700
 850800
 85082
 85560
 85600
 85570
 85130
 85160
 85180
 85190
 85200
 85210
 85220
 85230
 85240
 85250
 85260
 85270
 85280
 85290
 85300
 85310
 85320
 85330
 85340
 85350


```

000055 DESTROY HOME CALLED IDEV(IFLT(IRS))
000056 LET XR(IRS) = XI(IFLT(IRS))
000057 LET YS(IRS) = YI(IFLT(IRS))
000058 GO TO 140
000059
000060 114 CANCEL HOME# CALLED IDEV(IFLT(IRS))
000061 DESTROY HOME# CALLED IDEV(IFLT(IRS))
000062 GO TO 140
000063
000064 116 CANCEL DELAY CALLED IDEV(IFLT(IRS))
000065 DESTROY DELAY CALLED IDEV(IFLT(IRS))
000066 GO TO 140
000067
000068 118 CANCEL READY CALLED IDEV(IFLT(IRS))
000069 DESTROY READY CALLED IDEV(IFLT(IRS))
000070 GO TO 140
000071
000072 120 CANCEL CHEXN CALLED IDEV(IFLT(IRS))
000073 DESTROY CHEXN CALLED IDEV(IFLT(IRS))
000074 140 IF IWAIT(ACASE(IFLT(IRS))) EQ IRS, LET IWAIT(ACASE(IFLT(IRS))) =
000075
000076 145 LET ACASE(IFLT(IRS)) = CASE
000077 LET DEP(IFLT(IRS)) = TIME
000078
000079 155 LET XOE(IFLT(IRS)) = X(CASE)
000080 LET YOE(IFLT(IRS)) = Y(CASE)
000081 LET TRLS = TIME + TVEC(IRS)
000082 LET ROS(IFLT(IRS)) = TRLS
000083 IF INOTF NE 0, GO TO 160
000084 LET TRLS = TIME + TVEC(IRS) + OST(CASE)
000085 LET RLS(IFLT(IRS)) = TRLS
000086 IF IRS EQ -152(CASE), LET RLS(IFLT(IRS)) = ROS(IFLT(IRS)) +
000087 *TSM(CASE)/50A3(TYPE(IRS))
000088 LET FITON(IFLT(IRS)) = 0
000089 GO TO 165
000090
000091 160 LET RLS(IFLT(IRS)) = ROS(IFLT(IRS)) + OST(INOTF)
000092 LET FITON(IFLT(IRS)) = INOTF
000093
000094 165 STORE CASE IN CASNO(IDEV(IFLT(IRS)))
000095 STORE IRS IN RESNO(IDEV(IFLT(IRS)))
000096 CAUSE ARSN CALLED IOFV(IFLT(IRS)) AT TIME + TVEC(IRS)
000097 LET MFLG(IFLT(IRS)) = 1
000098 RETURN
000099 END
000100

```

85360

85370
85380
85390

85490
85500
85510
85520
85540
85550

85620
85630
85640
85650

85680
85690
85700

85740
85750
85760

```

000001      ENDOGENOUS EVENT DELAY
000002      STORE CASNO(DELAY) IN CASE
000003      STORE RESNO(DELAY) IN IRS
000004      DESTROY DELAY CALLED IDEV(IFLT(IRS))
000005      LET XDEST(IFLT(IRS)) = XC(CASE)
000006      LET YDEST(IFLT(IRS)) = YC(CASE)
000007      LET DEP(IFLT(IRS)) = TIME
000008      LET TFLT(IFLT(IRS)) = TIME
000009      LET TEMP = ROS(IFLT(IRS))
000010      IF TEMP LT TIME, LET TEMP = TIME
000011      CREATE ARVSN CALLED IDEV(IFLT(IRS))
000012      STORE CASE IN CASNO(IDEV(IFLT(IRS)))
000013      STORE IRS IN RESNO(IDEV(IFLT(IRS)))
000014      CAUSE ARVSN CALLED IDEV(IFLT(IRS)) AT TEMP
000015      LET MFLG(IFLT(IRS)) = 1
000016      RETURN
000017      END OF DELAY

```

```

000001      FNOGENOUS EVENT ARVSN
000002      STORE CASNO(ARVSN) IN CASE
000003      STORE RESNO(APVSN) IN IRS
000004      LET INOTF = FITON(IFLT(IRS))
000005      LET KFLAG = 0
000006      DESTROY ARVSN CALLED IDEV(IFLT(IRS))
000007      999 IF INOTF EQ 0, GO TO 12
000008      IF RESA(INOTF) EQ 0, LET RESA(INOTF) = IRS
000009      12 IF ITOL(CASE) NE 2, GO TO 50
000010      LET RESA(CASE) = IRS
000011      LET TWAIT(CASE) = TIME - OCCUR(CASE)
000012      LET NCAS(STN(IRS)) = NCAS(STN(IRS)) + 1
000013      LET AVGT#(STN(IRS)) = AVGT#(STN(IRS)) + TWAIT(CASE)
000014      CALL STATIS(CASE,IRS)
000015      IF INOTF EQ 0, GO TO 35
000016      IF NSET(CASE) IS EMPTY, GO TO 35
000017      00 TO 30, FOR EACH NOTIF IN NSET(CASE)
000018      IF SIGNL(NOTIF) NE 2, GO TO 40
000019      30 LOOP
000020      35 LET TQUE1(CASE) = TQUE(CASE)
000021      GO TO 42
000022      40 LET TQUE1(CASE) = TQUE(CASE) + TIME-TINQ(CASE)
000023      42 IF TWAIT(CASE) LE TOL(PRI(CASE)), GO TO 45
000024      LET FAIL3(STN(IRS)) = FAIL3(STN(IRS)) + 1
000025      LET ITOL(CASE) = 0
000026      GO TO 50
000027      45 LET ITOL(CASE) = 1
000028      50 LET OEP(IFLT(IRS)) = 0.0
000029      LET XR(IRS) = XDEST(IFLT(IRS))
000030      LET YR(IRS) = YDEST(IFLT(IRS))
000031      LET XDEST(IFLT(IRS)) = 0.
000032      LET YDEST(IFLT(IRS)) = 0.
000033      IF IWAIT(CASE) EQ 0, GO TO 300
000034      IF INOTF EQ 0, GO TO 301
000035      LET IR = IWAIT(CASE)
000036      IF NUMR(INOTF) NE 0, GO TO 299
000037      IF RLS(IFLT(IRS)) LS RLS(IFLT(IRS)), GO TO 301
000038      299 IF IR EQ IRS, GO TO 301
000039      LET IR(IRS) = 1
000040      IF MFLG(IFLT(IRS)) EQ 2, GO TO 303
000041      IF MFLG(IFLT(IRS)) NE 13, GO TO 298
000042      CANCEL CHEKN CALLED IDEV(IFLT(IRS))
000043      DESTROY CHEKN CALLED IDEV(IFLT(IRS))
000044      LET MFLG(IFLT(IRS)) = 9
000045      298 CALL RETN(CASE,IR)
000046      300 LET IWAIT(CASE) = IRS
000047      LET IR(IRS) = 2
000048      301 LET 0 = 0.
000049      302 IF MM(CASE) + NNN(CASE) GR 1, GO TO 310
000050      C SINGLE RESOURCE CASE
000051      IF MM(CASE) NE 0, GO TO 330
000052      C SINGLE RESOURCE NON-TOP CASE
000053      GO TO 315
000054      C MULTI-RESOURCE CASE

```

```

000055 310 IF NUMBR(INOTF) NE 0, GO TO 370
000056 C MULTI-RESOURCE, NON-TOW CASE
000057 315 LET JFLAG = 1
000058 GO TO 600
000059 316 IF KFLAG EQ 1, GO TO 317
000060 CREATE ONSCN CALLED IDEV(IFLT(IRS))
000061 STORE CASE IN CASNO(IDEV(IFLT(IRS)))
000062 STORF IRS IN RESNO(IDEV(IFLT(IRS)))
000063 CAUSE ONSCN CALLED IDEV(IFLT(IRS)) AT TIME
000064 317 LET MLG(IFLT(IRS)) = 2
000065 IF INOTF GR 7, LET COMPT(INOTF) = 2
000066 RETURN
000067 C SINGLE RESOURCE CASE TOW OR ESCORT
000068 330 CALL DTD(CASE,XDEST(IFLT(IRS)),YOEST(IFLT(IRS)),*0)
000069 350 IF NEED(CASE) NE 17, GO TO 360
000070 C SINGLE RESOURCE - AIR ESCORT
000071 LEFT OST(CASE) = O/SOAI(TYPE(IRS))
000072 LET TOWSP(IFLT(IRS)) = SOAI(TYPE(IRS))
000073 LET DEP(IFLT(IRS)) = TIME
000074 GO TO 367
000075 C SINGLE RESOURCE CASE TOW - NOT AIR ESCORT
000076 360 LET DEP(IFLT(IRS)) = TIME + THOOK
000077 IF L(CASE) LE 26, GO TO 365
000078 LET OST(CASE) = D/TSP2+THOOK
000079 LET TOWSP(IFLT(IRS)) = TSP2
000080 GO TO 367
000081 365 LET OST(CASE) = D/TSP1+THOOK
000082 LET TOWSP(IFLT(IRS)) = TSP1
000083 367 LET JFLAG = 2
000084 GO TO 600
000085 368 LET TRLS = TIME + OST(CASE)
000086 LET RLS(IFLT(IRS)) = TRLS
000087 GO TO 500
000088 C MULTI-RESOURCE TOW OR ESCORT
000089 370 LET O = SQRT((XC(CASE)-XHANO(INOTF))*2+(YC(CASE)-YHAND(INOTF))
000090 **2)
000091 LET XDEST(IFLT(IRS)) = XHANO(INOTF)
000092 LET YDEST(IFLT(IRS)) = YHANO(INOTF)
000093 IF NEED(INOTF) NE 17, GO TO 390
000094 LET OST(INOTF) = O/SOAI(TYPE(IRS))
000095 LET TOWSP(IFLT(IRS)) = SOAI(TYPE(IRS))
000096 LET DEP(IFLT(IRS)) = TIME
000097 GO TO 395
000098 390 LET DEP(IFLT(IRS)) = TIME + THOOK
000099 IF L(CASE) LE 26, GO TO 392
000100 LET OST(INOTF) = O/TSP2 + THOOK
000101 LET TOWSP(IFLT(IRS)) = TSP2
000102 GO TO 395
000103 392 LET OST(INOTF) = O/TSP1 + THOOK
000104 LET TOWSP(IFLT(IRS)) = TSP1
000105 395 LET JFLAG = 3
000106 GO TO 600
000107 396 LET TRLS = TIME + OST(INOTF)
000108 LET RLS(IFLT(IRS)) = TRLS
000109 500 IF KFLAG EQ 1, GO TO 515
000110 CREATE ONSCN CALLED IDEV(IFLT(IRS))
000111 STORE CASE IN CASNO(IDEV(IFLT(IRS)))
000112 STORF IRS IN RESNO(IDEV(IFLT(IRS)))

```

```

000113      CAUSE ONSCN CALLED IDEV(IFLT(IRS)) AT TRLS
000114      515 LET TOM(IFLT(IRS)) = 1
000115      LET KFLS(IFLT(IRS)) = 2
000116      IF INOTF GR 0, LET COMP(INOTF) = 2
000117      RETURN
000118      600 IF S2S(CASE) EQ 0, GO TO 650
000119      IF LOC(CASE) EQ 1, GO TO 450
000120
000121      C      A SHORT SEARCH IS NEEDED TO LOCATE THE CASE.
000122      C
000123      CALL SRCHF(CASE, INOTF, IRC)
000124      LET PRI(CASE) = S2PRI
000125      IF INOTF EQ 0, GO TO 70
000126      DO TO 60, FOR EACH N IN NSET(CASE)
000127      IF KPRES(N) NE 0, LET PRIOR(KRES(N)) = PRI(CASE)
000128      LET PRI(N) = PRI(CASE)
000129      60 LOOP
000130      GO TO 100
000131      70 LET PRIOR(IRS) = PRI(CASE)
000132      100 LET LOC(CASE) = 1
000133      IF IRS NE IS2(CASE), GO TO 650
000134      LET KFLAG = 1
000135      CREATE UNSCN CALLED IDEV(IFLT(IRS))
000136      STORE CASE IN CASNO(IDEV(IFLT(IRS)))
000137      STORE IRS IN RESNO(IDEV(IFLT(IRS)))
000138      LET TRLS = TIME+TSM(CASE)/SOA3(TYPE(IRS))
000139      CAUSE ONSCN CALLED IDEV(IFLT(IRS)) AT TRLS
000140      450 GO TO (316,368,396),JFLAG
000141      END

```

```

91080
91090
91100

90280
90290
90300

90310
90320
90330
90340
90350
90360

90380
90420

91110

```

```

000001 SUBROUTINE STATS(CASF,IRS)
000002 C
000003 C SUBROUTINE STATS COLLECTS CASE STATISTICS, STATISTICS
000004 C AND GROUP STATISTICS WHEN THE FIRST RESOURCE ARRIVES AT THE
000005 C SCENF OF A CASE.
000006 C
000007 LET DMERT(CASE) = 0.0
000008 LET X = TWAIT(CASE) - TOL(PRI(CASE))
000009 IF (X)LE(0.0), GO TO 10
000010 LET TMTOL(STN(IRS)) = T*TOL(STN(IRS)) + X
000011 LET MNTMT = MNTMT + X
000012 LET TMTAV(GRP(STN(IRS))) = TMTAV(GRP(STN(IRS))) + X
000013 IF PRI(CASE) GR 2, GO TO 40
000014 IF OFSHR(CASE) LE 20.0, LET BQ = 2.0
000015 IF OFSHR(CASE) GR 20.0, LET BQ = 1.0
000016 GO TO 41
000017 40 IF OFSHR(CASE) LE 20.0, LET BQ = 3.0
000018 IF OFSHR(CASE) GR 20.0, LET BQ = 2.0
000019 41 LET DMERT(CASE) = BQ * X
000020 LET DMRT(STN(IRS)) = DMRT(STN(IRS)) + DMERT(CASE)
000021 LET MEAND = MEAND + DMERT(CASE)
000022 LET AVDRT(GRP(STN(IRS))) = AVDRT(GRP(STN(IRS))) + DMERT(CASE)
000023 LET TMAVG(GRP(STN(IRS))) = TMAVG(GRP(STN(IRS))) + TWAIT(CASE)
000024 C
000025 C CALCULATE TVEC(IRS) TO COLLECT STATISTICS ON IT.
000026 C
000027 LET D = XC(CASE) - XR(IRS)
000028 LET Y = YC(CASE) - YR(IRS)
000029 LET D = SQRT(D*D+Y*Y)
000030 IF (SWELL(CASE))GR(SLIM(TYPE(IRS))), GO TO 20
000031 LET TVEC(IRS) = D/50AL(TYPE(IRS))
000032 GO TO 30
000033 20 LET TVEC(IRS) = D/50A2(TYPE(IRS))
000034 30 LET VCTR(STN(IRS)) = VCTR(STN(IRS)) + TVEC(IRS)
000035 LET MEANV = MEANV + TVEC(IRS)
000036 LET TMAVG(GRP(STN(IRS))) = TMAVG(GRP(STN(IRS))) + TVEC(IRS)
000037 C
000038 C DETERMINE IF IT IS A WEEKDAY OR WEEKEND.
000039 C
000040 LET I = 3
000041 IF (PSHFT)GR(NWD), LET I = 4
000042 LET MEEN(I) = MEEN(I) + TWAIT(CASE)
000043 LET MEEN(I+2) = MEEN(I+2) + TVEC(IRS)
000044 LET MEEN(I+4) = MEEN(I+4) + X
000045 IF X GR 0.0, LET MEEN(I+6) = MEEN(I+6) + X
000046 LET CNTR(I) = CNTR(I) + 1
000047 LET CNTR(I+2) = CNTR(I+2) + 1
000048 LET CNTR(I+4) = CNTR(I+4) + 1
000049 LET CNTR(I+6) = CNTR(I+6) + 1
000050 LET STDEV(I) = STDEV(I)+(TWAIT(CASE)*TWAIT(CASE))
000051 LET STDEV(I+2)=STDEV(I+2)+(TVEC(IRS)*TVEC(IRS))
000052 LET STDEV(I+4) = STDEV(I+4) + (X*X)
000053 IF X GR 0.0, LET STDEV(I+6) = STDEV(I+6) + (X*X)
000054 LET X = 0

```

```

000055      GO TO 100, F09 J=(1)(4)
000056      IF (JJEQ(1), LET Y = T*AT(CASE) * 24.0
000057      IF (JJEQ(2), LET Y = TVFC(IRS) * 24.0
000058      IF (JJEQ(3), LET Y = X*24.0
000059      IF (JJE(2), GO TO 101
000060      IF (Y)LE(0.1), GO TO 100
000061      IF Y GR 0.5, GO TO 102
000062      LET CATG1(I*K) = CATG1(I*K) + 1
000063      GO TO 99
000064      101 IF Y GR 1.0, GO TO 103
000065      LET CATG2(I*K) = CATG2(I*K) + 1
000066      GO TO 99
000067      102 IF Y GR 2.0, GO TO 104
000068      LET CATG3(I*K) = CATG3(I*K) + 1
000069      GO TO 99
000070      103 IF Y GR 3.0, GO TO 105
000071      LET CATG4(I*K) = CATG4(I*K) + 1
000072      GO TO 99
000073      104 IF Y GR 4.0, GO TO 106
000074      LET CATG5(I*K) = CATG5(I*K) + 1
000075      GO TO 99
000076      105 IF Y GR 5.0, GO TO 107
000077      LET CATG6(I*K) = CATG6(I*K) + 1
000078      GO TO 99
000079      106 IF Y GR 10.0, GO TO 108
000080      LET CATG7(I*K) = CATG7(I*K) + 1
000081      GO TO 99
000082      107 LET CATG8(I*K) = CATG8(I*K) + 1
000083      99 LET K = K + 2
000084      100 LOOP
000085      RETURN
000086      END

```

```

000001 SURROUTINE SRCHF(CASE,INOTF,IRSC)
000002 IF SPS(CASE) EQ 1, GO TO 200
000003 15 LET SMIL = TSM(CASE)
000004 CALL SASS(CASE,SMIL,IRUN,IRND)
000005 IF IND NE 0, GO TO 200
000006 50 LET IRAP = RAP
000007 IF RAP EQ 4, LET IRAP = 5
000008 IF RAP EQ 5, LET IRAP = 6
000009 CALL RESAP(CASE,IRAP,IRJRS)
000010 IF JRS EQ 0, GO TO 200
000011 100 LET IS2(CASE) = -JRS
000012 IF MMM(CASE) + NNN(CASE) LE 1, GO TO 150
000013 CREATE NOTIF
000014 FILE NOTIF IN NSET(CASE)
000015 LFT COMP(NOTIF) = 2
000016 LET KRES(NOTIF) = JRS
000017 LET PRI(NOTIF) = PRI(CASE)
000018 LET CAS(NOTIF) = CASE
000019 LET XHAND(NOTIF) = 0
000020 LET YHAND(NOTIF) = 0
000021 LET FEED(NOTIF) = IR
000022 LET RESA(NOTIF) = 0
000023 LET NUMBR(NOTIF) = 0
000024 LET FLG(NOTIF) = 2
000025 LET SIGNL(NOTIF) = 2
000026 LET OST(NOTIF) = TSM(CASE)/SOA3(TYPE(JRS))
000027 LET OST(INOTF)=OST(INOTF) + DVEC(JRS) + TSM(CASE)/SOA3(TYPE(JRS))
000028 IF NUMBR(INOTF) NE 0, LET DEP(IFLT(IRS)) = DEP(IFLT(IRS)) +
000029 * DVEC(JRS) + OST(NOTIF)
000030 LET RLS(IFLT(IRS)) = TIME + OST(INOTF)
000031 CALL SERVE (CASE,NOTIF,JRS)
000032 IF RLS(IFLT(IRS)) LE RLS(IFLT(JRS)), LET RLS(IFLT(IRS)) =
000033 *RLS(IFLT(IRS)) + 2.0/1440.0
000034 LET PRIOR(JRS) = S2PRI
000035 999 RETURN
000036 150 CALL SERVE(CASE,0,JRS)
000037 LET PRIOR(JRS) = S2PRI
000038 IF MMM(CASE) NE 0, LET DEP(IFLT(IRS)) = DEP(IFLT(IRS)) +
000039 * DVEC(JRS) + TSM(CASE)/SOA3(TYPE(JRS))
000040 LET TRLS = TIME + OST(CASE) + DVEC(JRS)+TSM(CASE)/SOA3(TYPE(JRS))
000041 LET RLS(IFLT(IRS)) = TRLS
000042 LET OST(CASE) = OST(CASE) + TSM(CASE)/SOA3(TYPE(JRS)) + DVEC(JRS)
000043 RETURN
000044 C
000045 C SEARCH IS DONE BY THE FIRST RESOURCE TO ARRIVE ON SCENE
000046 C
000047 200 LET IS2(CASE) = IPS
000048 IF MMM(CASE) + NNN(CASE) LE 1, GO TO 250
000049 LET OST(INOTF) = OST(INOTF) + TSM(CASE)/SOA3(TYPE(IRS))
000050 LET TRLS = TIME + OST(INOTF)
000051 LET RLS(IFLT(IRS)) = TRLS
000052 IF NUMBR(INOTF) NE 0, LET DEP(IFLT(IRS)) = DEP(IFLT(IRS)) +
000053 * TSM(CASE)/SOA3(TYPE(IRS))
000054 RETURN

```


000055	250 LET OST(CASE) = OST(CASE) + TSM(CASE)/SOA3(TYPE(INS))	110470
000056	LET TRLS = TIME + OST(CASE)	
000057	LET RLS(IFLT(IRS)) = TRLS	
000058	IF NMP(CASE) <= 0, LET DEP(IFLT(IRS)) = DEF(IFLT(INS)) +	
000059	* TSM(CASE)/SOA3(TYPE(IRS))	110480
000060	RETURN	110490
000061	END	


```

000055 IF COUNT(CASE) EQ MM(CASE) + MM(CASE), LET IR(IRS) = I
000056 IF MM(CASE) EQ 0, GO TO 100
000057 LET NI = COUNT(CASE)-MM(CASE) + 1
000058 IF NI LE 0, GO TO 103
000059 IF NI EQ 1, GO TO 61
000060 LET XC(CASE) = XDEST(IFL*(IRS))
000061 LET YC(CASE) = YDEST(IFL*(IRS))
000062 LET X(IRS) = XDEST(IFL(IRS))
000063 LET Y(IRS) = YDEST(IFL(IRS))
000064 LET XDEST(IFL(IRS)) = J,
000065 LET YDEST(IFL(IRS)) = J,
000066
000067 61 IF NI GR MM(CASE), GO TO 100
000068 DO TO 70, FOR EACH NOTIF IN NSFT(CASE)
000069 IF NUMR(NOTIF) NE NI, GO TO 70
000070 IF NI NE 1, GO TO 65
000071 CALL CRES(CASE,NEED(NOTIF),*KNT,*II)
000072 IF II EQ 0, GO TO 69
000073 IF KNT EQ 0, GO TO 68
000074 IF RQUE IS EMPTY, GO TO 48
000075 DO TO 62, FOR EACH I IN RQUE
000076 IF I NE IRS, GO TO 62
000077 LET TVEC(IRS) = 0,
000078 LET COST(IRS) = 0,
000079 LET IR(IRS) = I
000080 LET KRES(INOTF) = 0
000081 CALL SERVE(CASE,NOTIF,IRS)
000082 RETURN
000083 62 LOOP
000084 GO TO 68
000085 65 LET OFSHR(CASE) = *25
000086 68 CAUSE NOTIF AT TIME
000087 GO TO 100
000088 70 LOOP
000089 100 IF IR(IRS) NE 2, GO TO 148
000090 IF CQUE IS EMPTY, GO TO 900
000091 DO TO 130, FOR EACH ICNN IN CQUE
000092 IF FLG(ICNN) NE 2, GO TO 130
000093 IF CAS(ICNN) NE CASE, GO TO 130
000094 CALL CRES(CASE,NEED(ICNN),*KONTR,*II)
000095 IF II EQ 0, GO TO 120
000096 IF KONTR EQ 0, GO TO 120
000097 DO, FOR EACH IRES OF RQUE
000098 IF IRES EQ IRS, GO TO 105
000099 LOOP
000100 GO TO 130
000101 105 LET TVEC(IRS) = 0,
000102 LET KRES(IFTON(IFL(1>S))) = 0
000103 CALL SVQUE(ICNN,IRS)
000104 RETURN
000105 120 LET ITOL(CASE) = 7
000106 CALL ARECK(ICNN)
000107 130 LOOP
000108 900 CREATE CHEKN CALLED IDEV(IFL(1>S))
000109 STORE IRS IN RESNO(IDEV(IFL(IRS)))
000110 CAUSE CHEKN CALLED IDEV(IFL(IRS)) AT TIME + TCHEK
000111 LET MFLG(IFL(IRS)) = 13
000112 RETURN
000113 148 CALL RETN(CASE,IRS)

```

000113
000114

150 RETURN
END

95710
95720

```

000001 SUBROUTINE COVER (CASE,IRES)
000002 IF MMN(CASE) + MAX(CASE) EQ 1, GO TO 900
000003 LET IM = 0
000004 LET XRLS = 0.
000005 DO TO 100, FOR EACH NOT1, IN NSET(CASE)
000006 LET IR = KRES(NOT1P)
000007 IF IR EQ 0, GO TO 100
000008 IF MFLG(FLT(IR)) NE 2, GO TO 100
000009 IF IR EQ IRES, GO TO 100
000010 IF PLS(FLT(IR)) LE XRLS, GO TO 100
000011 LET XRLS = RLS(FLT(IR))
000012 LET IM = IR
000013 100 LOOP
000014 IF IR EQ 0, GO TO 900
000015 LET IWAIT(CASE) = IM
000016 LET IR(1) = 2
000017 RETURN
000018 900 LET IWAIT(CASE) = 0
000019 RETURN
000020 END

```

```

180001
180010
180020
190030
180040
180050
180060
180070
180080
180090
180100
180110
180120
180130
180140
180150
180160
180170
180180
180190

```

```

000001      EXOGENOUS EVENT CHECK
000002      STORE RESNO(CHEKN) IN IRS
000003      LET ^FLG(IFLT(IRS)) = 9
000004      DESTROY CHECK CALLID IDEV(IFLT(IRS))
000005      LET ICK = 1
000006      CALL EX2(IRS,**ICK)
000007      IF ICK EQ 1, GO TO 10
000008      RETURN
000009      10 CREATE CHEKN CALLED IDEV(IFLT(IRS))
000010      STORE IRS IN RESNO(IDEV(IFLT(IRS)))
000011      CAUSE CHEKN CALLED IDEV(IFLT(IRS)) AT TIME + TCHEK
000012      LET ^FLG(IFLT(IRS)) = 13
000013      RETURN
000014      END

```

```

000001 SUBROUTINE RETN(CASE,IRS,
000002 999 LET INOTF = F1TON(IFLT(IRS))
000003 1 IF I9(IRS) EQ 2, RETURN
000004 LET I9(IRS) = 0
000005 IF INOTF EQ 0, GO TO 20
000006 LET KRES(INOTF) = 0
000007 LET COMB(INOTF) = 3
000008 20 LET IVAR = 0
000009 CALL EX2(IRS,**IVAR)
000010 IF COUNT(CASE) LT MMM(CASE)+NN1(CASE), GO TO 71
000011 CALL TERN(CASE)
000012 RETURN
000013 21 IF IRS VE -1S2(CASE), RETURN
000014 LET S2S(CASE) = -S2S(CASE)
000015 LET PRI(CASE) = FPRI(CASE)
000016 IF MMM(CASE) + VNN(CASE) GR 1, GO TO 30
000017 22 DO TO 25, FOR I=(1)INRES)
000018 IF (IFLT(I)) EQ 0, GO TO 25
000019 IF ACASE(IFLT(I)) NE CASE, GO TO 25
000020 IF I EQ IRS, GO TO 25
000021 LET PRIOR(I) = FPRI(CASE)
000022 LET 1S2(CASE) = 0
000023 RETURN
000024 25 LOOP
000025 RETURN
000026 30 DO TO 35, FOR EACH N IN NSET(CASE)
000027 LET PRI(N) = PRI(CASE)
000028 IF KRES(N) NE 0, LET PRIOR(KRES(N)) = PRI(CASE)
000029 35 LOOP
000030 LET 1S2(CASE) = 0
000031 40 RETURN
000032 END
100003
100070
100102
100110
100180
100190
100200
100220
100230
100270
100280
100290
100300
100310
100320
100330
100370
100380
100390
100400
100410
100420
100430

```

```

000001 SUBROUTINE TERM(CASE)
000002 DIMENSION IPUFF(15),FRUFF(10)
000003 LET AMRQ(STATN(CASE)) = AMRQ(STATN(CASE)) + NQUE(C,SE)
000004 LET TSVC(CASE) = TIME-OCCUR(CASE)
000005 LET I = 1
000006 IF (PSHT)GR(NRD), LET I = 12
000007 LET MEEN(I) = MEEN(I) + TSVC(CASE)
000008 LET STDEV(I) = STDEV(I) + (TSVC(CASE)*TSVC(CASE))
000009 LET CNTR(I) = CNTR(I) + 1
000010 LET Y = TSVC(CASE) * 24.0
000011 IF Y GR 0.5, GO TO 102
000012 LET CATG1(I) = CATG1(I) + 1
000013 GO TO 99
000014 102 IF Y GR 1.0, GO TO 103
000015 LET CATG2(I) = CATG2(I) + 1
000016 GO TO 99
000017 103 IF Y GR 2.0, GO TO 104
000018 LET CATG3(I) = CATG3(I) + 1
000019 GO TO 99
000020 104 IF Y GR 3.0, GO TO 105
000021 LET CATG4(I) = CATG4(I) + 1
000022 GO TO 99
000023 105 IF Y GR 4.0, GO TO 106
000024 LET CATG5(I) = CATG5(I) + 1
000025 GO TO 99
000026 106 IF Y GR 5.0, GO TO 107
000027 LET CATG6(I) = CATG6(I) + 1
000028 GO TO 99
000029 107 IF Y GR 10.0, GO TO 108
000030 LET CATG7(I) = CATG7(I) + 1
000031 GO TO 99
000032 108 LET CATG8(I) = CATG8(I) + 1
000033 99 IF PRTO EQ 0, GO TO 10
000034 WRITE ON TAPE 6, CASE,NOCAS(CASE),TIME,OCCUR(CASE)
000035 FORMAT (S65,'CASE TERMINATED IS',16,' NOCAS=',15,' AT TIME',
000036 '2M4.2.2)
000037 WRITE ON TAPE 6, NNN(CASE),MM(CASE),SIS(CASE),S25(CASE),
000038 PRI(CASE),RESA(CASE),XC(CASE),YC(CASE),TSM(CASE),TQUE(CASE),
000039 TINT(CASE)
000040 FORMAT (S65,'414,204.2,D5.0,2M4.2.2)
000041 10 LET NBRCO = NBRCO + 1
000042 LET KOUNT = KOUNT - 1
000043 IF STAPE EQ 0, GO TO 29
000044 DO TO 11, FOR I=(1)(15)
000045 LET IPUFF(I) = 0
000046 IF I GR 10, GO TO 11
000047 LET FRUFF(I) = 0.
000048 11 LOOP
000049 WRITE ON TAPE STAPE, NBRCO,OPFAC(CASE),NOCAS(CASE),I,LOC(CASE),
000050 OCCUR(CASE),BOX(CASE),FPR(CASE),MM(CASE),NNN(CASE),GAMMA(CASE),
000051 NEE(CASE),ATH(CASE),OFSHR(CASE),VIS(CASE),AIND(CASE),SWELL(CASE),
000052 LIC(CASE),POB(CASE),SIS(CASE),S25(CASE),TSM(CASE),OST(CASE),
000053 DWERT(CASE)
000054 FORMAT (I15,I3.03,4,15,11,212,D3.2,12,15,D4.2,615,12,D5.0,201,4)

```



```

000055 WRITE ON TAPE STAPE, ITYPE(CASE), VALUE(CASE), XC(CASE), YC(CASE),
000056 XC(CASE), YC(CASE), STAT(CASE), CARF(CASE), RFSA(CASE), PRI(CASE),
000057 REAL(CASE), COST(CASE), ITOL(CASE), NOINT(CASE), NOUE(CASE),
000058 * FINT(CASE), TNOUE(CASE), TRUFF(CASE), TSVC(CASE), TAIT(CASE)
000059 FORMAT (15,110,435,2,15,12,13,211,67,2,11,212,503,4)
000060
000061 29 LET I = 11
000062 30 IF SRHS(CASE) IS EMPTY, GO TO 39
000063 REMOVE FIRST N FROM SRHS(CASE)
000064 IF I LE 15, LET IBUFF(I) = RESA(N)
000065 LET I = I+1
000066 IF PRTOT EQ 0, GO TO 1
000067 WRITE ON TAPE 6, RESA(N)
000068 FORMAT (565,15)
000069 1 DESTROY NOTE CALLED N
000070 GO TO 30
000071 19 LET I=1
000072 40 IF NSET(CASE) IS EMPTY, GO TO 100
000073 REMOVE FIRST N FROM NSET(CASE)
000074 IF PRTOT EQ 0, GO TO 2
000075 WRITE ON TAPE 6, RESA(N)
000076 FORMAT (565,15)
000077 2 IF I GR 10, GO TO 50
000078 LET IBUFF(I) = NEED(N)
000079 LET IBUFF(I+1) = RESA(N)
000080 LET FRUFF(I) = OST(N)
000081 LET FBUFF(I+1) = DELTA(N)
000082 LET I=I+2
000083 50 IF KRES(N) EQ 0, GO TO 52
000084 LET IB(KRES(N)) = 0
000085 LET IVV = 0
000086 CALL EXQ(KRES(N),*IVV)
000087 52 DESTROY NOTIF CALLED N
000088 GO TO 40
000089 100 DESTROY CASE
000090 IF STAPE EQ 0, RETURN
000091 WRITE ON TAPE STAPE,IRUFF(1),FRUFF(1),FBUFF(I+1),IBUFF(I+1), FOR
000092 * I = (1)(10)(2)
000093 FORMAT 5(12,01,4,01,2,13)
000094 WRITE ON TAPE STAPE, TRUFF(1), FOR I=(11)(15)
000095 FORMAT 5(13)
000096 RETURN
000097 END

```

105130

105160

105170
105180

```

000001      ENDogenous EVENT HOME
000002      STORE RESNO(HOME) IN IRS
000003      LET ETAT(IRS) = 0.
000004      DESTROY HOME CALLED IDEV(IFLT(IRS))
000005      LET ACRES(STN(IRS)) = BCREA(STN(IRS)) - 1
000006      IF TFLT(IFLT(IRS)) GR TLAST, GO TO 10
000007      LET TUTIL(IRS) = TUTIL(IRS) + TIME - TLAST
000008      GO TO 20
000009      10 LET TUTIL(IRS) = TUTIL(IRS) + TIME-TFLT(IFLT(IRS))
000010      20 LET XR(IRS) = XS(STN(IRS))
000011      LET YR(IRS) = YS(STN(IRS))
000012      1 DESTROY FLT CALLED IFLT(IRS)
000013      LET IFLT(IRS) = 0
000014      RETURN
000015      END

```

```

115000
115010
115060

115100
115110
115120
115130
115140

115150
115160

```

```

000001 SUBROUTINE SAC(IPS)
000002 C
000003 C SUBROUTINE SAC IS CALLED WHEN A CASE IS INTERRUPTED. AN ATTEMPT
000004 C IS MADE TO FIND AN IDLE RESOURCE TO SERVE IT; IF NONE IS FOUND,
000005 C THE CASE (OR NEED) IS QUEUED.
000006 C
000007 LET JK = ACASE(IFLT(IRS))
000008 IF (TOT(IFLT(IRS)))NE(1), GO TO 1001
000009 LET TOT(IFLT(IRS)) = 0
000010 LET XC(JK) = XI(IFLT(IRS))
000011 LET YC(JK) = YI(IFLT(IRS))
000012 LET XR(IRS) = XI(IFLT(IRS))
000013 LET YR(IRS) = YI(IFLT(IRS))
000014 LET NQNT(JK) = NQNT(JK) + 1
000015 LET TOTIN = TOTIN + 1
000016 LET NINTR(STN(IRS)) = NINTR(STN(IRS)) + 1
000017 IF IWAIT(JK) EQ IRS, CALL COVER(JK,IRS)
000018 IF (IR(IRS))NE(2), GO TO 1002
000019 IF (MFLG(IFLT(IRS)))EQ(2), GO TO 1002
000020 LET KRES(FITON(IFLT(IRS))) = 0
000021 RETURN
000022 IF (SIS(JK))GR(0), GO TO 1030
000023 IF (MMI(JK)+NNN(JK))EQ(1), GO TO 1010
000024 GO TO 1020
000025 C
000026 C A SINGLE RESOURCE CASE IS BEING INTERRUPTED: SEND IT BACK THROUGH
000027 C THE SYSTEM TO SEE IF AN IDLE RESOURCE CAN SERVE IT; IF NOT, QUEUE
000028 C THE CASE.
000029 C
000030 1010 IF (NEED(JK))NE(17), GO TO 1014
000031 LET ITOL(JK) = 5
000032 FILE JK IN EXCS
000033 RETURN
000034 1014 CALL CRFS(JK,NEED(JK),*INT,*IT)
000035 CALL VEC(JK)
000036 CALL DSET(JK,KNT)
000037 DO TO 1011, FOR EVERY 10ES IN RQUE
000038 IF (IB(IRS))NE(0), GO TO 1011
000039 IF (EJAT(IRS))EQ(0,0), GO TO 1012
000040 IF (HCREW(IFLT(IRS)))EQ(0), GO TO 1013
000041 GO TO 1011
000042 1012 CALL ROCA(JK,IRS,*IT)
000043 IF (I)EQ(1), GO TO 1011
000044 LOOP
000045 GO TO 1015
000046 1013 CALL SERVE(JK,0,IRS)
000047 RETURN
000048 1015 IF (FEA(JK))EQ(2), LET FEA(JK) = 0
000049 LET I = PRI(JK) + IDELT
000050 IF (I)GR(5), GO TO 1016
000051 LET PRI(JK) = PRI(JK) + IDELT
000052 LET TINO(JK) = TIME
000053 LET SIGNL(JK) = I
000054 LET NQUE(JK) = NQUE(JK) + 1

```

75140

```

000055 FILE JK IN QUEUE
000056 RETURN
000057 C
000058 C A MULTI-RESOURCE NEED IS BEING INTERRUPTED; CALL SUBROUTINE QUEUF.
000059 C
000060 1020 CALL QUEUF(IRS,I)
000061 RETURN
000062 C
000063 C A SEARCH NEED IS BEING INTERRUPTED; SEND IT BACK THROUGH THE
000064 C SYSTEM TO SEE IF AN IDLE RESOURCE CAN SERVE IT; IF NOT, QUEUE
000065 C THE CASE
000066 C
000067 1030 IF (MFLG(IFLT(IRS))EQ(3), GO TO 1031)
000068 IF (MFLG(IFLT(IRS))EQ(2), GO TO 1031)
000069 LET A = TIME - ROS(IFLT(IRS))
000070 LET A = A * SOA3(TYPE(IRS))
000071 LET SM(FITON(IFLT(IRS)))=SM(FITON(IFLT(IRS)))-A
000072 LET KDT = FITON(IFLT(IRS))
000073 CALL SASS(JK,SM(KDT),KDT,*IDC)
000074 GO TO 1032, FOR EVERY JRES IN RQUE
000075 IF (IB(IRES))NE(0), GO TO 1032
000076 IF (EJAT(IRES))EQ(0), GO TO 1033
000077 IF (HCREW(IFLT(RES))EQ(0), GO TO 1036
000078 GO TO 1032
000079 1033 CALL ROCA(JK,IRES,*II)
000080 IF (II)EQ(1), GO TO 1034
000081 1032 LOOP
000082 GO TO 1035
000083 1036 CALL SSS(KDT,IRES,*3)
000084 LET RSRC(KDT) = IRES
000085 RETURN
000086 1035 IF (REA(JK))EQ(2), LET REA(JK) = 0
000087 LET I = PRI(JK) + IDELT
000088 IF (I)GR(5), GO TO 1037
000089 LET PRI(JK) = PRI(JK) + IDELT
000090 DO, FOR EACH I IN SRHS(JK)
000091 LET PRI(I) = PRI(I) + IDELT
000092 IF (RSRC(I))NE(0), LET PRIOR(RSRC(I)) = PRIOR(RSRC(I)) + IDELT
000093 LOOP
000094 1037 LET NQUE(JK) = NQUE(JK) + 1
000095 LET J = 0
000096 LET J = 0
000097 DO, FOR EACH K IN SRHS(JK)
000098 IF (SIGML(K))EQ(0), LET I = 1
000099 IF (SIGNL(K))EQ(1), LET J = 1
000100 LOOP
000101 IF (I+J)EQ(0), LET TIMEQ(JK) = TIME
000102 IF (J)EQ(0), LET STING(JK) = TIME
000103 LET SIGNL(FITON(IFLT(IRS))) = I
000104 LET RSRC(FITON(IFLT(IRS))) = 0
000105 FILE FITON(IFLT(IRS)) IN QUEUE
000106 RETURN
000107 END

```

```

000001 SUBROUTINE QUEUE(NCODE,M) 125000
000002 LET NOTIF = NCODE 125010
000003 IF M EQ 1, LET NOTIF = FITON(FLT(NCODE)) 125020
000004 LET CASE = CAS(NOTIF) 125030
000005 IF NEED(NOTIF) NE 17, GO TO 10 125040
000006 LET ITOL(CASF) = 5 125050
000007 LET COMP(NOTIF) = 4 125060
000008 CALL MRECK(NOTIF) 125070
000009 RETURN 125080
000010 10 IF M EQ 0, GO TO 60 125090
000011 C 125100
000012 C INTERRUPT SITUATION. SEE IF SOME OTHER RESOURCE CAN SERVE THE CAS- 125110
000013 C 125120
000014 CALL CRES(CAS(NOTIF),NEE(NOTIF),KONTR,11) 125130
000015 IF 11 EQ 0, GO TO 20 125140
000016 IF KONTR EQ 0, GO TO 20 125150
000017 GO TO 30 125160
000018 20 LET ITOL(CAS(NOTIF)) = 7 125170
000019 CALL MRECK(NOTIF) 125180
000020 RETURN 125190
000021 30 CALL VEC(CAS(NOTIF)) 125200
000022 CALL OSET(CAS(NOTIF),KONTR) 125210
000023 IF RQUE IS EMPTY, GO TO 60 125220
000024 DO TO 50, FOR EACH IRS IN RQUE 125230
000025 IF 1A(IRS) NE 0, GO TO 50 125240
000026 IF 1A(IRS) EQ 0., GO TO 40 125250
000027 IF HCREW(FLT(IRS)) NE 0, GO TO 50 125260
000028 CALL SERVE(CAS(NOTIF),NOTIF,IRS) 125270
000029 RETURN 125280
000030 40 CALL ROCA(CAS(NOTIF),IRS,IVAL) 125290
000031 IF IVAL EQ 2, GO TO 50 125300
000032 CALL SERVE(CAS(NOTIF),NOTIF,IRS) 125310
000033 RETURN 125320
000034 50 LOOP 125330
000035 60 LET N1 = 0 125340
000036 LET N2 = 0 125350
000037 DO TO 100, FOR EACH INOTF IN NSET(CAS(NOTIF)) 125360
000038 IF COMP(INOTF) NE 1, GO TO 100 125370
000039 LET N1 = N1 + 1 125380
000040 IF SGNL(INOTF) EQ 0, GO TO 100 125390
000041 LET N2 = N2 + 1 125400
000042 100 LOOP 125410
000043 C 125420
000044 C N1 = NUMBER OF NOTIFS (FOR THIS CASE) ALREADY IN QUEUE. 125430
000045 C N2 = NUMBER OF NOTIFS IN QUEUE BECAUSE OF INTERRUPTS. 125440
000046 C 125450
000047 IF N1 EQ 0, LET TINQ(CAS(NOTIF)) = TIME 125460
000048 IF N2 GR 0, GO TO 110 125470
000049 IF M EQ 0, GO TO 110 125480
000050 LET STIND(CAS(NOTIF)) = TIME 125490
000051 110 IF RFA(CAS(NOTIF)) NE 2, GO TO 120 125500
000052 LET REAC(CAS(NOTIF)) = 1- 125510
000053 120 FILE NOTIF IN QUEUE 125520
000054 LET KRES(NOTIF) = 0 125530

```

125510
125520
125530
125540
125550
125560
125570
125580
125590
125600
125610
125620

```

LET COMP(INDIF) = I
LET NQUE(CASE) = NQUE(CASE) + 1
LET SIGNL(INDIF) = M
IF M EQ 0, RETURN
LET PRI(CAS(INDIF)) = PRI(CAS(INDIF)) + IDLT
IF PRI(CAS(INDIF)) GR 5, LET PRI(CAS(INDIF)) = 5
DO TO 130, FOR EACH INDIF IN NSET(CASE)
LET PRI(INDIF) = PRI(CAS(INDIF))
IF KRES(INDIF) NE 0, LET PRIOR(KRES(INDIF)) = PRI(CAS(INDIF))
130 LOOP
RETURN
END OF QUEUE

```

000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066

```

000001 SUBROUTINE EXO(IPES,IVAL)
000002 C
000003 C SUBROUTINE EXO EXAMINES THE CASE QUEUE (CQUE) FOR A SINGLE RESOURCE
000004 C CASE, A MULTI-RESOURCE NEED, OR A SEARCH NEED THAT CAN
000005 C BE SERVED BY THE RESOURCE BECOMING IDLE.
000006 C
000007 IF (IVAR)EQ(0), GO TO 10
000008 IF CQUE IS NOT EMPTY, GO TO 20
000009 RETURN
000010 C
000011 C COMPARE BCREW(STN(IRES)) WITH THE PRESENT SHIFT LEVEL.
000012 C
000013 10 LET K = SHIFT(STN(IRES),PSHFT) - BCREW(STN(IRES))
000014 IF (K)GE(0), GO TO 100
000015 C
000016 C RESOURCE MUST BE VECTORED TO HOME STATION DUE TO CREW AVAILABILITY.
000017 C CQUE IS NOT EXAMINED
000018 C
000019 LET HCREW(IFLT(IRES)) = 1
000020 LET IB(IRES) = 0
000021 LET DEP(IFLT(IRES)) = TIME
000022 LET XDEST(IFLT(IRES)) = XS(STN(IRES))
000023 LET YDEST(IFLT(IRES)) = YS(STN(IRES))
000024 LET X = XR(IRES) - XS(STN(IRES))
000025 LET Y = YR(IRES) - YS(STN(IRES))
000026 LET D = SORT(X*X+Y*Y)
000027 IF (SHELL(ACASE(IFLT(IRES))))GRISLIM(TYPE(IRES)), GO TO 16
000028 LET TVEC(IRES) = D/50AL(TYPE(IRES))
000029 GO TO 17
000030 16 LET TVEC(IRES) = D/50A2(TYPE(IRES))
000031 17 LET TARVL(IFLT(IRES)) = TIME + TVEC(IRES)
000032 C CREATE HOME CALLED IDEV(IFLT(IRES))
000033 LET RESNO(IDEV(IFLT(IRES))) = IRES
000034 LET MELG(IFLT(IRES)) = 7
000035 C CAUSE HOME CALLED IDEV(IFLT(IRES)) AT TIME + TVEC(IRES)
000036 RETURN
000037 100 IF CQUE IS EMPTY, GO TO 15
000038 C
000039 C EXAMINE CASE QUEUE
000040 C
000041 20 DO TO 999, FOR EACH ICN IN CQUE
000042 IF (FLG(ICNN))EQ(1), LET CASE = ICNN
000043 IF (FLG(ICNN))EQ(2), LET CASE = CAS(ICNN)
000044 IF (FLG(ICNN))EQ(3), LET CASE = ESAC(ICNN)
000045 C
000046 C CHECK TO SEE IF THE STATION OF THE RESOURCE IS EQUAL TO ONE OF THE
000047 C PRIMARY OR ADJACENT STATIONS OF THE CASE.
000048 C
000049 110 LET LSTA = STATN(CASE)
000050 IF (FLG(ICNN))EQ(2), GO TO 120
000051 IF (FLG(ICNN))EQ(3), GO TO 120
000052 GO TO (120,120,120,111,111,120), RAP
000053 IF (LSTA)EQ(STN(IRES)), GO TO 200
000054 DO TO 112, FOR I=(1)(NACS(LSTA))

```

```

000055 IF (STN(IPES))EQ(ACS(LSTA,1)), GO TO 200
000056 LOOP
000057 DO TO 113, FOR I=(1)(NCUT(LSTA))
000058 IF (STN(IRES))EQ(CUT(LSTA,1)), GO TO 200
000059 LOOP
000060 GO TO 999
000061 IF (LSTA)EQ(STN(IRES)), GO TO 200
000062 DO TO 121, FOR I=(1)(NADJS(LSTA))
000063 IF (STN(IRES))EQ(ADJS(LSTA,1)), GO TO 200
000064 LOOP
000065 DO TO 122, FOR I=(1)(NACS(LSTA))
000066 IF (STN(IRES))EQ(ACS(LSTA,1)), GO TO 200
000067 LOOP
000068 DO TO 123, FOR I=(1)(NCUT(LSTA))
000069 IF (STN(IRES))EQ(CUT(LSTA,1)), GO TO 200
000070 LOOP
000071 IF (ADJS(LSTA,1))EQ(0), GO TO 999
000072 DO TO 130, FOR I=(1)(NADJS(LSTA))
000073 DO TO 124, FOR KK=11)(NACS(ADJS(LSTA,1)))
000074 IF (STN(IRES))EQ(ACS(ADJS(LSTA,1),KK)), GO TO 200
000075 LOOP
000076 DO TO 125, FOR KK=11)(NCUT(ADJS(LSTA,1)))
000077 IF (STN(IRES))EQ(CUT(ADJS(LSTA,1),KK)), GO TO 200
000078 LOOP
000079 LOOP
000080 GO TO 999
000081 IF (FLG(ICNN))EQ(1), CALL CRES(ICNN,NEO(ICNN),*KONT,*11)
000082 IF (FLG(ICNN))EQ(2), CALL CRES(CAS(ICNN),NEO(ICNN),*KONT,*11)
000083 IF (FLG(ICNN))EQ(3), CALL CRES(ESAC(ICNN),18,*KONT,*11)
000084 DO TO 201, FOR EACH IR IN RQUE
000085 IF (IR)EQ(IRES), GO TO 210
000086 LOOP
000087 GO TO 999
000088 IF (SQTAG(TYPE(IRES)))EQ(0), GO TO 211
000089 IF (PRI(CASE))GE(TPRI), GO TO 211
000090 LET II = 1
000091 DO, FOR EACH IR IN RQUE
000092 IF (SQTAG(TYPE(IR)))EQ(0), LET II = 0
000093 LOOP
000094 IF (11)EQ(1), GO TO 211
000095 IF (FLG(ICNN))NE(3), GO TO 999
000096 DO TO 212, FOR EACH IR IN RQUE
000097 IF (SQTAG(TYPE(IR)))GR(0), GO TO 212
000098 LET X = XC(CASE) - XS(STN(IR))
000099 LET Y = YC(CASE) - YS(STN(IR))
000100 LET D = SQRT(X*X+Y*Y)
000101 IF (SMELL(CASE))GR(SLH(TYPE(IR))), GO TO 700
000102 LET XY = D/50A1(TYPE(IR))
000103 GO TO 701
000104 LET XY = D/50A2(TYPE(IR))
000105 IF (END(TYPE(IR)))GR(2.0*XY), LET II = 1
000106 LOOP
000107 IF (11)EQ(1), GO TO 999
000108 C
000109 C RESOURCE IS IN RQUE; CALCULATE TVEC(IRES)
000110 C
000111 LET X = XR(IRES) - XC(CASE)
000112 LET Y = YR(IRES) - YC(CASE)

```



```

000113 LET D = SORT(X*X+Y*Y)
000114 IF (SMALL(CASE))OR(SLIMITYPE(IRES)), GO TO 220
000115 LET TVEC(IRES) = D/50AL(TYPE(IRES))
000116 GO TO 230
000117 220 LET TVEC(IRES) = D/50A2(TYPE(IRES))
000118 230 IF (FLG(ICNN))EQ(2), GO TO 300
000119 IF (FLG(ICNN))EQ(3), GO TO 400
000120 C
000121 C FLG(ICNN) = 1: I.E., A SINGLE RESOURCE IS BEING REMOVED FROM CQUE
000122 C TO BE SERVED. ATTRIBUTES WILL BE UPDATED AND SUBROUTINE SERVE WILL
000123 C BE CALLED.
000124 C
000125 250 IF (IVAR)EQ(0), GO TO 260
000126 LET IVAR = 0
000127 LET IMAT(ACASE(IFLT(IRES))) = 0
000128 LET KRES(FITON(IFLT(IRES))) = 0
000129 260 LET TQUE(ICNN) = TQUE(ICNN) + (TIME-TINQ(ICNN))
000130 IF (SGLH(ICNN))EQ(1),LET TINT(ICNN)=TINT(ICNN)+(TIME-TINQ(ICNN))
000131 LET COST(IRES) = COST(TYPE(IRES)) * TVEC(IRES)
000132 LET PRIOR(IRES) = PRI(ICNN)
000133 STORE ICNN IN ACASE(IFLT(IRES))
000134 REMOVE ICNN FROM CQUE
000135 CALL SERVE(ICNN,0,IRES)
000136 RETURN
000137 C
000138 C FLG(ICNN) = 2: A MULTI-RESOURCE NEED IS BEING REMOVED FROM CQUE
000139 C TO BE SERVED. RES AND FLT ATTRIBUTES WILL BE UPDATED AND
000140 C SUBROUTINE SVQUE WILL BE CALLED.
000141 C
000142 300 IF (IVAR)EQ(0), GO TO 301
000143 LET IVAR = 0
000144 LET IMAT(ACASE(IFLT(IRES))) = 0
000145 LET KRES(FITON(IFLT(IRES))) = 0
000146 301 LET COST(IRES) = COST(TYPE(IRES))*TVEC(IRES)
000147 LET PRIOR(IRES) = PRI(ICNN)
000148 STORE CAS(ICNN) IN ACASE(IFLT(IRES))
000149 STORE ICNN IN FITON(IFLT(IRES))
000150 CALL SVQUE(ICNN,IRES)
000151 RETURN
000152 C
000153 C FLG(ICNN) = 3: A SEARCH NEED IS BEING REMOVED FROM CQUE TO BE
000154 C SERVED. ATTRIBUTES ARE UPDATED AND THE ENDOGENOUS EVENT AHSCH IS
000155 C CREATED AND CAUSED.
000156 C
000157 400 LET X = XC(FSAC(ICNN))-YS(STN(IRES))
000158 LET Y = YC(FSAC(ICNN))-YS(STN(IRES))
000159 LET D = SORT(X*X+Y*Y)
000160 IF (SMALL(CASE))OR(SLIMITYPE(IRES)), GO TO 401
000161 LET XY = D/50AL(TYPE(IRES))
000162 GO TO 402
000163 401 LET XY = D/50A2(TYPE(IRES))
000164 402 IF (END(TYPE(IRES)))(F(2,0*XY), GO TO 999
000165 450 LET X = AINT(TIME) + RISC
000166 IF (TIME)LE(X), GO TO 500
000167 LET X = AINT(TIME) + SET
000168 IF (TIME)GR(X), GO TO 500
000169 IF (TVEC(IRES))L(X-TIME), GO TO 500
000170 LET Y = AINT(TIME) + RISC + 1.0

```

```

000171 IF (TIME+TVEC(IRES))EQ(1), GO TO 500
000172 IF (VOP(TYPE(IRES)))EQ(1), GO TO 999
000173 IF (I)EQ(0), GO TO 501
000174 LET I)VAR = 0
000175 LET I)WAIT(ACASE(IFLT(IRES))) = 0
000176 LET KRES(FITON(IFLT(IRES))) = 0
000177 LET KRES(ESAC(ICNN)) = CNPES(ESAC(ICNN)) + 1
000178 LET TST = SN(ICNN)/50A3(TYPE(IRES))
000179 LET COST(IRES) = COSTD(TYPE(IRES))*(TVEC(IRES)+TST)
000180 LET COSTC(ESAC(ICNN)) = COSTD(TYPE(IRES)) + TVEC(IRES)+TST)
000181 LET RSRC(ICNN) = IRES
000182 LET I = SIGNL(ICNN)
000183 LET SIGNL(ICNN) = 2
000184 LET JIND = 2
000185 LET KIND = 2
000186 DO TO 506, FOR EACH J,1, SRHS(FSAC(ICNN))
000187 IF (SIGNL(J))EQ(0), LET JIND = 1
000188 IF (SIGNL(J))EQ(1), LET KIND = 1
000189 SO6 LOOP
000190 IF (I)EQ(1), GO TO 507
000191 IF (JIND)EQ(1), GO TO 520
000192 IF (KIND)EQ(1), GO TO 520
000193 LET TRUE(ESAC(ICNN)) = TQUF(ESAC(ICNN))+(TIME-TINQ(ESAC(ICNN)))
000194 GO TO 520
000195 IF (KIND)EQ(1), GO TO 520
000196 IF (JIND)EQ(2), GO TO 510
000197 LET TINT(ESAC(ICNN)) = TINT(ESAC(ICNN))+(TIME-STINQ(ESAC(ICNN)))
000198 GO TO 520
000199 LET TQUE(ESAC(ICNN)) = TQUE(ESAC(ICNN))+(TIME-TINQ(ESAC(ICNN)))
000200 LET TINT(ESAC(ICNN)) = TINT(ESAC(ICNN))+(TIME-STINQ(ESAC(ICNN)))
000201 LET SASG(ICNN) = 1
000202 LET IR(IRES) = 1
000203 LET NCASE(IRES) = NCASE(IRES) + 1
000204 LET PRIOR(IRES) = PRI(ICNN)
000205 STORE FSAC(ICNN) IN ACASE(IFLT(IRES))
000206 LET DEP(IFLT(IRES)) = TIME
000207 STORE ICNN IN FITON(IFLT(IRES))
000208 LET XDEST(IFLT(IRES)) = XC(ESAC(ICNN))
000209 LET YDEST(IFLT(IRES)) = YC(ESAC(ICNN))
000210 CREATE ARSCH CALLED IDEV(IFLT(IRES))
000211 LET LNOTE(IDEV(IFLT(IRES))) = ICNN
000212 LET LRES(IDEV(IFLT(IRES))) = IRES
000213 LET II = 0
000214 DO TO 1, FOR EACH LL IN SRHS(ESAC(ICNN))
000215 IF (SFLAG(LL))EQ(1), LET II = 1
000216 1 LOOP
000217 IF (I)EQ(1), GO TO 2
000218 LET SFLAG(ICNN) = 1
000219 LET LFLG(IDEV(IFLT(IRES))) = 1
000220 GO TO 3
000221 2 LET LFLG(IDEV(IFLT(IRES))) = 2
000222 3 LET MFLG(IFLT(IRES)) = 3
000223 LET ROS(IFLT(IRES)) = TIME + TVEC(IRES)
000224 CAUSE ARSCH CALLED IDEV(IFLT(IRES)) AT TIME + TVEC(IRES)
000225 LET NEEDS(STIN(IRES)) = NEEDS(STIN(IRES)) + 1
000226 REMOVE ICNN FROM CODE
000227 RETURN
000228 999 LOOP

```

```
000229  
000230  
000231  
000232  
000233  
000234  
000235  
  
C  
C  
C  
C  
  
C RESOURCE CANNOT SERVE ANY CASE IN QUEUE; GO TO 15 SO THAT RESOURCE  
C CAN BE VECTORED TO HOME STATION.  
  
C IF (I*VAR)EQ(0), GO TO 14  
C RETURN  
C END
```

```

000001      ENDOGENOUS EVENT STRAY
000002      C
000003      C      ENDOGENOUS EVE AT STRAY OCCURS WHEN A STAND-BY CREW ARRIVES AT A
000004      C      STATION. THE QUEUE IS EXAMINED. IF THERE IS NO CASE TO SERVICE,
000005      C      THE STAND-BY CREW IS NOT ADDED TO THE AVAILABLE CREWS AT THE STA.
000006      C
000007      STORE STAT(STRAY) IN ISTA
000008      DESTROY STRAY
000009      IF CQUE IS EMPTY, GO TO 510
000010      GO TO 100, FOR EACH IJK IN CQUE
000011      IF (FLG(IJK))EQ(1), LET CASE = IJK
000012      IF (FLG(IJK))EQ(2), LET CASE = CAS(IJK)
000013      IF (FLG(IJK))EQ(3), LET CASE = ESAC(IJK)
000014      IF (SIS(CASE))GR(0), GO TO 210
000015      IF (NNI(CASE)*MM(CASE))GR(1), GO TO 210
000016      GO TO (210,210,210,201,201,210), RAP
000017      IF (STATN(CASE))EQ(ISTA), GO TO 300
000018      GO TO 202, FOR I=1((NACS(STATN(CASE))))
000019      IF (ISTA)EQ(ACS(STATN(CASE),I)), GO TO 300
000020      LOOP
000021      GO TO 203, FOR I=1((NCUT(STATN(CASE))))
000022      IF (ISTAEQ(CUT(STATN(CASE),I)), GO TO 300
000023      LOOP
000024      GO TO 100
000025      IF (STATN(CASE))EQ(ISTA), GO TO 300
000026      GO TO 211, FOR I=1((NADJS(STATN(CASE))))
000027      IF (ISTAEQ(ADJS(STATN(CASE),I)), GO TO 300
000028      LOOP
000029      GO TO 212, FOR I=1((NACS(STATN(CASE))))
000030      IF (ISTAEQ(ACS(STATN(CASE),I)), GO TO 300
000031      LOOP
000032      GO TO 213, FOR I=1((NCUT(STATN(CASE))))
000033      IF (ISTAEQ(CUT(STATN(CASE),I)), GO TO 300
000034      LOOP
000035      GO TO 214, FOR I=1((NADJS(STATN(CASE))))
000036      GO TO 215, FOR K=1((NACS(ADJS(STATN(CASE),I))))
000037      IF (ISTAEQ(ACS(ADJS(STATN(CASE),I),K)), GO TO 300
000038      LOOP
000039      GO TO 217, FOR K=1((NCUT(ADJS(STATN(CASE),I))))
000040      IF (ISTAEQ(CUT(ADJS(STATN(CASE),I),K)), GO TO 300
000041      LOOP
000042      LOOP
000043      GO TO 100
000044      IF (FLG(IJK))EQ(1), CALL CRES(IJK,NEO(IJK),*KONTR,*11)
000045      IF (FLG(IJK))EQ(2), CALL CRES(CAS(IJK),NEO(IJK),*KONTR,*11)
000046      IF (FLG(IJK))EQ(3), CALL CRES(ESAC(IJK),18,*KONTR,*11)
000047      GO TO 400, FOR EACH IRS IN RQUE
000048      IF (STN(IRS))NE(ISTA), GO TO 400
000049      IF (E1AT(IRS))NE(0,0), GO TO 400
000050      IF (RAND)*GR(TYPE(IRS)), GO TO 400
000051      IF (SOTAG(TYPE(IRS))EQ(0), GO TO 350
000052      IF (PRI(CASE))GE(TPRI), GO TO 350
000053      LET I1 = 1
000054      DO, FOR EACH IR IN RQUE

```

```

000055 IF (SWTAG(TYPE(IRS))EQ(0)), LET I1 = 0
000056 LOOP
000057 IF (I1)EQ(1), GO TO 350
000058 IF (FLAG(IJK))NE(3), GO TO 400
000059 GO TO 340, FOR EACH IR IN CQUE
000060 IF (SWTAG(TYPE(IRS))NE(0)), GO TO 340
000061 LET X = XC(CASE) - XS(STA(IRS))
000062 LET Y = YC(CASE) - YS(STA(IRS))
000063 LET D = SQR(X*X+Y*Y)
000064 IF (SWELL(CASE))GR(SLIM(TYPE(IRS))), GO TO 338
000065 LET XY = D/50A1(TYPE(IRS))
000066 GO TO 339
000067 LET XY = D/50A2(TYPE(IRS))
000068 339 IF (END(TYPE(IRS)))GR(2.0*XY), LET I1 = 1
000069 340 LOOP
000070 IF (I1)EQ(1), GO TO 400
000071 350 LET PCREA(ISTA) = PCREA(ISTA) + 1
000072 C
000073 C A SINGLE RESOURCE NEED, A MULTI-RESOURCE NEED OR A SEARCH NEED
000074 C HAS BEEN LOCATED IN CQUE THAT CAN BE SERVED BY THE STANDBY CREW.
000075 C A RESOURCE THAT CAN SERVE THE NEED HAS BEEN LOCATED IDLE AT THE
000076 C STATION. TVEC(IRS) IS CALCULATED.
000077 C
000078 LET X = XR(IRS)-XC(CASE)
000079 LET Y = YR(IRS)-YC(CASE)
000080 IF (SWELL(CASE))GR(SLIM(TYPE(IRS))), GO TO 600
000081 LET D = SQR(X*X+Y*Y)
000082 LET TVEC(IRS) = D/50A1(TYPE(IRS))
000083 GO TO 601
000084 LET D = SQR(X*X+Y*Y)
000085 LET TVEC(IRS) = D/50A2(TYPE(IRS))
000086 601 IF (FLAG(IJK))EQ(2), GO TO 700
000087 IF (FLAG(IJK))EQ(3), GO TO 800
000088 C
000089 C FLAG(IJK)=1: I.E., A SINGLE RESOURCE NEED IS BEING REMOVED FROM
000090 C CQUE TO BE SERVED.
000091 C
000092 LET TQUE(CASE) = TQUE(CASE) + (TIME-TINQ(CASE))
000093 IF (SIGNL(CASE))EQ(1),LET TINT(CASE)=TINT(CASE)+(TIME-TINQ(CASE))
000094 LET COST(IRS) = COSTD(TYPE(IRS))*TVEC(IRS)
000095 LET PRIOR(IRS) = PRI(CASE)
000096 REMOVE CASE FROM CQUE
000097 CALL SERVE(CASE,0,IRS)
000098 RETURN
000099 C
000100 C FLAG(IJK)=2: A MULTI-RESOURCE NEED IS BEING REMOVED FROM CQUE TO
000101 C BE SERVED.
000102 C
000103 700 LET COST(IRS) = COSTD(TYPE(IRS))*TVEC(IRS)
000104 LET PRIOR(IRS) = PRI(CASE)
000105 CALL SVQUE(IJK,IRS)
000106 RETURN
000107 C
000108 C FLAG(IJK) = 3: A SEARCH NEED IS BEING REMOVED FROM CQUE TO BE
000109 C SERVED. ATTRIBUTES ARE UPDATED AND THE ENDOGENOUS EVENT ARSCH
000110 C IS CREATED AND CAUSED.
000111 C
000112 800 IF (END(TYPE(IRS)))GR(2.0*TVEC(IRS)), GO TO 850

```

```

000113 LET BCREW(ISTA) = BCREW(ISTA)-1
000114 GO TO 400
000115 IF (VDF(TYPE(IRS)))(EQ(0)), GO TO 860
000116 LET X = AINT(TIME) + RISE
000117 IF (TIME)EQ(X), GO TO 840
000118 LET X = AINT(TIME) + SET
000119 IF (TIME)GR(X), GO TO 840
000120 IF (TVEC(IRS)+DLAY(TYPE(IRS)))(LS(X-TIME), GO TO 860
000121 LET X = AINT(TIME) + RISE + 1.0
000122 IF (TIME+TVEC(IRS)+DLAY(TYPE(IRS)))(SE(X), GO TO 860
000123 LET BCREW(ISTA) = BCREW(ISTA) - 1
000124 GO TO 400
000125
000126
000127
000128
000129
000130
000131
000132
000133
000134
000135
000136
000137
000138
000139
000140
000141
000142
000143
000144
000145
000146
000147
000148
000149
000150
000151
000152
000153
000154
000155
000156
000157
000158
000159
000160
000161
000162
000163
000164
000165
000166
000167
000168
000169
000170

450
451
860
861
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

LET BCREW(ISTA) = BCREW(ISTA)-1
GO TO 400
IF (VDF(TYPE(IRS)))(EQ(0)), GO TO 860
LET X = AINT(TIME) + RISE
IF (TIME)EQ(X), GO TO 840
LET X = AINT(TIME) + SET
IF (TIME)GR(X), GO TO 840
IF (TVEC(IRS)+DLAY(TYPE(IRS)))(LS(X-TIME), GO TO 860
LET X = AINT(TIME) + RISE + 1.0
IF (TIME+TVEC(IRS)+DLAY(TYPE(IRS)))(SE(X), GO TO 860
LET BCREW(ISTA) = BCREW(ISTA) - 1
GO TO 400
CREATE FLT CALLED IFLT(IRS)
STORE CASE IN ACASE(IFLT(IRS))
LET DEPI(IFLT(IRS)) = 0.0
STORE IJK IN FITON(IFLT(IRS))
LET TELT(IFLT(IRS)) = 0.0
LET HCREW(IFLT(IRS)) = 0
LET XDEST(IFLT(IRS)) = 0.0
LET YDEST(IFLT(IRS)) = 0.0
LET MELG(IFLT(IRS)) = 12
LET POS(IFLT(IRS)) = 0.0
CREATE READY CALLED IDEV(IFLT(IRS))
LET NREAD(IDEV(IFLT(IRS))) = IJK
LET RREAD(IDEV(IFLT(IRS))) = IRS
LET II = 0
DO TO 1, FOR EACH LL IN SRHS(CASE)
IF (SFLAG(LL))EQ(1), LET II = 1
1 LOOP
IF (I)EQ(1), GO TO 2
LET SFLAG(IJK) = 1
LET FREAD(IDEV(IFLT(IRS))) = 1
GO TO 3
2 LET FREAD(IDEV(IFLT(IRS))) = 2
3 CAUSE READY CALLED IDEV(IFLT(IRS)) AT TIME + DLAY(TYPE(IRS))
LET CNPES(CASE) = CNRES(CASE) + 1
LET TST = S*(IJK)/SOA3(TYPE(IRS))
LET COST(IRS) = COSTD(TYPE(IRS))*(TVEC(IRS)+YST)
LET COSTC(CASE) = COSTC(CASE) + COST(IRS)
LET PSRC(IJK) = IRS
LET I = SIGNAL(IJK)
LET SIGNAL(IJK) = 2
LET KIND = 2
LET KIND = 2
DO TO 861, FOR EACH J IN SRHS(CASE)
IF (SIGNAL(J))EQ(0), LET JIND = 1
IF (SIGNAL(J))EQ(1), LET KIND = 1
LOOP
861 IF (I)EQ(1), GO TO 807
IF (JIND)EQ(1), GO TO 820
IF (KIND)EQ(1), GO TO 820
LET TQUE(CASE) = TQUE(CASE) + (TIME-TIND(CASE))
GO TO 820
867 IF (KIND)EQ(1), GO TO 820
IF (JIND)EQ(2), GO TO 810
LET TINT(CASE) = TINT(CASE) + (TIME-STIND(CASE))
GO TO 820
870 LET TQUE(CASE) = TQUE(CASE) + (TIME-TTIND(CASE))

```

```

000171
000172
000173
000174
000175
000176
000177
000178
000179
000180
000181
000182
000183
000184
000185

      LET TINT(CASE) = TINT(CASE) + (TIME-STIME(CASE))
      LET SASG(IJK) = 1
      LET FIAT(IRS) = 1.0
      LET IB(IRS) = 1
      LET NCASE(IRS) = NCASE(IRS) + 1
      LET PRIOR(IRS) = PRI(IJK)
      LET NEEDS(STN(IRS)) = NEEDS(STN(IRS)) + 1
      REMOVE IJK FROM QUEUE
      RETURN
400  LOOP
100  LOOP
510  LET UNPRO(ISTA) = UNPRO(ISTA) + 1
      LET TUNPR = TUNPR + 1
      RETURN
      END

```

```

000001      SUBROUTINE SVQUE(NOTIF,IRFS)
000002      LET CASE = CAS(NOTIF)
000003      REMOVE NOTIF FROM CQUE
000004      LET COMP(NOTIF) = 2
000005      LET KRES(NOTIF) = IRES
000006      LET N1 = 0
000007      LET N2 = 0
000008      DO TO 10, FOR EACH INOTF IN NSFT(CASE)
000009      IF COMP(INOTF) NE 1, GO TO 10
000010      LET N1 = N1 + 1
000011      IF SIGNAL(INOTF) EQ 0, GO TO 10
000012      LET N2 = N2 + 1
000013      10 LOOP
000014      IF N1 EQ 0, LET TQUE(CASE) = TQUE(CASE) + TIME-TINQ(CASE)
000015      IF N2 GR 0, GO TO 20
000016      IF SIGNAL(NOTIF) EQ 1, LET TINT(CASE) = TINT(CASE) + TIME -
000017      ,STING(CASE)
000018      20 LET SIGNAL(NOTIF) = 2
000019      CALL SERVE(CASE,NOTIF,IRFS)
000020      RETURN
000021      END OF SVQUE

```

```

130000
130010
130020
130030
130040
130050
130060
130070
130080
130090
130100
130110
130120
130130
130160
130170
130190
130200

```



```

000001 SUBROUTINE WRECK(NOTIF) 120300
000002 C THIS SUBROUTINE CANCELS AND DESTROYS NOTIFS, ASSOCIATED ONSCN A-D 120310
000003 C ARVSN EMERGENT EVENTS, AND CASES. 120320
000004 C 120330
000005 LET NTIFC = CAS(NOTIF) 120340
000006 1 IF NSET(NTIFC) IS EMPTY, GO TO 20 120350
000007 REMOVE FIRST NOTIF FROM NSET(NTIFC) 120360
000008 IF COMP(NOTIF) GE 3, GO TO 15 120370
000009 IF (COMP(NOTIF))EQ(1), GO TO 7 120380
000010 IF (COMP(NOTIF))EQ(2), GO TO 5 120390
000011 3 CANCEL NOTIF 120400
000012 GO TO 9 120410
000013 7 REMOVE NOTIF FROM CQUE 120420
000014 GO TO 9 120430
000015 5 IF KRES(NOTIF) EQ 0, GO TO 15 120440
000016 IF MFLG(IFLT(KRES(NOTIF))) EQ 1, GO TO 11 120450
000017 IF (MFLG(IFLT(KRES(NOTIF))))EQ(2), GO TO 13 120460
000018 IF MFLG(IFLT(KRES(NOTIF))) EQ 13, GO TO 6 120470
000019 IF MFLG(IFLT(KRES(NOTIF))) EQ 11, GO TO 4 120480
000020 GO TO 9
000021 11 CANCEL ARVSN CALLED IDEV(IFLT(KRES(NOTIF)))
000022 DESTROY ARVSN CALLED IDEV(IFLT(KRES(NOTIF)))
000023 GO TO 8
000024 6 CANCEL CHEKN CALLED IDEV(IFLT(KRES(NOTIF)))
000025 DESTROY CHEKN CALLED IDEV(IFLT(KRES(NOTIF)))
000026 GO TO 9
000027 4 CANCEL DELAY CALLED IDEV(IFLT(KRES(NOTIF)))
000028 DESTROY DELAY CALLED IDEV(IFLT(KRES(NOTIF)))
000029 GO TO 9
000030 13 CANCEL ONSCN CALLED IDEV(IFLT(KRES(NOTIF)))
000031 DESTROY ONSCN CALLED IDEV(IFLT(KRES(NOTIF)))
000032 IF NUMBR(NOTIF) EQ 0, GO TO 9
000033 8 LET I = KRES(NOTIF) 120240
000034 LET X = XR(I) - XDEST(IFLT(I)) 120250
000035 LET Y = YR(I) - YDEST(IFLT(I)) 120260
000036 LET D = SQR(X*X+Y*Y) 120270
000037 IF NEED(NOTIF) EQ 17, GO TO 30 120280
000038 IF NUMBR(NOTIF) NE 0, GO TO 35 120290
000039 IF SWELL(NTIFC) GR SLIM(TYPE(I)), GO TO 31 120300
000040 30 LET SPEED = 50A1(TYPE(I)) 120310
000041 GO TO 39 120320
000042 31 LET SPEED = 50A2(TYPE(I)) 120330
000043 GO TO 39 120340
000044 35 IF L(NTIFC) GR 26, GO TO 37 120350
000045 LET SPEED = TSP1 120360
000046 GO TO 39 120370
000047 37 LET SPED = TSP2 120380
000048 39 LET C = (TIME-DEP(IFLT(I)))*SPED 120390
000049 IF C LS 0, LET C = 0. 120400
000050 LET XR(I) = XR(I) - (C*A)/O 120410
000051 LET YR(I) = YR(I) - (C*Y)/O 120420
000052 LET XDEST(IFLT(I)) = 0. 120430
000053 LET YDEST(IFLT(I)) = 0. 120440
000054 LET DEP(IFLT(I)) = 0. 120450

```

000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065

```
9 LET IRES = KRES(NTIF)  
  IF (IRES)G(0),GO TO 15  
  LET JA(IRES) = 1  
  LET MFLG(1FL(IRES)) = 9  
  LET IVAR = 0  
  CALL EXQ(IRES,*IVAR)  
15 DESTROY NTIF  
   GO TO 1  
20 FILE NTIFC IN EXCS  
  RETURN  
  END
```

120560
120570
120490

120540
120550
120560
120570
120580

```

000001      EXOGENOUS EVENT ELUSIN
000002      DIMENSION Ibuff(15),PROFF(10)
000003      LET TOTIME(PSHFT) = TOTIME(PSHFT) + TIME - TLAST
000004      DO TO 10, FOR EACH RES IRS
000005      IF FIAT(IRS) EQ 0., GO TO 6
000006      IF TFLT(IFLT(IRS)) EQ 0.0, GO TO 6
000007      IF TFLT(IFLT(IRS)) GT TLAST, GO TO 5
000008      LET TUTIL(IRS) = TUTIL(IRS) + TIME - TLAST
000009      GO TO 6
000010      5 LET TUTIL(IRS) = TUTIL(IRS) + TIME - TFLT(IFLT(IRS))
000011      6 LET UTIL(IRS) = UTIL(IRS) + TUTIL(IRS)
000012      LET USHF(STN(IRS),PSHFT) = USHF(STN(IRS),PSHFT) + TUTIL(IRS)
000013      IO LOOP
000014      LET IJK = 0
000015      DO, FOR EACH RES I, WITH (SQTAG(TYPE(I)))EQ(0)
000016      LET RUTIL = RUTIL + UTIL(I)
000017      LET IJK = IJK + 1
000018      LOOP
000019      LET RUTIL = RUTIL*100./(TIME*FLOAT(IJK))
000020      LET IJK = 0
000021      DO, FOR EACH RES I, WITH (SQTAG(TYPE(I)))EQ(1)
000022      LET CUTIL = CUTIL + UTIL(I)
000023      LET IJK = IJK + 1
000024      LOOP
000025      LET CUTIL = CUTIL*100.0/(TIME*FLOAT(IJK))
000026      LET IJK = 0
000027      DO, FOR EACH RES I, WITH (SQTAG(TYPE(I)))EQ(3)
000028      LET AIRU = AIRU + UTIL(I)
000029      LET IJK = IJK + 1
000030      LOOP
000031      LET AIRU = AIRU*100./(TIME*FLOAT(IJK))
000032      LET R = NNTMT/MEAND
000033      DO, FOR EACH STA I
000034      LET CS(GRP(I)) = CS(GRP(I)) + NCAS(I)
000035      LET NDS(GRP(I)) = NDS(GRP(I)) + NEEDS(I)
000036      LET FL1(GRP(I)) = FL1(GRP(I)) + FAIL1(I)
000037      LET FL2(GRP(I)) = FL2(GRP(I)) + FAIL2(I)
000038      LET FL3(GRP(I)) = FL3(GRP(I)) + FAIL3(I)
000039      LET INTRP(GRP(I)) = INTRP(GRP(I)) + NINTR(I)
000040      LET NDS(GRP(I)) = NDS(GRP(I)) + NSTRY(I)
000041      LET NONPR(GRP(I)) = NONPR(GRP(I)) + UNPRO(I)
000042      DO, FOR EACH WEADS K
000043      LET USEAV(GRP(I)) = USEAV(GRP(I)) + USHF(I,K)
000044      LOOP
000045      LOOP
000046      DO TO 200, FOR EACH GROUP I
000047      LET NUM = 0
000048      DO TO 300, FOR EACH STA II, WITH (GRP(II))EQ(1)
000049      DO TO 30, FOR EACH RST III
000050      LET NUM = NUM + RST(II,III)
000051      330 LOOP
000052      300 LOOP
000053      IF NUM EQ 0, GO TO 200
000054      IF (USEAV(I))LE(0.0), GO TO 1

```

15000
15010
15020
15030

15040
15050
15060
15070
15080
15090
15100

```

000055 LET USEAV(1) = (USEAV(1)*100.0)/(TIME*FLOAT(NUM))
000056 IF CS(1) LE 0, GO TO 200
000057 LET X = FLOAT(CS(1))
000058 LET TAVG(1) = TAVG(1)*24./X
000059 LET TVAG(1) = TVAG(1)*24./X
000060 LET TMTAV(1) = TMTAV(1)*24./X
000061 LET AVDR(1) = (AVDR(1) * 24.0 * P)/X
000062 200 LOOP
000063 DO, FOR I=(1)(2)
000064 LET X = FLOAT(CNTR(I))*FLOAT(NRES)
000065 LET MEEN(I) = MEEN(I)/X
000066 LET X = X * FLOAT(NRES)
000067 LET STDEV(I) = ABS((STDEV(I)/X)-(MEEN(I)*MEEN(I)))
000068 LET STDEV(I) = SQRT(STDEV(I) * 100.0)
000069 LET MEEN(I) = MEEN(I) * 100.0
000070 LOOP
000071 DO, FOR I=(1)(10)
000072 LET K = 1 + 2
000073 LET MEEN(K) = MEEN(K)/FLOAT(CNTR(K))
000074 LET STDEV(K) = ABS((STDEV(K)/FLOAT(CNTR(K)))-(MEEN(K)*MEEN(K)))
000075 LET STDEV(K) = SQRT(STDEV(K) * 24.0)
000076 LET MEEN(K) = MEEN(K) * 24.0
000077 LOOP
000078 20 DO TO 40, FOR EACH WORDS J
000079 DO TO 30, FOR EACH STA I
000080 LET AVUS(J) = AVUS(J) + USHF(I,J)
000081 30 LOOP
000082 LET AVUS(J) = AVUS(J)*100./(TOTPE(J)*FLOAT(NRES))
000083 40 LOOP
000084 DO TO 50, FOR EACH RES I
000085 LET AUT(TYPE(I)) = AUT(TYPE(I)) + UTIL(I)
000086 50 LOOP
000087 DO TO 70, FOR EACH RST I
000088 LET NUM = 0
000089 DO TO 60, FOR EACH STA J
000090 LET NUM = NUM + REST(J,I)
000091 60 LOOP
000092 IF NUM LE 0, GO TO 70
000093 LET AUT(I) = AUT(I)*100./(TIME*FLOAT(NUM))
000094 IF (ISOTAG(1))EQ(2), LET C300 = AUT(I)
000095 70 LOOP
000096 LET NUMB = 0
000097 LET MEANB = 0
000098 LET COUNT = 0
000099 LET LIMIT = 0
000100 DO TO 100, FOR EACH STA J
000101 LET SNEED = SNEED + NEEDS(J)
000102 LET NBRFA = NBRFA + FAIL1(J)
000103 LET NBRFB = NBRFB + FAIL2(J)
000104 LET NBRFC = NBRFC + FAIL3(J)
000105 LET MEANB = MEANB + AVGT(J)
000106 LET LIMIT = LIMIT + NBRRO(J)
000107 LET NUMB = NUMB + NCAS(J)
000108 IF NCAS(J) LE 0, GO TO 70
000109 LET COUNT = COUNT + NCAS(J)
000110 LET AVGT(J) = AVGT(J)*24./(FLOAT(NCAS(J)))
000111 IF FAIL3(J) GR 0, LET CFTT(J)=T/TOTL(J)*24./FLOAT(FAIL3(J))
000112 LET T/TOTL(J) = T/TOTL(J)*24./(FLOAT(NCAS(J)))

```

15110
15120
15130
15140
15160
15170
15190
15200
15210
15220
15230
15240
15250
15260
15270
15280

15290
15298
15299

15300

15310
15320
15330
15302

15304

```

000113 LET VCTR(J) = VCTR(J)*24./ (FLOAT(NCAS(J)))
000114 LET ONP1(J) = (ONP1(J)*24.0*4)/(FLOAT(NCAS(J)))
000115 79 LET NUM = 7
000116 DO TO 83, FOR EACH RST I
000117 LET NUM = NUM + RST(J,I)
000118 80 LOOP
000119 DO TO 93, FOR EACH REPS I
000120 LET USE(J) = USE(J) + USHF(J,I)
000121 IF TOTIME(I) LE 0., GO TO 90
000122 IF NUM LE 0, GO TO 90
000123 LET USHF(J,I) = USHF(J,I)*100./ (TOTIME(I)*FLOAT(NUM))
000124 90 LOOP
000125 IF NUM LE 0, GO TO 100
000126 LET USE(J) = USE(J)*100./ (TIME*FLOAT(NUM))
000127 LET AVUTO = AVUTO + USE(J)*FLOAT(NUM)
000128 100 LOOP
000129 LET AVUTO = AVUTO/FLOAT(REPS)
000130 LET MEANV = MEANV*24./FLOAT(NUM*3)
000131 IF NRREC GR 0, LET NCFTT = MNTMT*24.0/FLOAT(NRPFEC)
000132 LET MNTMT = MNTMT*24.0/FLOAT(NUM*3)
000133 LET MEANV = MEANV*24.0/FLOAT(NUM*3)
000134 LET MEAND = MEAND*24.0*4/FLOAT(NUM*3)
000135 DO TO 110, FOR EACH RES I
000136 LET UTIL(I) = UTIL(I)*100./TIME
000137 110 LOOP
000138 CALL DRIVE(R)
000139 IF STAPE EN 0, STOP
000140 IF EXCS IS EMPTY, GO TO 875
000141 DO TO 800, FOR I = (1)(15)
000142 LET Ibuff(I) = 0
000143 IF I GR 10, GO TO 800
000144 LET Fbuff(I) = 0.
000145 800 LOOP
000146 LET NUMB8 = NRRCO + 1
000147 DO TO 850, FOR EACH CASE IN EXCS
000148 WRITE ON TAPE STAPE, NUMB8,OPFAC(CASE),NOCAS(CASE),IDLOC(CASE),
000149 .OCCUR(CASE),ROX(CASE),FPR1(CASE),MM(CASE),NNN(CASE),GAMMA(CASE),
000150 .NEED(CASE),ATR(CASE),OFSPR(CASE),VIS(CASE),WIND(CASE),SWELL(CASE),
000151 .L(CASE),PUB(CASE),SIS(CASE),S2S(CASE),TSM(CASE),OST(CASE)
000152 FORMAT (I5,I3,D3.4,I5,I1,2I2,D3.2,I2,I5,D4.2,6I5,I2,D5.0,I1.4)
000153 WRITE ON TAPE STAPE, IUTYPE(CASE),VALUE(CASE),XCX(CASE),YCY(CASE),
000154 .XC(CASE),YC(CASE),STATN(CASE),CMRES(CASE),RESA(CASE),PRI(CASE),
000155 .REA(CASE),COSTC(CASE),ITOL(CASE),NOINT(CASE),NQUE(CASE),
000156 .TINT(CASE),TRUE(CASE),TQUE1(CASE),TSVC(CASE),TWAIT(CASE)
000157 FORMAT (I5,I10,4D5.2,I5,I2,I3,2I1,D7.2,I1,2I2,5D3.4)
000158 WRITE ON TAPE STAPE, Ibuff(I),Fbuff(I),Fbuff(I+1),Fbuff(I+1), FOR
000159 . I = (1)(10)(2)
000160 FORMAT 5(I2,2I.4,0I.2,I3)
000161 WRITE ON TAPE STAPE, Ibuff(I), FOR I=(11)(15)
000162 FORMAT 5(I3)
000163 LET NUMB8 = NUMB8 + 1
000164 850 LOOP
000165 875 IF STAPE EQ 4, STOP
000166 ENDFILE STAPE
000167 STOP
000168 END

```

15610
15620

```

000001 SUBROUTINE DRIVE(R)
000002 CALL JUMPER
000003     X
000004     NCF 0,6969
000005     CALL SAKSIN
000006     WRITE ON TAPE 6, R
000007     FORMAT(/, NOTE: 1, SCALING FACTOR FOR NORMALIZED DEMERIT (12TH COLU
000008 *N ABOVE) = ,D1,3//,SA,2, COEFFICIENTS USED IN THE CALCULATION,
000009 * OF CASE DEMERIT: 1,2,3,.)
000010     CALL GPPRES
000011     CALL HEADER
000012     DO TO 600, FOR EACH STA N
000013     IF (XS(N))NE(0,0), GO TO 500
000014     IF (YS(N))NE(0,0), GO TO 500
000015     GO TO 600
000016     500 CALL TITLE(N)
000017     DO TO 601, FOR EACH RST J, WITH (REST(N,J))GR(D)
000018     CALL RESULT(N,J)
000019     601 LOOP
000020     600 LOOP
000021     CALL HEAD
000022     DO TO 602, FOR EACH CASE IN EXCS
000023     CALL EXCASE(CASE)
000024     602 LOOP
000025     CALL DSTIR
000026     IF CQUE IS EMPTY, GO TO 730
000027     WRITE ON TAPE 6, TIME
000028     FORMAT (,1,56,*,THE FOLLOWING CASES WERE IN THE QUEUE WHEN THE SIM-
000029     ULATION ENDED AT, M4,2,2//,S9,*,TIME OF, S48,*,QUEUED, S10,*,S6,
000030     *,*,*,*, CASE OCCURRENCE STATION NNN MMH STS S2S AC YC
000031     *, AT TIME REASON TYPE,.)
000032     DO TO 720, FOR EACH ICNN IN CQUE
000033     IF FLG(ICNN) NE 0, GO TO (702,704,706),FLG(ICNN)
000034     702 LET CASE = ICNN
000035     GO TO 710
000036     704 LET CASE = CAS(ICNN)
000037     GO TO 710
000038     706 LET CASE = ESAC(ICNN)
000039     710 WRITE ON TAPE 6, OCCAS(CASE),OCCUR(CASE),STAT(CASE),NNN(CASE),
000040     *MMH(CASE),STS(CASE),S2S(CASE),AC(CASE),YC(CASE),TINQ(CASE),
000041     *SINGL(ICNN),FLG(ICNN)
000042     FORMAT (16,53,M3,2,2,217,315,205,1,52,M3,2,2,16,18)
000043     720 LOOP
000044     LET II = I
000045     WRITE ON TAPE 6, II
000046     FORMAT (,0 *, 0 = NO AVAILABLE RESOURCES,*,12,*, INTERRUPT,/,0,*,
000047     *, 1 = SINGLE RESOURCE CASE, 2 = NEED OR TOW NOTIF, 3 = SEARCH NOTE,
000048     *)
000049     GO TO 740
000050     730 WRITE ON TAPE 6, TIME
000051     FORMAT (,1,59,*,THERE WERE NO CASES IN THE QUEUE WHEN THE SIMULATI
000052     *ON ENDED AT, M4,2,2)
000053     740 LET II = 0
000054     GO TO 790, FOR EACH RFS IRS
000055     IF IR(IRS) EQ 0, GO TO 790

```

```

000055 IF I1 EQ 1, GO TO 750
000056 WRITE ON TAPE 6, TIME
000057 FORMAT (I1,S5,'THE FOLLOWING RESOURCES WERE BUSY WHEN THE SIMULAT
000058 ION ENDED AT,M4.2.2//SIC,RESOURCE,S43,CASE,/,RESOURCE, STATI
000059 ON, X1, YR IR CASE OCCUR STATION AND MPM SIS
000060 ,S25 XC YC TAIT')
000061 LET I1 = 1
000062 750 LET CASE = ACASE(IFLT(IHS))
000063 WRITE ON TAPE 6, IHS,STH(IHS),XR(IHS),YR(IHS),IS(IHS),OPFAC(CASE),
000064 ,NOCAS(CASE),OCCUR(CASE),STATN(CASE),NNN(CASE),NPM(CASE),SIS(CASE),
000065 ,S2S(CASE),XC(CASE),YC(CASE),TAIT(CASE)
000066 FORMAT (I7,I9,S3,2D5.1,2I4,I5,M5.2,2I7,3I5,2D6.1,M5.2.2)
000067 790 LOOP
000068 IF I1 EQ 1, RETURN
000069 WRITE ON TAPE 6, TIME
000070 FORMAT (///S9,'THERE WERE NO BUSY RESOURCES WHEN THE SIMULATION
000071 ENDED AT,M4.2.2)
000072 RETURN
000073 END

```


X	CIRCUIT UTILIZATION OF OTHER AIRCRAFT - *****
X	AIRU
X	C. STATION SUMMARY
X	TOTAL NUMBER OF TIMES A STANDBY WAS CALLED (ALL STATIONS) =
X	TOTAL NUMBER OF TIMES A STANDBY WAS CALLED BUT NOT USED (-L)
X	III. STATION
X	NUMBER NUMBER FAILURE FAILURE NO.
X	OF TYPE TYPE OF INTERPRETED TVEC
X	CASES NEEDS A R C Q NEEDS HOUR
X	STA ** *** ** ** *
X	I, NCAS(1), NEEDS(1), FAIL1(1), FAIL2(1), FAIL3(1), NBRMQ(1), NINTR(1), VC
X	FOR EACH STA I
X	TOTAL / AVG **** * ** ** **
X	KOUNT SNEED NHRFA NBRFB NBRFC LIMIT TOTIN MEANY
X	END

X

2x 1

T STATISTICS

2

1

160^x

```

000113
000114
000115
000116
000117
000118
000119
000120
000121
000122
000123
000124
000125
000126
000127
000128
000129
000130
000131
000132
000133
000134
000135
000136
000137
000138
000139
000140
000141
000142
000143
000144

1 X
1
1 X
1 X
1 X
12
11
1
1
12X
1
1
1
12X

*****
TOSBY
STATIONS) = *****
TUNPR
RESPONSE
GE AVERAGE AVG. CFIL AVG. POS. NORMLZO STANOBY5 AVERAGE
TWT TWT-TOL TWT-TOL OMERIT CALLS/UNPRO. UTILIZATION
S) (HOURS) (HOURS) (HOURS) **/ * *****
TR(1),AVGTW(1),CFTT(1),TWTOL(1),DMRT(1),NSTBY(1),UNPRO(1),USE(1)
*****
MEANW MCFTT MNTWT MEANO TOSBY TUNPR AVUTO
END

```

```

000001      REPORT GRPMS
000002      X
000003      X
000004      X
000005      X
000006      X
000007      X
000008      X
000009      X
000010      X
000011      X
000012      X
000013      X
000014      X
000015      X
000016      X
000017      X
000018      X

      GROUP **
      I,CS(I),INDS(I),FL1(I),FL2(I),FL3(I),INTRP(I),TVAVG(I),TWAVG(I),TMT
      FOR EACH GROUP I
      END

      OUP RESPONSE
      ERAGE AVERAGE AVG. POS. NORMALIZED TIMES UNPRO AVERAGE
      VEC TRAIT TRAIT-TOL DEMERIT STANDBY UTILIZATION
      OURS) (HOURS) (HOURS) (HOURS) CALLED CALLS
      ** **
      AV(I),AVDRT(I),NOSR(I),NONPR(I),USEAV(I)
      END
  
```

@ ELT HEADR,1,710520, 65045

000001
000002
000003
000004
000005
000006

REPORT HEADR

X

END

LIZATION (PERCENT)
END

IV • RESOURCE UTI

2

```

000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012

      B      F O R      E A C H      V E V O S      J
      X
      X
      X
      S T A T N   * * *
      N
      E N D

      S H I F T 1
      * * * * *
      S ( U S H F ( , J ) )

      S H I F T 2
      * * * * *

      S H I F T 3
      * * * * *

      S H I F T 4
      * * * * *

      X

      1

      S H I F T 5
      * * * * *

      S H I F T 6
      * * * * *

      S H I F T 7
      * * * * *

      S H I F T 8
      * * * * *

      E N D

```

4) ELT RESULT,1,710520, 65845

```
000001
000002      REPORT RESULT(NN,JJ)
000003      RST ***
000004      X
000005      X
000006      X
000007      X
000008      X
000009      X
000010      X
000011      X
000012      X
000013      X
000014      X

      JJ
      RES *** (ASSIGNED TO *** NREDS) = ***
      JCASE(K) UTIL(K)
      FOR EACH RES K, WITH (TYPE(K))EQ(JJ), AND (STN(K))EQ(NN)
      END
```

1 X

Q ELT HEAD,1,710,20, 65945

000001
000002
000003
000004
000005
000006

X

END

ASES

END

REPORT HEAD

V. EXCEPTIONAL C

3

```

000001 SUBROUTINE EXCASE(CASE)
000002   WRITE ON TAPE 6,NDCAS(CASE),STATN(CASE),SZS(CASE),L(CASE),
000003   • POB(CASE),UTYPE(CASE),OFSHR(CASE),OST(CASE),XC(CASE),TSM(CASE)
000004   FORMAT (SI,CASE NO.,16, STATN,14, SZS,13, LENGTH,15,
000005   1 • POB,15, UTYPE,15, OFSHR,15, OST,13,2, XC,
000006   2 DS,1, TSM,14,1)
000007   ARITE ON TAPE 6,NNN(CASE),FPRI(CASE),ITOL(CASE),SWELL(CASE),
000008   • NOINT(CASE),GAMMA(CASE),TINT(CASE),XCX(CASE),TSVC(CASE)
000009   FORMAT (SI9,NNN,14, FPRI,13, ITOL,15, SWELL,15,
000010   1 • NOINT,15, GAMMA,15,2, TINT,13,2, XCX,15,1,
000011   2 • TSVC,14,1)
000012   WRITE ON TAPE 6,MMN(CASE),PRI(CASE),NEED(CASE),VIS(CASE),
000013   • NQUE(CASE),COSTC(CASE),TQUE(CASE),YC(CASE),TWAIT(CASE)
000014   FORMAT (SI9,MMN,14, PRI,13, NEED,15, VIS,15,
000015   1 • NQUE,15, COST,15,2, TQUE,13,2, YC,15,1,
000016   2 • TWAIT,14,1)
000017   WRITE ON TAPE 6,SIS(CASE),IDLOC(CASE),AIR(CASE),WIND(CASE),
000018   • CNRES(CASE),OCCUR(CASE),TQUEI(CASE),YCY(CASE)
000019   FORMAT (SI9,SIS,14, IDLOC,13, AIR,15, WIND,15,
000020   1 • CNRES,15, OCCUR,15,2, TQUEI,13,2, YCY,15,1,/)
000021   RETURN
000022   END

```



```

000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054

REPORT DSTRIP
A. SYSTEM
VI. DIST?

-----
MEAN STD. DEV. 0
DAILY UTILIZATION OVERALL (WEEKDAYS) - .000 .000
MEFN(1),STDEV(1),CATG1(1),CATG2(1),CATG3(1),CATG4(1),CATG5(1),CATG
DAILY UTILIZATION OVERALL (WEEKENDS) - .000 .000
MEFN(2),STDEV(2),CATG1(2),CATG2(2),CATG3(2),CATG4(2),CATG5(2),CATG
A. CASES
-----
MEAN STD. DEV. 0
T-WAIT (ALL CASES) WEEKDAYS - .000 .000
MEFN(3),STDEV(3),CATG1(3),CATG2(3),CATG3(3),CATG4(3),CATG5(3),CATG
T-WAIT (ALL CASES) WEEKENDS - .000 .000
MEFN(4),STDEV(4),CATG1(4),CATG2(4),CATG3(4),CATG4(4),CATG5(4),CATG
TVEC (ALL CASES) WEEKDAYS - .000 .000
MEFN(5),STDEV(5),CATG1(5),CATG2(5),CATG3(5),CATG4(5),CATG5(5),CATG
TVEC (ALL CASES) WEEKENDS - .000 .000
MEFN(6),STDEV(6),CATG1(6),CATG2(6),CATG3(6),CATG4(6),CATG5(6),CATG
T-WAIT-TOL (ALL CASES) WEEKDAYS - .000 .000
MEFN(7),STDEV(7),CATG1(7),CATG2(7),CATG3(7),CATG4(7),CATG5(7),CATG
T-WAIT-TOL (ALL CASES) WEEKENDS - .000 .000
MEFN(8),STDEV(8),CATG1(8),CATG2(8),CATG3(8),CATG4(8),CATG5(8),CATG
DEMERT (ALL CASES) WEEKDAYS - .000 .000
MEFN(9),STDEV(9),CATG1(9),CATG2(9),CATG3(9),CATG4(9),CATG5(9),CATG
DEMERT (ALL CASES) WEEKENDS - .000 .000
MEFN(10),STDEV(10),CATG1(10),CATG2(10),CATG3(10),CATG4(10),CATG5(1
TSVC (ALL CASES) WEEKDAYS - .000 .000
MEFN(11),STDEV(11),CATG1(11),CATG2(11),CATG3(11),CATG4(11),CATG5(1
TSVC (ALL CASES) WEEKENDS - .000 .000
MEFN(12),STDEV(12),CATG1(12),CATG2(12),CATG3(12),CATG4(12),CATG5(1
END

IBUTIONS
-----
PERCENT-----
-0.5 0.5-1.0 1.0-2.0 2.0-3.0 3.0-4.0 4.0-5.0 5.0-10.0 >10 1
.00 .00 .00 .00 .00 .00 .00
6(1),CATG7(1),CATG9(1)
.00 .00 .00 .00 .00 .00 .00
6(2),CATG7(2),CATG9(2)
-----
HOURS-----
-0.5 0.5-1.0 1.0-2.0 2.0-3.0 3.0-4.0 4.0-5.0 5.0-10.0 >10 1
.00 .00 .00 .00 .00 .00 .00
6(3),CATG7(3),CATG9(3)
.00 .00 .00 .00 .00 .00 .00
6(4),CATG7(4),CATG9(4)
.00 .00 .00 .00 .00 .00 .00
6(5),CATG7(5),CATG9(5)
.00 .00 .00 .00 .00 .00 .00
6(6),CATG7(6),CATG9(6)
.00 .00 .00 .00 .00 .00 .00

```

000055	6(7),CATG7(7),CATG8(7)
000056
000057	6(8),CATG7(8),CATG8(8)
000058
000059	6(9),CATG7(9),CATG8(9)
000060
000061	0),CATG6(10),CATG7(10),CATG8(10)
000062
000063	1),CATG6(11),CATG7(11),CATG8(11)
000064
000065	2),CATG6(12),CATG7(12),CATG8(12)
000066	END

```

000001 SUPROUTINE JUMPER
000002 READ (5,1) N,A,B
000003 FORMAT(12,2A6)
000004 WRITE(6,2)N,A,B
000005 1 FORMAT(1H,152,'SANSI4 - DISTR(CT',12,32X,'DATE ',2A4)
000006 READ(5,3)A,H,N
000007 2 FORMAT(245,12)
000008 WRITE(6,4)A,H,N
000009 3 FORMAT(1H,154,2A6,2X,'1',12)
000010 READ (5,5)N,M
000011 4 FORMAT(212)
000012 WRITE(6,6)N,M
000013 5 FORMAT(1H,151,'FORECAST FROM 19',12,2X,'TO',2X,12)
000014 READ(5,8)N
000015 6 FORMAT(12)
000016 IF(N) 300,300,301
000017 7 WRITE(6,7)
000018 FORMAT(160,'WITH')
000019 DO 9 I=1,N
000020 11 FORMAT(149,F5.1,' PCT CHANGE FOR ',3A6)
000021 READ(5,10)A,B,C,D
000022 10 FORMAT(F5.1,3A6)
000023 WRITE(6,11)A,B,C,D
000024 9 CONTINUE
000025 300 WRITE(6,12)
000026 12 FORMAT(1H,151,'1. INPUT CONDITIONS')
000027 WRITE(6,13)
000028 13 FORMAT(1H,112,'A. STATION CHANGES')
000029 WRITE(6,14)
000030 14 FORMAT(15,'1. ADD')
000031 READ(5,15)N
000032 15 FORMAT(12)
000033 IF(N)35,35,34
000034 DO 16 I=1,N
000035 36 READ(5,18)K,A,B,I,J,L,M,T,A,IR,IC,IO,IE,IF,IG,IJ,IK
000036 18 FORMAT(13,2F9.1,13I3)
000037 WRITE(6,17)K,A,B,I,J,L,M,T,A,IR,IC,IO,IE,IF,IG,IJ,IK
000038 17 FORMAT(1R,'OPFAC ',13,2X,'XS = ',F8.1,'YS = ',F8.1,' ACS = ',13,'
000039 1',13,'',13,'',13,2X,'ADJS = ',13,'',13,'',13,'',13,2X,'CUT =
000040 2 ',13,'',13,'',13,'',13,'',13)
000041 16 CONTINUE
000042 GO TO 55
000043 35 WRITE(6,51)
000044 51 FORMAT(1R,'NO ADDITIONS')
000045 WRITE(6,19)
000046 19 FORMAT(15,'2. DELETE')
000047 READ(5,20)N
000048 20 FORMAT(12)
000049 IF(N)21,21,22
000050 22 DO 25 I=1,N
000051 READ(5,23)K
000052 23 FORMAT(13)
000053 WRITE(6,24)K
000054 24 FORMAT(1R,'OPFAC ',13)

```

```

000055 25 CONTINUE
000056 GO TO 57
000057 21 WRITE(6,56)
000058 56 FORMAT(118,'NO DELETIONS.')
000059 57 WRITE(6,56)
000060 26 FORMAT(15,'1. ATTRIBUTE CHANGE')
000061 READ(5,27)N
000062 27 FORMAT(12)
000063 IF(N)28,28,29
000064 29 DO 30 I=1,N
000065 READ(5,31)K,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12
000066 31 FORMAT(13,12A6)
000067 WRITE(6,32)K,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12
000068 32 FORMAT(118,'OFFAC ',I3,3X,12A6)
000069 30 CONTINUE
000070 GO TO 59
000071 28 WRITE(6,58)
000072 58 FORMAT(118,'NO ATTRIBUTE CHANGES')
000073 59 WRITE(6,33)
000074 33 FORMAT(14,'12, 8. RESOURCE TYPE CHANGES')
000075 WRITE(6,42)
000076 42 FORMAT(15,'1. NEW (SEE INITIALIZATION DATA FOR CHANGES TO CAPABILI-
000077 TILITY MATRIX(CPBL))')
000078 READ(5,34)N
000079 34 FORMAT(12)
000080 IF(N)37,37,38
000081 38 DO 41 I=1,N
000082 READ(5,39)K,A1,A2,A3,A4,A5,A6,A7,A8
000083 39 FORMAT(12,2F7.1,F5.2,F4.1,3F5.1,F6.1)
000084 WRITE(6,40)K,A1,A2,A3,A4,A5,A6,A7,A8
000085 40 FORMAT(118,'COST=',I2,6X,'COSTO=',F7.1,3X,'END=',F7.1,2X,'MR=',F5.2
000086 1,2X,'SLIM=',F4.1,2X,'SOA1=',F5.1,2X,'SOA2=',F5.1,2X,'SOA3=',F5.1,2
000087 2X,'TF=',F6.1)
000088 CONTINUE
000089 GO TO 60
000090 37 WRITE(6,61)
000091 61 FORMAT(118,'NO NEW ADDITIONS')
000092 60 WRITE(6,43)
000093 43 FORMAT(15,'2. ATTRIBUTE CHANGE')
000094 READ(5,44)N
000095 44 FORMAT(12)
000096 IF(N)45,45,46
000097 46 DO 49 I=1,N
000098 READ(5,47)K,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12
000099 47 FORMAT(12,12A6)
000100 WRITE(6,48)K,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12
000101 48 FORMAT(118,'RSI',I2,6X,12A6)
000102 49 CONTINUE
000103 GO TO 63
000104 45 WRITE(6,50)
000105 50 FORMAT(118,'NO ATTRIBUTE CHANGES')
000106 63 WRITE(6,64)
000107 64 FORMAT(15,'3. CAPABILITY CHANGE')
000108 READ(5,66)N
000109 66 FORMAT(12)
000110 IF(N)67,67,68
000111 68 WRITE(6,65)
000112 65 FORMAT(118,'SEE INITIALIZATION DATA FOR CHANGES TO CAPABILITY MATR

```

```

000113      11*(CPBL) FOR,*)
000114      DO 69 I=1,N
000115      READ(5,70)K
000116      FORMAT(13)
000117      WRITE(6,71)
000118      71  FORMAT(18,'DST=',13)
000119      69  CONTINUE
000120      GO TO 72
000121      67  WRITE(6,73)
000122      73  FORMAT(18,'O CAPABILITY CHANGES')
000123      72  WRITE(6,74)
000124      74  FORMAT(18,'T12,C. RESOURCE INVENTORY CHANGES')
000125      WRITE(6,75)
000126      75  FORMAT(15,'1. NEW OFFACS')
000127      READ(5,76)N
000128      76  FORMAT(12)
000129      IF(N)77,77,78
000130      78  DO 79 I=1,N
000131      READ(5,80)A,B,C,D,E,F,G,H,O,P,Q,R
000132      80  FORMAT(12A6)
000133      WRITE(6,81)A,B,C,D,E,F,G,H,O,P,Q,R
000134      81  FORMAT(18,12A6)
000135      79  CONTINUE
000136      GO TO 82
000137      77  WRITE(6,83)
000138      83  FORMAT(18,'NO NEW OFFACS')
000139      82  WRITE(6,84)
000140      84  FORMAT(15,'2. EXISTING OFFACS')
000141      READ(5,85)N
000142      85  FORMAT(12)
000143      IF(N)86,86,87
000144      87  DO 88 I=1,N
000145      READ(5,89)A,B,C,D,E,F,G,H,O,P,Q,R
000146      89  FORMAT(12A6)
000147      WRITE(6,90)A,B,C,D,E,F,G,H,O,P,Q,R
000148      90  FORMAT(18,12A6)
000149      88  CONTINUE
000150      GO TO 91
000151      86  WRITE(6,92)
000152      92  FORMAT(18,'NO RESOURCE CHANGES TO EXISTING OFFACS')
000153      91  WRITE(6,93)
000154      93  FORMAT(18,'T12,O. CRE# MANNING LEVEL CHANGES')
000155      WRITE(6,94)
000156      94  FORMAT(15,'1. NEW OFFACS')
000157      READ(5,95)N
000158      95  FORMAT(12)
000159      IF(N)96,96,97
000160      97  DO 125 I=1,N
000161      READ(5,98)N1,N2,N3,N4,N5,N6,N7,N8,N9
000162      98  FORMAT(9I3)
000163      WRITE(6,99)N1,N2,N3,N4,N5,N6,N7,N8,N9
000164      99  FORMAT(18,'OFFAC ',13,2X,'SHIFT1=',13,2X,'SHIFT2=',13,2X,'SHIFT3=',13,2X,'SHIFT4=',13,2X,'SHIFT5=',13,2X,'SHIFT6=',13,2X,'SHIFT7=',13,2X,'SHIFT8=',13)
000165      125  CONTINUE
000166      GO TO 100
000167      96  WRITE(6,101)
000168      101  FORMAT(18,'O CHANGES')
000169
000170

```


